Engineering with Computers
https://doi.org/10.1007/s00366-020-01229-y

ORIGINAL ARTICLE

Numerical solution of fractal-fractional Mittag–Leffler differential equations with variable-order using artificial neural networks

C. J. Zúñiga-Aguilar[1] · J. F. Gómez-Aguilar[2] · H. M. Romero-Ugalde[3,4] · R. F. Escobar-Jiménez[1] · G. Fernández-Anaya[5] · Fawaz E. Alsaadi[6]

Received: 30 August 2020 / Accepted: 23 November 2020

Abstract
In this work, a methodology based on a neural network to solve fractal-fractional differential equations with a nonsingular and nonlocal kernel is proposed, the neural network is optimized by the Levenberg–Marquardt algorithm. For evaluating the neural network, different chaotic oscillators of variable order are solved and compared with algorithms of numeric approximation.

Keywords Fractional calculus · Fractal-fractional definition · Nonlinear fractional differential equations · Variable-order fractal-fractional derivative · Artificial neural networks

1 Introduction

Fractional calculus (FC) is the generalization of integration and differentiation of noninteger order. In recent years, the FC has been used to describe precisely the physical systems dynamic through the use of fractional differential equations. For example, to describe viscoelastic systems [1–3], diffusion [4–6], electromagnetism [7–9], mechanics [10–12], bioengineering [8], biochemistry [13], etc.

Sometimes, it is complicated to solve a differential equation of fractal order analytically; therefore, to analyze or observe the performance of the fractional systems (referring to FDE that describe the diverse systems), it is utilized different techniques to approximate the FDE solution. For example, variational Laplace method [14], Laplace transforms [15], Lagrange multipliers [16, 17], Laplace homotopy [18], homotopy perturbation technique [19–21], and homotopy analysis transform method [22].

The FC is composed of several definitions of fractional derivative, for instance, Günwald–Letnikov (GL) [23], Riemann–Liouville (RL) [24], Liouville–Caputo (LC) [25], Caputo–Fabrizio (CF) [26]. The difference between these definitions is their kernel. The kernel is the term that allows the fractional derivative to have memory effects. The RL and LC derivatives are formed by a potency nucleus which has a singularity when $t = \tau$.

Due to the singularity of the RL and LC definitions, it was proposed using an exponential nucleus as in the CF definition. Highlighting that, in FC, different functions can be used to generalize primitive functions; for example, the factorial and the exponential functions are generalized by the Gamma and Mittag–Leffler functions, respectively. The term generalization means that the primitive functions (factorial, exponential, etc) are only a case of superior functions (Gamma function, Mittag–Leffler). As mentioned above, the CF definition took care of the singularities that the RL and LC derivative have introducing an exponential function as the kernel. In [27], the authors proposed to utilize a superior function as kernel and this way to face the singularities and localities that the exponential function express.

✉ J. F. Gómez-Aguilar
jose.ga@cenidet.tecnm.mx

1 Tecnológico Nacional de México/CENIDET, Interior Internado Palmira S/N, Col. Palmira, 62490 Cuernavaca, MOR, Mexico

2 CONACyT-Tecnológico Nacional de México/CENIDET, Interior Internado Palmira S/N, Col. Palmira, C.P. 62490, 62490 Cuernavaca, MOR, Mexico

3 Université Grenoble Alpes, Grenoble 38000, France

4 CEA LETI MINATEC Campus, Grenoble 38054, France

5 Departamento de Física y Matemáticas., Universidad Iberoamericana, 01219 Mexico, CDMX, Mexico

6 Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Published online: 02 January 2021

Several works justify the use of variable orders $\alpha(t)$ in the formulation of the fractional derivatives in the Atangana–Baleanu (AB) sense to find mathematical representations that describe in a better way a physical system [28–30]. The fact of using a variable order adds a stronger index of nonlocality, allowing to find more general solution of the physical system.

The artificial neural networks (ANNs) are nonlinear maps systems [31], and they have been used in several science fields thanks to their great adaptability in several problems [32–34]. Some of the applications of the neural network are the systems identification [35], fractional-order system identification [36], classification [37, 38], control of fractional systems [39, 40], image encryption [41], to solve differential equations [42, 43], among others.

In [44], it is proposed a methodology to solve FDEs in the LC sense; in this work, the neural network is used for establishing the FDE's initial conditions. The neural networks presented problems when the equation has highly nonlinear components. In [45, 46], it is proposed a diffuse neural network to solve diffuse differential equations. In [47], the authors propose using a neural network to find the optimal control law in such a way that a fractional system is stable. This proposal was explored for solving FDE's obtaining promising results. In [48], a methodology to find the solution of the model of combustible ignition in a dimension was proposed, and the neural network was optimized using a particle swarm optimization algorithm (PSO). The authors in [49] used a recurring neural network using a PSO for the optimization of synaptic weights of the neural networks to solve FDEs in the RL sense. In [50], it was presented a methodology to solve optimal control problems with an infinite horizon using the definition of LC derivative leaning in a neural networks. In [42], a methodology based on neural network proposed by [44] was used to solve FDEs in the ABC sense with variable-order, it was demonstrated the robustness of the neural network despite the variability of the FDE's order. In [43], a neural network to solve FDE's in the ABC sense with variable order and delay was proposed. Several examples were presented to demonstrate that, no matter the delay of the variable, as well as the variable-order through time, the neural network can characterize the solution of the FDEs.

2 Preliminaries and notation

2.1 Fractal-fractional operator

Definition 1 Let $\Xi \in C^1$ in $[t_0, T]$, α and β bounded continuous functions, such that $\alpha = \{\alpha(t) : 0 < \alpha(t) \leq 1\}$ and $\beta = \{\beta(t) : 0 < \beta(t) \leq 1\}$, and then, the fractional derivative of Ξ with fractional variable-order $\alpha(t)$ and fractal variable-order $\beta(t)$ is defined as:

$$\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha(t),\beta(t)}\Xi\right)(t) = \frac{AB(\alpha(t))}{1-\alpha(t)}\frac{\mathrm{d}}{\mathrm{d}t^{\beta(t)}}\int_{t_0}^{T}\Xi(t)E_{\alpha(\tau)}\left(-\frac{\alpha(\tau)}{1-\alpha(\tau)}(t-\tau)^{\alpha(\tau)}\right)\mathrm{d}\tau, \tag{1}$$

where,

$$\frac{\mathrm{d}}{\mathrm{d}t^{\beta(t)}}\Phi(t) = \lim_{t=t_1}\frac{\Phi(t)-\Phi(t_1)}{t^{\beta(t)}-t_1^{\beta(t_1)}} = \dot{\Phi}(t)\frac{t_1^{\beta(t_1)}}{\dot{\beta}(t_1)\ln t_1 + \frac{\beta(t_1)}{t1}}, \tag{2}$$

$E_{\alpha(\cdot)}(\cdot)$ represents the function of Mittag–Leffler which generalizes the behavior of the exponential function. As is shown in Eq. (1), the kernel Mittag–Leffler is completely stable in the sense that it does not show singularities in its domain [51] and $AB(\cdot)$ is a function of normalization expressed in Eq. (3):

$$AB(\alpha(t)) = 1-\alpha(t) + \frac{\alpha(t)}{\Gamma(\alpha(t))}, \tag{3}$$

where $\Gamma(x)$ is the Gamma function of Euler which generalizes the factorials and computes the factorial noninteger number whose domain is $\Omega_\Gamma = \{\Gamma(x) : x \in \mathbb{R}\backslash\mathbb{Z}^-\}$.

The integral associated with the fractal-fractional derivative previously shown in (1) is presented in Eq. (4):

$$
\begin{aligned}
{}^{\mathrm{FFM}}_{t_0}\mathcal{I}_T^{\alpha(t),\beta(t)}\Xi(t) =& \frac{1-\alpha(t)}{AB(\alpha(t))}t^{\beta(t)}\left[\dot{\beta}(t)\ln t+\frac{\beta(t)}{t}\right]\Xi(t)\\
&+\frac{\alpha(t)}{AB(\alpha(t))\Gamma(\alpha(t))}\int_{t_0}^{T}(t-\tau)^{\alpha(\tau)-1}\Xi(\tau)\\
&\left[\dot{\beta}(\tau)\ln\tau+\frac{\beta(\tau)}{\tau}\right]\tau^{\beta(\tau)}\mathrm{d}\tau.
\end{aligned}
\tag{4}
$$

It is worth to highlight that the proof of Eq. (2), the integral associated (4), existence, and uniqueness of Eq. (1) have been documented in [51].

$$
\begin{aligned}
\Xi(t) =& \frac{1-\alpha(t)}{AB(\alpha(t))}t^{\beta(t)}\left(\dot{\beta}(t)\ln t+\frac{\beta(t)}{t}\right)\xi(t,\Xi(t))\\
&+\frac{\alpha(t)}{\Gamma(\alpha(t))AB(\alpha(t))}\int_{t_0}^{T}(t-\tau)^{\alpha(\tau)-1}\tau^{\beta(\tau)}\\
&\times\left(\dot{\beta}(\tau)\ln\tau+\frac{\beta(\tau)}{\tau}\right)\xi(\tau,\Xi(\tau))d\tau;
\end{aligned}
\tag{6}
$$

for simplicity, let us consider the following function:

$$
\Sigma(\tau,\Xi(\tau)) = \tau^{\beta(\tau)}\left(\dot{\beta}(\tau)\ln\tau+\frac{\beta(\tau)}{\tau}\right)\xi(\tau,\Xi(\tau));
\tag{7}
$$

then, evaluating at $t = t_{m+1}$, we have:

$$
\begin{aligned}
\Xi(t_{m+1}) =& \frac{1-\alpha(t_{m+1})}{AB(\alpha(t_{m+1}))}t^{\beta(t_{m+1})}\left(\frac{\beta(t_{m+1})-\beta(t_m)}{\Delta t}\ln t_{m+1}+\frac{\beta(t_{m+1})}{t_{m+1}}\right)\xi(t_m,\Xi(t_m))\\
&+\frac{\alpha(t_{m+1})}{\Gamma(\alpha(t_{m+1}))AB(\alpha(t_{m+1}))}\int_{t_0}^{T}(t_{m+1}-\tau)^{\alpha(\tau)-1}\Sigma(\tau,\Xi(\tau)),
\end{aligned}
\tag{8}
$$

$$
\begin{aligned}
\Xi(t_{m+1}) =& \frac{1-\alpha(t_{m+1})}{AB(\alpha(t_{m+1}))}t^{\beta(t_{m+1})}\left(\frac{\beta(t_{m+1})-\beta(t_m)}{\Delta t}\ln t_{m+1}+\frac{\beta(t_{m+1})}{t_{m+1}}\right)\xi(t_m,\Xi(t_m))\\
&+\frac{\alpha(t_{m+1})}{\Gamma(\alpha(t_{m+1}))AB(\alpha(t_{m+1}))}\sum_{j=0}^{m}\int_{t_j}^{t_{j+1}}(t_{m+1}-\tau)^{\alpha(\tau)-1}\Sigma(\tau,\Xi(\tau)).
\end{aligned}
\tag{9}
$$

2.2 Numerical scheme to solve FFDE

In several cases, it is not possible to find the analytical solution of the fractional differential equation due to the high complexity of these. Therefore, the solution of the FDE is approximated by numerical methods to find the FDE trajectory. In the present work, a modification of the numerical scheme proposed in [51] is presented to find the FDE solution with fractal dimensions with variable order, as shown in Eq. (5):

$$
\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha(t),\beta(t)}\Xi\right)(t) = \xi(t,\Xi(t)),\quad \Xi(0)=\Xi_0.
\tag{5}
$$

Then, using the Lagrange interpolation, we approximate the auxiliary function $\Sigma(\tau,\Xi(\tau))$ within the interval $[t_j,t_{j+1}]$:

$$
\begin{aligned}
\gamma_j(\tau) =& \frac{\tau-t_{j-1}}{t_j-t_{j-1}}\Sigma(t_j,\Xi_j)-\frac{\tau-t_j}{t_j-t_{j-1}}\Sigma(t_{j-1},\Xi(t_{j-1}))\\
=& \frac{\Sigma(t_j,\Xi_j)}{\Delta t}(\tau-t_{j-1})-\frac{\Sigma(t_{j-1},\Xi(t_{j-1}))}{\Delta t}(\tau-t_j).
\end{aligned}
\tag{10}
$$

As shown above, the interpolation is used to approximate the function $\Sigma(\tau,\Xi(\tau))$ in an interval of Δt. Considering that approximation, we replace γ_j and its values by $\Sigma(\tau,\Xi(\tau))$, and integrating, we obtain the following expression:

$$
\begin{aligned}
\Xi(t_{m+1}) =& \frac{1-\alpha(t_{m+1})}{AB(\alpha(t_{m+1}))}t^{\beta(t_{m+1})}\left(\frac{\beta(t_{m+1})-\beta(t_m)}{\Delta t}\ln t_{m+1}+\frac{\beta(t_{m+1})}{t_{m+1}}\right)\xi(t_m,\Xi(t_m))\\
&+\frac{\alpha(t_{m+1})}{AB(\alpha(t_{m+1}))}\sum_{j=0}^{m}\left\{\frac{(\Delta t)^{\alpha(t_j)}}{\Gamma(\alpha(t_j)+2)}\Sigma(t_j,\Xi_j)\left[(m-j+1)^{\alpha(t_j)}(m-j+2+\alpha(t_j))\right.\right.\\
&\left.\left.-(n-j)^{\alpha(t_j)}(m-j+1+\alpha(t_j))\right]-\frac{(\Delta t)^{\alpha(t_j)}}{\Gamma(\alpha(t_j)+2)}\Sigma(t_{j-1},\Xi_{j-1})\left[(m-j+1)^{\alpha(t_{j-1})}-(m-j)^{\alpha(t_{j-1})}(m-j+1+\alpha(t_{j-1}))\right]\right\}.
\end{aligned}
\tag{11}
$$

The numerical scheme that solves the fractal-fractional differential equations is presented below. Let us consider a general nonlinear equation as is shown in Eq. (6):

Now, replacing Eqs. (7) into (11), we have:

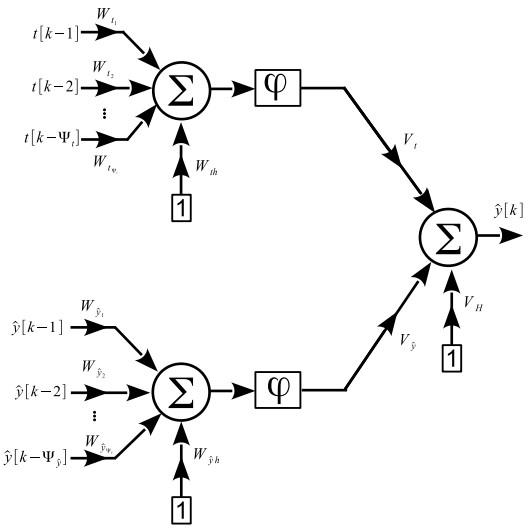


Fig. 1 Artificial neural-network proposed to solve variable-order FFDEs. The ANN is composed by two layers, two neurons in the input layer, and one neuron in the output layer

$$\hat{y}[k] = V_t \varphi_1\left(\sum_{i=1}^{\Psi_t} W_{t_i} t[k-i] + W_{\text{th}}\right) + V_{\hat{y}} \varphi_1\left(\sum_{j=1}^{\Psi_{\hat{y}}} W_{\hat{y}_j} \hat{y}[k-j] + W_{\hat{y}h}\right) + V_{\text{H}}, \tag{13}$$

where φ represents activation functions used to process the linear combination of the synaptic weights and the inputs of the ANN commonly chosen as sigmoid functions. The synaptic weights V_t, W_{th}, $V_{\hat{y}}$, $W_{\hat{y}h}$, $V_{\text{H}} \in \mathbb{R}$, $W_{ti} \in \mathbb{R}^{\Psi_t}$, and $W_{\hat{y}j} \in \mathbb{R}^{\Psi_{\hat{y}}}$ are the parameters of the model which are optimized by a learning algorithm.

This work aims to solve FFDEs of variable order with neural networks in such a way that the initial conditions Ξ_0 of the fractional-fractal differential equation are fulfilled, as shown in Eq. (5). In the proposed method in [42–44], it was established a general equation to solve differential equations using a neural network in the Caputo and Atangana–Baleanu–Caputo sense with variable order. Unlike

$$\begin{aligned}
\Sigma(t_j, \Xi_j) =& t_j^{\beta(t_j)}\left(\frac{\beta(t_{j+1}) - \beta(t_j)}{\Delta t}\ln t_j + \frac{\beta(t_j)}{t_j}\right)\xi(t_j, \Xi_j),\\
\Sigma(t_{j-1}, \Xi_{j-1}) =& t_{j-1}^{\beta(t_{j-1})}\left(\frac{\beta(t_j) - \beta(t_{j-1})}{\Delta t}\ln t_{j-1} + \frac{\beta(t_{j-1})}{t_{j-1}}\right)\xi(t_{j-1}, \Xi_{j-1}),\\
\Xi(t_{n+1}) =& \frac{1-\alpha(t_m)}{AB(\alpha(t_m))} t_{m+1}^{\beta(t_{m+1})}\left(\frac{\beta(t_{m+1}) - \beta(t_m)}{\Delta t}\ln t_{m+1} + \frac{\beta(t_{m+1})}{t_{m+1}}\right)\xi(t_m, \Xi_m)\\
&\frac{\alpha(t_m)}{AB(\alpha(t_m))}\sum_{j=0}^{m}\left\{\frac{(\Delta t)^{\alpha(t_j)}}{\Gamma(\alpha(t_j)+2)} t_j^{\beta(t_j)}\left(\frac{\beta(t_{j+1}) - \beta(t_j)}{\Delta t}\ln t_j + \frac{\beta(t_j)}{t_j}\right)\xi(t_j, \Xi_j)\right.\\
&\times\left[(m-1+1)^{\alpha(t_j)}(m-j+2+\alpha(t_j)) - (m-j)^{\alpha(t_j)}(m-j+1+\alpha(t_j))\right]\\
&-\frac{(\Delta t)^{\alpha(t_j)}}{\Gamma(\alpha(t_{j-1})+2)} t_{j-1}^{\beta(t_{j-1})}\left(\frac{\beta(t_j) - \beta(t_{j-1})}{\Delta t}\ln t_{j-1} + \frac{\beta(t_{j-1})}{t_{j-1}}\right)\xi(t_{j-1}, \Xi_{j-1})\\
&\left.\times\left[(m-j+1)^{\alpha(t_{j-1})+1} - (n-j)^{\alpha(t_{j-1})}(m-j+1+\alpha(t_{j-1}))\right]\right\}.
\end{aligned} \tag{12}$$

3 Proposed ANN and method for solving FFDEs

The neural network used in this work is shown in Fig. 1. This architecture of the neural network has been presented in several investigations to develop nonlinear systems identification [36], and to find the solution of systems of differential equations with variable order in the sense of Atangana–Baleanu–Caputo [42, 43]. The ANN is of two layers with two neurons in the input layer and one neuron in the exit layer, meaning, a neural networks $2-1$.

The neuromodel shown in Fig. 1 is described in Eq. (13):

[42, 43], in the present work, it will be proposed a general solution in such a way that it is reflected the contribution of fractality in the differential equation like the effect provided by the variable orders $\alpha(t)$ and $\beta(t)$, as shown in Eq. (14):

$$\hat{y} = \Xi_0 + \sum_{j=1}^{\lceil\alpha(t)\rceil} t^j + \lceil\beta(t)\rceil t^{\lceil\alpha(t)\rceil+\lceil\beta(t)\rceil-1}\hat{y}_N(t, \Theta), \tag{14}$$

where $\lceil\alpha(t)\rceil$ is the fractional-order rounded up to its next superior value, $\lceil\beta(t)\rceil$ is the fractal order rounded up to its superior value, $\hat{y}$ is the solution of the differential equation, $\hat{y}_N(t_k, \Theta)$ is the neuromodel proposed in (13), and Θ is the vector of parameters provided by the artificial neural network. It is clear that Eq. (14) satisfies the behavior of the

FFDE's initial condition. Replacing Eqs. (14) into (5), we have the following expression:

$$\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha(t),\beta(t)}\hat{y}(t,\Theta)\right)(t) = f(t,\hat{y}(t,\Theta)). \tag{15}$$

In this way, Eq. (15) becomes an optimization problem of the form:

$$\Theta^* = \min_{\Theta}\{C(e)\} = \min_{\Theta}\left\{\frac{1}{2}\sum_{j=1}^{N} e_j^2\right\}, \tag{16}$$

where $C(e)$ is the objective function to minimize by searching the optimal parameter through optimization algorithm, e is the error defined as $e := \left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha(t),\beta(t)}\hat{y}(t,\Theta)\right)(t) - f(t,\hat{y}(t,\Theta))$. Now, to approximate the fractal-fractional derivative of fractal variable and fractional variable order of Eq. (15), it is used the definition shown in (1), as shown in Eq. (17):

$$\begin{aligned}
\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha(t),\beta(t)}\hat{y}\right)(t,\Theta) &= \frac{AB(\alpha(t))}{1-\alpha(t)}\frac{\mathrm{d}}{\mathrm{d}t^{\beta(t)}}\int_{t_0}^{T}\hat{y}(\tau,\Theta)E_{\alpha(\tau)}\left(-\frac{\alpha(\tau)}{1-\alpha(\tau)}(t-\tau)^{\alpha(\tau)}\right)\mathrm{d}\tau,\\
\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha(t),\beta(t)}\hat{y}\right)(t,\Theta) &= \frac{AB(\alpha(t))}{1-\alpha(t)}\frac{\mathrm{d}}{\mathrm{d}t}\int_{t_0}^{T}\hat{y}(\tau,\Theta)E_{\alpha(\tau)}\left(-\frac{\alpha(\tau)}{1-\alpha(\tau)}(t-\tau)^{\alpha(\tau)}\right)\frac{1}{\dot{\beta}(t)\ln t + \dfrac{\beta(t)}{t}}\mathrm{d}\tau.
\end{aligned} \tag{17}$$

For simplicity, a change of variable is made in Eq. (17), as shown in Eq. (18):

$$P(\tau,t,\alpha(t),\beta(t),\Theta) = \hat{y}(\tau,\Theta)E_{\alpha(\tau)}\left(-\frac{\alpha(\tau)}{1-\alpha(\tau)}(t-\tau)^{\alpha(\tau)}\right)\frac{1}{\dot{\beta}(t)\ln t + \dfrac{\beta(t)}{t}}. \tag{18}$$

Substituting Eqs. (18) into (17), we obtain:

$$\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha_t,\beta_t}\hat{y}\right)(t,\Theta) = \frac{AB(\alpha(t))}{1-\alpha(t)}\frac{\mathrm{d}}{\mathrm{d}t}\int_{t_0}^{T}P(\tau,t,\alpha_t,\beta_t,\Theta)\mathrm{d}\tau, \tag{19}$$

where $\alpha_t = \alpha(t)$ y $\beta_t = \beta(t)$. The integral can be solved by any numerical integration method. For simplicity, it is proposed to use the trapezoidal method in such a way that:

$$\begin{aligned}
f_{\mathrm{p}}(t,\Theta) &= \frac{hAB(\alpha_t)}{2(1-\alpha_t)}\frac{\mathrm{d}}{\mathrm{d}t}\left[P(0,t,\alpha_t,\beta_t,\Theta) + 2P(h,t,\alpha_t,\beta_t,\Theta) + \cdots + 2P(h(N-1),t,\alpha_t,\beta_t,\Theta) + P(T,t,\alpha_t,\beta_t,\Theta)\right],\\
f_{\mathrm{p}}(t,\Theta) &\approx \frac{AB(\alpha_t)}{2(1-\alpha_t)}\Delta P.
\end{aligned} \tag{20}$$

According to Eq. (20), we have the following optimization problem:

$$\Theta^* = \min_{\Theta}\left\{\frac{1}{2}\sum_{i=1}^{N}\left[f_{\mathrm{p}}(t_i,\Theta) - f\left(t_i,\hat{y}(t_i,\Theta)\right)\right]^2\right\}. \tag{21}$$

With the purpose of resolve the optimization problem shown in Eq. (21), it is used the Levenberg–Marquardt algorithm (LM) for the estimation of the synaptic weights of the neural network as shown in Eq. (22):

$$\Delta W = \left[J(W_k)^T J(W_k) + \mu I\right]^{-1} J(W_k)\left(f_{\mathrm{p}}\left(t_k,\Theta\right) - f\left(t_k,\hat{y}\left(y_k,\Theta\right)\right)\right), \tag{22}$$

where $J(W_k)$ is the Jacobian matrix of the synaptic weight in the instant kth, $\mu > 0$ represents the learning coefficient.

4 Results

This section presents a series of examples used for demonstrating that using Eq. (14), the FFDEs can be solved. It is worth highlighting that the proposed neural network is of three neurons, two neurons in the input layer to process the time and to process the behavior of the FFDE solution. The number of parameters of the neural network varies according to the estimation performed, i.e., we will obtain $\Psi_t + \Psi_{\hat{y}+5}$ synaptic weights. The synaptic weights of the neural network are optimized by the LM algorithm. The performance of each solution is measured by applying Eq. (23):

$$\mathrm{FIT} = 100\left(1 - \frac{||y-\hat{y}||}{||y-\bar{y}||}\right)\%, \tag{23}$$

where y represents the real solution of the FFDE calculated using the numerical scheme proposed in Eq. (12), $\bar{y}$ is the mean of y, and $\hat{y}$ represents the neural network solution. The FIT is a performance index that allows us to measure the relationship between the real and estimated signals, i.e., if the FIT FIT $\to 100\%$, both signals are equal, but if FIT $\to 0$, both signals are different.

Example 1 Consider the following set of fractal-fractional differential equations with $\alpha(t) = \tanh(t+5)$ and $\beta(t) = 0.98$:

$$\begin{aligned}
\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha_t,\beta} x\right)(t) &= (20a+40)(y-x) + \frac{5a+4}{25}xz, \quad x(0) = 2,\\
\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha_t,\beta} y\right)(t) &= (55-90a)x + (5a+20)y - xz, \quad y(0) = 1,\\
\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha_t,\beta} z\right)(t) &= -\frac{13}{20}x^2 + xy + \frac{11-6a}{6}z, \quad z(0) = 4,
\end{aligned} \tag{24}$$

where $a = 0$ y $b = 0.2$ [52]. It is proposed three different solutions corresponding to each of the system status shown in (32). According to Eq. (14), we obtain the neuronal solution given by Eq. (25):

$$\begin{aligned}
\hat{x} &= 2 + t + t\hat{x}_N,\\
\hat{y} &= 1 + t + t\hat{y}_N,\\
\hat{z} &= 4 + t + t\hat{z}_N.
\end{aligned} \tag{25}$$

According to the neuronal model shown in Eq. (13), two regression parameters must be estimated Ψ_t y $\Psi_{\hat{x},\hat{y},\hat{z}}$. For that, a brute force algorithm was used to find the optimal solution $\left(\Psi_t^*\ \mathrm{y}\ \Psi_{\hat{x},\hat{y},\hat{z}}^*\right)$; we proposed 20 values for each of the regressors, giving a total of 400 different solutions for each of the states. The objective function that was minimized is the FIT shown in Eq. (23). Figure 2 shows the behavior of the FIT changing the value of Ψ_t y $\Psi_{\hat{x},\hat{y},\hat{z}}$ between 1 and 20; it is worth mentioning that a mark has been put with the purpose of highlight that value with a better performance.

According to Fig. 2, it is showing that for the state x, we define $\Psi_t = 9$ and $\Psi_{\hat{x}} = 2$. Then, the neural network used to find the solution of the fractal-fractional differential equation for the state x is shown in Eq. (26), where the neural network used needs a total of 16 parameters. The efficiency of the neural network can be seen in Fig. 2a, obtaining 99.57% of the relationship. Figure 3 shows the behavior of the numerical scheme proposed in (12) and the solution found for the neural network:

$$\hat{x}[k] = V_t\varphi_1\left(\sum_{i=1}^{9} W_{t_i}t[k-i] + W_{\mathrm{th}}\right) + V_{\hat{x}}\varphi_1\left(\sum_{j=1}^{2} W_{\hat{x}_j}\hat{x}[k-j] + W_{\hat{x}h}\right) + V_{\mathrm{H}}. \tag{26}$$

In the same way, Fig. 2b shows the optimal values of Ψ_t and $\Psi_{\hat{y}}$ to reach 99.32% of the FIT in the estimation of the state y. According to Fig. 2b, it must be defined $\Psi_t = 2$ and $\Psi_{\hat{x}} = 2$ to find the performance. Thus, the neuronal model to estimate the state y is shown in Eq. (27) with only 9 parameters. Figure 4 shows the comparison between the numerical scheme and the neural network proposed:

$$\begin{aligned}
\hat{y}[k] =& V_t\varphi_1\left(\sum_{i=1}^{2} W_{t_i}t[k-i] + W_{\mathrm{th}}\right)\\
&+ V_{\hat{y}}\varphi_1\left(\sum_{j=1}^{2} W_{\hat{y}_j}\hat{y}[k-j] + W_{\hat{y}h}\right) + V_{\mathrm{H}}.
\end{aligned} \tag{27}$$

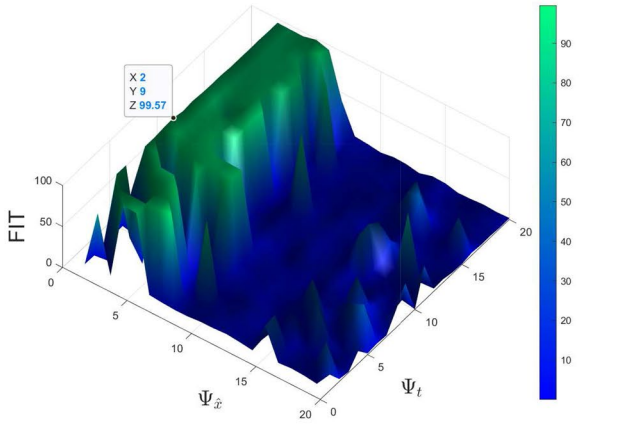
Figure 2c shows the values of Ψ_t and $\Psi_{\hat{z}}$, so the proposed solution by the neural network reaches a 99.76% of FIT. Thus, with $\Psi_t = 9$ and $\Psi_{\hat{z}} = 15$, it is reached the maximum performance with a total of 29 parameters. The neural network proposed to find the solution of the state z is shown in Eq. (28). Figure 5 shows the behavior of the numerical scheme and the neural network:

$$\hat{z}[k] = V_t\varphi_1\left(\sum_{i=1}^{9} W_{t_i}t[k-i] + W_{\mathrm{th}}\right) + V_{\hat{z}}\varphi_1\left(\sum_{j=1}^{15} W_{\hat{z}_j}\hat{z}[k-j] + W_{\hat{z}h}\right) + V_{\mathrm{H}}. \tag{28}$$

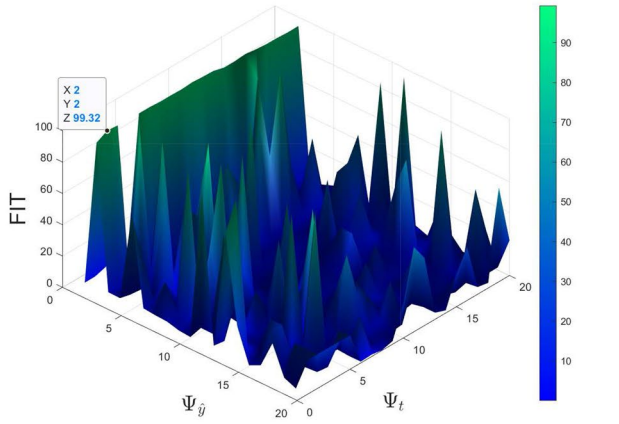
Let us remark a zoom-in was made at the start of each trajectory to show that the general solution proposed in Eq. (14) respects the initial conditions of the FFDE. Figure 6 shows the phase portrait of the states to perceive the system's chaotic behavior.

After finding the values or the orders with the brute force algorithm, it was repeated 1000 times each one of the estimates made to demonstrate that the algorithm proposed based in neuronal networks is more efficient in computational cost than the numerical method proposed in Eq. (12). It is important to denote that both experiments (numerical method and neuronal networks) were tested under the same conditions, meaning, on the same computer (Mac with MATLAB 2016B with 32 GB RAM) and without programs executing in the background. After the experimentation with the 1000 estimates, the numerical algorithm shown in Eq. (12) showed an execution average of $\mathrm{ST}_{\mathrm{nm}} = 0.37297\mathrm{s} \pm 0.1725$ s and the neuronal networks a time of $\mathrm{ST}_{\mathrm{nn}} = 0.01678 \pm 0.0097844$.

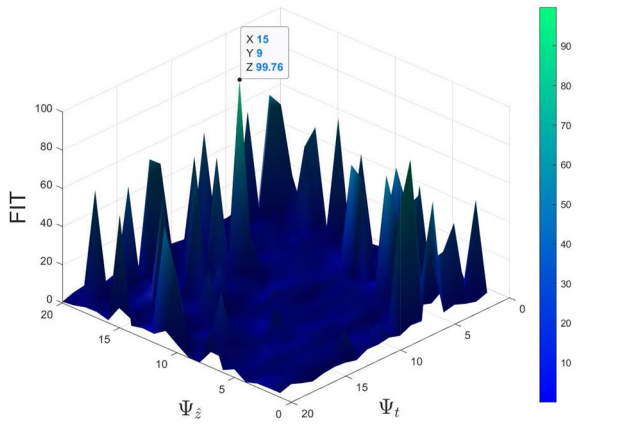
Example 2 Consider the following set of FFDEs that represent the chaotic behavior of the Rössler system [42]:



(a) Behavior of the state x to find the optimal value of Ψ_t and $\Psi_{\hat{x}}$.



(b) Behavior of the state x to find the optimal value of Ψ_t and $\Psi_{\hat{y}}$.



(c) Behavior of the state x to find the optimal value of Ψ_t and $\Psi_{\hat{z}}$.

Fig. 2 Behavior of the FIT changing the value of each regressor between 1 and 20 to find the optimal value of Ψ_t and $\Psi_{\hat{x},\hat{y},\hat{z}}$

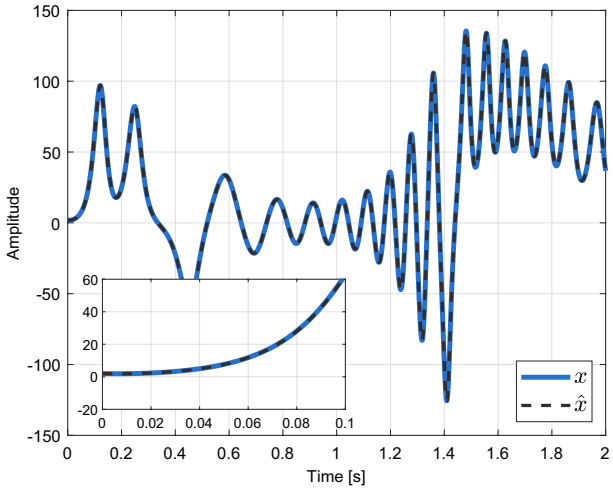


Fig. 3 The comparison between the ANN with the numerical scheme proposed to find the solution of the state x

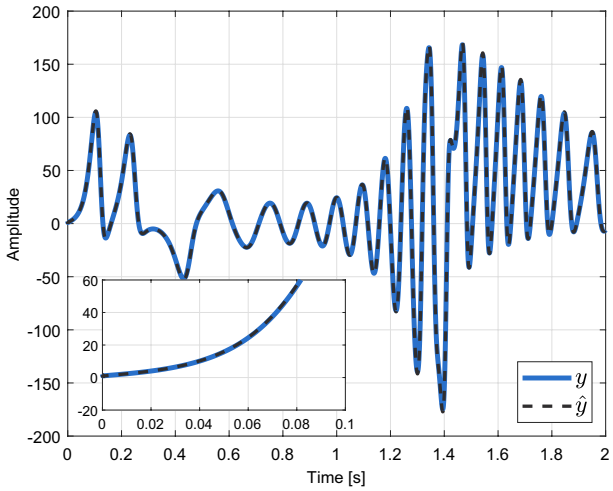


Fig. 4 The comparison between the ANN with the numerical scheme proposed to find the solution of the state y

$$
\begin{aligned}
\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}^{\alpha,\beta_t}_{T}x\right)(t) &= -(z+y) \quad x(0) = 4.87623,\\
\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}^{\alpha,\beta_t}_{T}y\right)(t) &= x + \varepsilon_1 y, \quad y(0) = 4.87623,\\
\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}^{\alpha,\beta_t}_{T}z\right)(t) &= \varepsilon_2 + z(x-\varepsilon_3), \quad z(0) = 0.1278,
\end{aligned}
\tag{29}
$$

where $\varepsilon_1 = \varepsilon_2 = 0.1$ and $\varepsilon_3 = 14$. For this example, it was considered keeping constant the fractional derivative order

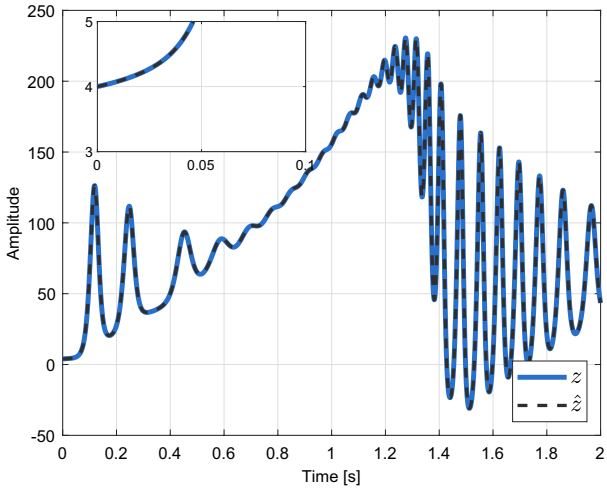
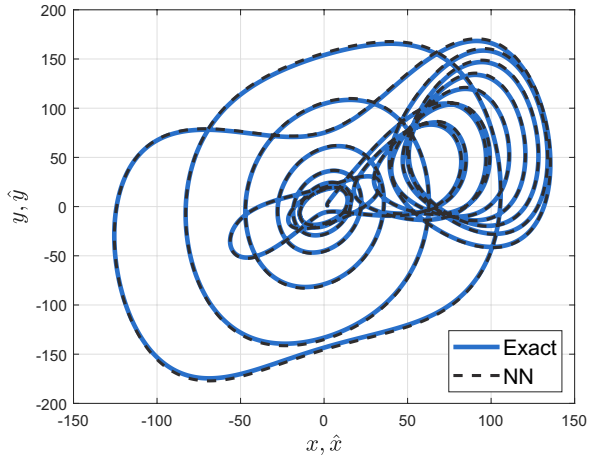


Fig. 5 The comparison between the ANN with the numerical scheme proposed to find the solution of the state z

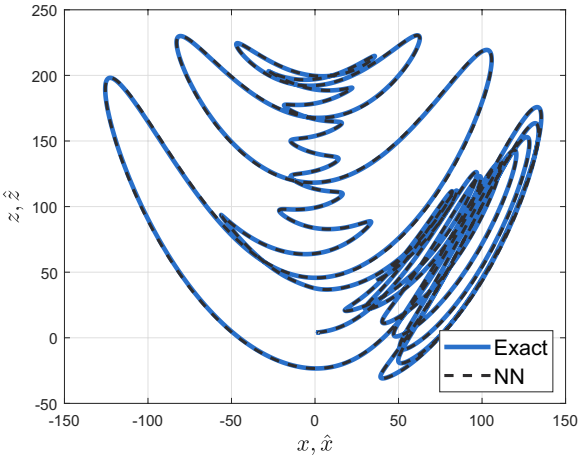
and changing through time, the order which represents the fractality is given by $\beta(t) = 0.99 + \frac{0.01}{2}(\sin(t) + 1)$. The proposed generalized solutions are shown in Eq. (30):

$$\begin{aligned} \hat{x} &= 4.87623 + t + t\hat{x}_N, \\ \hat{y} &= 4.87623 + t + t\hat{y}_N, \\ \hat{z} &= 0.1278 + t + t\hat{z}_N. \end{aligned} \tag{30}$$

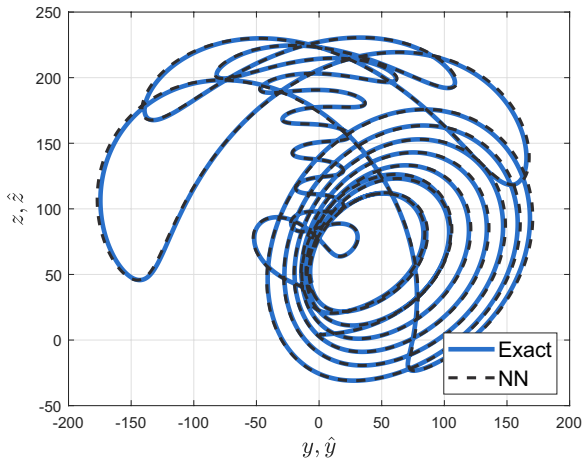
As in the previous example, it was used the brute force algorithm with the purpose of find the optimal regression values. The parameters found are shown in Table 1. In Table 1, it is shown a column n_p that represents the number of parameters that makes each one of the neural networks proposed. The equations for the neural networks used are shown in (31).



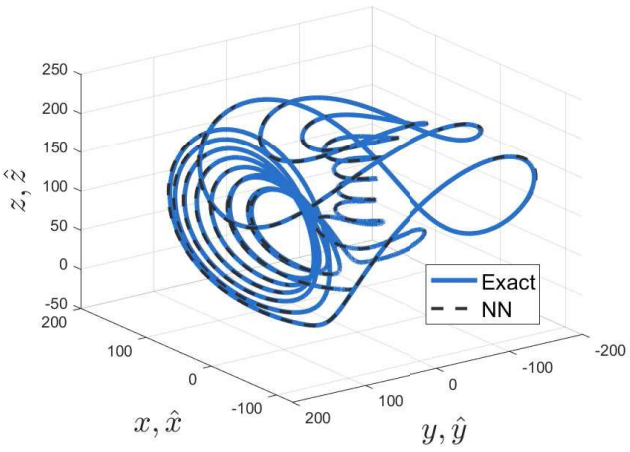
(a) Behavior of the states x and y



(b) Behavior of the states x and z.



(c) Behavior of the states y and z.



(d) Chaotic behavior of all the states.

Fig. 6 Chaotic behavior of all the states against the others

Table 1 Efficiency and parameters of the ANN

	$\Psi_{\hat{x},\hat{y},\hat{z}}$	Ψ_t	n_{p}	FIT (%)
$\hat{x}$	2	9	16	99.7783
$\hat{y}$	2	4	11	99.8915
$\hat{z}$	5	9	19	99.6847

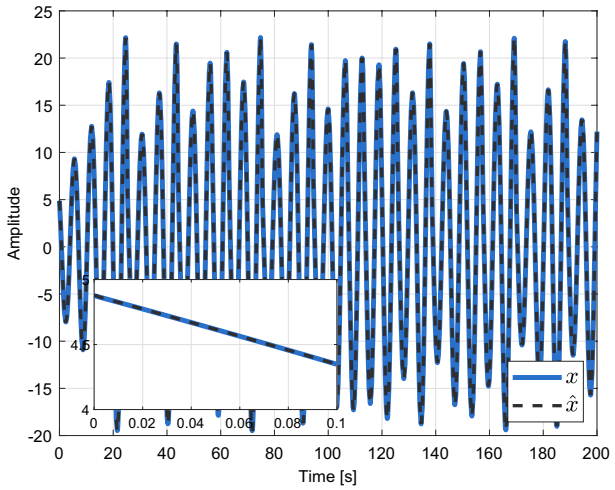


Fig. 7 Comparison between the ANN with the numerical scheme proposed to find the solution of the state x

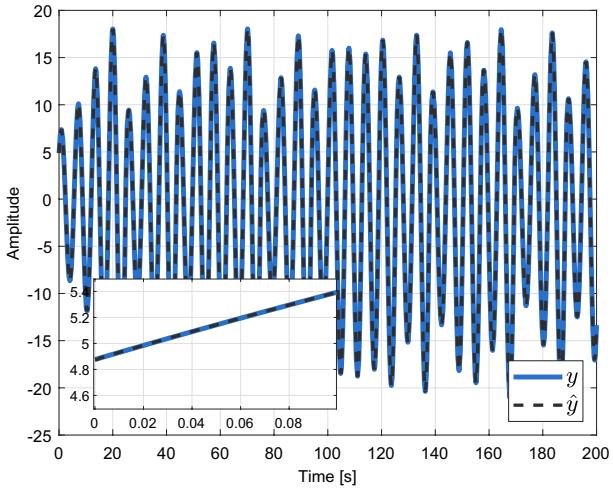


Fig. 8 Comparison between the ANN with the numerical scheme proposed to find the solution of the state y

conditions. In Fig. 10, are shown the different chaotic behaviors of each of the status of the Rössler chaotic system.

As the previous example, a test was made to measure the computational cost. Unlike the previous example, the repetitions were duplicated to get a more precise measure of the simulation time with $\mathrm{ST}_{\mathrm{nm}} = 0.302919\mathrm{s} \pm 0.03789$ s and $\mathrm{ST}_{\mathrm{nn}} = 0.00035561\mathrm{s} \pm 0.0001572$ s.

$$
\begin{aligned}
\hat{x}[k] &= V_t \varphi_1\left(\sum_{i=1}^{9} W_{t_i} t[k-i] + W_{\mathrm{th}}\right) + V_{\hat{x}} \varphi_1\left(\sum_{j=1}^{2} W_{\hat{x}_j} \hat{x}[k-j] + W_{\hat{x}h}\right) + V_{\mathrm{H}},\\
\hat{y}[k] &= V_t \varphi_1\left(\sum_{i=1}^{4} W_{t_i} t[k-i] + W_{\mathrm{th}}\right) + V_{\hat{y}} \varphi_1\left(\sum_{j=1}^{2} W_{\hat{y}_j} \hat{y}[k-j] + W_{\hat{y}h}\right) + V_{\mathrm{H}},\\
\hat{z}[k] &= V_t \varphi_1\left(\sum_{i=1}^{9} W_{t_i} t[k-i] + W_{\mathrm{th}}\right) + V_{\hat{z}} \varphi_1\left(\sum_{j=1}^{5} W_{\hat{z}_j} \hat{z}[k-j] + W_{\hat{z}h}\right) + V_{\mathrm{H}}.
\end{aligned}
\tag{31}
$$

Figures 7, 8, and 9 show the temporal behaviors of the given solution by the numerical scheme, and the general solution obtained with the neural networks. As in the previous examples, a zoom-in was made on the figure at $t = 0$ to highlight that the general solutions given by Eq. (31) respect the initial

Example 3 Consider the set of FFDEs that represent the hyperchaotic system of Van Der Pol:

$$
\begin{aligned}
&\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha_t,\beta_t} x\right)(t) = y, \quad x(0) = 1.5,\\
&\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha_t,\beta_t} y\right)(t) = -(\varepsilon_1 + bz)x - (\varepsilon_1 + \varepsilon_2 z)x^3 - \varepsilon_3 y + \varepsilon_4 z, \quad y(0) = 1.5,\\
&\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha_t,\beta_t} z\right)(t) = v, \quad z(0) = 1.5,\\
&\left({}^{\mathrm{FFM}}_{t_0}\mathcal{D}_T^{\alpha_t,\beta_t} v\right)(t) = -z + \varepsilon_5(1 - z^2)v + \varepsilon_6 x, \quad v(0) = 1.5,
\end{aligned}
\tag{32}
$$

where $\varepsilon_1 = 10$, $\varepsilon_2 = 3$, $\varepsilon_3 = 0.4$, $\varepsilon_4 = 70$, $\varepsilon_5 = 5$, $\varepsilon_q = 0.1$. According to the FFDE's initial conditions and the general solution proposed in Eq. (14), we obtain the neuronal solutions shown in Eq. (33). In this example, both orders were

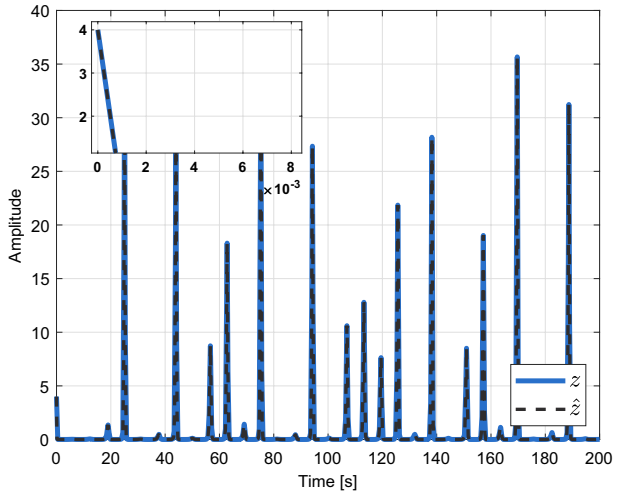


Fig. 9 Comparison between the ANN with the numerical scheme proposed to find the solution of the state z

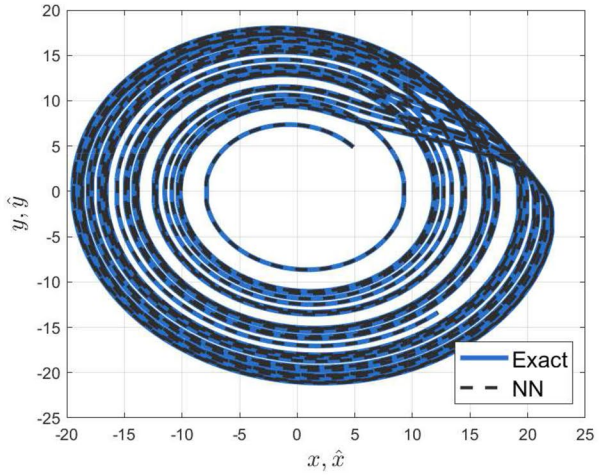
Table 2 Efficiency and parameters of the ANN

	$\Psi_{\hat{x},\hat{y},\hat{z}}$	Ψ_t	n_p	FIT (%)
$\hat{x}$	2	1	8	99.4679
$\hat{y}$	8	10	23	98.97
$\hat{z}$	5	3	13	99.85
$\hat{v}$	10	5	20	99.2

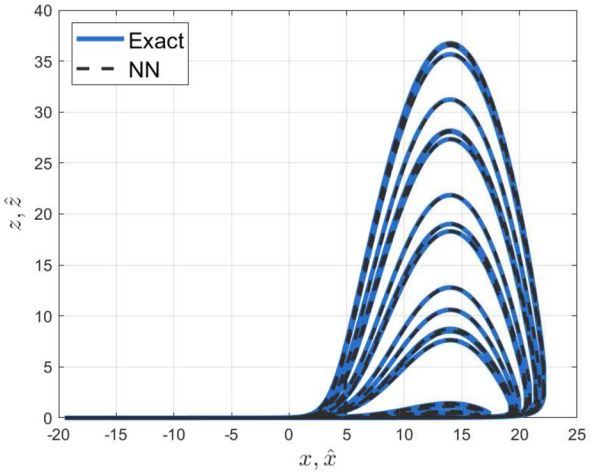
considered as variables considering that $\alpha_t = \tanh t + 5$ y $\beta(t) = 0.99 + \frac{0.01}{2}(1 + \sin(t))$:

$$\begin{aligned} \hat{x} &= 1.5 + t + t\hat{x}_N, \\ \hat{y} &= 1.5 + t + t\hat{y}_N, \\ \hat{z} &= 1.5 + t + t\hat{z}_N, \\ \hat{v} &= 1.5 + t + t\hat{v}_N. \end{aligned} \tag{33}$$

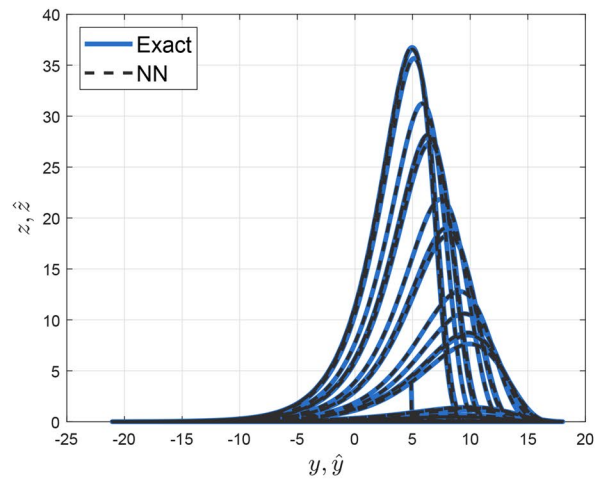
Just as we made it above, the brute force algorithm was performed to find the optimal values of the regressors. In Table 2,



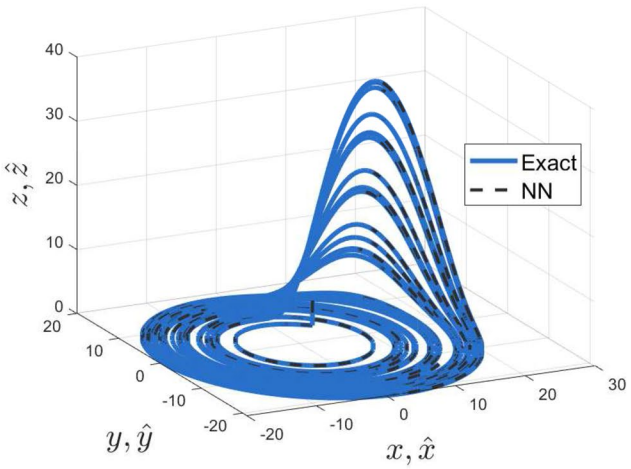
(a) Behavior of the states x and y



(b) Behavior of the states x and z.



(c) Behavior of the states y and z.



(d) Chaotic behavior of all the states.

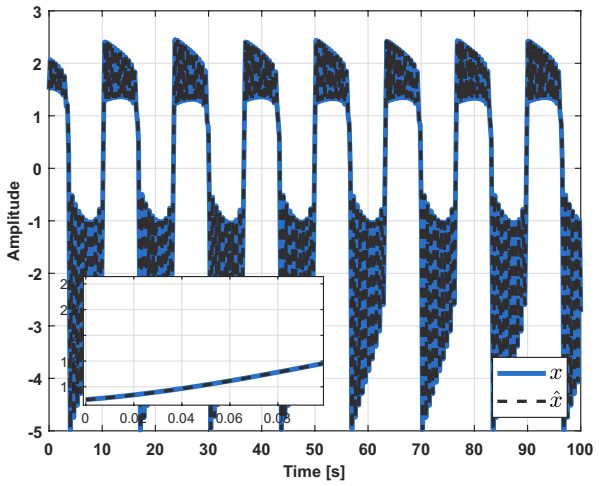
Fig. 10 Chaotic behavior of all the states against the others

it is shown the optimal values found and the FIT calculated for each one of the states of the Van Der Pol system.

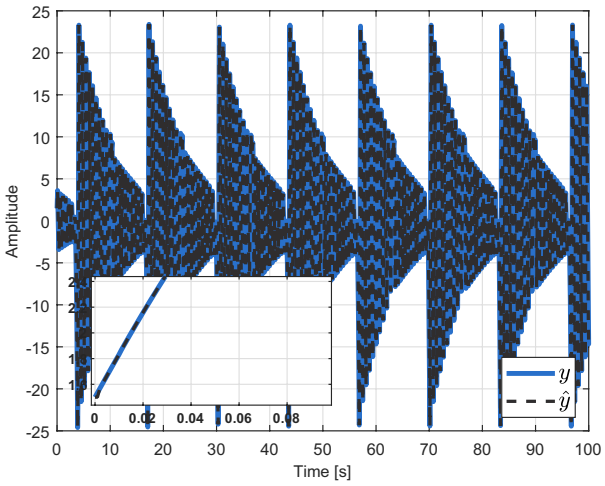
$$
\begin{aligned}
\hat{x}[k] &= V_t\varphi_1\left(\sum_{i=1}^{1} W_{t_i}t[k-i] + W_{\text{th}}\right) + V_{\hat{x}}\varphi_1\left(\sum_{j=1}^{2} W_{\hat{x}_j}\hat{x}[k-j] + W_{\hat{x}h}\right) + V_{\text{H}},\\
\hat{y}[k] &= V_t\varphi_1\left(\sum_{i=1}^{10} W_{t_i}t[k-i] + W_{\text{th}}\right) + V_{\hat{y}}\varphi_1\left(\sum_{j=1}^{8} W_{\hat{y}_j}\hat{y}[k-j] + W_{\hat{y}h}\right) + V_{\text{H}},\\
\hat{z}[k] &= V_t\varphi_1\left(\sum_{i=1}^{3} W_{t_i}t[k-i] + W_{\text{th}}\right) + V_{\hat{z}}\varphi_1\left(\sum_{j=1}^{5} W_{\hat{z}_j}\hat{z}[k-j] + W_{\hat{z}h}\right) + V_{\text{H}},\\
\hat{v}[k] &= V_t\varphi_1\left(\sum_{i=1}^{5} W_{t_i}t[k-i] + W_{\text{th}}\right) + V_{\hat{v}}\varphi_1\left(\sum_{j=1}^{10} W_{\hat{v}_j}\hat{v}[k-j] + W_{\hat{v}h}\right) + V_{\text{H}}.
\end{aligned}
\tag{34}
$$

In Eq. (34), it is shown the neural networks used to approximate the behavior of the FFDEs of the hyperchaotic system.

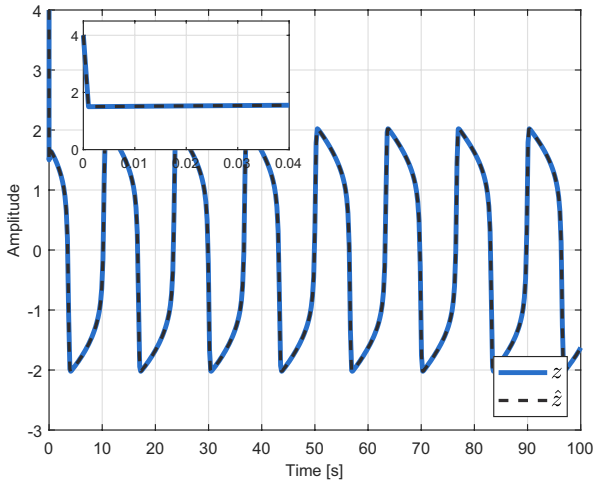
In Fig. 11, it is shown the temporal behavior of each one of the states respect to time, as in the other examples, an approach was made at the start of the simulation time to show that the neuronal solutions respect the original



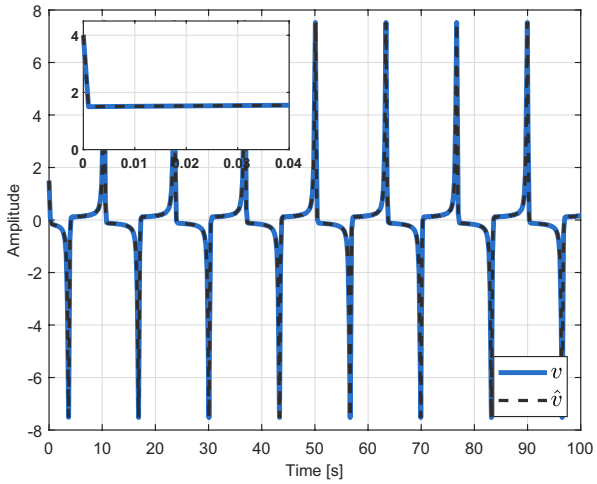
(a) Behavior of the state x and $\hat{x}$.



(b) Behavior of the state x and $\hat{y}$.

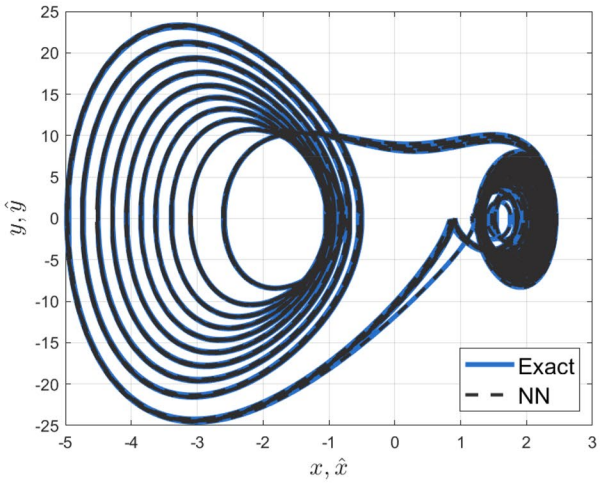


(c) Behavior of the state x and $\hat{z}$.

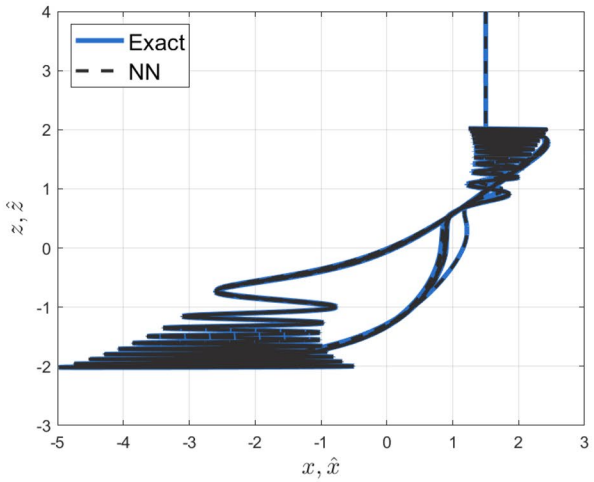


(d) Behavior of the state x and $\hat{v}$.

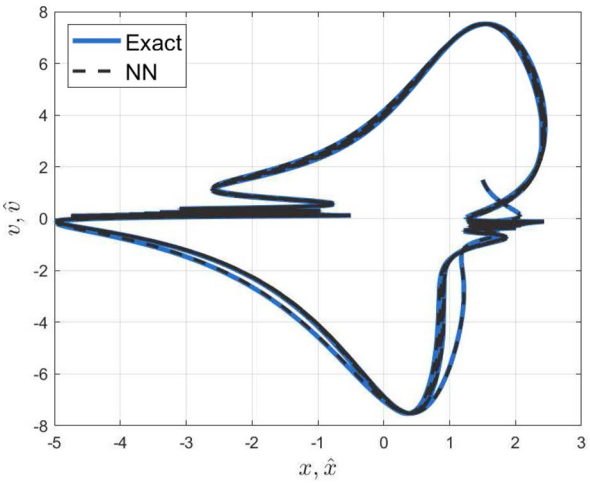
Fig. 11 Chaotic behavior of all the states against the others



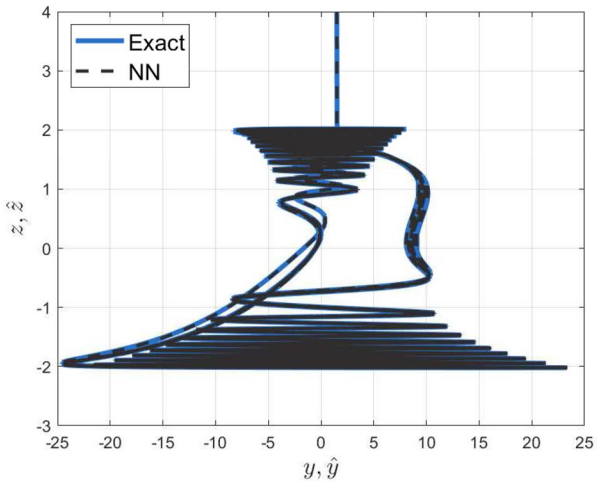
(a) Behavior of the states x and y.



(b) Behavior of the states x and z.



(c) Behavior of the states x and v.



(d) Behavior of the states y and z.

Fig. 12 Chaotic behavior of all the states against others

system's initial conditions. In Fig. 12, it is shown some phase portraits of the states to show the chaotic behavior that the system presents. Figure 13 shows the chaotic behavior of 3 out of 4 variables of Van Der Pol's hyperchaotic system just like the solutions found by the neural networks.

Lastly, the computational cost test was made for this example with 10, 000 repetitions giving as result $ST_{nm} = 1.41192s \pm 0.760691$ s and $ST_{nn} = 0.016019s \pm 0.009657$ s.

5 Conclusions

In this work, it has been demonstrated that it is possible to carry out the approximations of the behavior of fractional-fractal differential equations with fractional derivation and fractality orders variable in time. It was considered three different systems with chaotic behavior, and it

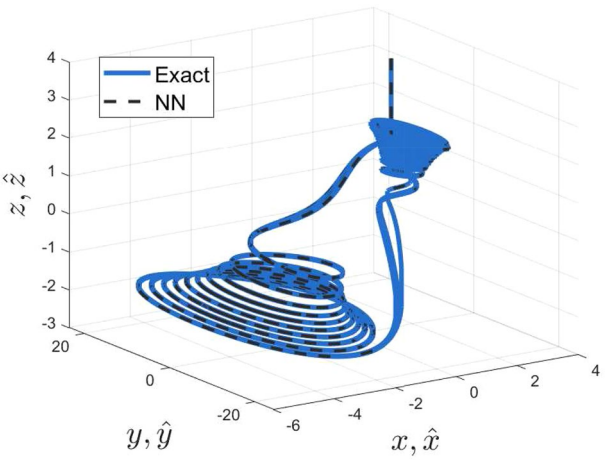


Fig. 13 Chaos behavior for all the states and its approximation with the neural networks

was demonstrated that even when the order is a function of time (either for the fractional derivative like the fractality), the neural network is able to find the optimal solution. A modification of the numerical scheme was also performed presented in [51] to consider no just one variable order else both.

The advantages of the proposed method can be seen reflected in the computational cost that it requires; there were made repetitions to the estimates made by the neuronal network and by the numerical approximation algorithm, as shown in Eq. (12). The comparison was made under the same conditions meaning, with the same computer under the same background (MATLAB 2016B). The results of the computational cost tests have shown that the neuronal network is more feasible and faster to find the solution of the FFDEs. On the other hand, the neuronal network needs in the beginning of the approximation the numerical method to be able to be trained; however, the neuronal network shows better performance than the numerical method due to the computational cost that it requires.

Acknowledgements José Francisco Gómez Aguilar acknowledges the support provided by CONACyT: Cátedras CONACyT para jóvenes investigadores 2014 and SNI-CONACyT. The Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia funded this project, under Grant no. (FP-110-42).

Author Contributions All authors have contributed equally and significantly in writing this article. All authors read and approved the final manuscript.

Compliance with ethical standards

Conflict of interests The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Bagley RL, Torvik PJ (1983) A theoretical basis for the application of fractional calculus to viscoelasticity. J Rheol 27(3):201–210
2. Meral FC, Royston TJ, Magin R (2010) Fractional calculus in viscoelasticity: an experimental study. Commun Nonlinear Sci Numer Simul 15(4):939–945
3. Su X, Xu W, Chen W, Yang H (2020) Fractional creep and relaxation models of viscoelastic materials via a non-Newtonian time-varying viscosity: physical interpretation. Mech Mater 140:103222
4. Bouharguane A, Seloula N (2020) The local discontinuous Galerkin method for convection-diffusion-fractional anti-diffusion equations. Appl Numer Math 148:61–78
5. Arqub OA, Al-Smadi M (2020) An adaptive numerical approach for the solutions of fractional advection-diffusion and dispersion equations in singular case under Riesz's derivative operator. Phys A 540:123257
6. Bai ZZ, Lu KY (2020) On regularized Hermitian splitting iteration methods for solving discretized almost isotropic spatial fractional diffusion equations. Numer Linear Algebra Appl 27(1):e2274
7. Engheta N (1996) On fractional calculus and fractional multipoles in electromagnetism. IEEE Trans Antennas Propag 44(4):554–566
8. Gómez F, Bernal J, Rosales J, Cordova T (2019) Modeling and simulation of equivalent circuits in description of biological systems—a fractional calculus approach. J Electr Bioimpedance 3(1):2–11
9. Stefański TP, Gulgowski J (2020) Signal propagation in electromagnetic media described by fractional-order models. Commun Nonlinear Sci Numer Simul 82:105029
10. Gómez-Aguilar JF, Rosales-García JJ, Bernal-Alvarado JJ, Córdova-Fraga T, Guzmán-Cabrera R (2012) Fractional mechanical oscillators. Rev Mexicana física 58(4):348–352
11. Grzesikiewicz W, Wakulicz A, Zbiciak A (2013) Non-linear problems of fractional calculus in modeling of mechanical systems. Int J Mech Sci 70:90–98
12. Jiménez F, Ober-Blöbaum S (2019) Fractional damping through restricted calculus of variations. arXiv:1905.05608
13. Coelho RM, Neto JP, Valério D, Vinga S (2020) Dynamic biochemical and cellular models of bone physiology: integrating remodeling processes, tumor growth, and therapy. The computational mechanics of bone tissue. Springer, Cham, pp 95–128
14. Yépez-Martínez H, Gómez-Aguilar JF (2020) Laplace variational iteration method for modified fractional derivatives with nonsingular kernel. J Archive 2020:6
15. Goswami P, Alqahtani RT (2016) On the solution of local fractional differential equations using local fractional Laplace variational iteration method. Math Probl Eng 2016:2
16. Wu GC, Baleanu D (2013) Variational iteration method for fractional calculus-a universal approach by Laplace transform. Adv Differ Equ 2013(1):1–9
17. Inokuti M, Sekine H, Mura T (1978) General use of the Lagrange multiplier in nonlinear mathematical physics. Variation Method Mech Solids 33(5):156–162
18. Morales-Delgado VF, Gómez-Aguilar JF, Yépez-Martínez H, Baleanu D, Escobar-Jimenez RF, Olivares-Peregrino VH (2016) Laplace homotopy analysis method for solving linear partial differential equations using a fractional derivative with and without kernel singular. Adv Differ Equ 2016(1):1–17
19. Ghorbani A, Saberi-Nadjafi J (2008) Exact solutions for nonlinear integral equations by a modified homotopy perturbation method. Comput Math Appl 56(4):1032–1039
20. Filobello-Nino U, Vázquez-Leal H, Khan Y, Pereyra-Díaz D, Pérez-Sesma A, Díaz-Sánchez A (2014) Modified nonlinearities distribution homotopy perturbation method as a tool to find power series solutions to ordinary differential equations. Nova Scientia 6(12):13–38
21. Cuce E, Cuce PM (2015) A successful application of homotopy perturbation method for efficiency and effectiveness assessment of longitudinal porous fins. Energy Convers Manage 93:92–99
22. Saad KM, AL-Shareef EH, Alomari AK, Baleanu D, Gómez-Aguilar JF (2020) On exact solutions for time-fractional Korteweg-de Vries and Korteweg-de Vries-Burger's equations using homotopy analysis transform method. Chin J Phys 63:149–162
23. Murio DA (2009) Stable numerical evaluation of Grünwald-Letnikov fractional derivatives applied to a fractional IHCP. Inverse Probl Sci Eng 17(2):229–243
24. Zhang S (2009) Monotone iterative method for initial value problem involving Riemann-Liouville fractional derivatives. Nonlinear Anal Theory Methods Appl 71(5–6):2087–2093
25. Baleanu D, Agrawal OP (2006) Fractional Hamilton formalism within Caputo's derivative. Czech J Phys 56(10–11):1087–1092
26. Yang XJ, Srivastava HM, Machado JA (2015) A new fractional derivative without singular kernel: application to the modelling of the steady heat flow. arXiv:1601.01623

27. Atangana A, Baleanu D (2016) New fractional derivatives with nonlocal and non-singular Kernel. Theory and application to heat transfer model. Therm Sci 20(2):763–769
28. Yang XJ (2016) Fractional derivatives of constant and variable orders applied to anomalous relaxation models in heat-transfer problems. arXiv:1612.03202
29. Ghanbari B, Gómez-Aguilar JF (2018) Modeling the dynamics of nutrient-phytoplankton-zooplankton system with variable-order fractional derivatives. Chaos Solit Fract 116:114–120
30. Gómez-Aguilar JF (2018) Analytical and numerical solutions of a nonlinear alcoholism model via variable-order fractional differential equations. Phys A 494:52–75
31. Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals Syst 2(4):303–314
32. Sahnoun MA, Ugalde HMR, Carmona JC, Gomand J (2013) Maximum power point tracking using P&O control optimized by a neural network approach: a good compromise between accuracy and complexity. Energy Procedia 42:650–659
33. Corbier C, Ugalde HMR (2016) Low-order control-oriented modeling of piezoelectric actuator using Huberian function with low threshold: pseudolinear and neural network models. Nonlinear Dyn 85(2):923–940
34. Romero-Ugalde HM, Corbier C (2016) Robust estimation of balanced simplicity-accuracy neural networks-based models. J Dyn Syst Meas Control 138:5
35. Ugalde HMR, Carmona JC, Alvarado VM, Reyes-Reyes J (2013) Neural network design and model reduction approach for black box nonlinear system identification with reduced number of parameters. Neurocomputing 101:170–180
36. Zúñiga-Aguilar CJ, Gómez-Aguilar JF, Alvarado-Martínez VM, Romero-Ugalde HM (2020) Fractional order neural networks for system identification. Chaos Solit Fract 130:109444
37. Arora P, Srivastava S, Singhal S (2020) Analysis of gait flow image and gait Gaussian image using extension neural network for gait recognition. In: Deep learning and neural networks: concepts, methodologies, tools, and applications (pp 429–449). IGI Global
38. Abduh Z, Nehary EA, Wahed MA, Kadah YM (2019) Classification of heart sounds using fractional Fourier transform based mel-frequency spectral coefficients and stacked autoencoder deep neural network. J Med Imaging Health Informatics 9(1):1–8
39. Liu H, Pan Y, Cao J, Wang H, Zhou Y (2020) Adaptive neural network Backstepping control of fractional-order nonlinear systems with actuator faults. IEEE Trans Neural Netw Learn Syst 2020:1
40. Sharafian A, Ghasemi R (2019) Fractional neural observer design for a class of nonlinear fractional chaotic systems. Neural Comput Appl 31(4):1201–1213
41. Mani P, Rajan R, Shanmugam L, Joo YH (2019) Adaptive control for fractional order induced chaotic fuzzy cellular neural networks and its application to image encryption. Inf Sci 491:74–89
42. Zúñiga-Aguilar CJ, Romero-Ugalde HM, Gómez-Aguilar JF, Escobar-Jiménez RF, Valtierra-Rodríguez M (2017) Solving fractional differential equations of variable-order involving operators with Mittag-Leffler kernel using artificial neural networks. Chaos Solit Fract 103:382–403
43. Zuniga-Aguilar CJ, Coronel-Escamilla A, Gómez-Aguilar JF, Alvarado-Martínez VM, Romero-Ugalde HM (2018) New numerical approximation for solving fractional delay differential equations of variable order using artificial neural networks. Eur Phys J Plus 133(2):75
44. Pakdaman M, Ahmadian A, Effati S, Salahshour S, Baleanu D (2017) Solving differential equations of fractional order using an optimization technique based on training artificial neural network. Appl Math Comput 293:81–95
45. Effati S, Pakdaman M, Ranjbar M (2011) A new fuzzy neural network model for solving fuzzy linear programming problems and its applications. Neural Comput Appl 20(8):1285–1294
46. Effati S, Pakdaman M (2010) Artificial neural network approach for solving fuzzy differential equations. Inf Sci 180(8):1434–1457
47. Sabouri J, Effati S, Pakdaman M (2017) A neural network approach for solving a class of fractional optimal control problems. Neural Process Lett 45(1):59–74
48. Raja MAZ (2014) Solution of the one-dimensional Bratu equation arising in the fuel ignition model using ANN optimised with PSO and SQP. Connect Sci 26(3):195–214
49. Raja MAZ, Khan JA, Qureshi IM (2011) Swarm Intelligent optimized neural networks for solving fractional differential equations. Int J Innovat Comput Inf Control 7(11):6301–6318
50. Yavari M, Nazemi A (2019) An efficient numerical scheme for solving fractional infinite-horizon optimal control problems. ISA Trans 94:108–118
51. Atangana A, Shafiq A (2019) Differential and integral operators with constant fractional order and variable fractional dimension. Chaos Solit Fract 127:226–243
52. Tirandaz H, Hajipour A (2017) Adaptive synchronization and anti-synchronization of TSUCS and Lü unified chaotic systems with unknown parameters. Optik 130:543–549