Neural minimization methods (NMM) for solving variable order fractional delay differential equations (FDDEs) with simulated annealing (SA)

Amber Shaikh1*, M. Asif Jamal2, Fozia Hanif3, M. Sadiq Ali Khan4, Syed Inayatullah5

1 Department of Humanities and Sciences, National University of Computer and Emerging Sciences, Karachi, Pakistan, 2 Department of Basic Sciences Federal Urdu University of Art, Science and technology Karachi & Cadet College, Karachi, Pakistan, 3 Department of Mathematics, University of Karachi, Karachi, Pakistan, 4 Department of Computer Sciences, University of Karachi, Karachi, Pakistan, 5 Department of Mathematics, University of Karachi, Karachi, Pakistan

* amb_shaikh@hotmail.com

Abstract

To enrich any model and its dynamics introduction of delay is useful, that models a precise description of real-life phenomena. Differential equations in which current time derivatives count on the solution and its derivatives at a prior time are known as delay differential equations (DDEs). In this study, we are introducing new techniques for finding the numerical solution of fractional delay differential equations (FDDEs) based on the application of neural minimization (NM) by utilizing Chebyshev simulated annealing neural network (ChSANN) and Legendre simulated annealing neural network (LSANN). The main purpose of using Chebyshev and Legendre polynomials, along with simulated annealing (SA), is to reduce mean square error (MSE) that leads to more accurate numerical approximations. This study provides the application of ChSANN and LSANN for solving DDEs and FDDEs. Proposed schemes can be effortlessly executed by using Mathematica or MATLAB software to get explicit solutions. Computational outcomes are depicted, for various numerical experiments, numerically and graphically with error analysis to demonstrate the accuracy and efficiency of the methods.

Introduction

In the old days, fractional calculus was only used by pure mathematicians due to its imperceptible applications at that time. When mathematicians were trying to implement fractional calculus in modeling of physical phenomena then applicability of this marvelous tool was successfully revealed. Therefore, in recent years, fractional derivatives have been used in many phenomena in electromagnetic theory, fluid mechanics, viscoelasticity, circuit theory, control theory, biology, atmospheric physics, etc. Many real-world problems can be accurately modeled by fractional differential equations (FDEs) such as damping laws, fluid mechanics, rheology, physics, mathematical biology, diffusion processes, electrochemistry, and so on.
Nowadays, there has been a tremendous increase of interest in theory of FDDEs, due to the fact that it can express the system of many dynamics population with more accuracy as demonstrated by past research in science and engineering. In the systems of real world, delays can be recognized and implemented everywhere and due to this advancement in modeling it has captured a lot of attention of the scientific community towards it.

**Literature review**

Many researchers were inspired by the subject of fractional order and addressed these phenomena in different scenarios. For example, Lie et al. [1] used phase portraits, time domain waveform and bifurcation to examine the behavior of nonlinear system of fractional order. Modified Kalman filter was proposed by [2] to deal with fractional order system. To achieve the suitable generalized projective synchronization (GPS) of incommensurate fractional order system,[3] has introduced fuzzy approach. Coronel-Escamilla et al.[4] used Euler–Lagrange and Hamilton formalisms for describing the the fractional modeling and control of an industrial selective compliant assembly robot arm (SCARA).

Many researchers have made their efforts to investigate DEs and FDEs such as Zhang et al. [5] studied generalized Burgers equation and a generalized Kupershmidt equation through lie-group analysis method for similarity reductions and exact solutions, Zhang and Zhou [6] also carried out analysis of Drinfeld-Sokolov-Wilson system through symmetry analysis method, Yang et al.[7, 8] implemented travelling wave solution to local fractional two-dimensional Burgers-type equations and Boussinesq equation in fractal domain. Atangana and Gómez-Aguilar [9] presented exact solution and semi group principle for evolutions equation by using three different definitions of fractional derivatives. Lie et al. [10] calculated the Haar wavelet operational matrix together with Block phase function to find the solution of FDEs. Li et al. [11] have also made a successful attempt to approximate the same problem by using Chebyshev wavelet operational matrix method. Adomian decomposition method and variational iteration method were implemented in [12–16] to solve a variety of FDEs. While differential transform method and power series method are also noteworthy for the solution of FDEs [17–22].

In recent years the approach of neural architecture has been used to solve Des and FDEs. In [23] Aarts and Veer implemented the multilayer neural algorithm for the solution of partial differential equations (PDEs) with evolutionary algorithm for training of weights. Feed forward NN technique, with the blend of piece splines of Lagrange polynomial, was proposed by [24]. Same approach was applied by [25] in which genetic algorithm was used as an evolutionary algorithm for training of network of nonbed-catalytic gas reactor system. For solving PDEs [26] has put an effort by implementing NN together with Broden-Flecher-Goldfarb-Shanno algorithm. Nelder Meade optimization procedure with hybrid neural network (HNN) was adopted by [27] for the numerical simulation of higher order DEs. Levenberg-Marquardt algorithm with ANN and Mittag-Leffler kernel was implemented by [28] to solve FDEs.

Those systems that are governed by their past are modeled in the form of delay differential equations. DDEs are proved useful in control systems [29], lasers, traffic models [30], metal cutting, epidemiology, neuroscience, population dynamics [31], and chemical kinetics [32] condition. Due to the infinite dimensionality of delay systems it is very challenging to analytically analyze DDEs, therefore numerical simulations of DDEs play a key role for study of such systems. A noteworthy study of FDDEs through neural network can be visualized in [33] Existence and uniqueness theorems on FDDEs are discussed in [34–36]

This study will generate an approximate solution for solving the DDEs and FDDEs by using ChSANN and LSANN, which were first developed by Khan et al. [37, 38] to solve Lane Emden equations and fractional differential equations on a discrete domain. In the following paper we
have developed an approach to solve the higher order DDEs and FDDEs on a continuous domain. This paper is organized as follows: first section of paper describes introduction and literature review while second section concerns with details of the methodologies with well explained algorithm and implication procedure. Error analysis procedure is explained in third section where as fourth, fifth and sixth sections describe the numerical experiments along with results and their discussions.

Methodology

The proposed methodologies are based on functional link neural network with optimization through thermal minimization. In this study, the Caputo definition will be used for working out the fractional derivative in the subsequent procedure. These definitions of commonly used fractional differential operators are discussed in [39].

Initially introduced by Pao [40], ChSANN and LSANN are the revised version of functional link artificial NN coupled with optimization strategy for learning. Functional link architecture of NN was design to build the connection between the linearity in a single layer NN and the computationally challenging multilayer NN.

Inspired by the physical process of annealing, SA is basically a kind of combinatorial optimization process. This process is based on two steps: first to perturb and then to measure quality of the solution. MSE is basically the fitness function denoted by $E_r$, that can be minimized by the use of SA.

Algorithm

Due to the structural similarity in ChSANN and LSANN, the steps of algorithm are described in combination for both methods. Accuracy of results depends on the selection of base polynomial.

**Step 1:** Initialize the network by applying number of Chebyshev polynomials or Legendre polynomials (to independent variable $x$) $k = 0$ to $n$.

**Step 2:** Provide in each polynomial a network adaptive coefficient (NAC).

**Step 3:** Calculate the summation of product of NAC and Chebyshev polynomial or Legendre polynomial and store the value in $\phi$ or $\psi$ respectively.

**Step 4:** Activate $\phi$ and $\psi$, by first three terms of Taylor Series expansion of $\tanh$ function.

**Step 5:** As given by Lagaris and Fotiadias [41] trial solution will be generated by the help of initial conditions and activated $\phi$ (in case of LSANN $\psi$)

**Step 6:** Calculate the value of delay trial solution by repeating step 1 to 5 with delay independent variable.

**Step 7:** Calculate the MSE of DDEs or FDDEs by discretizing the domain in $\beta$ number of points.

**Step 8:** Set Tolerance for accepting minimized value of MSE.

**Step 9:** Minimize the MSE by thermal minimizing methodology with the following settings from Mathematica 11.0.

> Level iterations $\rightarrow$ 50

> Perturbation Scale $\rightarrow$ 1.0

> Probability function $\rightarrow e^{(-\log(i+1)/T)+\log(T)}$

> Random seed $\rightarrow$ 0

> Tolerance for accepting constraint violations $\rightarrow$ 0.001
Step 10: If the value of MSE falls in the range of pre-defined criteria, then substitute the value of NAC in trial solution to get the output otherwise go to step 1 change the value of n and repeat whole procedure till the acceptable MSE is obtained.

Pictorial presentation of above algorithm can be observed in Fig 1

Employment on delay differential equation

Now we apply ChSANN and LSANN on DDEs of the following type,

\[ f^{(n)}(x) = F(f(x - \tau x), f'(x), f''(x), \ldots, f^{(n-1)}(x)) + g(x). \]  

(1)

With initial conditions as follows,

\[ f(0) = a_0, f'(0) = a_1, \ldots, f^{(n-1)}(0) = a_{n-1}. \]

(2)

For implementation of ChSANN and LSANN Eq (1) can be written as

\[ f^{(n)}(x) - F(f(x - \tau x), f'(x), f''(x), \ldots, f^{(n-1)}(x)) - g(x) = 0. \]

(3)

For trial and delay trial solution of above differential equation consider,

\[ \psi = \sum_{i=1}^{n} w_i T_{k-1}(x), \]

(4)

where \( T_k \) is Chebyshev polynomial with the following recursive formula defined as

\[ T_{k+1}(x) = 2xT_k(x) - T_{k-1}, \quad k \geq 2 \]

Here \( T_0 = 1 \) and \( T_1 = x \) are the fundamental values of Chebyshev polynomials and

\[ \phi = \sum_{j=1}^{n} w_j L_{j-1}(x), \]

(5)

where \( L_j \) is Legendre polynomial with the following recursive formula defined as:

\[ L_{j+1}(x) = \frac{1}{(j+1)(2j+1)} (2j+1) x L_j(x) - \frac{1}{(j+1)} L_{j-1}(x), \quad j \geq 2, \]

(6)

where, \( L_0(x) = 1 \) and \( L_1(x) = x \) are the fundamental values of Legendre polynomials.

Here we are using first three terms of Taylor series expansion of \( \tanh \) function to activate \( \phi \) and \( \psi \). As defined by Lagaris and Fotiadis [41], the trial solution of Eq (1) can be written as,

\[ f_{trial}(x, w) = a_0 + a_1 x + \frac{x^2}{2!} \alpha_2 + \ldots + N(x, w) \frac{x^k}{k!}, \]

(7)

Where \( N \) is the activated \( \phi \) or \( \psi \), depends on the method, while the delay trial solution can be given by replacing the \( x \) by \( x-\tau x \).

The MSE of the Eq (1) will be calculated from the following:

\[ E_i(w) = \sum_{j=1}^{\beta} (f_{trial}^{(n)}(x_i, w) - F(f_{trial}(x_i, w), f_{trial}(x_i - \tau x_i, w), f_{trial}(x_i, w), f_{trial}(x_i, w), \ldots, f_{trial}^{(n-1)}(x_i, w))) - g(x_i))^2 \]

(8)

here \( \beta \) represents the number of trial points while Eq (8) will be the fitness function for learning of NAC. For implementation on FDDEs only the method of computing the fractional derivative of trial solution will vary, which will be taken in this study according to Caputo definition as follows. 
Definition

According to [39] Caputo operator for \( \lambda > 0 \) can be defined as:

\[
D^\lambda g(\eta) = \frac{1}{\Gamma(m - \lambda)} \int_0^\eta (\eta - t)^{m-\lambda-1} g^{(n)}(t) dt, \quad m - 1 < \lambda < m, m \in \mathbb{N},
\]

with
\[
D^\lambda c = 0, \text{ where } c \text{ is a constant}
\]
\[
D^\lambda (\eta^\beta) = \begin{cases} 
0, & \lambda \in \mathbb{N}, \beta < \lambda \\
\frac{\Gamma(\beta + 1)}{\Gamma(\beta + 1 - \lambda)} \eta^{\beta - \lambda}, & \text{otherwise}
\end{cases}
\]

Mathematica 11.0 is the minimization implementation tool in this study but details can be seen from [42].

**Error analysis**

The error analysis of numerical experiments for ChSANN and LSANN methodologies can be observed by following procedure. By substituting the values of NAC, after learning from SA algorithm, into the trial solution it will become ChSANN or LSANN solution that can be further substituted into Eq (9) for analyzing the accuracy of method on the domain of [0,1].

\[
E(x) = |f^n(x) - F(f(x - \tau x), f'(x), f''(x), \ldots, f^{(n-1)}(x)) - g(x)| \cong 0 \tag{9}
\]

While \( f(x) \) is the obtained approximated continuous solution by ChSANN or LSANN. \( E(x) \) tends to 0 as the value of MSE obtained by ChSANN and LSANN is in the predefined range. Convergence of solution is totally dependent on the learning methodology of respected NN architecture which is SA in the current case.
Numerical experiments

Experiment 1. Consider 2nd order DDE along with the initial conditions as:

\[ y(x) = \frac{3}{4} y(x) + y\left(\frac{x}{2}\right) - x^2 + 2, \quad y(0) = 0; \quad y'(0) = 0; \quad x = 2. \]

The exact solution when \( \alpha = 1 \) is given as:

\[ y(x) = x^2. \]

In this experiment we employed proposed methodologies on above second order linear DDE on domain of \([0,1]\). Both the methods were employed by dividing the domain with 10 equidistant training points and 6 NAC. For ChSANN and LSANN at \( \alpha = 2 \), the MSE at defined conditions is found to be \( 1.89855 \times 10^{-11} \) and \( 1.32344 \times 10^{-14} \) respectively. (Fig 2) depicts the comparison of both methods with true solution at continuous domain of \([0,1]\), while (Fig 3) displays the error analysis for both methods at \( \alpha = 2 \). For the above experiment the trial and delay trial solutions are found to be following:

\[ y_{trial} = \frac{x^2}{2} N \]

and

\[ y_{dtrial} = \frac{(x/2)^2}{2} M. \]

Where, \( N \) and \( M \) are the structural outputs of both NNs. Table 1 displays the final values of NAC after training by SA algorithm and Figs 4 and 5 is displaying the data for 100 independent runs by altering the scale for random jumps.

| SNo | NAC | ChSANN     | LSANN     |
|-----|-----|------------|-----------|
| 1   | \( w_1 \) | 1.70693523 | 1.710066834 |
| 2   | \( w_2 \) | -0.00101189 | -0.00004054 |
| 3   | \( w_3 \) | -0.00220052 | -0.00372515 |
| 4   | \( w_4 \) | -0.00028488 | -0.00002618 |
| 5   | \( w_5 \) | 0.000058430 | -0.00003209 |
| 6   | \( w_6 \) | -0.00001175 | -0.00002803 |

Experiment 2. Consider 3rd order nonlinear DDE along with the initial condition as:

\[ y^{(x)}(x) = 2 \sqrt{\frac{x}{2}} - 1, \quad y(0) = 0, \quad y'(0) = 1, \quad y''(0) = 0; \quad x = 3. \]

The exact solution when \( \alpha = 3 \) is given below:

\[ y(x) = Sin(x). \]

3rd order nonlinear DDE is solved by ChSANN and LSANN on the continuous domain of \([0,1]\). ChSANN and LSANN were run for 10 and 6 NAC respectively while training points were 20 for both. With given predefined conditions MSE is found to be \( 2.5679 \times 10^{-5} \) for ChSANN and \( 5.4843 \times 10^{-7} \) for LSANN. Comparison of the methods with true solution can be visualized in Fig 6 and error analysis can be observed in Fig 7. Table 2 represents the final
values of weights after training by SA algorithm while Figs 8 and 9 are displaying the results for 100 independent runs for elapsed time in seconds, fitness and number of iterations. Following are trial and delay trial solution for the current experiment:

\[ y_{\text{trial}} = x + \frac{x^3}{6} M \]

and

\[ y_{\text{dtrial}} = \frac{x}{2} + \frac{(x/2)^3}{6} N. \]

While \( M \) and \( N \) are structural NN outputs of both the methods for in progress experiment. 

**Example 3.** Consider FDDE along with the initial conditions as:

\[ y^{(\alpha)}(x) = \frac{1}{2} e^{x} y(x) + \frac{1}{2} y(x), \quad y(0) = 1, \quad 0 \leq x \leq 1, \quad 0 < \alpha \leq 1. \]

The exact solution at \( \alpha = 1 \) is given by:

\[ y(x) = e^{x} \]

ChSANN and LSANN have been employed successfully on above FDDE with 10 training points and 6 NAC. (Fig 10) depicts the comparison of ChSANN and LSANN solutions with true values at \( \alpha = 1 \) while values of final NAC after learning by SA algorithm can be visualized.
in Table 3. Both the methods were executed for different fractional values of $\alpha$ for which results can be visualized in Tables 4–6. Error analysis for all fractional values can be seen in (Fig 11) and Figs 12 and 13 is displaying the results for 100 independent runs for both the proposed methods.

![Fig 5. Results for 100 independent runs.](https://doi.org/10.1371/journal.pone.0223476.g005)

![Fig 6. Comparison of ChSANN and LSANN with true solution.](https://doi.org/10.1371/journal.pone.0223476.g006)
Trial and delay trial solutions are taken to be:

\[ y_{\text{trial}} = 1 + xN \]

and

\[ y_{\text{del trial}} = 1 + \frac{x}{2}M. \]

**Example 4.** Consider Non-Linear FDDE along with the initial condition as:

\[ y''(x) = 1 - 2y^2 \left(\frac{x}{2}\right) , \quad y(0) = 1, \quad y'(0) = 0, \quad 0 \leq x \leq 1, \quad 1 < \alpha \leq 2 \]

The exact solution at \( \alpha = 2 \) is given by:

\[ y(x) = \cos(x) \]

ChSANN and LSANN are implemented on the above experiment of nonlinear FDDE. (Fig 14) shows the comparison of both methods with true values at \( \alpha = 2 \). While (Fig 15) represents the error analysis for different fractional values of \( \alpha \). Trial solution was taken in the same manner as in previous experiments. ChSANN and LSANN solutions with obtained MSE at

| SNo | NAC | ChSANN  | LSANN   |
|-----|-----|---------|---------|
| 1   | \( w_1 \) | -0.874541110 | -1.2104577306 |
| 2   | \( w_2 \) | -0.478671752 | -0.0384815109 |
| 3   | \( w_3 \) | 0.0165454550 | 0.06652279388 |
| 4   | \( w_4 \) | 0.4425599703 | -0.0172612429 |
| 5   | \( w_5 \) | -0.594007418 | 0.00543949906 |
| 6   | \( w_6 \) | 0.4767150261 | -0.00064910894 |
| 7   | \( w_7 \) | -0.261651890 | 0.9618236111 |
| 8   | \( w_8 \) | 0.0976657350 | -1.0110550149 |
| 9   | \( w_9 \) | -0.022615861 | 2.3259303584 |
| 10  | \( w_{10} \) | 0.0024649552 | -2.0381716440 |

https://doi.org/10.1371/journal.pone.0223476.t002
different fractional values of $\alpha$ can be visualized in Tables 7–9 and Figs 16 and 17 are displaying the results for 100 independent runs.

**Discussion**

The above work is concerned with the successful implementation of ChSANN and LSANN on higher order DDEs and FDDEs. Some benchmark examples were considered for experimental cases and validity of implementation has been judged by standard error analysis procedure, data analysis of 100 number of runs of algorithm and comparison with other methods.

**Error analysis**

For test experiment 1, MSE for ChSANN and LSANN for $\alpha = 2$ were found to be $1.89855 \times 10^{-11}$ and $1.32344 \times 10^{-14}$ that gave the minimum error for ChSANN is $2.8 \times 10^{-5}$ and for LSANN is $2.6 \times 10^{-7}$ that can be easily visualized from (Fig 3). It shows that accuracy of both the methods is inversely proportional to the value of MSE and it can also be noticed that change of polynomials in both methods has strongly influenced the learning of NAC by SA algorithm that can be witnessed from Table 1. For experiment no 1, at similar conditions of training points and NAC, LSANN gave more promising results.

Similar trends can be depicted from the results of test experiment 2, in which MSE for ChSANN and LSANN for $\alpha = 3$ were found to be $2.5679 \times 10^{-5}$ and $2.20049 \times 10^{-7}$.
Analysis from (Fig 7) exhibiting the better performance of LSANN with less MSE as described above. Moreover it can also be visualized that this structure of neural network can be easily implemented on higher order nonlinear differential equations with ease.

In experiment no 3, both the proposed architectures were employed on linear delay fractional differential equation. MSE for ChSANN and LSANN at $\alpha = 1$ were found to be $4.63009 \times 10^{-11}$ and $3.88356 \times 10^{-11}$ respectively that gave excellent results that can be seen in (Fig 10).

Fig 9. Results for 100 independent runs.
https://doi.org/10.1371/journal.pone.0223476.g009

Fig 10. Comparison of ChSANN and LSANN with true solution at $\alpha = 1$.
https://doi.org/10.1371/journal.pone.0223476.g010
Table 3. Values of NAC at $\alpha = 1$.

| SNo | NAC  | ChSANN       | LSANN       |
|-----|------|--------------|-------------|
| 1   | $w_1$| 1.1669374428 | 1.1917375672|
| 2   | $w_2$| 0.5269817424 | 0.5257487109|
| 3   | $w_3$| -0.079021346 | -0.107447861|
| 4   | $w_4$| 0.0160510892 | 0.0265364411|
| 5   | $w_5$| -0.0011808281| -0.003520220 |
| 6   | $w_6$| 0.0008888300 | 0.0002011224|

https://doi.org/10.1371/journal.pone.0223476.t003

Table 4. ChSANN and LSANN results at $\alpha = 0.5$.

| X Values | ChSANN | LSANN |
|----------|--------|-------|
| 0.1      | 1.403383 | 1.403323 |
| 0.2      | 1.70248  | 1.702377|
| 0.3      | 1.981465 | 1.981438|
| 0.4      | 2.269594 | 2.269551|
| 0.5      | 2.576318 | 2.576261|
| 0.6      | 2.904964 | 2.904908|
| 0.7      | 3.258834 | 3.258791|
| 0.8      | 3.643645 | 3.643603|
| 0.9      | 4.066385 | 4.066317|
| 1.0      | 4.529678 | 4.529594|

MSE For ChSANN $9.94644\times10^{-6}$
MSE For LSANN $9.947919\times10^{-6}$

https://doi.org/10.1371/journal.pone.0223476.t004

Table 5. ChSANN and LSANN results at $\alpha = 0.7$.

| X Values | ChSANN       | LSANN       |
|----------|--------------|-------------|
| 0.1      | 1.2261657    | 1.22608151  |
| 0.2      | 1.4256231    | 1.4255953   |
| 0.3      | 1.62462243   | 1.62460268  |
| 0.4      | 1.83454966   | 1.83452219  |
| 0.5      | 2.06067256   | 2.06063551  |
| 0.6      | 2.30584961   | 2.30580978  |
| 0.7      | 2.57260285   | 2.57256722  |
| 0.8      | 2.86419774   | 2.86413699  |
| 0.9      | 3.1841364    | 3.18416933  |
| 1.0      | 3.5369036    | 3.53690470  |

MSE For ChSANN $2.86781\times10^{-6}$
MSE For LSANN $2.86881\times10^{-6}$

https://doi.org/10.1371/journal.pone.0223476.t005

Table 6. ChSANN and LSANN at $\alpha = 0.8$.

| X Values | ChSANN       | LSANN       |
|----------|--------------|-------------|
| 0.1      | 1.17320141   | 1.1731957   |
| 0.2      | 1.33893581   | 1.33893313  |
| 0.3      | 1.50993727   | 1.5099335   |
| 0.4      | 1.69280737   | 1.6928027   |
| 0.5      | 1.89124814   | 1.8912421   |
| 0.6      | 2.10770277   | 2.1076959   |
| 0.7      | 2.34439824   | 2.3443915   |
| 0.8      | 2.60397962   | 2.6039730   |
| 0.9      | 2.88960028   | 2.8895924   |
| 1.0      | 3.20417343   | 3.2041639   |

MSE For ChSANN $1.2802637\times10^{-6}$
MSE For LSANN $1.28041\times10^{-6}$

https://doi.org/10.1371/journal.pone.0223476.t006
Further both the methods were also executed for $\alpha = 0.7, 0.8$ and 0.9. Obtained MSE for all the fractional values obtained by both the methods were in the same range so the accuracy of results for all the fractional values is approximately similar that can be visualized in (Fig 11).

Experiment no 4 is a case of nonlinear FDDE. For $\alpha = 2$ values of MSE with 6 NAC were found to be $7.82099 \times 10^{-6}$ and $1.050097 \times 10^{-5}$. For fractional values of $\alpha$ ChSANN is showing better results than LSANN at $\alpha = 1.5$ and $\alpha = 1.9$, with better values of MSE for ChSANN and for $\alpha = 1.7$ both the methods are exhibiting the similar accuracy. Accuracy at fractional values can be visualized from (Fig 15).

Data analysis for 100 numbers of independent runs. For each test experiment, algorithms of proposed techniques were executed 100 times by altering the scale of random jumps to assess the precision, performance and reliability. Results of obtained data can be visualized
in form of figures which shows that for test experiment 1 fitness function is revolving between $10^{-4}$ to $10^{-14}$ and $10^{-6}$ to $10^{-14}$ for CHSANN and LSANN respectively. Elapsed time in second is found to be within three seconds for both the methods while number of iterations were between 600–1200 and 400–1000 for CHSANN and LSANN respectively. Results of 100 independent runs for test experiments 2–4 can be visualized in Figs 7 and 8, Figs 11 and 12 and Figs 16 and 17 respectively, which demonstrate a similar trend except for the case of nonlinear models for which the maximum elapsed time is found to be 20 seconds due to computational complexity.

Comparison with other methods. We compared the proposed techniques in terms of accuracy, elapsed time, ease of calculation and error prediction with the methods presented in [43], [44] and [45]. Methods in [43–45] have been implemented on similar type of problems as in current study. Test example number 3 by Radial basis method presented in [43] and test example no 4 presented in [45] is similar to test experiment no 2 presented in following paper, Test example number 5 presented in [44] is similar to test experiment number 4 presented above and test experiment number 2 presented in [45] is similar to test experiment no 1 by proposed methods.

Following key points of comparison can be noticed.

➢ Radial basis method, method in [44] and method in [45] are providing results on collocation points while proposed schemes are providing a continuous solution.
On the other hand method in [43] is taking 10 to 85 seconds for solving a linear problem while proposed techniques are consuming 6 to 12 seconds and 3 to 5 seconds (Figs 8 and 9) for solving nonlinear problem by ChSANN and LSANN respectively.

Fig 13. Results for 100 independent runs.
https://doi.org/10.1371/journal.pone.0223476.g013

Fig 14. Comparison of ChSANN and LSANN with true values.
https://doi.org/10.1371/journal.pone.0223476.g014
Computational complexity of method presented in [43–45] is very large due to solving a large system of nonlinear equations while proposed techniques are too simple in terms of implementation that can be observed through the computational time difference.

There is no way to predict the accuracy in method proposed in [43–45] when there is no exact solution present while proposed schemes can predict accuracy of solution through fitness function. In terms of accuracy methods in [43–45] is giving more accurate results than proposed schemes but limitations of [43–45] is making proposed schemes more powerful.
Table 7. ChSANN and LSANN results for FDDE at $\alpha = 1.5$.

| $x$  | ChSANN         | LSANN         |
|-----|----------------|---------------|
| 0.1 | 0.9819344      | 0.9822488     |
| 0.2 | 0.9422867      | 0.9427229     |
| 0.3 | 0.8913654      | 0.8918057     |
| 0.4 | 0.8333956      | 0.8338889     |
| 0.5 | 0.7710079      | 0.7715448     |
| 0.6 | 0.7063767      | 0.7068606     |
| 0.7 | 0.6413646      | 0.6417259     |
| 0.8 | 0.5774801      | 0.5777550     |
| 0.9 | 0.5159194      | 0.5161458     |
| 1.0 | 0.4577416      | 0.4576439     |

MSE For ChSANN $3.32555 \times 10^{-6}$
MSE For LSANN $1.05097 \times 10^{-5}$

https://doi.org/10.1371/journal.pone.0223476.t007

Table 8. ChSANN and LSANN results for FDDE at $\alpha = 1.7$.

| $x$  | ChSANN   | LSANN   |
|-----|----------|---------|
| 0.1 | 0.989296 | 0.9893501 |
| 0.2 | 0.962232 | 0.9623276 |
| 0.3 | 0.923181 | 0.9232965 |
| 0.4 | 0.874549 | 0.8746877 |
| 0.5 | 0.818103 | 0.8182648 |
| 0.6 | 0.755417 | 0.7555877 |
| 0.7 | 0.687973 | 0.6881368 |
| 0.8 | 0.617155 | 0.6173071 |
| 0.9 | 0.544234 | 0.5443786 |
| 1.0 | 0.470418 | 0.4705211 |

MSE For ChSANN $1.09296 \times 10^{-6}$
MSE For LSANN $1.73111 \times 10^{-6}$

https://doi.org/10.1371/journal.pone.0223476.t008

Table 9. ChSANN and LSANN results for FDDE at $\alpha = 1.9$.

| X Values | ChSANN   | LSANN   |
|----------|----------|---------|
| 0.1      | 0.9935924 | 0.99359236 |
| 0.2      | 0.9753719 | 0.97537194 |
| 0.3      | 0.9464374 | 0.94643741 |
| 0.4      | 0.9076268 | 0.90762677 |
| 0.5      | 0.8597139 | 0.87971386 |
| 0.6      | 0.8034944 | 0.80349434 |
| 0.7      | 0.7398116 | 0.73981153 |
| 0.8      | 0.6695533 | 0.66955325 |
| 0.9      | 0.5936416 | 0.59364162 |
| 1.0      | 0.5130313 | 0.51303120 |

MSE For ChSANN $1.26583 \times 10^{-7}$
MSE For LNN $1.296043 \times 10^{-7}$

https://doi.org/10.1371/journal.pone.0223476.t009
Conclusion

In above study we have developed two methods ChSANN and LSANN for simulation of fractional delay differential equation. After analyzing procedure and numerical experiments following points can be concluded.

➢ Proposed methods can be successfully implemented on linear and nonlinear FDDEs with ease of calculation.

➢ Accuracy of method can be increased by improving the learning methodology of NAC.

➢ Accuracy of both the methods is inversely proportional to MSE.

➢ Both the methods can easily handle the nonlinear terms.

➢ Accuracy prediction can be obtained for fractional values of derivatives by observing the MSE values.

In future the proposed schemes can be further developed for accuracy by refining the learning methodology of NAC and by improving the neural architecture. However, it can also be successfully implemented on partial differential equations with some alterations in methodology.
Fig 17. Results for 100 independent runs.
https://doi.org/10.1371/journal.pone.0223476.g017

Author Contributions

Conceptualization: M. Asif Jamal.
Data curation: Amber Shaikh, M. Asif Jamal.
Formal analysis: Amber Shaikh, M. Sadiq Ali Khan, Syed Inayatullah.
Methodology: Amber Shaikh, M. Sadiq Ali Khan.
Project administration: Fozia Hanif, Syed Inayatullah.
Resources: M. Asif Jamal, Fozia Hanif, M. Sadiq Ali Khan, Syed Inayatullah.
Software: Amber Shaikh, M. Asif Jamal, M. Sadiq Ali Khan.
Supervision: Fozia Hanif, Syed Inayatullah.
Validation: Fozia Hanif.
Visualization: Amber Shaikh.
Writing – original draft: M. Asif Jamal.
Writing – review & editing: Amber Shaikh, Fozia Hanif, M. Sadiq Ali Khan, Syed Inayatullah.

References

1. Li Z, Chen D, Zhu J, Liu Y. Nonlinear dynamics of fractional order Duffing system. Chaos, Solitons & Fractals. 2015 Dec 1; 81:111–6.
2. Pourdehi S, Azami A, Shabaninia F. Fuzzy Kalman-type filter for interval fractional-order systems with finite-step auto-correlated process noises. Neurocomputing. 2015 Jul 2; 159:44–9.

3. Boukroune A, Bouzeriba A, Bouden T. Fuzzy generalized projective synchronization of incommensurate fractional-order chaotic systems. Neurocomputing. 2016 Jan 15; 173:606–14.

4. Coronel-Escamilla A, Torres F, Gomez-Aguilar JF, Escobar-Jimenez RF, Guerrero-Ramirez GV. On the trajectory tracking control for an SCARA robot manipulator in a fractional model driven by induction motors with PSO tuning. Multibody System Dynamics. 2018 Jul 15; 43(3):257–77.

5. Zhang Y, Mei J, Zhang X. Symmetry properties and explicit solutions of some nonlinear differential and fractional equations. Applied Mathematics and Computation. 2018 Nov 15; 337:408–18.

6. Zhang Y, Zhao Z. Lie symmetry analysis, Lie-Bäcklund symmetries, explicit solutions, and conservation laws of Drinfeld-Sokolov-Wilson system. Boundary Value Problems. 2017 Dec 1; 2017(1):154.

7. Yang XJ, Gao F, Srivastava HM. Exact travelling wave solutions for the local fractional two-dimensional Burgers-type equations. Computers & Mathematics with Applications. 2017 Jan 15; 73(2):203–10.

8. Yang XJ, Machado JT, Baleanu D. Exact traveling-wave solution for local fractional Boussinesq equation in fractal domain. Fractals. 2017 Aug; 25(04):1740006.

9. Atangana A, Gómez-Aguilar JF. Decolonization of fractional calculus rules: Breaking commutativity and associativity to capture more natural phenomena. The European Physical Journal Plus. 2018 Apr; 133:1–22.

10. Li Y, Zhao W. Haar wavelet operational matrix of fractional order integration and its applications in solving the fractional order differential equations. Applied Mathematics and Computation. 2010 Jun 15; 216 (8):2276–85.

11. Yuanlu L. Solving a nonlinear fractional differential equation using Chebyshev wavelets. Communications in Nonlinear Science and Numerical Simulation. 2010 Sep 1; 15(9):2284–92.

12. Odibat Z, Momani S. Numerical methods for nonlinear partial differential equations of fractional order. Applied Mathematical Modelling. 2008 Jan 1; 32(1):28–39.

13. Momani S, Odibat Z. Numerical approach to differential equations of fractional order. Journal of Computational and Applied Mathematics. 2007 Oct 1; 207(1):96–110.

14. El-Wakil SA, Elhanbaly A, Abdou MA. Adomian decomposition method for solving fractional nonlinear differential equations. Applied Mathematics and Computation. 2006 Nov 1; 182(1):313–24.

15. Hosseininia SH, Ranjbbar A, Momani S. Using an enhanced homotopy perturbation method in fractional differential equations via deforming the linear part. Computers & Mathematics with Applications. 2008 Dec 1; 56(12):3138–49.

16. Dhaigude DB, Birajdar GA. Numerical solution of system of fractional partial differential equations by discrete Adomian decomposition method. J. Frac. Cal. Appl. 2012 Jul; 3(12):1–1.

17. Arikoglu A, Ozkol I. Solution of fractional differential equations by using differential transform method. Chaos, Solitons & Fractals. 2007 Dec 1; 34(5):1473–81.

18. Arikoglu A, Ozkol I. Solution of fractional integro-differential equations by using fractional differential transform method. Chaos, Solitons & Fractals. 2009 Apr 30; 40(2):521–9.

19. Darania P, Ebadian A. A method for the numerical solution of the integro-differential equations. Applied Mathematics and Computation. 2007 May 1; 188(1):657–68.

20. Ertürk VS, Momani S. Solving systems of fractional differential equations using differential transform method. Journal of Computational and Applied Mathematics. 2008 May 15; 215(1):142–51.

21. Ertürk VS, Momani S, Odibat Z. Application of generalized differential transform method to multi-order fractional differential equations. Communications in Nonlinear Science and Numerical Simulation. 2008 Oct 1; 13(8):1642–54.

22. Odibat ZM, Shawagfeh NT. Generalized Taylor’s formula. Applied Mathematics and Computation. 2007 Mar 1; 186(1):286–93.

23. Aarts LP, Van Der Veer P. Neural network method for solving partial differential equations. Neural Processing Letters. 2001 Dec 1; 14(3):261–71.

24. Meade AJ, Fernandez AA. The numerical solution of linear ordinary differential equations by feedforward neural networks. Mathematical and Computer Modelling. 1994 Jun 1; 19(12):1–25.

25. Parisi DR, Mariani MC, Laborde MA. Solving differential equations with unsupervised neural networks. Chemical Engineering and Processing: Process Intensification. 2003 Aug 1; 42(8–9):715–21.

26. Lagaris IE, Likas A, Fotiadis DI. Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks. 1998 Sep; 9(5):987–1000. https://doi.org/10.1109/72.712178 PMID: 18255782
27. Malek A, Beidokht RS. Numerical solution for high order differential equations using a hybrid neural network—optimization method. Applied Mathematics and Computation. 2006 Dec 1; 183(1):260–71.

28. Zúñiga-Aguilar CJ, Romero-Ugalde HM, Gómez-Aguilar JF, Escobar-Jiménez RF, Valtierra-Rodríguez M. Solving fractional differential equations of variable-order involving operators with Mittag-Leffler kernel using artificial neural networks. Chaos, Solitons & Fractals. 2017 Oct 1; 103:382–403.

29. Davis LC. Modifications of the optimal velocity traffic model to include delay due to driver reaction time. Physica A: Statistical Mechanics and its Applications. 2003 Mar 1; 319:557–67.

30. Epstein IR, Luo Y. Differential delay equations in chemical kinetics. Nonlinear models: The cross-shaped phase diagram and the Oregonator. The Journal of chemical physics. 1991 Jul 1; 95(1):244–54.

31. Kuang Y, editor. Delay differential equations: with applications in population dynamics. Academic Press; 1993 Mar 5.

32. Benchchhra M, Henderson J, Ntouyas SK, Ouahab A. Existence results for fractional order functional differential equations with infinite delay. Journal of Mathematical Analysis and Applications. 2008 Feb 15; 338(2):1340–50.

33. Zúñiga-Aguilar CJ, Coronel-Escamilla A, Gómez-Aguilar JF, Alvarado-Martínez VM, Romero-Ugalde HM. New numerical approximation for solving fractional delay differential equations of variable order using artificial neural networks. The European Physical Journal Plus. 2018 Feb 1; 133(2):75.

34. Henderson J, Ouahab A. Fractional functional differential inclusions with finite delay. Nonlinear Analysis; Theory, Methods & Applications. 2009 Mar 1; 70(5):2091–105.

35. Maraaba TA, Jarad F, Baleanu D. On the existence and the uniqueness theorem for fractional differential equations with bounded delay within Caputo derivatives. Science in China Series A: Mathematics. 2008 Oct 1; 51(10):1775–86.

36. Maraaba T, Baleanu D, Jarad F. Existence and uniqueness theorem for a class of delay differential equations with left and right Caputo fractional derivatives. Journal of Mathematical Physics. 2008 Aug; 49(8):083507.

37. Khan NA, Shaihk A. A smart amalgamation of spectral neural algorithm for nonlinear Lane-Emden equations with simulated annealing. Journal of Artificial Intelligence and Soft Computing Research. 2017 Jul 1; 7(3):215–24.

38. Khan NA, Shaikh A, Sultan F, Ara A. Numerical Simulation Using Artificial Neural Network on Fractional Differential Equations. In Numerical Simulation-From Brain Imaging to Turbulent Flows 2016. InTech.

39. Yang XJ, Baleanu D, Srivastava HM. Local fractional integral transforms and their applications. Academic Press; 2015 Oct 22.

40. Pao YH, Takefuji Y. Functional-link net computing: theory, system architecture, and functionalities. Computer. 1992 May; 25(5):76–9.

41. Lagaris IE, Likas A, Fotiadis DI. Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks. 1998 Sep; 9(5):987–1000. https://doi.org/10.1109/72.712178 PMID: 18255782

42. Ledesma S, Aviña G, Sanchez R. Practical considerations for simulated annealing implementation. InSimulated Annealing 2008. InTech.

43. Saeed U. Radial basis function networks for delay differential equation. Arabian Journal of Mathematics. 2016 Sep 1; 5(3):139–44.

44. Saeed U. Hermite wavelet method for fractional delay differential equations. Journal of Difference Equations. 2014; 2014.

45. Iqbal MA, Saeed U, Mohyud-Din ST. Modified Laguerre wavelets method for delay differential equations of fractional-order. Egypt. J. Basic Appl. Sci. 2015 Mar 1; 2:50.