

Article

# The Numerical Solution of Nonlinear Fractional Lienard and Duffing Equations Using Orthogonal Perceptron

Akanksha Verma <sup>1</sup>, Wojciech Sumelka <sup>2,\*</sup> and Pramod Kumar Yadav <sup>3</sup>

<sup>1</sup> Department of Mathematics, Dyal Singh College, University of Delhi, New Delhi 110003, India; akankshaverma.maths@dsc.du.ac.in

<sup>2</sup> Institute of Structural Analysis, Poznan University of Technology, Piotrowo 5 Street, 60-965 Poznan, Poland

<sup>3</sup> Department of Mathematics, Motilal Nehru National Institute of Technology Allahabad, Prayagraj 211004, India; pramodky@mnnit.ac.in

\* Correspondence: wojciech.sumelka@put.poznan.pl

**Abstract:** This paper proposes an approximation algorithm based on the Legendre and Chebyshev artificial neural network to explore the approximate solution of fractional Lienard and Duffing equations with a Caputo fractional derivative. These equations show the oscillating circuit and generalize the spring–mass device equation. The proposed approach transforms the given nonlinear fractional differential equation (FDE) into an unconstrained minimization problem. The simulated annealing (SA) algorithm minimizes the mean square error. The proposed techniques examine various non-integer order problems to verify the theoretical results. The numerical results show that the proposed approach yields better results than existing methods.

**Keywords:** orthogonal neural network; simulated annealing optimization technique; fractional differential equations; Caputo derivative



**Citation:** Verma, A.; Sumelka, W.; Yadav, P.K. The Numerical Solution of Nonlinear Fractional Lienard and Duffing Equations Using Orthogonal Perceptron. *Symmetry* **2023**, *15*, 1753. <https://doi.org/10.3390/sym15091753>

Academic Editor: Sergio Elaskar

Received: 20 July 2023

Revised: 29 August 2023

Accepted: 6 September 2023

Published: 13 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In ancient times, fractional calculus was used by mathematicians due to its several applications in applied mathematics as well as mathematical physics. Recently, fractional differential equations have been used to model many real-world problems in circuit theory, fluid dynamics, physics, mathematical biology, quantum mechanics, electrochemistry, etc. Also, it is well known that non-integer-order derivatives control the models efficiently. Talebi et al. [1] explored the application of fractional calculus to filtering structures for  $\alpha$ -stable systems, where  $\alpha$ -stable distributions are a class of probability distributions that generalize the Gaussian distribution and can describe asymmetric and heavy-tailed behavior. These distributions are encountered in real-world scenarios, including financial time series and communication channels. Fractional-order filters and processing methods might provide better tools for dealing with such systems. Therefore, studying these equations and finding their solutions is necessary. The general form of the Lienard equation [2] is given by

$$z''(t) + f(z)z'(t) + g(z) = h(t). \quad (1)$$

Various types of selection of the functions  $f$ ,  $g$ , and  $h$  give distinct models. For example, if  $f(z)z'(t)$  is the damping force,  $g(z)$  is the restoring force, and  $h(t)$  is the external force, then it forms the damped pendulum equation. Moreover, if we have  $f(z) = \epsilon(z^2 - 1)$ ,  $g(z) = z$ , and  $h(t) = 0$ , then Equation (1) is transferred to the Van der Pol equation [3], representing a nonlinear electronic oscillation model. However, it is well known that the exact solution of Equation (1) is a complex problem.

Kong [4] and Feng [5] investigated the exact solution of the Lienard equation in the form

$$z''(t) + Lz'(t) + Mz^3 + Nz^5 = 0, \quad (2)$$

where,  $L$ ,  $M$ , and  $N$  are constant coefficients.

The general form of the Duffing oscillator equation is

$$z''(t) + Lz'(t) + Pz + Mz^3 = 0, \quad (3)$$

where,  $L$ ,  $M$ , and  $P$  are constant coefficients.

Some recent works in the literature, such as [6–9], focused on generalized forms of the Lienard and Duffing equations using fractional calculus. Fractional-order derivatives can explore various physical methods that vary with time and space [10,11]. Also, using fractional calculus principles is well established in several scientific fields. Bohner and Tunç [12] conducted a qualitative analysis of Caputo fractional integro-differential equations with constant delays. This work delves into the dynamics of such equations, shedding light on their behavior and properties and contributing to the advancement of their understanding in this specialized area of mathematics. In [13], the authors dived into fractional calculus and delay integro-differential equations. Their work, which focuses on Caputo proportional fractional derivatives, presents novel solution estimation techniques. By addressing these intricate equations, the authors contributed to advancing analytical methods in the context of fractional calculus and its applications. Many real-life phenomena are represented by the fractional Lienard equation and Duffing equation, such as oscillating circuit theory [14,15], the mass damping effect [16], and pipelines and fluid dynamics [17].

The general form of the fractional order Lienard equation is given as follows:

$$D^\nu z(t) + LD'z(t) + Mz^3 + Nz^5 = 0, \quad 1 < \nu \leq 2, \quad t \in [0, 1], \quad (4)$$

with respect to

$$z(0) = \alpha, \quad z'(0) = \beta, \quad \text{where } \alpha \text{ and } \beta \text{ are constants.} \quad (5)$$

Also, the fractional Duffing equation with the damping effect is given as follows:

$$D^\eta z(t) + LD'z(t) + Pz(t) + Mz^3 = 0, \quad 1 < \eta \leq 2, \quad t \in [0, 1], \quad (6)$$

subject to

$$z(0) = \gamma \quad z'(0) = \delta, \quad \text{where } \gamma \text{ and } \delta \text{ are constants.} \quad (7)$$

In the literature, many analytical and numerical approaches exist for solving Equations (4) and (6). In 2004, Feng [18] explicitly presented the exact solution of the Lienard equation and provided some applications. In 2008, Matinfar et al. [19] used the variation iteration technique to solve the Lienard equation and compared the numerical solutions obtained with the analytic solution. Furthermore, Xu [20] acquired the eight types of explicit analytical solutions of the Lienard equation, which included periodic wave solutions and solitary wave solutions in terms of elliptic Jacobian and trigonometric functions. Janiczek [21] demonstrated the modulating functions method for all models described by fractional differential equations. Modulating functions are used to reduce the order of derivatives in an equation, generate equations without derivatives of output signals, and eliminate the need to solve differential equations. Chebyshev's operational matrix method for solving the multi-term fractional-order ordinary differential equation was proposed by Atabakzadeh et al. [22]. To apply this approach, they first converted the given problem into a system of fractional ODEs and then used the Chebyshev operational matrix method. Again, Kazem [23] analyzed fractional ODEs via an integral operational matrix approach based on Jacobi polynomials. Nourazar and Mirzabeigy [16] proposed a modified differential transform technique to deal with the fractional Duffing equation with a damping effect. In 2016, Ezz-Eldien [24] discovered a new numerical approach to solving fractional variational problems. Furthermore, Gómez-Aguilar et al. [17] used the Laplace homotopy analysis technique with a new fractional derivative without a singular kernel to solve the fractional Lienard equation that describes the fluid dynamics of the pipeline. The

homotopy analysis method was implemented in [25,26] to solve the Duffing oscillator and the Lienard problem with a fractional derivative. Recently, Singh et al. [14,15,27,28] also made several effective attempts to find the solutions to Equations (4) and (6) by using a variety of techniques. Also, Kumar et al. [29] used the Rabotnov fractional exponential kernel to solve the nonlinear Lienard equation numerically. More recently, Adel [30] demonstrated an approach based on Bernoulli collocation and shifted Chebyshev collocation points to solve Equations (4) and (6).

In recent years, neural architecture-based approximation schemes have been used to solve FDEs, ODEs, PDEs, and delay differential equations (DDEs) [31–38]. In 2013, Lefik [39] illustrated that an ANN performs the numerical representation of the inverse relation. It can be used as many times as needed in the same application, replacing traditional “ad hoc” back computation for any new piece of experimental data. Malik et al. [40] proposed a hybrid heuristic approach to solve the Lienard equation based on genetic algorithms, such as memetic computation, combining genetic algorithms, the interior-point algorithm, and the active set algorithm. Furthermore, Mall and Chakraverty [33,36] used the multilayer perceptron and functional connection neural network with regression-based parameters to solve ODEs. In [34,38], the authors also used the multilayer perceptron technique with quasi-Newton and Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithms to solve the singular initial and boundary value problems. Kumar et al. [41] presented a comparative analysis of two distinct neural modeling approaches to approximate the multidimensional poverty levels within an Indian state. This study sought to provide useful information for choosing the best modeling approaches to determine poverty levels in the Indian setting by examining their performance and precision. Sahoo and Chakraverty [42] proposed a symplectic artificial neural network to handle nonlinear systems arising in dusty plasma models. They presented the dynamics of Van der Pol–Mathieu–Duffing oscillator problems for different excitation functions using the proposed method, and numerical simulations and graphical representations were carried out to establish the accuracy of the presented algorithm.

Motivated by the above, in this manuscript, we discuss the functional link neural network architecture, which is a single-layer neural network. This article aims to find the solutions to fractional Lienard and Duffing equations using functional link neural networks. This technique offers us the following attractive features:

- The proposed technique gives us the solution in a closed analytic form.
- The functional link neural network consists of a single layer, and thus the number of network parameters is less than the traditional multilayer ANN and works with low computational complexity.
- It is capable of fast learning and is computationally efficient.
- This process does not need linearization to solve a nonlinear problem.

We have organized the present article as follows. Section 2 includes some important preliminaries and discusses the structure of the Chebyshev and Legendre neural networks. Section 3 discusses the methodology, including a well-explained algorithm and the implication protocol, while Section 4 discusses the numerical experiments and their results. Section 5 deals with the error analysis of the technique, while Section 6 concludes the work.

## 2. Preliminaries

This section provides some basic definitions and results related to the Chebyshev and Legendre artificial neural network models used in this paper. We begin with the following definitions:

**Definition 1.** Suppose that  $r$  and  $s$  are positive integers such that  $r - 1 < s < r$ . Then, for a continuous function  $p(x)$ , the Riemann–Liouville derivative and integral of a fractional order  $s$  are given by

$${}^R D_x^s p(x) = \frac{1}{\Gamma(r-s)} \frac{d^r}{dx^r} \int_a^x (x-\tau)^{r-s-1} p(\tau) d\tau,$$

and

$${}^R D_x^{-s} p(x) = \frac{1}{\Gamma_s} \int_a^x (x - \tau)^{s-1} p(\tau) d\tau,$$

respectively.

**Definition 2.** Suppose that  $r$  and  $s$  are positive integers such that  $r - 1 < s < r$ . Then, for a continuous function  $p(x)$ , the Caputo derivative and integral of a fractional order  $s$  are given by

$${}^C D_x^s p(x) = \frac{1}{\Gamma(r-s)} \int_a^x \frac{p^{(r)}(\tau)}{(x-\tau)^{s+1-r}} d\tau, \quad r-1 \leq s \leq r,$$

and

$${}^C D_x^{-s} p(x) = \frac{1}{\Gamma_s} \int_a^x (x - \tau)^{s-1} p(\tau) d\tau,$$

respectively.

**Remark 1.** Note that the Caputo definition has the advantage of being bound, which means that the derivative of a constant is equal to zero. It also allows for the consideration of easily understood initial conditions.

### 2.1. Chebyshev and Legendre Artificial Neural Network Models

Chebyshev and Legendre artificial neural networks are functional link neural networks with one single layer. Both models can learn fast and have functional expansion blocks to enhance input patterns. In 1995, Pao and Philips [43] presented an approach based on a single-layer functional connection neural network. The architectures of the Chebyshev and Legendre neural network models are given as follows.

**The architecture of the Chebyshev neural network model:** The structure of the ChNN model is shown in Figure 1. It consists of a single input node, a Chebyshev polynomial function expansion block, and an output node. Assume that  $t$  is a single input node with  $k$  data; that is,  $(t = t_1, t_2, \dots, t_k)^T$ .  $N(t, p)$  is the output of the feedforward neural network.

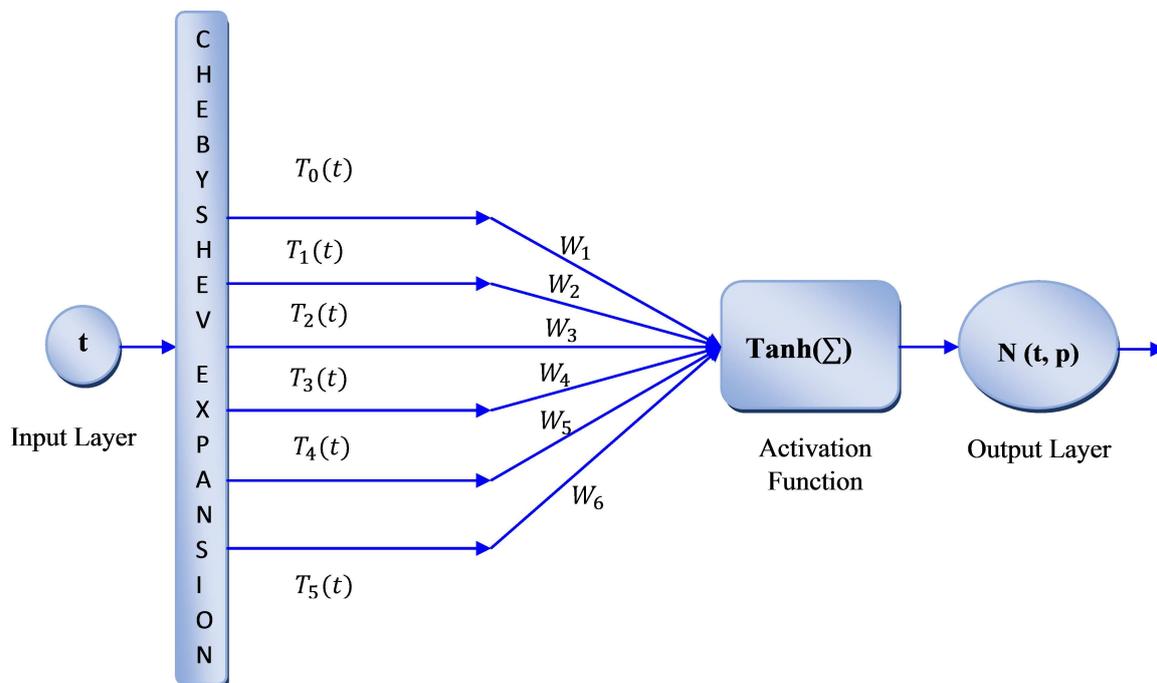


Figure 1. Structure of Chebyshev neural network.

The following are the first two Chebyshev polynomials:

$$T_0[t] = 1,$$

$$T_1[t] = t.$$

The well-known recurrence relation can be used to obtain higher-order Chebyshev polynomials:

$$T_{n+1}(t) = 2tT_n(t) - T_{n-1}(t).$$

**The architecture of the Legendre neural network model:** The structure of the LeNN model is depicted in Figure 2. It has a single input node, a Legendre polynomial function expansion block, and an output node. Assume that  $t$  is a single input node with  $l$  data; that is,  $(t = t_1, t_2, \dots, t_l)^T$ .  $N(t, p)$  is the output of the feedforward neural network.

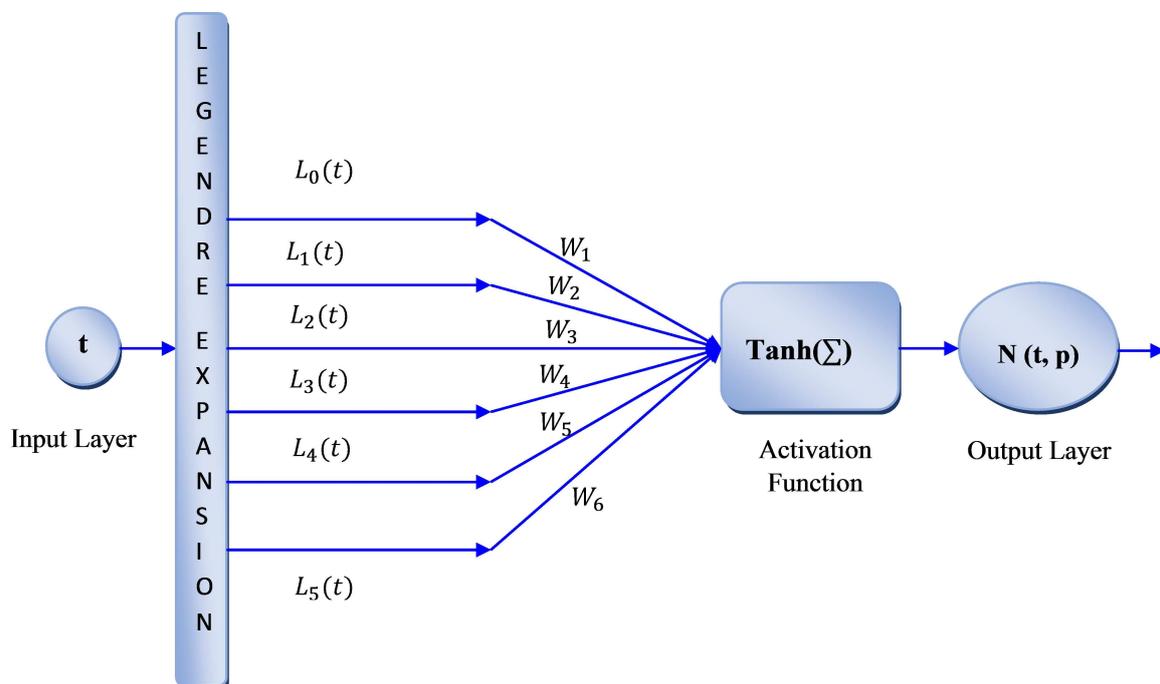


Figure 2. Structure of Legendre neural network.

Some Legendre polynomials are as follows:

$$L_0[t] = 1,$$

$$L_1[t] = t,$$

$$L_2[t] = \frac{1}{2}(3t^2 - 1).$$

The well-known recurrence relation can be used to obtain higher-order Legendre polynomials:

$$L_{n+1}(t) = \frac{1}{n+1}[(2n+1)tL_n(t) - nL_{n-1}(t)].$$

## 2.2. Simulated Annealing Algorithm

Simulated annealing is a straightforward stochastic function minimizer. It is inspired by the annealing process, which involves heating a metal piece to a high temperature and letting it cool gently. This process enables the metal's atomic structure to settle into a lower energy state, making it harder. Using optimization terminology, annealing enables the structure to leave a local minimum and to search for and arrive at a better, ideally global minimum. A new point is generated randomly near the present point with each iteration

of the simulated annealing technique. The radius of the neighborhood decreases with each iteration.

### 3. Description of the Method

#### 3.1. General Remarks

The orthogonal perceptron expands upon the classical perceptron algorithm utilized for binary classification applications. The perceptron is a fundamental building element that accepts input values, adds weights to them, and generates output in the context of machine learning and neural networks. The orthogonal perceptron introduces a significant change: it requires orthogonality between the weight vectors connected to various classes. The weight vectors are modified in conventional perceptrons to reduce misclassification errors. However, in the orthogonal perceptron, the weight vectors for various classes are restricted to be orthogonal to one another, in addition to being modified to minimize mistakes. In higher-dimensional spaces, the orthogonal constraint provides a discriminating component that can better explain class separation. Orthogonal weight vectors can lead to better decision boundaries and enhanced generalization.

A flow chart demonstrates the proposed technique in Figure 3. We have also explained the methodology stepwise.

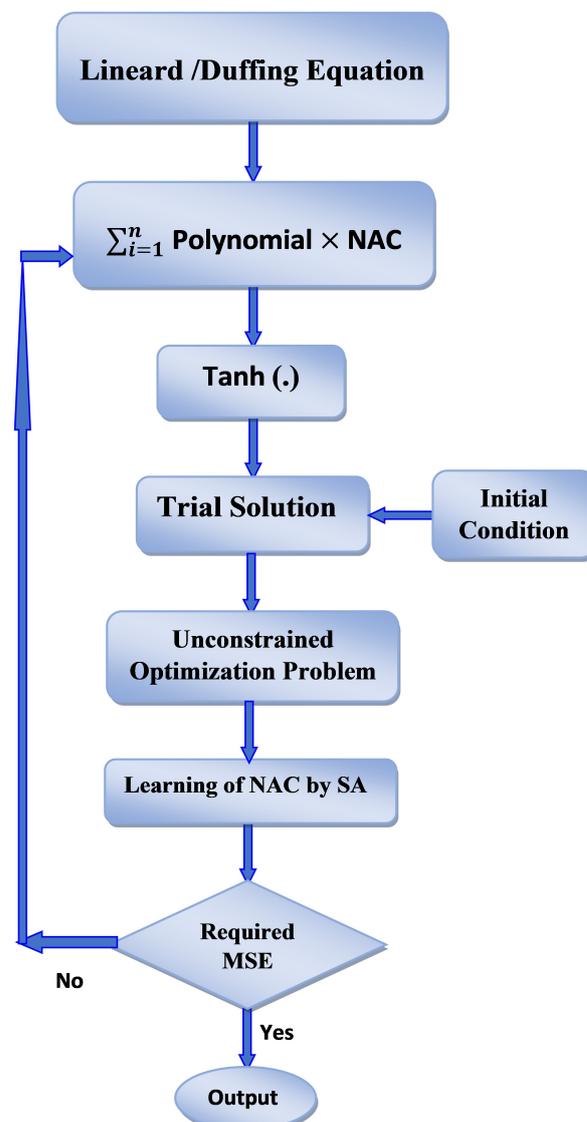


Figure 3. Pictorial presentation of the algorithm.

Due to the similarity of both techniques, we discuss the combined steps for the Chebyshev and Legendre ANN techniques. For both methods, the entire procedure is the same except for the polynomial selection

- Prepare the network by using the Chebyshev or Legendre polynomials.
- Provide the network adaptive coefficients (NACs) for each polynomial.
- Calculate and save the sum of the products of the NACs and Chebyshev polynomials or Legendre polynomials as a function  $\mu$  or  $\delta$ .
- Use the Taylor series extension of the activation function  $\tanh(\cdot)$  to activate the function  $\mu$  or  $\delta$ .
- Now, construct the problem's trial solution, which satisfies the given initial or boundary conditions.
- Substitute the derivatives of the trial solution into the given situation and obtain the error function.
- To minimize the error function, use simulated annealing optimization techniques.
- If the assessment of the mean square error is in an acceptable range, then collect the value of the adaptive coefficients of the network and substitute it into the trial solution to obtain the results; otherwise, repeat the same process for various values of the NACs until the agreeable MSE is acquired.

### 3.2. Employment on Nonlinear FDEs

Here, we apply the proposed technique to fractional-order nonlinear differential equations of the following type:

$$D^\nu z(t) = F(t, z, z'(t)), \quad (8)$$

subjected to the following conditions:

$$z(0) = \alpha, \quad z'(0) = \beta.$$

Equation (8) can be written in the following form to apply the Chebyshev and Legendre ANN techniques:

$$D^\nu z_{tr}(t, \psi) - F(t, z_{tr}(t, \psi), Dz_{tr}(t, \psi)) = 0, \quad t \in [0, 1], \quad 1 < \nu \leq 2, \quad (9)$$

where  $z_{tr}$  is the trial solution of Equation (8) which satisfies the given initial or boundary conditions and  $\psi$  is network adaptive coefficients known as bias and weight. Let us try the trial solution for Equation (8) expressed as

$$z_{tr}(t, \psi) = A + tN(t, \psi), \quad (10)$$

where  $N$  is the network output for the Chebyshev ANN. We apply the Taylor series to the  $\tanh(\cdot)$  activation function to activate the sum of the product of the weights and orthogonal polynomials such that

$$N = \tanh(\mu) = \mu - \frac{\mu^3}{3} + \frac{2\mu^5}{15}, \quad (11)$$

Here,  $\mu$  is given by

$$\mu = \sum_{i=1}^n \psi_i T_{i-1}, \quad (12)$$

where  $T_{i-1}$  represents the Chebyshev polynomials, which can be characterized with the following recursive formula:

$$T_{i+1}(t) = 2tT_i(t) - T_{i-1}(t), \quad i \geq 2, \quad (13)$$

Here,  $T_0(t) = 1$  and  $T_1(t) = t$  are the fundamental values of the Chebyshev polynomials.

For the Legendre ANN, the activation function  $N$  is defined by

$$N = \tanh(\delta) = \delta - \frac{\delta^3}{3} + \frac{2\delta^5}{15}, \quad (14)$$

Here,  $\delta$  is given by

$$\delta = \sum_{i=1}^n \psi_i L_{i-1}, \quad (15)$$

where  $L_{i-1}$  represents the Legendre polynomials, which are defined by the following recursive formula:

$$L_{i+1}(t) = \frac{1}{(i+1)}(2i+1)tL_i(t) - \frac{1}{(i+1)}iL_{i-1}(t), \quad i \geq 2, \quad (16)$$

Here,  $L_0(t) = 1$  and  $L_1(t) = t$  are the fundamental values of the Legendre polynomial.

Now, we can write the trial solution in terms of  $\mu$  and  $\delta$  for the Chebyshev and Legendre ANNs, respectively, and obtain the following:

$$z_{tr}(t, \psi) = A + t \left( \mu - \frac{\mu^3}{3} + \frac{2\mu^5}{15} \right) \quad \text{and} \quad = A + t \left( \delta - \frac{\delta^3}{3} + \frac{2\delta^5}{15} \right), \quad (17)$$

Now, we put the values of  $\mu$  and  $\delta$  into Equation (17) for  $n = 2$  and obtain

$$z_{tr}(t, \psi) = A + t \left( (\psi_1 + t\psi_2) - \frac{(\psi_1 + t\psi_2)^3}{3} + \frac{2(\psi_1 + t\psi_2)^5}{15} \right). \quad (18)$$

By applying the Caputo fractional-order derivative, we obtain

$$\begin{aligned} D^\nu z_{tr}(t, \psi) &= \frac{\Gamma 2}{\Gamma(2-\nu)} t^{1-\nu} \left( \psi_1 - \frac{\psi_1^3}{3} + \frac{2\psi_1^5}{15} \right) + \frac{2}{3} \frac{\Gamma 6}{\Gamma(6-\nu)} t^{5-\nu} (\psi_1 \psi_2^4) + \frac{2}{15} \frac{\Gamma 7}{\Gamma(7-\nu)} t^{6-\nu} (\psi_2^5) \\ &+ \frac{\Gamma 3}{\Gamma(3-\nu)} t^{2-\nu} \left( \psi_2 - \psi_1^2 \psi_2 + \frac{2}{3} \psi_1^4 \psi_2 \right) + \frac{\Gamma 4}{\Gamma(4-\nu)} t^{3-\nu} \left( \frac{4}{3} \psi_1^3 \psi_2^2 - \psi_1 \psi_2^2 \right) + \\ &\frac{\Gamma 5}{\Gamma(5-\nu)} t^{4-\nu} \left( \frac{4}{3} \psi_1^2 \psi_2^3 - \frac{\psi_2^3}{3} \right). \end{aligned} \quad (19)$$

Also, the mean square error for Equation (8) can be calculated as follows:

$$E(\psi) = \sum_{j=1}^m \frac{1}{m} (D^\nu z_{tr}(t_j, \psi_i) - F(t_j, z_{tr}(t_j, \psi_i), Dz_{tr}(t_j, \psi_i)))^2, \quad t \in [0, 1]. \quad (20)$$

Here, we refer to Equation (20) as a fitness function and  $m$  as the number of trial points. We used a thermal minimization process known as simulated annealing to minimize the fitness function. This is a probabilistic method used to approximate the global optimum of a function. This method consists of three steps: perturb the solution, determine the consistency of the solution, and accept the solution if it is better than the improved one. The learning of NACs will be performed from Equation (18) by minimizing the MSE to the lowest possible acceptable minimum value.

### 3.3. Advantages of the Proposed Technique

1. To approximate the complex nonlinear interactions present in fractional differential equations, the orthogonal perceptron-based method uses the adaptability of artificial neural networks. It can capture complex behaviors that are exceedingly difficult to express using conventional numerical techniques.
2. Once trained, the orthogonal perceptron may generalize its learned patterns to fresh input data. This is extremely helpful when solving fractional differential equations with various initial conditions or parameters.

3. Traditional numerical techniques are frequently built with well-established convergence features. However, the neural network approach's convergence is determined by the quality of the data and the architecture used.

#### 4. Numerical Implementation

In this section, two fractional-order problems are solved using the ChNN and LeNN architectures. The numerical results show that the proposed technique is highly efficient and robust. All the computations were performed on a computer with an Intel Core i3 processor (Intel Corporation, Santa Clara, CA, USA) with 8 gigabytes of RAM, and the simulation was conducted with Mathematica 11.1.0 for each problem.

**Problem 1.** Consider the particular choice of the parameters  $L = -1$ ,  $M = 4$ , and  $N = 3$  in Equation (4). The fractional Lienard problem is given as follows [14,15,28]:

$$D^\nu z(t) - D'z(t) + 4z^3 + 3z^5 = 0, \quad 1 < \nu \leq 2, \quad t \in [0, 1], \quad (21)$$

$$z(0) = \alpha = \sqrt{\frac{\tau}{2 + \kappa}} \quad \text{and} \quad z'(0) = \beta = 0,$$

where

$$\tau = 4\sqrt{\frac{3a^2}{3b^2 - 16ac}} \quad \text{and} \quad \kappa = -1 + \frac{\sqrt{3}b}{\sqrt{(3b^2 - 16ac)}}. \quad (22)$$

For  $\nu = 2$ , the exact solution is already known with the given conditions

$$z(t) = \sqrt{\frac{\tau \sec h^2 \sqrt{-at}}{2 + \kappa \sec h^2 \sqrt{-at}}} \quad (23)$$

Equation (22) presents the values of  $\tau$  and  $\kappa$ .

As we discussed in Section 4, we constructed the trial solution as

$$z_{tr}(t, \psi) = \alpha + t^2 N(t, \psi). \quad (24)$$

The given Lienard problem was solved by ChNN and LeNN techniques for the various values of  $\nu$  and employed by dividing the domain into 10 equidistant training points with 6 NACs. The acquired appropriate MSEs were  $1.39515 \times 10^{-10}$  and  $5.27835 \times 10^{-10}$ , respectively. The computational time for Problem 1 was 0.05 s and 0.09 s, respectively. Table 1 shows the accurate values of the NACs after training by the SA algorithm. In Table 2, we have listed the approximated solution by our methods (ChNN and LeNN), the methods of Singh [14,15,28], and the exact solution. Table 2 shows the good agreement with these methods.

**Table 1.** The ideal estimations of NAC (Problem 1).

NAC	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$
ChNN	-0.41174	0.379264	0.463621	0.386701	0.0344511	0.0811897
LeNN	-1.53305	1.02509	-2.11092	1.1266	-0.969482	0.15995

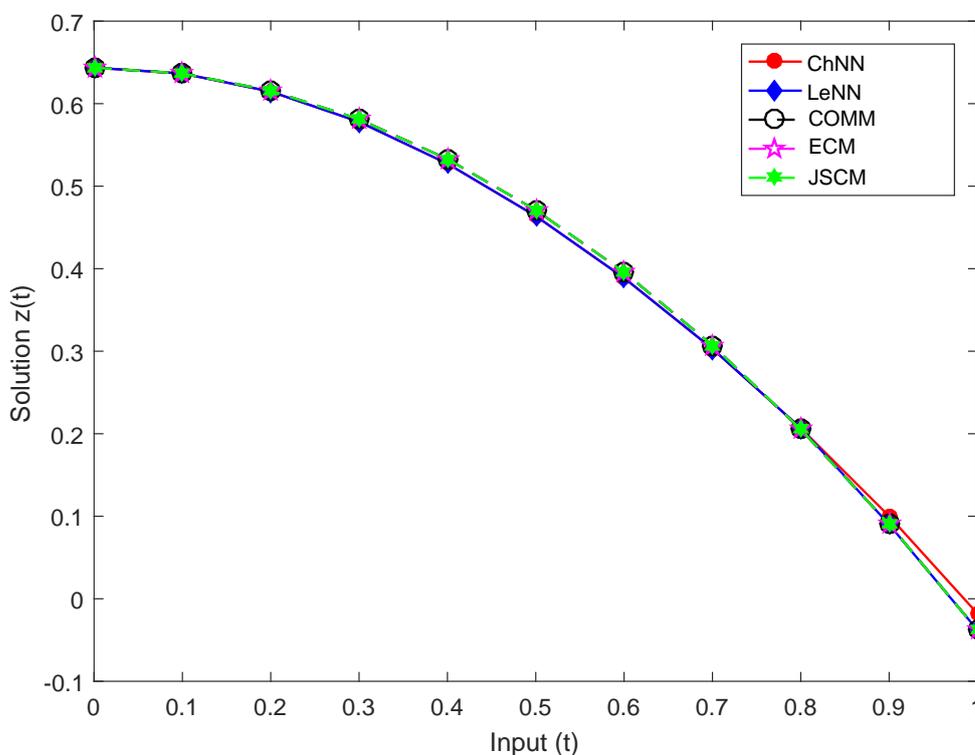
**Table 2.** Comparison of numerical results at  $\nu = 2$  for Problem 1.

$t$	Exact Solution	ChNN	LeNN	[28]	[14]	[15]
0.00	0.643594	0.643594	0.643594	0.643594	0.643594	0.643594
0.01	0.643556	0.643524	0.643524	0.643524	0.643524	0.643524
0.02	0.643443	0.643313	0.643313	0.643313	0.643313	0.643314
0.03	0.643255	0.642959	0.642959	0.642959	0.642959	0.642965
0.04	0.642991	0.642462	0.642462	0.642461	0.642461	0.642477

**Table 2.** Cont.

$t$	Exact Solution	ChNN	LeNN	[28]	[14]	[15]
0.05	0.642653	0.641821	0.641821	0.641818	0.641818	0.641894
0.06	0.642239	0.641033	0.641033	0.641029	0.641028	0.641082
0.07	0.641751	0.640100	0.640100	0.640093	0.640092	0.640176
0.08	0.641189	0.639019	0.639019	0.639009	0.639009	0.639130
0.09	0.640553	0.637790	0.637790	0.637777	0.637776	0.637946
0.1	0.639844	0.636413	0.636413	0.636395	0.636395	0.636623

In Figure 4, we have compared the approximate solutions by the LeNN and ChNN methods with the solutions obtained by the methods given in [14,15,28]. Figures 5 and 6 show the approximate solutions for the various values of  $\nu$ .



**Figure 4.** Comparison of approximate solutions at  $\nu = 2$  by ChNN, LeNN Method with [28], [14], [15] (Problem 1).

From Figures 5 and 6, we observed that the solution varied continuously from the fractional-order solution to the integer order. Therefore, we can say that the behaviors of approximate solutions for different fractional orders converge to integer-order solutions.

**Problem 2.** Consider the particular choice of the parameters  $L = 0.5, P = 25,$  and  $M = 25$  in Equation (6). The fractional duffing equation is given as follows [14,15,28]:

$$D^\eta z(t) + 0.5D'z(t) + 25z + 25z^3 = 0, \quad 1 < \eta \leq 2, \tag{25}$$

$$z(0) = \gamma = 0.1 \quad \text{and} \quad z'(0) = \delta = 0.$$

The exact solution of Problem 2 at  $\eta = 2$  when using the differential transform method [16] is given as follows:

$$z(t) = 0.1 - 1.2625t^2 + 0.2104t^3 + 2.6828t^4 - 0.5392t^5 - 2.6563t^6 + 0.6152t^7. \tag{26}$$

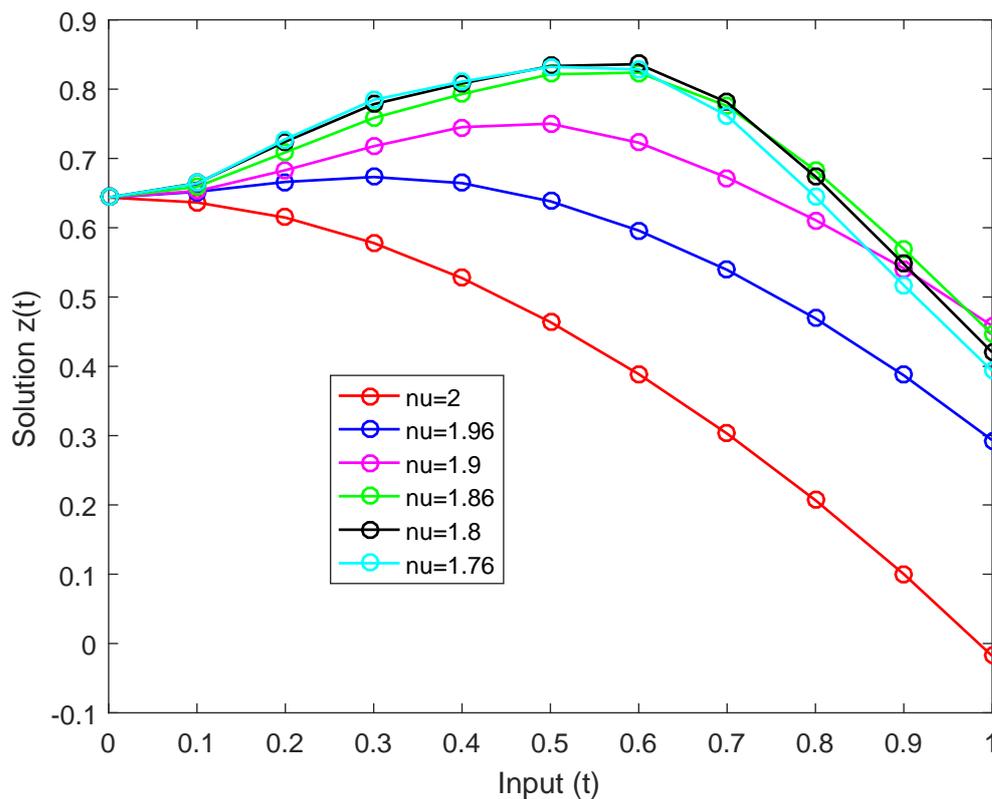


Figure 5. Nature of the approximate results with the LeNN method at  $\nu = 2, 1.96, 1.9, 1.86, 1.8, 1.76$  (Problem 1).

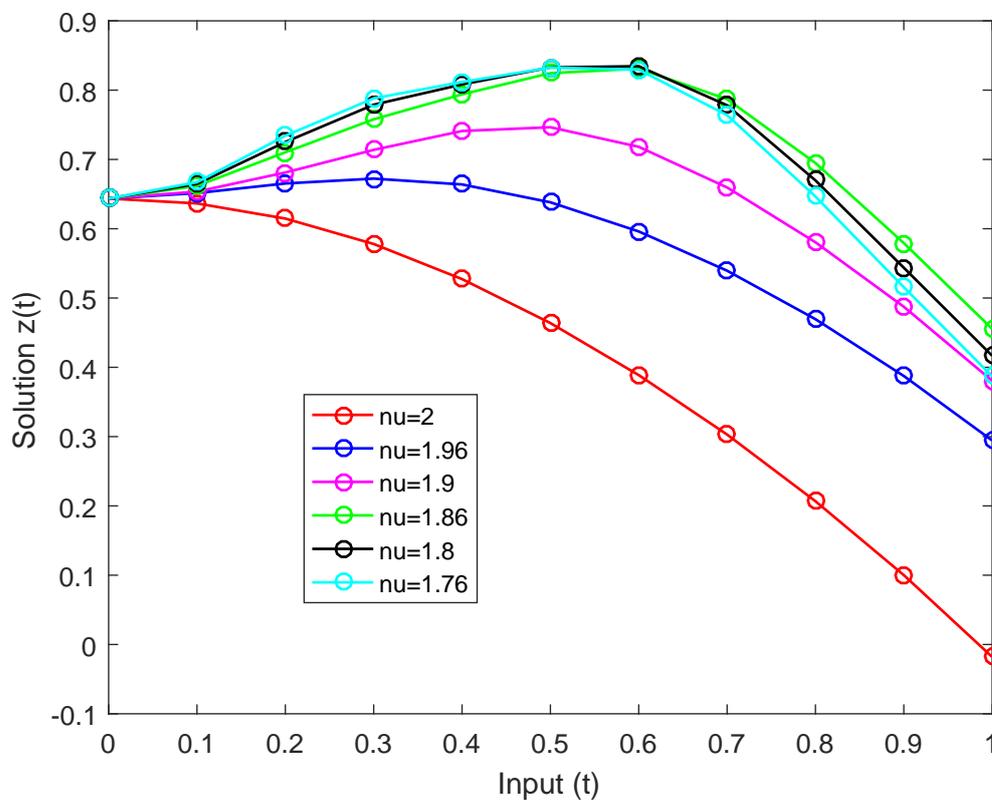


Figure 6. Nature of the approximate results with the ChNN method at  $\nu = 2, 1.96, 1.9, 1.86, 1.8, 1.76$  (Problem 1).

The trial solution can be written as

$$z_{tr}(t, \psi) = 0.1 + t^2 N(t, \psi). \tag{27}$$

The Duffing equation was solved with the ChNN and LeNN techniques for the various values of  $\eta$ . We trained the network using 10 equidistant points in the domain  $[0, 0.1]$  with 6 NACs. The obtained MSEs were  $1.61568 \times 10^{-9}$  and  $1.73644 \times 10^{-8}$ , respectively. The appropriate values of the NACs using the SA algorithm are given in Table 3. The computational time for Problem 2 was 0.05 s and 0.03 s, respectively. In Table 4, we have listed the obtained solutions by the proposed methods (ChNN and LeNN) and the existing method's solutions from [15,16,28]. Table 4 shows the good accuracy for the acquired results and the results given in [16].

**Table 3.** The ideal estimations of the NAC (Problem 2).

NAC	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$
ChNN	-0.84541	2.43209	0.54458	1.04621	-0.0457733	0.165949
LeNN	-1.54176	0.143329	-0.674402	-0.433421	-0.617103	-0.35283

**Table 4.** Comparison of numerical results at  $\eta = 2$  for Problem 2.

$t$	DTM [16]	ChNN	LeNN	[28]	[15]
0.00	0.100000	0.100000	0.100000	0.1000000	0.100000
0.01	0.099874	0.099874	0.099874	0.0998745	0.099874
0.02	0.099497	0.0994971	0.099497	0.0995025	0.099502
0.03	0.098871	0.0988715	0.0988715	0.0988894	0.098889
0.04	0.098002	0.0980002	0.0980001	0.0980414	0.098041
0.05	0.096886	0.0968865	0.0968864	0.0969644	0.096964
0.06	0.095534	0.0955346	0.0955344	0.0956646	0.095664
0.07	0.093949	0.093949	0.0939488	0.0941484	0.094148
0.08	0.092135	0.0921351	0.0921348	0.0924222	0.092422
0.09	0.090098	0.0900985	0.0900982	0.0904924	0.090492
0.1	0.087845	0.0878456	0.0878453	0.0883660	0.088366

In Figure 7, we have compared the approximate solutions with the proposed method and the solutions given in [15,16,28]. Figures 8 and 9 show the solutions for the different values of  $\eta$ .

From Figures 8 and 9, we observed that the solutions varied continuously from fractional-order solutions to integer-order solutions. Therefore, we can say that the behaviors of the approximate solutions for different fractional orders converged to an integer-order solution, and the periodic behavior of the solution can be seen.

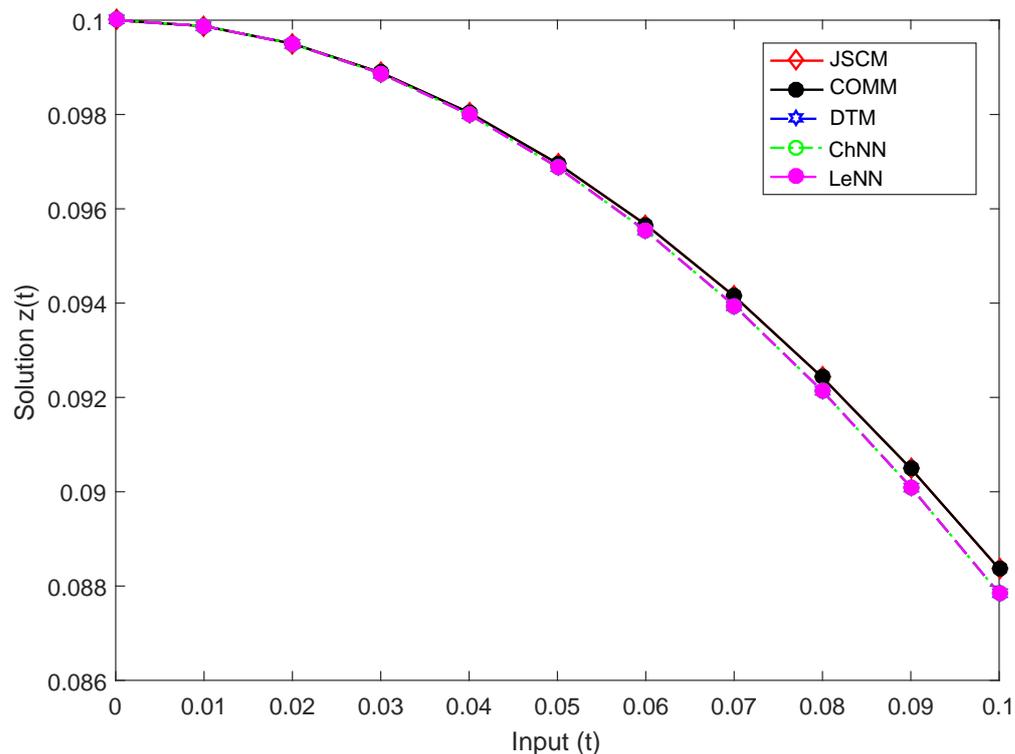
**Problem 3.** Consider the particular choice of the parameters  $L = -1$ ,  $M = 4$ , and  $N = -3$  in Equation (4). The fractional duffing equation is given as follows [14,44]:

$$D^\nu z(t) - D'z(t) + 4z^3 - 3z^5 = 0, \quad 1 < \nu \leq 2, \tag{28}$$

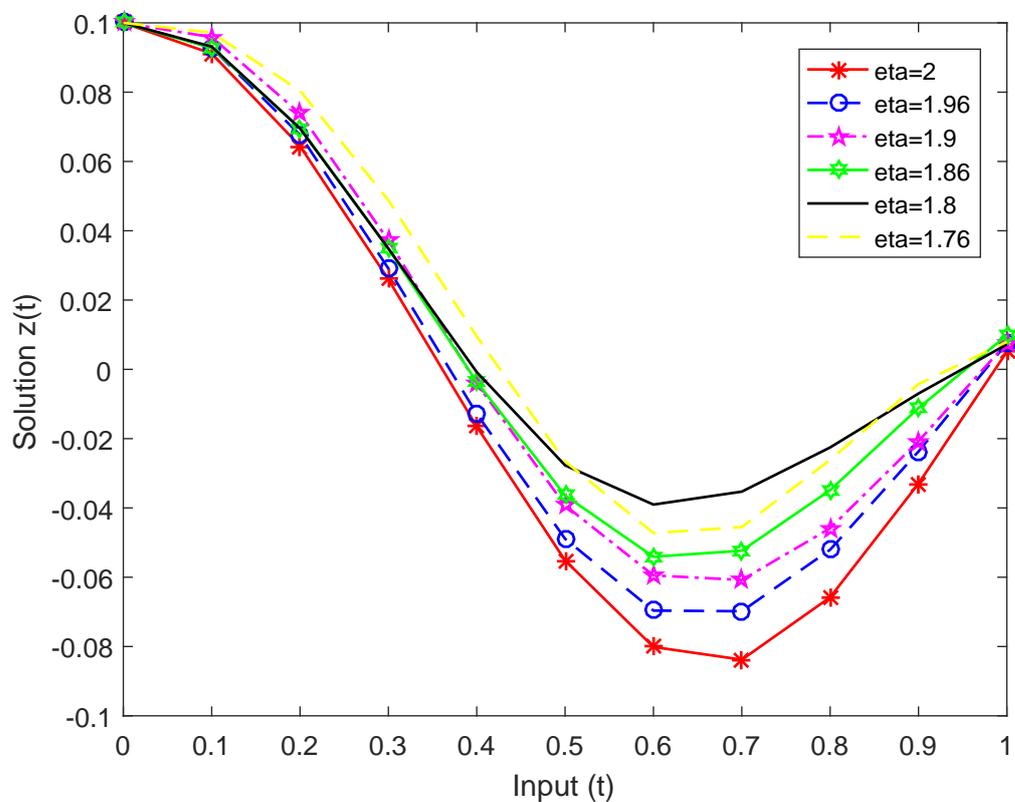
$$z(0) = c_1 = \sqrt{\frac{-2L}{M}} \quad \text{and} \quad z'(0) = c_2 = -\frac{L\sqrt{-L}}{M\sqrt{\frac{-2L}{M}}}.$$

The exact solution for the considered problem (Problem 3) for  $\nu = 2$  is given as follows:

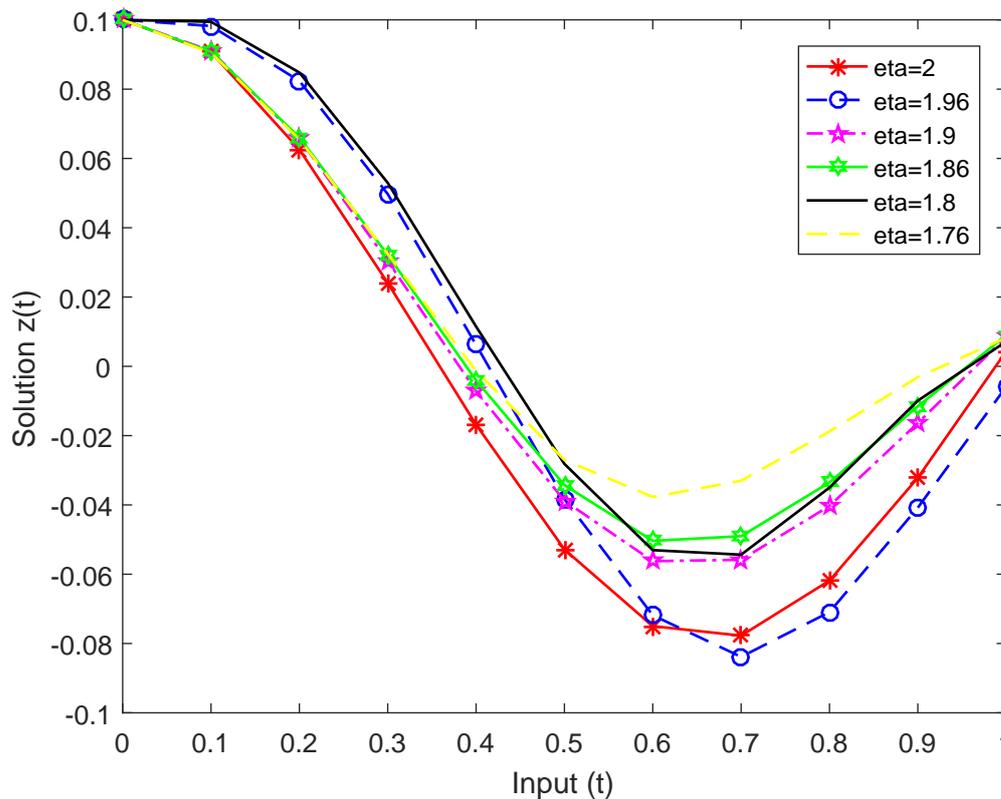
$$z(t) = \sqrt{\frac{-2L(1 + \tanh\sqrt{-Lt})}{M}}. \tag{29}$$



**Figure 7.** Comparison of approximate solutions at  $\eta = 2$  by ChNN, LeNN method with [16], [28], [15] (Problem 2).



**Figure 8.** Nature of the approximate results with the LeNN method at  $\eta = 2, 1.96, 1.9, 1.86, 1.8, 1.76$  (Problem 2).



**Figure 9.** Nature of the approximate results with the ChNN method at  $\eta = 2, 1.96, 1.9, 1.86, 1.8, 1.76$  (Problem 2).

The trial solution can be written as

$$z_{tr}(t, \psi) = c_1 + t^2 N(t, \psi). \tag{30}$$

The given Duffing equation was solved with the ChNN and LeNN techniques for the various values of  $\nu$ . We trained the network by taking 10 equidistant points in the domain  $[0, 0.1]$  with 6 NACs. The obtained MSEs were  $5.49991 \times 10^{-9}$  and  $2.69692 \times 10^{-10}$ , respectively. The computational time for the problem was 0.08 s and 0.09 s, respectively. The appropriate values for the NACs using the SA algorithm are given in Table 5. In Table 6, we listed the outcomes by the proposed method, analytic method and the solutions obtained by other existing numerical methods [15,44]. In Table 7, we have shown the absolute error between the exact solution and solution obtained by the proposed technique and other existing techniques.

**Table 5.** The ideal estimations of NAC (Problem 3).

NAC	$W_1$	$W_2$	$W_3$	$W_4$	$W_5$	$W_6$
ChNN	0.815413	1.23726	2.53108	1.03956	0.498807	0.307813
LeNN	-1.04706	-1.33436	0.254833	-1.04073	-0.113977	-0.295584

In Figure 10, we have compared the approximate solutions with the proposed methods with the exact solution and the solutions given in [15]. In Figures 11 and 12, we have presented the solutions to Problem 3 for the various values of  $\nu$ .

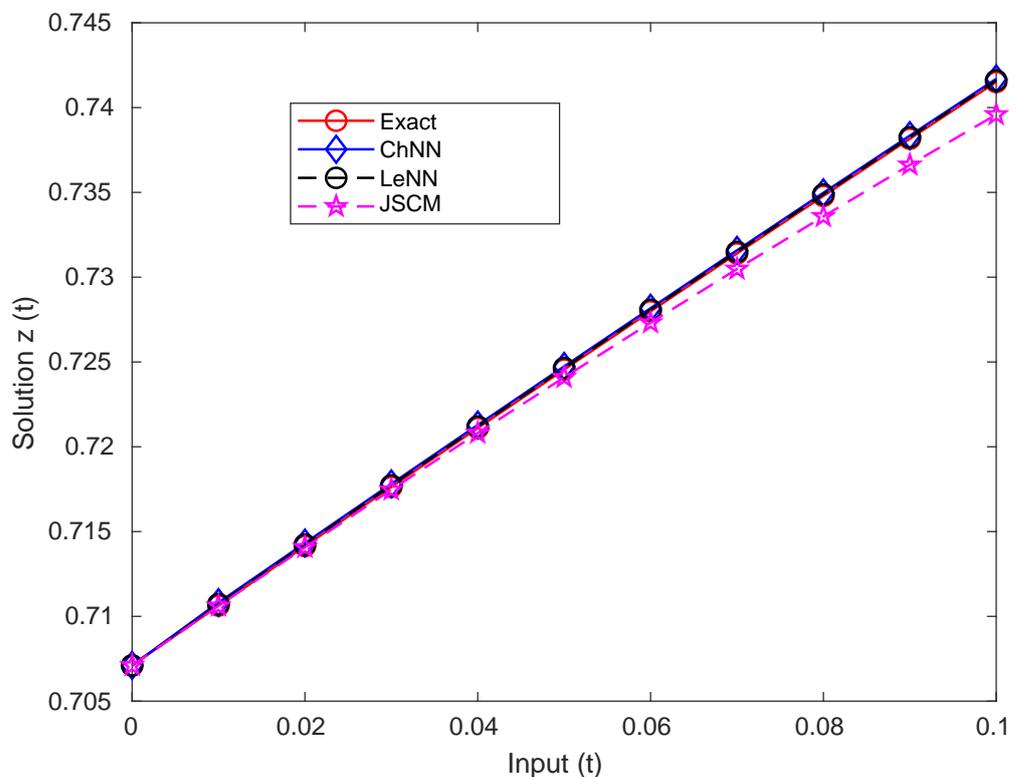
In Figures 11 and 12, we can see that the solution varied continuously from fractional order to integer order.

**Table 6.** Comparison of numerical results at  $\nu = 2$  for Problem 3.

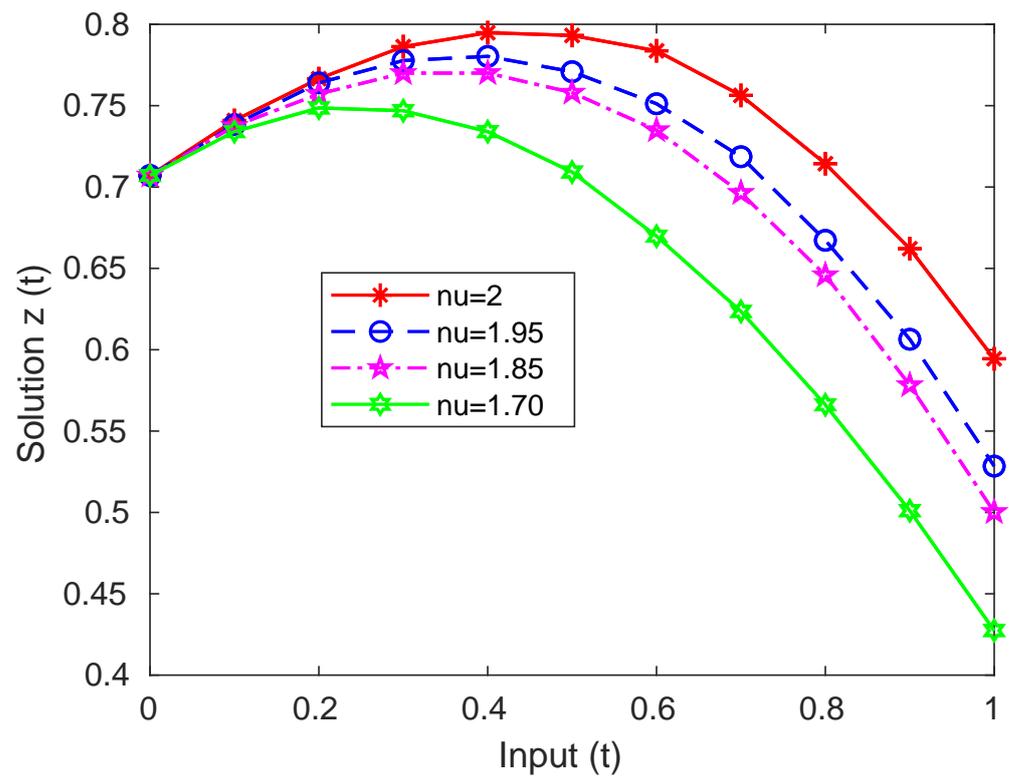
$t$	Exact	ChNN	LeNN	[15]	[44]
0.00	0.7071067	0.7071067	0.7071067	0.7071067	0.7071067
0.01	0.7106334	0.710811	0.710722	0.7106155	
0.02	0.7141419	0.71432	0.714231	0.7140699	0.7141419094
0.03	0.7176318	0.717811	0.717722	0.7174686	
0.04	0.7211028	0.721283	0.721193	0.7208102	0.7211028634
0.05	0.7245544	0.724736	0.724645	0.7240935	
0.06	0.7279862	0.728168	0.728077	0.7273171	0.7279862988
0.07	0.7313979	0.731581	0.731489	0.7304797	
0.08	0.7347890	0.734973	0.734881	0.7335800	0.7347890065
0.09	0.7381591	0.738344	0.738251	0.7366167	
0.1	0.7415079	0.741693	0.741601	0.7395886	0.7415079207

**Table 7.** Absolute error comparison from various numerical technique for Problem 3 at  $\nu = 2$ .

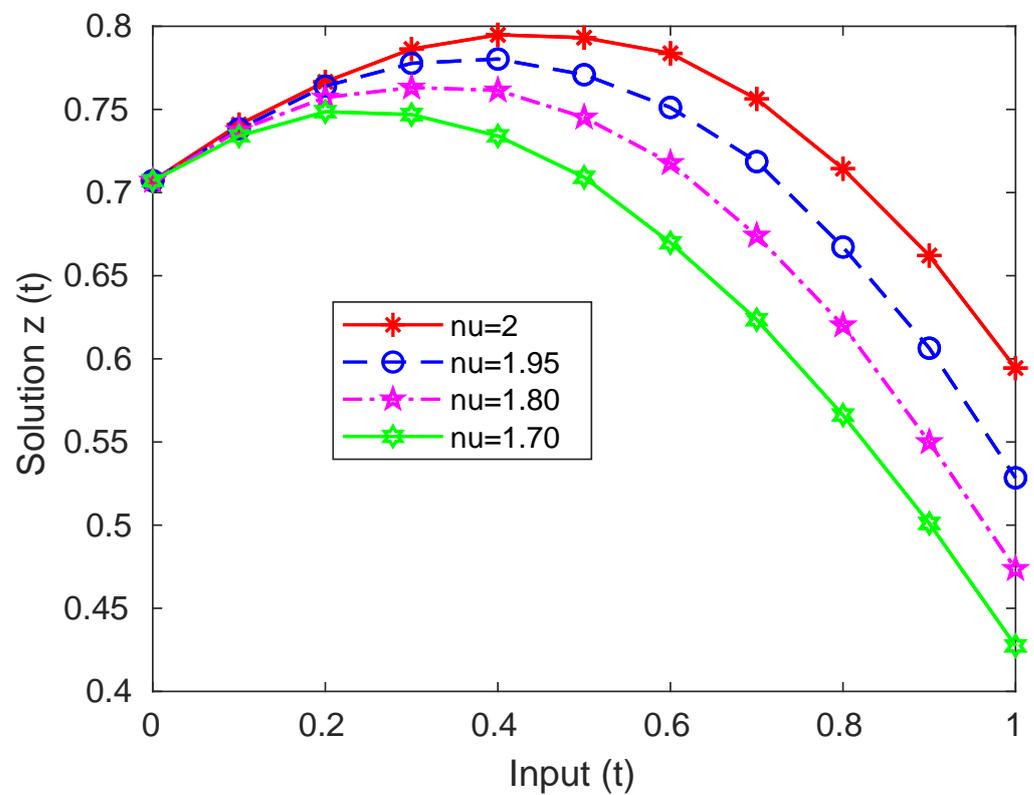
$t$	Abs. Error (ChNN)	Abs. Error (LeNN)	Abs. Error [15]	Abs. Error [44]
0.00	0	0	0	0
0.01	$1.776 \times 10^{-4}$	$8.85 \times 10^{-5}$	$1.79 \times 10^{-5}$	
0.02	$1.780 \times 10^{-4}$	$8.90 \times 10^{-5}$	$7.19 \times 10^{-5}$	$1.8669 \times 10^{-6}$
0.03	$1.790 \times 10^{-4}$	$9.02 \times 10^{-5}$	$1.63 \times 10^{-4}$	
0.04	$1.801 \times 10^{-4}$	$9.01 \times 10^{-5}$	$2.93 \times 10^{-4}$	$6.2706 \times 10^{-6}$
0.05	$1.810 \times 10^{-4}$	$9.06 \times 10^{-5}$	$4.61 \times 10^{-4}$	
0.06	$1.818 \times 10^{-4}$	$9.08 \times 10^{-5}$	$6.69 \times 10^{-4}$	$4.94502 \times 10^{-5}$
0.07	$1.831 \times 10^{-4}$	$9.12 \times 10^{-5}$	$9.18 \times 10^{-4}$	
0.08	$1.839 \times 10^{-4}$	$9.20 \times 10^{-5}$	$1.21 \times 10^{-3}$	$1.161249 \times 10^{-4}$
0.09	$1.849 \times 10^{-4}$	$9.19 \times 10^{-5}$	$1.54 \times 10^{-3}$	
0.1	$1.851 \times 10^{-4}$	$9.31 \times 10^{-5}$	$1.91 \times 10^{-3}$	$2.24737 \times 10^{-4}$



**Figure 10.** Comparison of approximate solutions at  $\eta = 2$  by ChNN, LeNN Method with exact solution and [15] (Problem 3).



**Figure 11.** Nature of the approximate results with the ChNN method at  $\nu = 2, 1.95, 1.85, 1.70$  (Problem 3).



**Figure 12.** Nature of the approximate results with the ChNN method at  $\nu = 2, 1.95, 1.80, 1.70$  (Problem 3).

## 5. Error Analysis

For the above problems, we have presented the error analysis of the numerical solutions of the ChNN and LeNN techniques. Initially, we trained the neural network using the SA algorithm and collected the appropriate values of the network parameters. After that, we substituted the NAC values into the trial solution and obtained the results for the ChNN or LeNN techniques (according to the polynomial). To analyze the precision of the method within the domain  $[0,1]$ , we also substituted it into Equation (31):

$$E(t, \psi) = |D^\nu z_{tr}(t, \psi) - F(t, z_{tr}(t, \psi), Dz_{tr}(t, \psi))| \cong 0 \quad (31)$$

where we found the  $z(t)$  approximated continuous results through the ChNN and LeNN techniques.  $E(t)$  approached 0 as the value of the MSE acquired with the ChNN and LeNN with the SA algorithm changed. The solution's convergence depends upon the optimization algorithm, the number of network adaptive coefficients, and the neural network's architecture, which we used.

For Problem 1, the mean square errors for the ChNN and LeNN techniques at  $\nu = 2$  were  $1.39515 \times 10^{-10}$  and  $5.27835 \times 10^{-10}$ , respectively, which showed that the minimum error for the ChNN and LeNN was  $1.3 \times 10^{-5}$ . This presents that both the strategies' precisions were inversely proportional to the mean square error value. When we exchanged the polynomials and used the SA algorithm for the network training, we observed that both techniques were strongly affected, as can be seen in Table 1.

Problem 2 is known as the fractional-order Duffing equation. For  $\eta = 2$ , the values of the MSE with 6 NACs by using ChNN and LeNN methods are  $1.61568 \times 10^{-9}$  and  $1.73644 \times 10^{-8}$  respectively. For the fractional value of  $\eta = 1.96, 1.9$ , the LeNN shows better results with minimum value for the MSE, however for  $\eta = 1.86, 1.8$  and  $1.76$  both techniques yielded a similar MSE. Problem 3 is also a fractional Lienard equation. We solved it approximately for the various values of  $\nu$  and obtained better results with less computational time than other existing numerical techniques [15,44].

## 6. Conclusions

This article has solved the Lienard and Duffing fractional- and integer-order equations using the ChNN and LeNN techniques with the SA algorithm. The proposed approach is easy to implement on nonlinear FDEs with simplicity of evaluation. The method's accuracy can be improved by enhancing the NAC learning methodology. The numerical results show that the proposed strategies give better results compared with other existing numerical techniques, such as DTM [16], the spectral collocation method [15], the Chebyshev operational matrix method [28], HAM [25], and the fractional homotopy analysis transform method [44]. For the solution of nonlinear problems, this technique does not require any linearization process. As a result, we can conclude that the method is exceptional and applicable to a broad range of nonlinear fractional-order differential equations that arise in engineering and science. The proposed technique can also be improved for precision in the future by improving the neural architecture and the learning technique of NACs. In the proposed technique, once the network has been trained, it allows continuous evaluation of the solution inside the domain. Also, it can be considered a powerful tool for the computation of nonlinear problems.

**Author Contributions:** Methodology and Simulation, A.V.; Conceptualization, W.S.; Supervision P.K.Y.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Science Centre, Poland under Grant No. 2017/27/B/ST8/00351.

**Data Availability Statement:** Not Applicable

**Acknowledgments:** We express our sincere thanks to Late Manoj Kumar, who provided the inspiration for this research.

**Conflicts of Interest:** The authors declare no conflicts of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

FDE	Fractional differential equation
SA	Simulated annealing
ChNN	Chebyshev neural network
LeNN	Legendre neural network
NAC	Network adaptive coefficient
COMM	Chebyshev Operational Matrix Method
JSCM	Jacobi Spectral Collocation Method
DTM	Differential Transform Method
ECM	Efficient Computational Method

### References

1. Talebi, S.P.; Godsill, S.J.; Mandic, D.P. Filtering Structures for  $\alpha$ -Stable Systems. *IEEE Control Syst. Lett.* **2023**, *7*, 553–558. [\[CrossRef\]](#)
2. Liénard, A. Etude des oscillations entretenues. *Rev. Gen. l'Electr.* **1928**, *23*, 901–912.
3. Guckenheimer, J. Dynamics of the Van der pol equation. *IEEE Trans. Circuit Syst.* **1980**, *27*, 938–989. [\[CrossRef\]](#)
4. Kong, D.X. Explicit exact solutions for the Lienard equation and its applications. *Phys. Lett. A* **1995**, *196*, 301–306. [\[CrossRef\]](#)
5. Feng, Z. On explicit exact solutions for the Lienard equation and its applications. *Phys. Lett. A* **2002**, *293*, 50–56. [\[CrossRef\]](#)
6. Diethelm, K.; Walz, G. Numerical solution of fractional order differential equations by extrapolation. *Numer. Algorithms* **1997**, *16*, 231–253. [\[CrossRef\]](#)
7. Esmaeili, S.; Shamsi, M.; Luchko, Y. Numerical solution of fractional differential equations with a collocation method based on Müntz polynomials. *Comput. Math. Appl.* **2011**, *62*, 918–929. [\[CrossRef\]](#)
8. Demirci, E.; Ozalp, N. A method for solving differential equations of fractional order. *J. Comput. Appl. Math.* **2012**, *236*, 2754–2762. [\[CrossRef\]](#)
9. Han, W.; Chen, Y.M.; Liu, D.Y.; Boutat, D. Numerical solution for a class of multi-order fractional differential equations with error correction and convergence analysis. *Adv. Differ. Equ.* **2018**, *253*, 253. [\[CrossRef\]](#)
10. Sumelka, W.; Luczak, B.; Gajewski, T.; Voyiadjis, G.Z. Modelling of AAA in the framework of time-fractional damage hyperelasticity. *Int. J. Solids Struct.* **2020**, *206*, 30–42. [\[CrossRef\]](#)
11. Sumelka, W. Fractional calculus for continuum mechanics—Anisotropic non-locality. *Bull. Pol. Acad. Sci. Tech. Sci.* **2016**, *64*, 361–372. [\[CrossRef\]](#)
12. Bohner, M.; Tunç, O.; Tunç, C. Qualitative analysis of caputo fractional integro-differential equations with constant delays. *Comput. Appl. Math.* **2021**, *40*, 214. [\[CrossRef\]](#)
13. Tunç, O.; Tunç, C. Solution estimates to Caputo proportional fractional derivative delay integro-differential equations. *Rev. R. Acad. Cienc. Exactas Fis. Nat. Ser. A Mat.* **2023**, *117*, 12. [\[CrossRef\]](#)
14. Singh, H. An efficient computational method for non-linear fractional Lienard equation arising in oscillating circuits. In *Methods of Mathematical Modelling: Fractional Differential Equations*; CRC Press : London, UK; New York, NY, USA, 2019.
15. Singh, H.; Srivastava, H.M. Numerical Investigation of the Fractional-Order Liénard and Duffing Equations Arising in Oscillating Circuit Theory. *Front. Phys.* **2020**, *8*, 120. [\[CrossRef\]](#)
16. Nourazar, S.; Mirzabeigy, A. Approximate solution for nonlinear Duffing oscillator with damping effect using the modified differential transform method. *Sci. Iran. B* **2013**, *20*, 364–368.
17. Gómez-Aguilar, J.F.; Torres, L.; Yépez-Martínez, H.; Baleanu, D.; Reyes, J.M.; Sosa, I.O. Fractional Liénard type model of a pipeline within the fractional derivative without singular kernel. *Adv. Differ. Equ.* **2016**, *2016*, 173. [\[CrossRef\]](#)
18. Feng, Z. Exact solutions for the Lienard equation and its applications. *Chaos Solitons Fractals* **2004**, *21*, 343–348. [\[CrossRef\]](#)
19. Matinfar, M.; Hosseinzadeh, H.; Ghanbari, M. A numerical implementation of the variational iteration method for the Lienard equation. *World J. Model. Simul.* **2008**, *4*, 205–210.
20. Xu, G.Q. New explicit exact solutions for the Liénard equation and its applications. *arXiv* **2010**, arxiv:1003.2921v1.
21. Janiczek, T. Generalisation of the modulating functions method into the fractional differential equations. *Bull. Pol. Acad. Sci. Tech. Sci.* **2010**, *58*, 593–599. [\[CrossRef\]](#)
22. Atabakzadeh, M.H.; Akrami, M.H.; Erjaee, G.H. Chebyshev Operational Matrix Method for Solving Multi-order Fractional Ordinary Differential Equations. *Appl. Math. Model.* **2013**, *37*, 8903–8911. [\[CrossRef\]](#)
23. Kazem, S. An integral operational matrix based on Jacobi polynomials for solving fractional-order differential equations. *Appl. Math. Model.* **2013**, *37*, 1126–1136. [\[CrossRef\]](#)
24. Ezz-Eldien, S.S. New quadrature approach based on operational matrix for solving a class of fractional variational problems. *J. Comput. Phys.* **2016**, *317*, 362–381. [\[CrossRef\]](#)
25. Ejikeme, C.L.; Oyesanya, M.O.; Agbebaku, D.F.; Okofu, M.B. Solution to nonlinear Duffing Oscillator with fractional derivatives using Homotopy Analysis Method (HAM). *Glob. J. Pure Appl. Math.* **2018**, *14*, 1363–1388.

26. Morales-Delgado, V.F.; Gómez-Aguilar, J.F.; Torres, L.; Escobar-Jiménez, R.F.; Taneco-Hernandez, M.A. Exact Solutions for the Liénard Type Model via Fractional Homotopy Methods. In *Fractional Derivatives with Mittag-Leffler Kernel*; Springer: Cham, Switzerland, 2019; Volume 194, pp. 269–291. [[CrossRef](#)]
27. Singh, H.; Srivastava, H.M.; Kumar, D. A reliable numerical algorithm for the fractional vibration equation. *Chaos Solitons Fractals* **2017**, *103*, 131–138. [[CrossRef](#)]
28. Singh, H. Solution of fractional Lienard equation using Chebyshev operational matrix method. *Nonlinear Sci. Lett. A* **2017**, *8*, 397–404.
29. Kumar, S.; Gomez-Aguilar, J.F.; Lavin-Delgado, J.E.; Baleanu, D. Derivation of operational matrix of Rabotnov fractional-exponential kernel and its application to fractional Lienard equation. *Alex. Eng. J.* **2020**, *59*, 2991–2997. [[CrossRef](#)]
30. Adel, W. A fast and efficient scheme for solving a class of non-linear Lienard's equations. *Math. Sci.* **2020**, *14*, 167–175. [[CrossRef](#)]
31. Kumar, M.; Yadav, N. Multilayer perceptrons and radial basis function neural network methods for the solution of differential equations: A survey. *Comput. Math. Appl.* **2011**, *62*, 3796–3811. [[CrossRef](#)]
32. Khan, N.A.; 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*; IntechOpen: London, UK, 2017. [[CrossRef](#)]
33. Chakraverty, S.; Mall, S. *Artificial Neural Networks for Engineers and Scientists Solving Ordinary Differential Equations*; CRC Press: London, UK; New York, NY, USA, 2017.
34. Verma, A.; Kumar, M. Numerical solution of Lane–Emden type equations using multilayer perceptron neural network method. *Int. J. Appl. Comput. Math.* **2019**, *5*, 141. [[CrossRef](#)]
35. Shaikh, A.; Jamal, M.A.; Hanif, F.; Khan, M.S.A.; Inayatullah, S. Neural minimisation methods for solving variable order fractional delay differential equations with simulated annealing. *PLoS ONE* **2019**, *14*, e0223476. [[CrossRef](#)] [[PubMed](#)]
36. Chakraverty, S.; Mall, S. Single layer Chebyshev neural network model with regression-based weights for solving nonlinear ordinary differential equations. *Evol. Intell.* **2022**, *13*, 687–694. [[CrossRef](#)]
37. Verma, A.; Kumar, M. Numerical solution of Bagley–Torvik equations using Legendre artificial neural network method. *Evol. Intell.* **2020**, *14*, 2027–2037. [[CrossRef](#)]
38. Verma, A.; Kumar, M. Numerical solution of third-order Emden–Fowler type equations using artificial neural network technique. *Eur. Phys. J. Plus* **2020**, *135*, 751. [[CrossRef](#)]
39. Lefik, M. Some aspects of application of artificial neural network for numerical modeling in civil engineering. *Bull. Pol. Acad. Sci. Sci.* **2013**, *61*, 39–50. [[CrossRef](#)]
40. Malik, S.A.; Qureshi, I.M.; Amir, M.; Haq, I. Numerical Solution of Lienard Equation Using Hybrid Heuristic Computation. *World Appl. Sci. J.* **2013**, *28*, 636–643.
41. Kumar, S.; Sahoo, A.K.; Chakraverty, S. A Comparative study of Chebyshev and Legendre Polynomial-based Neural Models for Approximating Multidimensional Poverty for an Indian State. In *Polynomial Paradigms: Trends and Applications in Science and Engineering*; IOP Publishing: Bristol, UK, 2023. [[CrossRef](#)]
42. Sahoo, A.K.; Chakraverty, S. A neural network approach for the solution of Van der Pol–Mathieu–Duffing oscillator model. *Evol. Intell.* **2023**. [[CrossRef](#)]
43. Pao, Y.H.; Philips, S.M. The functional link net and learning optimal control. *Neurocomputing* **1995**, *9*, 149–164. [[CrossRef](#)]
44. Kumar, D.; Agarwal, R.P.; Singh, J. A modified numerical scheme and convergence analysis for fractional model of Lienard's equation. *J. Comput. Appl. Math.* **2018**, *339*, 405–413. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.