We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter 5

Numerical Simulation Using Artificial Neural Network on Fractional Differential Equations

Najeeb Alam Khan, Amber Shaikh,
Faqiha Sultan and Asmat Ara

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/64151

Abstract

This chapter offers a numerical simulation of fractional differential equations by utilizing Chebyshev-simulated annealing neural network (ChSANN) and Legendre-simulated annealing neural network (LSANN). The use of Chebyshev and Legendre polynomials with simulated annealing reduces the mean square error and leads to more accurate numerical approximation. The comparison of proposed methods with previous methods confirms the accuracy of ChSANN and LSANN.

Keywords: neural network, fractional Riccati, Legendre polynomial, Chebyshev polynomial, simulated annealing

1. Introduction

During the last few decades, fractional calculus has gained massive attention of physicists and mathematicians because of its numerous interdisciplinary applications. Many recent researches are ended up demonstrating the significance of fractional-order differential equations as valuable instruments to model several physical phenomena such as the nonlinear oscillation of earthquake and the fluid dynamics traffic can be elegantly modelled with fractional derivatives [1, 2]. Various physical processes show fractional-order behaviour that might change with respect to time or space. The adoption of fractional calculus concepts is well known in many scientific areas such as physics, diffusion and wave propagation, heat

INTECH
open science | open minds

© 2016 The Author(s). Licensee InTech. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

transfer, viscoelasticity and damping, electronics, robotics, electromagnetism, signal processing, telecommunications, control systems, traffic systems, system identification, chaos and fractals, biology, genetic algorithms, filtration, modelling and identification, chemistry, irreversibility, as well as economy and finance [3–5].

Modelling of different physical phenomena gave rise to a special differential equation known as Riccati differential equation that was named after an Italian mathematician Count Jacopo Francesco Riccati. Due to many applications of fractional Riccati differential equations such as in stochastic controls and pattern formation, many researchers studied it to get the exact or approximate solutions. Such as fractional variational iteration method was applied in [6] to give an approximate analytical solution of non-linear fractional Riccati differential equation. His modified homotopy perturbation method (MHPM) was used on quadratic Riccati differential equation of fractional order [7]. The results of fractional Riccati differential equation were also obtained on the basis of Taylor collocation method [8]. Fractional Riccati differential equations were solved by means of variational iteration method and homotopy perturbation Pade technique [9, 10], and the numerical results were attained by using Chebyshev finite difference method [11]. Adomian decomposition method was presented for fractional Riccati differential equation [12], the problem was described by means of Bernstein collocation method [13], and enhanced homotopy perturbation method (EHPM) was used to study this problem [14]. Recently, artificial neural network and sequential quadratic programming have been utilized to obtain the solution of Riccati differential equation [15]. The problem was also explained by Legendre wavelet operational matrix method [16] and the results of fractional Riccati differential equation by new homotopy perturbation method (NHPM) [17] were obtained.

In recent years, artificial neural network (ANN) is one of the methods that are attaining massive attention of researchers in the area of mathematics as well as in different physical sciences. The concept of ANN started to develop in 1943 when a neurophysiologist and a young mathematician [18] gave the idea on working of a neuron with the help of an electric circuit. Later, a book [19] was written to clarify the working of neurons then in 1949. Bernard Widrow and Marcian Hoff developed a model MEDALIN that was used to study the first real-world problem of neural network. Researchers continued to study the single-layered neural network, but in 1975, the concept of multilayer perceptron (MLP) was introduced, which was computationally exhaustive due to multilayer architecture. The excessive training time and high computational complexity of MLP gave rise to functional neural network by which the complexity of multilayers was overcome by introducing variable functions [20]. Functional link neural network has been implemented to several problems such as modified functional link neural network for denoising of image [21], active control of non-linear noise processes through functional link neural network [22], and the problem of channel equalization in a digital communication system was solved by functional link neural network [23]. Due to less computational effort with easy to implement procedure, functional link neural network was also implemented to solve differential equations [24, 25].

http://dx.doi.org/10.5772/64151

2. Definitions and preliminaries

The Riemann-Liouville, Grünwald-Letnikov and Caputo definitions of fractional derivatives of order $\alpha > 0$ are used more frequently among several definitions of fractional derivatives and integrals, but in this chapter, the Caputo definition will be used for working out the fractional derivative in a subsequent procedure. The definitions of commonly used fractional differential operators are discussed in the study of Sontakke and Shaikh [26].

Definition 1: The Riemann-Liouville fractional derivative operator can be defined as follows:

$$D^{\alpha} g(x) = \frac{1}{\Gamma(\xi - \alpha)} \frac{d^{\xi}}{dx^{\xi}} \int_{a}^{x} \frac{g(\beta)}{(x-\beta)^{\alpha+1-\xi}} d\beta, \quad \xi\text{-}1 < \alpha < \xi$$

where here $\alpha > 0,\ x > a,\ \alpha,\ a$ and $x \in R$.

Definition 2: The definition of fractional differential operator was introduced by Caputo in late 1960s that can be defined as follows [27]:

$$D_{*}^{\alpha} g(x) = \frac{1}{\Gamma(\xi - \alpha)} \int_{a}^{x} \frac{g^{(\xi)}(\beta)}{(x-\beta)^{\alpha+1-\xi}} d\beta, \quad \xi\text{-}1 < \alpha < \xi$$

where here $\alpha > 0,\ x > a,\ \alpha,\ a$ and $x \in R$.

The Caputo fractional derivative satisfies the important attribute of being zero when applied to a constant. In addition, it is attained by computing an ordinary derivative followed by the fractional integral, while the Riemann-Liouville is acquired in the contrary order.

3. ChSANN and LSANN

3.1. Methodology

The functional mapping of (LSANN) and (ChSANN) is shown in **Figure 1** demonstrating the structure of both methods, but for the convenience of the reader, a stepwise explanation of both the methods is also presented.

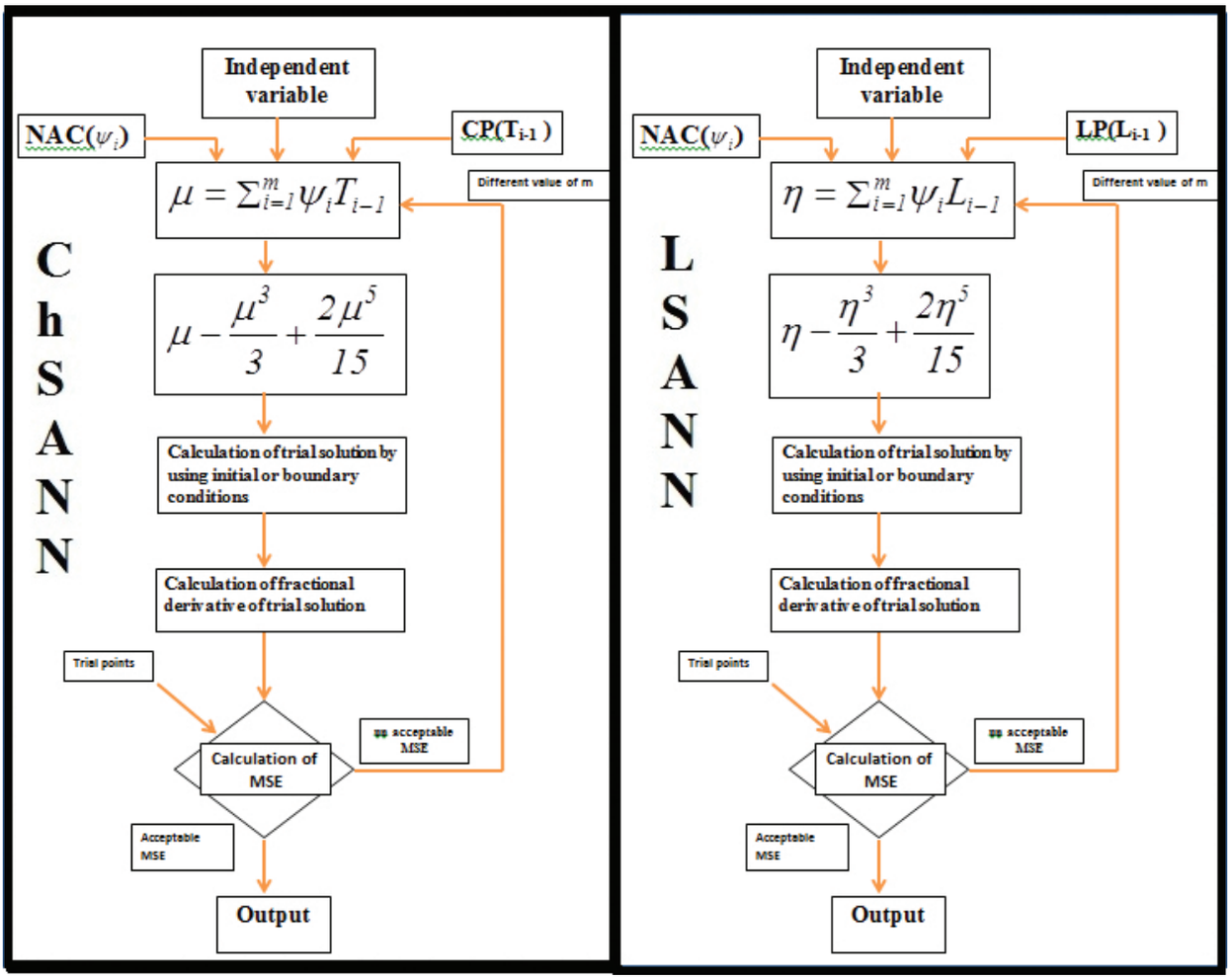


Figure 1. Model diagram of ChSANN and LSANN.

The combined steps for both the methods are explained because of the structural similarity in them, except the polynomial basis that affects the accuracy of the results.

Step 1: The summation of the product of network adaptive coefficients (NAC) and Chebyshev or Legendre polynomials is calculated for the independent variable of fractional differential equation for an arbitrary value of m as shown in **Figure 1**.

Step 2: The activation of μ or η will be performed by the first three terms of the series of tangent hyperbolic function $\tanh(\bullet)$, these terms have been mentioned in **Figure 1**.

Step 3: The trial solution will be calculated by using initial conditions as in the study of Lagaris and Fotiadis [28].

Step 4: Required derivatives of the trial solution will be calculated.

Step 5: The optimization of mean square error (MSE) or learning of NAC will be executed by the thermal minimization methodology known as simulated annealing. The equation used to calculate MSE would be discussed in next section. Before optimization, the independent variable will be discretized by an array of trial points.

http://dx.doi.org/10.5772/64151

Step 6: If the value of MSE is in an acceptable range, then the values of trial points and NAC will be replaced in trial solution to get the output. On the other hand, the procedure will be repeated from step 1 with a different value of m.

3.2. Implementation on fractional Riccati differential equation

In this section, the ChSANN and LSANN are employed for the fractional Riccati differential equation of the type:

$$\frac{d^{\alpha}y(t)}{dt^{\alpha}} = f\left(t, y\right), \quad y(0) = A\ , \ \ 0 < \alpha \le 1 \tag{1}$$

For implementing both methodologies, Eq. (1) can be written in the following form:

$$\nabla^{\alpha} y_{tr}(t,\psi) - F(t, y_{tr}(t,\psi)) = 0, \quad t \in \left[0,1\right] \tag{2}$$

$y_{tr}(t, \psi)$ can be defined as trial solution, where ψ is defined as NAC, generally known as weights, and ∇ is defined as differential operator. Trial solution will be obtained by applying Taylor series on the activation of μ by using the initial conditions, while μ being the sum of the product of network adaptive coefficients and Chebyshev polynomials. For obtaining the trial solution of LSANN, the above procedure will be pursued, but η will be calculated in spite of μ, that is the sum of the product of NAC and Legendre polynomials. Here, $\tanh(\bullet)$is used as activation function, but for fractional derivative based on Caputo sense, first three terms of the Taylor series of $\tanh(\bullet)$ are considered that are given for ChSANN as follows:

$$N = \mu - \frac{\mu^3}{3} + \frac{2\mu^5}{15} \tag{3}$$

while for ChSANN, μ can be defined as follows:

$$\mu = \sum_{i=1}^{m} \psi_i T_{i-1} \tag{4}$$

where here T_{i-1} are Chebyshev polynomials with the following recursive formula:

$$T_{m+1}\left(x\right) = 2xT_m\left(x\right) - T_{m-1}\left(x\right),\ m \ge 2 \tag{5}$$

while hile $T_0(x)=1$ and $T_1(x)=x$.

For LSANN, the activation function and η can be defined as follows:

$$N = \eta - \frac{\eta^3}{3} + \frac{2\eta^5}{15} \tag{6}$$

$$\eta = \sum_{i=1}^{m} \psi_i L_{i-1} \tag{7}$$

whereas hereas L_{i-1} are the Legendre polynomials with the recursive formula:

$$L_{m+1} = \frac{1}{(m+1)}\ (2m+1)\ x\ L_m(x) - \frac{1}{(m+1)} m\ L_{m-1}(x),\ \ m \geq 2 \tag{8}$$

where here $L_0(x)=1$ and $L_1(x)=x$, and value of m is adjustable to reach the utmost accuracy. For Eq. (1), the trial solution can be written as defined in the study of Lagaris and Fotiadis [28], but N will be used according to the method.

$$y_{tr}(t,\psi) = A + t\ N \tag{9}$$

Trial solution can be written in expanded form for ChSANN at $m=2$ as follows:

$$y_{tr}(t,\psi) = A + t\left(\psi_1 + t\psi_2 - \frac{1}{3}\left(\psi_1 + t\psi_2\right)^3 + \frac{2}{15}\left(\psi_1 + t\psi_2\right)^5\right) \tag{10}$$

Fractional derivative in Caputo sense of Eq. (10) is as follows:

$$\begin{aligned}\nabla^{\alpha} y_{tr}(t,\psi) = {} & \frac{\Gamma 2}{\Gamma(2-\alpha)} t^{1-\alpha}\left(\psi_1 - \frac{\psi_1^3}{3} + \frac{2\psi_1^5}{15}\right) + \left(\frac{2}{3}\right)\frac{\Gamma 6}{\Gamma(7-\alpha)} t^{5-\alpha}\left(\psi_1\psi_2^4\right) \\ & + \left(\frac{2}{15}\right)\frac{\Gamma 7}{\Gamma(7-\alpha)} t^{6-\alpha}\left(\psi_2^5\right) + \frac{\Gamma 3}{\Gamma(3-\alpha)}\ \ t^{2-\alpha}\left(\psi_2 - \psi_1^2\psi_2 + \frac{2}{3}\psi_1^4\psi_2\right) \\ & + \frac{\Gamma 4}{\Gamma(4-\alpha)} t^{3-\alpha}\left(\frac{4}{3}\psi_1^3\psi_2^2 - \psi_1\psi_2^2\right) + \frac{\Gamma 5}{\Gamma(5-\alpha)} t^{4-\alpha}\left(\frac{4}{3}\psi_1^2\psi_2^3 - \frac{\psi_2^3}{3}\right)\end{aligned} \tag{11}$$

The mean square error (MSE) of the Eq. (1) will be calculated from the following:

$$MSE(\psi_i) = \sum_{j=1}^{n} \frac{1}{n}\left(\nabla^{\alpha} y_{tr}\left(t_j,\psi_i\right) - F\left(t_j, y_{tr}\left(t_j,\psi_i\right)\right)\right)^2, \quad t \in [0,1] \tag{12}$$

http://dx.doi.org/10.5772/64151

whereas here as n can be defined as number of trial points. The learning of NAC will be performed from Eq. (10) by minimizing the MSE to the lowest possible acceptable minimum value. The thermal minimization methodology and simulated annealing is applied here for the learning of NAC. The process of simulated annealing can be described as a physical model of annealing, where a metal object is first heated and then slowly cooled down to minimize the system energy. Here, we have implemented the procedure by Mathematica 10, but the interested readers can learn the details of simulated annealing from the study of Ledesma et al. [29].

Example 1:

Consider the following Riccati differential equation with initial condition as:

$$\frac{d^{\alpha}y(t)}{dt^{\alpha}} + y^{2}(t) - 1 = 0, \quad y(0) = 0\ ,\ 0 < \alpha \le 1$$

The exact solution for $\alpha = 1$ is given by the following:

$$y(t) = \frac{e^{2t} - 1}{e^{2t} + 1}$$

The above fractional Riccati differential equation is solved by implementing the ChSANN and LSANN algorithms for various values of α and the results are compared with several methods to exhibit the strength of proposed neural network algorithms. The ChSANN and LSANN methods are employed on the above equation for $\alpha = 1$ with 20 equidistant training points and 6 NAC and attained the mean square error up to $5.501631 \times 10^{-9}$and 1.21928×10^{-9} for ChSANN and LSANN, respectively. **Figure 2** shows the combined results of ChSANN for different

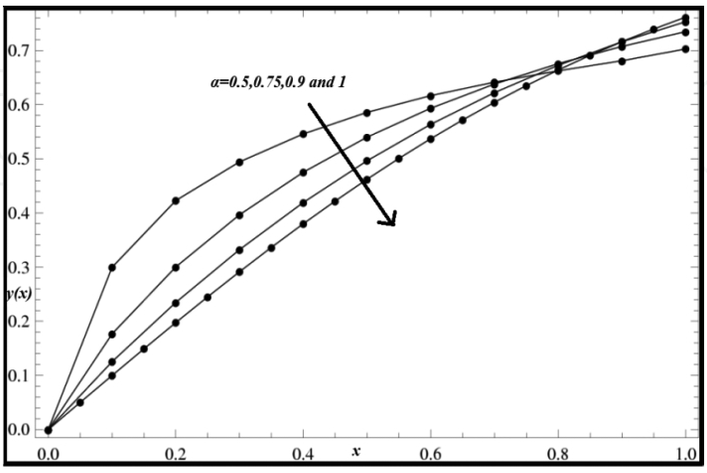


Figure 2. ChSANN results at different values of $\alpha = 1$.

values of α. **Table 1** depicts the comparison of results obtained from both the methods with exact solution and the absolute error values for both the methods. Absolute error (AE) values for ChSANN and LSANN can be viewed in **Table 1** but can be better visualized in **Figure 3**. Implementation of ChSANN and LSANN for $\alpha=0.75$ with 10 equidistant training points and 6 NAC on the above equation gave the mean square error up to 1.66032×10^{-7} for ChSANN and 4.8089×10^{-7} for LSANN. **Table 2** shows the numerical comparison for $\alpha=0.75$ with 10 equidistant training points of ChSANN and LSANN with the methods in [13, 14], while **Tables 3** and **4** demonstrate the numerical comparison of the proposed methods with the methods in [7, 13, 14] for $\alpha=0.5$and $\alpha=0.9$ correspondingly. Numerical values of ChSANN for $\alpha=1$ at $t=1$ are presented in **Table 5**.

x	Exact	ChSANN	LSANN	AE of ChSANN	AE of LSANN
0.05	0.049884	0.0499572	0.0499441	1.20167×10^{-6}	1.49267×10^{-5}
0.10	0.099668	0.0996676	0.0996552	4.32269×10^{-6}	1.27675×10^{-5}
0.15	0.148885	0.148884	0.148876	1.13944×10^{-6}	9.07723×10^{-6}
0.20	0.197375	0.197372	0.197367	2.91448×10^{-6}	7.92946×10^{-6}
0.25	0.244919	0.244915	0.244909	4.13612×10^{-6}	9.23796×10^{-6}
0.30	0.291313	0.291309	0.291301	3.63699×10^{-6}	1.12070×10^{-5}
0.35	0.336376	0.336374	0.336363	1.41863×10^{-6}	1.22430×10^{-5}
0.40	0.379949	0.379949	0.379937	1.42762×10^{-6}	1.17881×10^{-5}
0.45	0.421899	0.421902	0.421889	3.33093×10^{-6}	1.03088×10^{-5}
0.50	0.462117	0.462117	0.462108	3.05535×10^{-6}	8.76900×10^{-6}
0.55	0.500520	0.500521	0.500512	3.81254×10^{-6}	7.95700×10^{-6}
0.60	0.537055	0.537046	0.537042	3.62984×10^{-6}	8.00729×10^{-6}
0.65	0.571670	0.571663	0.571662	6.95133×10^{-6}	8.33268×10^{-6}
0.70	0.604368	0.604360	0.604360	7.47479×10^{-6}	8.04113×10^{-6}
0.75	0.635149	0.635145	0.635142	4.22458×10^{-6}	6.62579×10^{-6}
0.80	0.664037	0.664038	0.664032	1.57001×10^{-6}	4.48924×10^{-6}
0.85	0.691069	0.691076	0.691067	6.23989×10^{-6}	2.58549×10^{-6}
0.90	0.716298	0.716303	0.716298	5.13615×10^{-6}	2.66632×10^{-7}
0.95	0.739783	0.739780	0.739793	3.29187×10^{-6}	9.67625×10^{-6}
1.00	0.761594	0.761584	0.761644	9.94216×10^{-6}	4.97369×10^{-5}

Table 1. Numerical comparisons of ChSANN and LSANN values with exact values for fractional Riccati differential equation.

http://dx.doi.org/10.5772/64151

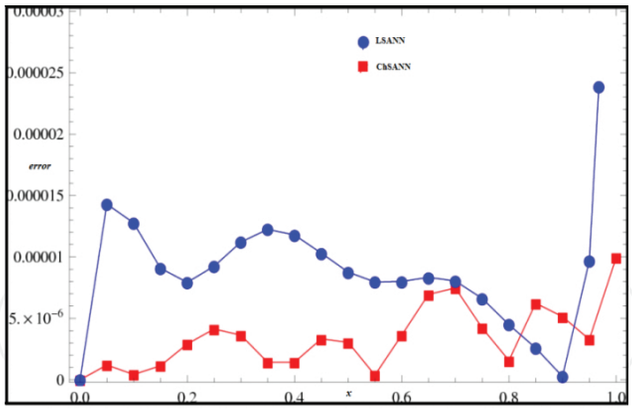


Figure 3. Absolute error of ChSANN and LSANN at $\alpha = 1$ for test example 1.

x	ChSANN	LSANN	IABMM [14]	EHPM [14]	MHPM [7]	Bernstein [13]
0	0	0	0	0	0	0
0.2	0.30018	0.29937	0.3117	0.3214	0.3138	0.30997
0.4	0.47512	0.47486	0.4885	0.5077	0.4929	0.48163
0.6	0.59334	0.59320	0.6045	0.6259	0.5974	0.59778
0.8	0.67572	0.67571	0.6880	0.7028	0.6604	0.67884
1.0	0.73748	0.73430	0.7478	0.7542	0.7183	0.73683

Table 2. Numerical comparison for $\alpha = 0.75$.

x	ChSANN	LSANN	MHPM [7]
0	0	0	0
0.1	0.299658	0.299416	0.273875
0.2	0.422837	0.422808	0.454125
0.3	0.494204	0.494145	0.573932
0.4	0.545856	0.545773	0.644422
0.5	0.585660	0.585619	0.674137
0.6	0.616648	0.616647	0.671987
0.7	0.641558	0.641543	0.648003
0.8	0.662486	0.662452	0.613306
0.9	0.681101	0.681237	0.579641
1.0	0.702813	0.703857	0.558557

Table 3. Numerical comparison for $\alpha = 0.5$.

x	ChSANN	LSANN	IABMM [14]	EHPM [14]	MHPM [7]	Bernstein [13]
0	0	0				
0.2	0.234602	0.236053	0.2393	0.2647	0.2391	0.23878
0.4	0.419229	0.419898	0.4234	0.4591	0.4229	0.42258
0.6	0.563627	0.564474	0.5679	0.6031	0.5653	0.56617
0.8	0.672722	0.673241	0.6774	0.7068	0.6740	0.67462
1.0	0.753188	0.755002	0.7584	0.7806	0.7569	0.75458

Table 4. Numerical comparison for $\alpha = 0.9$.

No of NAC	No of training points	Mean square error	$y(t)$	Absolute error
4	10	9.7679×10^{-5}	0.760078	1.51570×10^{-3}
5	20	2.3504×10^{-7}	0.761644	5.02121×10^{-5}
6	20	5.5016×10^{-9}	0.761584	9.94216×10^{-6}

Table 5. Numerical values of ChSANN at $t = 1$ and for $\alpha = 1$.

Example 2:

Consider the nonlinear Riccati differential equation along with the following initial condition:

$$\frac{d^{\alpha} y(t)}{dt^{\alpha}} + y^2(t) - 2y(t) - 1 = 0, \quad y(0) = 0\ , \quad 0 < \alpha \le 1$$

The exact solution when $\alpha = 1$ is given by [7]:

$$y(t) = 1 + \sqrt{2}\ \tanh\left(\sqrt{2}\ t + \frac{1}{2}\log\left(\frac{\sqrt{2}\ -1}{\sqrt{2}\ +1}\right)\right)$$

ChSANN and LSANN algorithms are executed on the above test experiment with 6NAC, $\alpha = 1$ and 20 equidistant points that gave the mean square error up to $1.6127 \times 10^{-7}$and 4.68641×10^{-6} for ChSANN and LSANN, respectively. **Table 6** shows the absolute errors and the numerical comparison with exact values for both the methods, while graphical comparison can be better envisioned through **Figure 4**. **Tables 7** and **8** display the numerical comparison of the proposed methods with the results obtained in [7] for $\alpha = 0.75$and [13] for $\alpha = 0.9$, respectively, whereas the mean square error, number of training points, and NAC for different values of α are presented in **Table 9**. The effects on accuracy of results with variable NAC and training points can be understood through **Table 10**.

http://dx.doi.org/10.5772/64151

x	ChSANN	LSANN	Exact	AE of ChSANN
0.05	0.052620	0.053034	0.052539	8.0725×10^{-5}
0.10	0.110385	0.110899	0.110295	8.9742×10^{-5}
0.15	0.173488	0.173936	0.173419	6.8944×10^{-5}
0.20	0.242027	0.242359	0.241977	5.0509×10^{-5}
0.25	0.315977	0.316226	0.315926	5.0577×10^{-5}
0.30	0.395175	0.395419	0.395105	6.9946×10^{-5}
0.35	0.479313	0.479634	0.479214	9.9086×10^{-5}
0.40	0.567937	0.568390	0.567812	1.2488×10^{-4}
0.45	0.660455	0.661048	0.660318	1.3690×10^{-4}
0.50	0.756146	0.756840	0.756014	1.3149×10^{-4}
0.55	0.854184	0.854907	0.854071	1.1284×10^{-4}
0.60	0.953657	0.954329	0.953566	9.0888×10^{-5}
0.65	1.053601	1.054165	1.053524	7.6914×10^{-5}
0.70	1.153027	1.153472	1.152949	7.8337×10^{-5}
0.75	1.250962	1.251332	1.250867	9.4552×10^{-5}
0.80	1.346479	1.346852	1.346364	1.1576×10^{-4}
0.85	1.438740	1.439172	1.438614	1.2625×10^{-4}
0.90	1.527024	1.527452	1.526911	1.1292×10^{-4}
0.95	1.610762	1.610859	1.610683	7.8852×10^{-5}
1.00	1.689559	1.688555	1.689498	6.1063×10^{-5}

Table 6. Numerical comparison of ChSANN and LSANN values with exact values at $\alpha = 1$ for fractional Riccati differential equation test example 2.

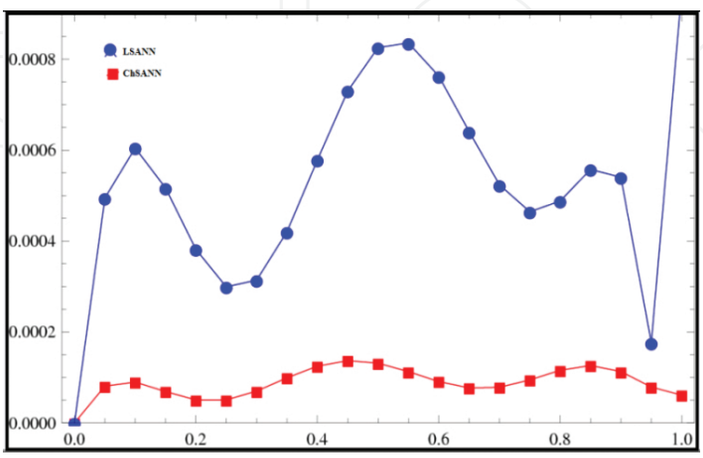


Figure 4. Absolute error of ChSANN and LSANN at $\alpha = 1$ for test example 2.

x	ChSANN	LSANN	MHPM [7]
0	0	0	0
0.1	0.22983	0.21885	0.216866
0.2	0.46136	0.45018	0.482292
0.3	0.69478	0.68545	0.654614
0.4	0.92279	0.91423	0.891404
0.5	1.13531	1.12664	1.132763
0.6	1.32357	1.31532	1.370240
0.7	1.48314	1.47660	1.594278
0.8	1.61485	1.61045	1.794879
0.9	1.72401	1.71972	1.962239
1.0	1.81844	1.80882	2.087384

Table 7. Numerical comparison for $\alpha = 0.75$.

x	ChSANN	LSANN	Reference [13]
0.2	0.31018	0.30567	0.314869
0.4	0.69146	0.68661	0.697544
0.5	0.89758	0.89230	0.903673
0.6	1.10220	1.09708	1.107866
0.8	1.47288	1.46889	1.477707
1.0	1.76276	1.76355	1.765290

Table 8. Numerical comparison for $\alpha = 0.9$.

α	ChSANN			LSANN		
	NAC	Training points	MSE	NAC	Training points	MSE
1	6	20	1.6127×10^{-7}	6	20	4.68641×10^{-6}
0.9	6	10	7.2486×10^{-6}	6	10	7.36985×10^{-6}
0.75	6	20	6.9229×10^{-5}	6	10	1.91318×10^{-5}

Table 9. Value of mean square error at different values of α.

No. of NAC	No of training points	MSE	$y(t)$	AE
4	10	1.1531×10^{-5}	1.67997	9.52973×10^{-3}
5	10	4.9226×10^{-7}	1.69016	6.61306×10^{-4}
6	20	1.6127×10^{-7}	1.68956	6.10629×10^{-5}

Table 10. Numerical values of ChSANN at $t = 1$ and for $\alpha = 1$.

http://dx.doi.org/10.5772/64151

4. Results and discussion

In this chapter, two new algorithms have been developed and verified for the Riccati differential equation with fractional order, based on the functional neural network, Chebyshev and Legendre polynomials and simulated annealing for fractional differential equations. Substantiation of the methods is carried out by examining two benchmark examples that were already solved by some previously renowned methods. The numerical evaluation with previously obtained results for fractional-order derivative exhibited the achievement of proposed methods.

For test example 1, better results with less value of mean square error were obtained for each method. Comparison of the mean square errors 5.501631×10^{-9} and 1.21928×10^{-9} for ChSANN and LSANN, respectively, showed that the mean square error is less for LSANN when $\alpha = 1$. However, it can be observed from **Table 1** and **Figure 2** that ChSANN gave the better results with slightly more mean square error than LSANN. It can be noted from **Table 5** that better results can be attained with variable number of weights and training points, while the trend witnessed from **Table 5** indicated that for ChSANN, decreasing value of mean square error is directly proportional to the absolute error for $\alpha = 1$.

The test example 2 showed quite similar trends as of example 1. **Tables 6** and **9** exhibited that for $\alpha = 1$, less mean square error for ChSANN than LSANN was noted due to which, more accurate results were achieved by ChSANN at $\alpha = 0.9$ as compared to LSANN that can be viewed in **Figure 4**. The results obtained for fractional values of derivatives are compared with MHPM for $\alpha = 0.75$ and a collocation-based method of Bernstein polynomials for $\alpha = 0.9$ as presented in **Tables 7** and **8**. The comparison showed that the results achieved by ChSANN and LSANN are quite similar to the results obtained by MHPM and collocation-based method of Bernstein polynomials. However, according to the observations from the case of $\alpha = 1$, it can be assumed that the results obtained for $\alpha = 0.75$ will be accurate up to 2–3 decimal places because the MSE was detected up to 6.9229×10^{-5} for ChSANN and 1.91318×10^{-5} for LSANN. While the results achieved for $\alpha = 0.9$ will be accurate up to 3–4 decimal places as the MSE was noticed up to 7.2486×10^{-6} for ChSANN and 7.36985×10^{-6} for LSANN.

The methods proposed in this study are capable of handling highly non-linear systems. Both the proposed neural architectures are less computational and exhaustive than MLP. With ease of computation, the suggested activation function has made fractional differential equations possible to solve. Training of NAC by simulated annealing with Chebyshev and Legendre neural architecture minimized the MSE up to a tolerable level that leads to more accurate numerical approximation. Simulated annealing is a probabilistic procedure that is mostly free of initial values and can easily escape from local optimum to global optimum unlike other methods. As well as it can successfully optimize the functions with crests and plateaus. The methods can be enhanced by introducing more advanced optimization techniques. The motivation behind the work is the successful implementation of neural algorithms in the field of calculus that gave the solution of fractional differential equations a new direction with ease of implementation.

5. Conclusion

In this chapter, ChSANN and LSANN have been developed for fractional differential equations and successfully employed on two benchmark examples of Riccati differential equations. The proposed methods gave the excellent numerical approximation of the Riccati differential equation of fractional order. The most remarkable advantage of the proposed methods is the accurate prediction of the result for the fractional values of derivative. These procedures are easy to implement and can be used to find the exact solution in the fractional values of derivative. ChSANN displayed more accurate results than LSANN for the similar applied conditions. Both the proposed algorithms are non-iterative and can be implemented through mathematical software and Mathematica 10 was used in this study to obtain all the results displayed in **Tables 1–10** and **Figures 2–4**.

Author details

Najeeb Alam Khan[1*], Amber Shaikh[1], Faqiha Sultan[2] and Asmat Ara[3]

*Address all correspondence to: njbalam@yahoo.com

1 Department of Mathematics, University of Karachi, Karachi, Pakistan

2 Department of Sciences and Humanities, National University of Computer and Emerging Sciences, Karachi, Pakistan

3 Department of Computer Science, Mohammad Ali Jinnah University, Karachi, Pakistan

References

[1] He JH. Nonlinear oscillation with fractional derivative and its applications. International Conference on Vibrating Engineering. 1998;98:288–91.

[2] He JH. Some applications of nonlinear fractional differential equations and their approximations. Bulletin of Science, Technology & Society. 1999;15(2):86–90.

[3] Grigorenko I, Grigorenko E. Chaotic dynamics of the fractional Lorenz system. Physical Review Letters. 2003;91(3):034101.

[4] Podlubny I. Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications. Academic Press; 1998. ISBN 0125588402.

[5] Podlubny I. Geometric and physical interpretation of fractional integration and fractional differentiation. Fractional Calculus and Applied Analysis. 2002;5(4):367–86.

[6] Merdan M. On the solutions fractional Riccati differential equation with modified Riemann-Liouville derivative. International Journal of Differential Equations. 2012;2012:1–17.

[7] Odibat Z, Momani S. Modified homotopy perturbation method: application to quadratic Riccati differential equation of fractional order. Chaos, Solitons & Fractals. 2008;36(1):167–74.

[8] Öztürk Y, Anapalı A, Gülsu M, Sezer M. A collocation method for solving fractional Riccati differential equation. Journal of Applied Mathematics. 2013;2013:1–8.

[9] Jafari H, Tajadodi H. He's variational iteration method for solving fractional Riccati differential equation. International Journal of Differential Equations. 2010;2010:1–8.

[10] Jafari H, Tajadodi H, Baleanu D. A modified variational iteration method for solving fractional Riccati differential equation by Adomian polynomials. Fractional Calculus and Applied Analysis. 2013;16(1):109–22 .

[11] Khader MM. Numerical treatment for solving fractional Riccati differential equation. Journal of the Egyptian Mathematical Society. 2013;21(1):32–7.

[12] Momani S, Shawagfeh N. Decomposition method for solving fractional Riccati differential equations. Applied Mathematics and Computation. 2006;182(2):1083–92.

[13] Yüzbaşı Ş. Numerical solutions of fractional Riccati type differential equations by means of the Bernstein polynomials. Applied Mathematics and Computation. 2013;219(11):6328–43.

[14] Hosseinnia SH, Ranjbar A, Momani S. Using an enhanced homotopy perturbation method in fractional differential equations via deforming the linear part. Computers & Mathematics with Applications. 2008;56(12):3138–49.

[15] Raja MAZ, Manzar MA, Samar R. An efficient computational intelligence approach for solving fractional order Riccati equations using ANN and SQP. Applied Mathematical Modelling. 2015;39(10–11):3075–93.

[16] Balaji S. Legendre wavelet operational matrix method for solution of fractional order Riccati differential equation. Journal of the Egyptian Mathematical Society. 2015;23(2): 263–70.

[17] Khan NA, Ara A, Jamil M. An efficient approach for solving the Riccati equation with fractional orders. Computers & Mathematics with Applications. 2011;61(9):2683–9.

[18] McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics. 1943;5:115–33.

[19] Hebb DO. The organization of behavior: A neuropsychological theory. New York: Wiely; 1949.

[20] Pao Y-H, Phillips SM, Sobajic DJ. Neural-net computing and the intelligent control of systems. International Journal of Control. 1992;56(2):263–89.

[21] Pandey C, Singh V, Singh O, Kumar S. Functional link artificial neural network for denoising of image. Journal of Electronics and Communication Engineering. 2013;4(6): 109–15.

[22] Panda G, Das DP. Functional link artificial neural network for active control of nonlinear noise processes. International Workshop on Acoustic Echo and Noise Control. 2003;2003:163–6.

[23] Patra JC, Pal RN, editors. Functional link artificial neural network-based adaptive channel equalization of nonlinear channels with QAM signal. Systems, Man and Cybernetics, 1995 Intelligent Systems for the 21st Century, IEEE International Conference on; 22–25 October 1995.

[24] Mall S, Chakraverty S. Comparison of artificial neural network architecture in solving ordinary differential equations. Advances in Artificial Neural Systems. 2013;2013:1–12.

[25] Mall S, Chakraverty S. Chebyshev Neural Network based model for solving Lane-Emden type equations. Applied Mathematics and Computation. 2014;247:100–14.

[26] Sontakke BR, Shaikh AS. Properties of Caputo operator and its applications to linear fractional differential equations. International Journal of Engineering Research and Applications. 2015;5(5):22–7.

[27] Caputo M. Linear models of dissipation whose Q is almost frequency independent-II. Geophysical Journal International. 1967;13(5):529–39.

[28] Lagaris IE, Fotiadis DI. Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks. 1998;9(5):987–1000.

[29] Ledesma S, Aviña G, Sanchez R. Practical considerations for simulated annealing implementation. In: Tan CM, editor. Simulated Annealing: InTech; 2008. ISBN 9789537619077.

IntechOpen