Iranian Journal of Numerical Analysis and Optimization
Vol. 15, No. 1, 2025, pp 375–395
https://doi.org/10.22067/ijnao.2024.88927.1479
https://ijnao.um.ac.ir/

Research Article

Numerical simulation of nonlinear Liénard's equation via Morgan–Voyce even Fibonacci neural network

P. Rahimkhani and R. Moeti*

Abstract

In the current study, we design a new computational method to solve a class of Liénard's equations. This equation originates from advancements in radio and vacuum tube technology. To attain the proposed goal, we develop a method using a three-layer artificial neural network, consisting of an input layer, a hidden layer, and an output layer. We use the Morgan–Voyce even Fibonacci polynomials and sinh function as activation functions for the hidden layer and the output layer, respectively. Then, the neural network is trained using a classical optimization method. Finally, we analyze four examples using graphs and tables to demonstrate the accuracy and effectiveness of the numerical approach.

*Corresponding author

Received 13 July 2024; revised 26 October 2024; accepted 28 October 2024

Parisa Rahimkhani
Faculty of Science, Mahallat Institute of Higher Education, Mahallat, Iran. e-mail: rahimkhani.parisa@mahallat.ac.ir

Reza Moeti
Department of Mathematics Education, Faculty of Mathematics and Computer, Shahid Bahonar University, Kerman, Iran. e-mail: rezamoeti@math.uk.ac.ir

How to cite this article
Rahimkhani, P. and Moeti, R., Numerical simulation of nonlinear Liénard's equation via Morgan–Voyce even Fibonacci neural network. *Iran. J. Numer. Anal. Optim.*, 2025; 15(1): 375-395. https://doi.org/10.22067/ijnao.2024.88927.1479

AMS subject classifications (2020): 92B20, 41A10, 35E15.

Keywords: Morgan–Voyce even Fibonacci polynomials, Neural network, Liénard's equation, Numerical method.

1 Introduction

Nonlinear differential equations have significant implications in various fields of science and engineering, including fluid mechanics, solid state physics, chemical kinetics, plasma physics, mathematical biology, and so on. Most of the nonlinear differential equations are complicated to be solved using analytical methods, so different numerical methods are proposed for solving these problems. For example,

- The Bernoulli collocation approach to solve the nonlinear Liénard's equations (LEs) [2].
- The Taylor wavelets method to solve the Bratu-type equation [8].
- The orthonormal Bernoulli wavelets neural network approach to solve the Lane-Emden equation [25].
- The Hermite wavelets method to solve the nonlinear Rosenau–Hyman equation [12].
- The Laguerre wavelets to solve the Hunter Saxton equation [33].
- The cardinal B-spline wavelets to solve the generalized Burgers–Huxley equation [31].
- Hahn hybrid functions to solve distributed order fractional Black–Scholes European [27].
- Chelyshkov least squares support vector regression to solve nonlinear stochastic differential equations [26].

The standard LE is a generalization of the damped pendulum equation or spring-mass system. Because this equation can be applied to describe the oscillating circuits, it is used in developing radio and vacuum-tube technology. The LE, formulated by Liénard, is a second-order nonlinear differential equation [13] and is given as

$$f''(t) + \eta f'(t) + \psi f(t)f'(t) + \sigma f(t) + \delta f^2(t) + \zeta f^3(t) + \gamma f^5(t) = g(t), \quad (1)$$

subject to

$$f(0) = \nu_1, \qquad f'(0) = \nu_2, \quad (2)$$

where $0 \leq t \leq 1, \eta, \psi, \sigma, \delta, \zeta, \gamma, \nu_1, \nu_2$ are real coefficients, and $g(t)$ is a continuous function.

Authors in [3, 10, 34] introduced a precise expression for the exact solution of LEs applying the direct scheme. Matinfar, Hosseinzadeh, and Ghanbari [18] proposed a method based on a variational iteration approach to obtain a closed-form solution of LEs. Heydari, Hooshmandasl, and Ghaini [4] utilized the block-pulse functions method for solving the LEs. Kaya and El-Sayed [7] applied the Adomian decomposition method to solve the LEs. Adel [2] adopted the Bernoulli collocation method to solve the LEs. The stability of the periodic solution of the LEs was conducted in [35].

Over the past few years, significant focus has been placed on exploring artificial neural networks for studying various differential equations, including: linear and nonlinear differential equations (see [21, 20]), higher order differential equations [14], elliptical partial differential equation [6], Emden–Fowler equation [15], doubly singular nonlinear systems [28], nonlinear Bratu type equation [16], Riccati differential equation [29], fractal-fractional pantograph differential equations [24], nonlinear stochastic differential equations with fractional Brownian motion [22], and stochastic biological systems [23].

The objective of this study is to adapt the Morgan–Voyce even Fibonacci polynomials (MVEFPs) and neural network technique for obtaining the approximate solution of the nonlinear LEs using the given initial conditions. We use the MVEFPs and sinh functions as activation functions of the hidden layer and the output layer, respectively. Then, we use the classical optimization method to train the presented neural network. The approximate solutions by neural networks have many advantages. Some of the main features of neural network schemes are listed as follows [25]:

- The approximate solution of this scheme is continuous and differentiable.
- By increasing the number of neurons, the accuracy of the method can be increased.
- At any arbitrary point even between training points, the solution can be obtained.
- Neural networks can be used for solving linear, nonlinear, and system of differential equations, fractional differential equations, and so on.

The paper's organization is as follows: In Section 2, some preliminaries about the MVEFPs and the structure of neural networks are presented. In Section 3, we design the MVEFPs neural network method to solve (1)–(2). The convergence analysis and estimated error (EE) are discussed in Section 4. Section 5 offers some numerical examples to show the method's effectiveness, and conclusions are drawn in Section 6.

2 The requirement concepts

In this section, we recall the MVEFPs and Morgan–Voyce even Fibonacci neural network (MVEFNN).

2.1 Morgan–Voyce even Fibonacci polynomials

The MVEFPs $B_j(t)$ are defined by the following recurrence relation [5]:

$$B_j(t) = (2+t)B_{j-1}(t) - B_{j-2}(t), \tag{3}$$

with the initial conditions

$$B_1(t) = 1, \qquad B_2(t) = 2+t. \tag{4}$$

Also, we have

$$B_0(t) = 0.$$

The explicit summation expressions of the MVEFPs for ($n \geq 1$) on the interval $[a, b]$ are defined as [5]

$$B_j(t) = \sum_{i=0}^{j-1} \binom{j+i}{2i+1} t^i. \tag{5}$$

The coefficients and sums associated to MVEFPs [30] are presented in Table 1.

Table 1: Coefficients and sums associated with the MVEFPs.

j	Diagonal Sums	Row	Partial
	0 1 2 4 8 16 32 64 128	Sums	Column Sums
0	0	0	0
1	1	1	1
2	2 1	3	3 1
3	3 4 1	8	6 5 1
4	4 10 6 1	21	10 15 7 1
5	5 20 21 8 1	55	15 35 28 9 1
6	6 35 56 36 10 1	144	21 70 84 45 11 1
7	7 56 126 120 55 12 1	377	28 126 210 165 66 13 1

2.2 Structure of Morgan–Voyce even Fibonacci neural network

We consider an MVEFNN for finding the numerical approximation of the nonlinear LEs.

- The first layer is the network's input layer, containing a single node t with ϱ data points as $\{t_1, t_2, \ldots, t_\varrho\}$.

- The second layer of the network is the hidden layer, which uses a class of polynomials as activation functions. In this study, the MVEFPs are utilized as the activation functions in the hidden layer.

- The output layer of the network receives a linear combination of the MVEFPs from the second layer as input and produces an output by applying an activation function ($AF(\cdot)$) to this input.

Therefore, the output of the MVEFNN with the input data t and parameter C is as

$$N(t, C) = AF(\Theta). \tag{6}$$

Here Θ is a linear combination of the MVEFPs, expressed as

$$\Theta = \sum_{j=0}^{N} c_j B_j(t) = C^T B(t). \tag{7}$$

The vectors C and $B(t)$ are established by the following expressions:

$$C = [c_0, c_1, c_2, \dots, c_N]^T,$$

and

$$B(t) = [B_0(t), B_1(t), B_2(t), \dots, B_N(t)]^T.$$

Additionally, $AF(\cdot)$ is another activation function that affects on the combination of the MVEFPs.

3 Description of the strategy

To solve the problem (1)–(2), we approximate the function $f(t)$ as follows:

$$f(t) \simeq \nu_1 + \nu_2 t + t^2 N(t, C) = \tilde{f}(t). \tag{8}$$

It is evident that the function $\tilde{f}(t)$ meets the specified initial conditions. Here we have

$$N(t, C) = AF(C^T B(t)).$$

We use sinh function as the activation function as

$$N(t, C) = \sinh(C^T B(t)).$$

Since we require the approximate solution $\tilde{f}(t)$ to satisfy (1), we define the residual function ($Res(t, C)$) as follows:

$$Res(t,C) = \tilde{f}''(t) + \eta \tilde{f}'(t) + \psi \tilde{f}(t)\tilde{f}'(t) + \sigma \tilde{f}(t) + \delta \tilde{f}^2(t) + \zeta \tilde{f}^3(t) + \gamma \tilde{f}^5(t) - g(t). \quad (9)$$

For each training point (t_i), it yields

$$Res(t_i,C) = \tilde{f}''(t_i) + \eta \tilde{f}'(t_i) + \psi \tilde{f}(t_i)\tilde{f}'(t_i) + \sigma \tilde{f}(t_i) + \delta \tilde{f}^2(t_i) + \zeta \tilde{f}^3(t_i) + \gamma \tilde{f}^5(t_i) - g(t_i), \quad (10)$$

where $i = 1, 2, \ldots, \varrho$.

So, a data set $\{t_1, t_2, \ldots, t_\varrho\}$ is considered, and the following optimization problem needs to be achieved

$$C^* = \min \frac{1}{2} \sum_{i=1}^{\varrho} Res^2(t_i, C). \quad (11)$$

Thus, (11) represents an unconstrained parametric optimization problem that can be expressed as follows: Find the vector C that minimizes C^*. The necessary conditions to determine the minimum value of C^* are

$$\frac{\partial}{\partial c_j} C^* = 0, \qquad j = 0, 1, 2, \ldots, N. \quad (12)$$

Using Newton's iterative method, (12) can be solved for C. Once we obtain C, we can determine the approximate solution $\tilde{f}(t)$ as given in (8).

Remark 1. The training data are chosen according to the behavior of the solution function. To have an adaptive algorithm, the points should be accumulated near the high frequencies of the underlying function. Because of the nature of this problem, the training points should be near zero and so, we used zeros of the shifted Legendre polynomials. In the numerical results (NRs), we use ϱ roots of the shifted Legendre polynomials as the training points of the method.

4 Error analysis

In this section, an error estimation will be given by means of the error function and residual error function.

4.1 Estimation of the error function

This section will provide an error estimation of the MVEFNN solution (1) using the residual error function. To achieve this aim, we express the error function as

$$\mathfrak{e}(t) = f(t) - \tilde{f}(t), \tag{13}$$

where $f(t)$ and $\tilde{f}(t)$ are the exact and numerical solution of relation (1), respectively. The numerical solution $\tilde{f}(t)$ satisfies the following problem:

$$\begin{aligned}&\tilde{f}''(t) + \eta\tilde{f}'(t) + \psi\tilde{f}(t)\tilde{f}'(t) + \sigma\tilde{f}(t) + \delta\tilde{f}^2(t)\\&+\zeta\tilde{f}^3(t) + \gamma\tilde{f}^5(t) = g(t) + \mathfrak{R}(t),\end{aligned} \tag{14}$$

subject to

$$\tilde{f}(0) = \nu_1, \qquad \tilde{f}'(0) = \nu_2, \tag{15}$$

where $\mathfrak{R}(t)$ is the residual function. By subtracting (14) from (1), we get

$$\begin{aligned}&\mathfrak{e}''(t) + \eta\mathfrak{e}'(t) + \psi(f(t)f'(t) - \tilde{f}(t)\tilde{f}'(t)) + \sigma\mathfrak{e}(t)\\&+\delta(f^2(t) - \tilde{f}^2(t)) + \zeta(f^3(t) - \tilde{f}^3(t)) + \gamma(f^5(t) - \tilde{f}^5(t)) = 0.\end{aligned} \tag{16}$$

By using (13), the above equations can be written as

$$\begin{aligned}&\mathfrak{e}''(t) + \eta\mathfrak{e}'(t) + \psi(\mathfrak{e}(t)\mathfrak{e}'(t) + \tilde{f}'(t)\mathfrak{e}(t) + \tilde{f}(t)\mathfrak{e}'(t)) + \sigma\mathfrak{e}(t)\\&+\delta(\mathfrak{e}^2(t) + 2\tilde{f}(t)\mathfrak{e}(t) + \tilde{f}^2(t)) + \zeta(\mathfrak{e}^3(t) + 3\tilde{f}^2(t)\mathfrak{e}(t) + 3\tilde{f}(t)\mathfrak{e}^2(t) + \tilde{f}^3(t))\\&+\gamma(\mathfrak{e}^5(t) + 5\tilde{f}(t)\mathfrak{e}^4(t) + 10\tilde{f}^2(t)\mathfrak{e}^3(t) + 10\tilde{f}^3(t)\mathfrak{e}^2(t) + 5\tilde{f}^3(t)\mathfrak{e}^2(t)\\&+5\tilde{f}^4(t)\mathfrak{e}(t) + \tilde{f}^5(t)) = 0,\end{aligned} \tag{17}$$

subject to

$$\mathfrak{e}(0) = 0, \qquad \mathfrak{e}'(0) = 0. \tag{18}$$

We can obtain an approximation of the error function $\mathfrak{e}(t)$ by the MVEFNN as

$$\mathfrak{e}(t) \simeq t^2 N(t, A) = \tilde{\mathfrak{e}}(t). \tag{19}$$

Substituting (19) into (17), we yield

$$\mathfrak{E}(t) = \tilde{\mathfrak{e}}''(t) + \eta\tilde{\mathfrak{e}}'(t) + \psi(\tilde{\mathfrak{e}}(t)\tilde{\mathfrak{e}}'(t) + \tilde{f}'(t)\tilde{\mathfrak{e}}(t) + \tilde{f}(t)\tilde{\mathfrak{e}}'(t))$$

$$
\begin{aligned}
&+\sigma\tilde{\mathfrak{e}}(t)+\delta(\tilde{\mathfrak{e}}^2(t)+2\tilde{f}(t)\tilde{\mathfrak{e}}(t)+\tilde{f}^2(t))\\
&+\zeta(\tilde{\mathfrak{e}}^3(t)+3\tilde{f}^2(t)\tilde{\mathfrak{e}}(t)+3\tilde{f}(t)\tilde{\mathfrak{e}}^2(t)+\tilde{f}^3(t))\\
&+\gamma(\tilde{\mathfrak{e}}^5(t)+5\tilde{f}(t)\tilde{\mathfrak{e}}^4(t)+10\tilde{f}^2(t)\tilde{\mathfrak{e}}^3(t)\\
&+10\tilde{f}^3(t)\tilde{\mathfrak{e}}^2(t)+5\tilde{f}^3(t)\tilde{\mathfrak{e}}^2(t)+5\tilde{f}^4(t)\tilde{\mathfrak{e}}(t)+\tilde{f}^5(t))=0.
\end{aligned}
\tag{20}
$$

Then, we obtain the following optimization problem:

$$A^* = \min\ \frac{1}{2}\sum_{i=1}^{\varrho}\mathfrak{E}^2(t_i, A). \tag{21}$$

By solving problem (21), we obtain an approximation of the error function.

Remark 2. In cases where the exact solution of the problem is unknown, the error estimation (21) can be utilized to assess the accuracy of the obtained results.

4.2 Residual error

When the analytical solution of the problem is unknown, the reliability and accuracy of the technique are checked via the residual error function as

$$E(t_i) = |\tilde{f}''(t_i)+\eta\tilde{f}'(t_i)+\psi\tilde{f}(t_i)\tilde{f}'(t_i)+\sigma\tilde{f}(t_i)+\delta\tilde{f}^2(t_i)+\gamma\tilde{f}^5(t_i)-g(t_i)|, \tag{22}$$

where

$$t_i \in [0,1], \qquad i = 1,2,3,\ldots.$$

5 Computational experiments

Here, four numerical examples are provided to show the validity and applicability of the presented scheme in Section 3.

Example 1. Consider the problem (1)–(2) with $\eta = 0, \psi = 1, \sigma = \delta = 1, \zeta = 0, \gamma = 0, \nu_1 = 1, \nu_2 = 0, g(t) = \cos^2(t) - \sin(t)\cos(t)$ as [9]

$$f''(t) + f(t)f'(t) + f(t) + f^2(t) = \cos^2 t - \sin t \cos t, \tag{23}$$

$$f(0) = 1, \qquad f'(0) = 0. \tag{24}$$

Equation (23)–(24) has the exact solution given by

$$f(t) = \cos(t). \tag{25}$$

We compare the absolute error (AE) of the mentioned strategy for $N = 4, 5, 7, 8$, with the predictor-corrector method [9] for $N = 10$ in Table 2. Table 3 compares the results of the present method with the MVEFPs method without neural network and activation function. The results show that using a small number of basis functions, our approach yields the numerical solution with high accuracy. Moreover, a graphical illustration of the NR and the EE in (22) for $N = 8$, is shown in Figure 1.

Table 2: The comparison of AE of the mentioned method with [9] (Example 1).

t	Ref. [9]	mentioned method				
	$N=10$	$N=4$	$N=5$	$N=6$	$N=7$	$N=8$
0.1	2.68×10^{-7}	9.16×10^{-7}	1.03×10^{-11}	1.37×10^{-10}	9.09×10^{-14}	7.11×10^{-13}
0.2	5.72×10^{-7}	1.12×10^{-6}	1.52×10^{-10}	2.62×10^{-10}	5.43×10^{-11}	9.63×10^{-14}
0.3	8.88×10^{-7}	1.94×10^{-7}	4.02×10^{-10}	3.02×10^{-10}	1.96×10^{-11}	9.22×10^{-13}
0.4	1.20×10^{-6}	1.82×10^{-6}	1.47×10^{-10}	2.92×10^{-10}	7.95×10^{-11}	1.20×10^{-11}
0.5	1.28×10^{-6}	2.35×10^{-6}	8.21×10^{-10}	6.17×10^{-10}	9.95×10^{-14}	1.69×10^{-14}
0.6	–	1.29×10^{-6}	1.61×10^{-9}	7.06×10^{-11}	8.88×10^{-11}	1.68×10^{-11}
0.7	–	6.33×10^{-7}	8.47×10^{-10}	6.34×10^{-10}	7.07×10^{-14}	1.01×10^{-11}
0.8	–	1.90×10^{-6}	1.36×10^{-9}	3.57×10^{-10}	6.98×10^{-11}	7.26×10^{-14}
0.9	–	1.39×10^{-6}	2.07×10^{-9}	3.71×10^{-10}	1.69×10^{-11}	2.13×10^{-14}
1	–	2.51×10^{-7}	1.20×10^{-10}	9.49×10^{-14}	8.07×10^{-14}	3.08×10^{-14}

Example 2. Consider the problem (1)–(2) with $\eta = 0.5, \psi = 0, \sigma = 25, \delta = 0, \zeta = 0, \gamma = 25, \nu_1 = 0.1, \nu_2 = 0, g(t) = 0$, as (see [17, 32]):

$$f''(t) + 0.5f'(t) + 25f(t) + 25f^5(t) = 0, \tag{26}$$

$$f(0) = 0.1, \qquad f'(0) = 0. \tag{27}$$

A closed-form exact solution for problem (26)–(27) is not available.

Table 4 shows the NR of the presented method for $N = 18$ with the differential transform scheme [17] and the Chebyshev matrix approach [32].

Table 3: The comparison of the AE of the mentioned method for $N = 6, 8$ with the MVEFPs. (Example 1).

t	$N=6$		$N=8$	
	$MVEFPs$	$mentioned\ method$	$MVEFPs$	$mentioned\ method$
0.1	8.94×10^{-10}	1.37×10^{-10}	1.16×10^{-11}	7.11×10^{-13}
0.2	1.32×10^{-9}	2.62×10^{-10}	1.08×10^{-10}	9.63×10^{-14}
0.3	1.71×10^{-9}	3.02×10^{-10}	1.86×10^{-11}	9.22×10^{-13}
0.4	1.02×10^{-9}	2.92×10^{-10}	1.21×10^{-10}	1.20×10^{-11}
0.5	2.62×10^{-9}	6.17×10^{-10}	1.49×10^{-11}	1.69×10^{-14}
0.6	6.22×10^{-10}	7.06×10^{-11}	1.62×10^{-10}	1.68×10^{-11}
0.7	2.04×10^{-9}	6.34×10^{-10}	7.41×10^{-11}	1.01×10^{-11}
0.8	1.28×10^{-9}	3.57×10^{-10}	5.82×10^{-11}	7.26×10^{-14}
0.9	1.03×10^{-9}	3.71×10^{-10}	2.48×10^{-11}	2.13×10^{-14}
1.0	9.50×10^{-12}	9.49×10^{-14}	3.05×10^{-11}	3.08×10^{-14}

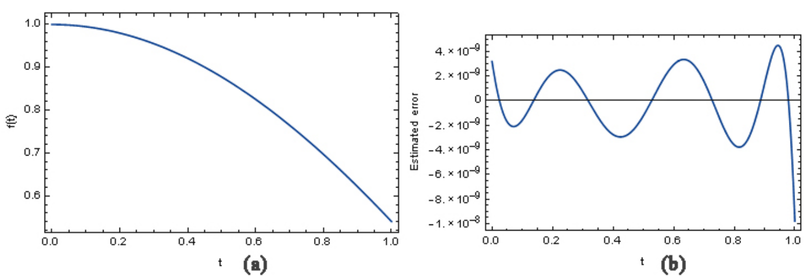

Figure 1: (a) : The NRs and (b) : The EE with $N = 8$ (Example 1).

Also, theEE in (22) of our method is reported in this table. In Table 5, we compare the CPU times (seconds) of Examples 1 and 2 for different choices of N. Also, a graphical illustration of the NR and the EE in (22) for $N = 18$, is depicted in Figure 2.

Example 3. Consider the problem (1)–(2) with $\eta = \psi = 0, \sigma = -1, \delta = 0, \zeta = 4, \gamma = -3, \nu_1 = \frac{1}{\sqrt{2}}, \nu_2 = \frac{\sqrt{2}}{4}, g(t) = 0$ as (see [2, 17, 32, 18, 7, 19, 1, 11])

$$f''(t) - f(t) + 4f^3(t) - 3f^5(t) = 0, \tag{28}$$

$$f(0) = \frac{1}{\sqrt{2}}, \qquad f'(0) = \frac{\sqrt{2}}{4}. \tag{29}$$

Equation (28)–(29) has the exact solution given by

Table 4: The comparison of the NRs of the mentioned method for $N = 18$ with other methods. (Example 2).

t	Ref. [17]	Ref. [32]	mentioned method	EE
0.01	0.099873987171430	0.0998745	0.0998752	5.39×10^{-5}
0.02	0.099497110553344	0.0995025	0.0995020	1.30×10^{-4}
0.03	0.098871588842452	0.0988894	0.0988826	1.10×10^{-4}
0.04	0.098000267574510	0.0980414	0.0980197	4.51×10^{-5}
0.05	0.096886607975937	0.0969644	0.0969167	3.21×10^{-5}
0.06	0.095534673995913	0.0956646	0.0955776	9.93×10^{-5}
0.07	0.093949117549958	0.0941484	0.0940068	1.44×10^{-4}
0.08	0.092135162006012	0.0924222	0.0922094	1.60×10^{-4}
0.09	0.090098583944017	0.0904924	0.0901910	1.51×10^{-4}
0.1	0.087845693220000	0.0883660	0.0178866	7.64×10^{-4}

Table 5: The CPU times of the mentioned method for Examples 1 and 2 and different choices of N.

	$N = 4$	$N = 6$	$N = 8$	$N = 10$
Example 1	0.469	1.702	12.234	43.469
Example 2	0.297	1.703	11.844	28.578

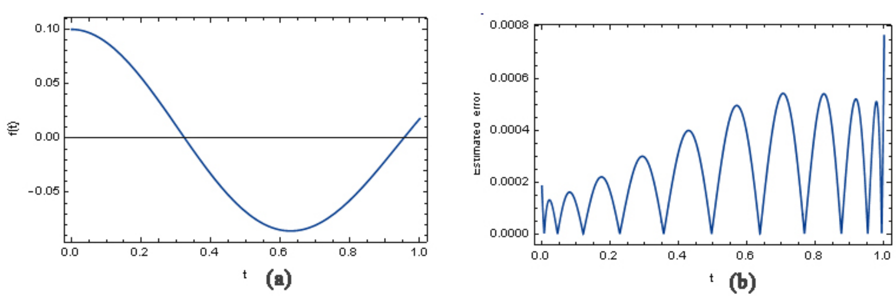

Figure 2: (a) : NRs and (b) : EE with $N = 18$ (Example 2).

$$f(t) = \sqrt{\frac{1 + \tanh(t)}{2}}. \tag{30}$$

Table 6 shows a comparison between the maximum absolute error (MAE) of the mentioned strategy with variational iteration [18], Adomian decomposition [7], variational Homotopy perturbation [19], differential transform [17], and genetic algorithm [1] methods. The AE of the mentioned method for

$N = 8$ with a modified numerical scheme [11], and Chebyshev operational matrix method [32] are reported in Table 7. Also, the MAE ($\|E_N\|_\infty$), maximum relative error (MRE) ($\|E_N\|_R$), and maximum error function (MEF) ($\|e_N\|_\infty$) of the mentioned method for different values of N with the Bernoulli operational matrix method [2] are presented in Table 8. A graphical illustration of the NR and the EE in (22) for $N = 18$, is demonstrated in Figure 3.

Table 6: The comparison of MAE for $N = 8$ with other methods. (Example 3).

Variational iteration [18]	1.9575×10^{-4}
Adomian decomposition [7]	2.1346×10^{-7}
Variational Homotopy perturbation [19]	4.5009×10^{-5}
Differential transform [17]	1.3692×10^{-2}
Genetic algorithm [1]	2.5110×10^{-5}
Mentioned method	5.1126×10^{-9}

Table 7: The comparison of the AE of the mentioned method for $N = 8$ with other methods. (Example 3).

t	Ref. [11]	Ref. [32]	mentioned method
0.01	–	1.75×10^{-5}	6.43×10^{-10}
0.02	1.87×10^{-6}	7.16×10^{-5}	1.93×10^{-9}
0.03	–	1.63×10^{-4}	3.79×10^{-9}
0.04	6.27×10^{-6}	2.92×10^{-4}	7.62×10^{-10}
0.05	–	4.61×10^{-4}	5.11×10^{-9}
0.06	4.95×10^{-5}	6.69×10^{-4}	4.24×10^{-10}
0.07	–	9.18×10^{-4}	3.91×10^{-9}
0.08	1.16×10^{-4}	1.21×10^{-3}	2.00×10^{-9}
0.09	–	1.54×10^{-3}	1.05×10^{-9}
0.1	2.25×10^{-4}	1.92×10^{-3}	4.69×10^{-10}

Example 4. Consider the problem (1)–(2) with $\eta = \psi = 0, \sigma = -1, \delta = 0, \zeta = 4, \gamma = 3, \nu_1 = \frac{1}{\sqrt{1+\sqrt{2}}}, \nu_2 = 0, g(t) = 0$ as (see [2, 17, 32, 18, 7, 19, 1, 11])

Table 8: The comparison of the MAE, MRE, and MEF of the mentioned method for different values of N with [2]. (Example 3).

N	Ref. [2]			mentioned method		
	$\|E_N\|_\infty$	$\|E_N\|_R$	$\|e_N\|_\infty$	$\|E_N\|_\infty$	$\|E_N\|_R$	$\|e_N\|_\infty$
4	2.62×10^{-4}	2.79×10^{-4}	1.89×10^{-2}	1.32×10^{-5}	1.66×10^{-5}	1.32×10^{-3}
6	1.96×10^{-5}	2.09×10^{-5}	1.33×10^{-3}	9.21×10^{-8}	1.03×10^{-7}	7.23×10^{-5}
8	5.86×10^{-7}	6.24×10^{-7}	7.10×10^{-5}	5.11×10^{-9}	5.97×10^{-9}	1.40×10^{-6}

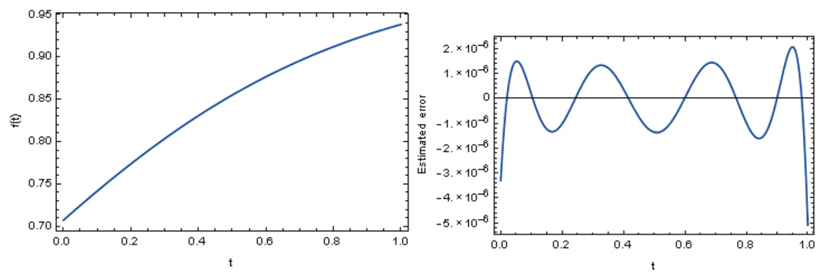

Figure 3: (a) : NRs and (b) : EE with $N = 18$ (Example 3).

$$f''(t) - f(t) + 4f^3(t) + 3f^5(t) = 0, \tag{31}$$

$$f(0) = \frac{1}{\sqrt{1+\sqrt{2}}}, \qquad f'(0) = 0. \tag{32}$$

Equation (31)–(32) has the exact solution given by

$$f(t) = \sqrt{\frac{sech^2(t)}{2\sqrt{2} + (1-\sqrt{2})sech^2(t)}}. \tag{33}$$

Table 9 exhibits a comparison between the MAE of the mentioned process with variational iteration [18], Adomian decomposition [7], variational Homotopy perturbation [19], differential transform [17], and genetic algorithm [1]. The AE of our method at $N = 8$ and a modified numerical scheme [11], and Chebyshev operational matrix method [32] are reported in Table 10. Also, the MAE ($\|E_N\|_\infty$), MRE ($\|E_N\|_R$), and MEF ($\|e_N\|_\infty$) of the mentioned method for various values of N with the Bernoulli operational matrix method [2] are presented in Table 11. In Table 12, we compare the CPU times (seconds) of Examples 3 and 4 for different choices of N.

Table 9: The comparison of MAE for $N = 8$ with other methods. (Example 4).

Variational iteration [18]	1.3667×10^{-2}
Adomian decomposition [7]	5.4649×10^{-3}
Variational Homotopy perturbation [19]	5.1335×10^{-4}
Differential transform [17]	4.4891×10^{-2}
Genetic algorithm [1]	9.3400×10^{-5}
Mentioned method	6.9955×10^{-8}

Table 10: The comparison of the AE of the mentioned method for $N = 8$ with other methods. (Example 4).

t	Ref. [11]	Ref. [32]	mentioned method
0.01	–	3.22×10^{-5}	3.65×10^{-9}
0.02	7.10×10^{-5}	1.31×10^{-5}	1.12×10^{-8}
0.03	–	2.96×10^{-4}	1.87×10^{-8}
0.04	2.84×10^{-4}	5.31×10^{-4}	2.40×10^{-8}
0.05	–	8.35×10^{-4}	2.59×10^{-8}
0.06	6.37×10^{-4}	1.21×10^{-3}	2.40×10^{-8}
0.07	–	1.66×10^{-3}	1.87×10^{-8}
0.08	1.13×10^{-3}	2.18×10^{-3}	1.06×10^{-8}
0.09	–	2.78×10^{-3}	4.89×10^{-10}
0.1	1.76×10^{-3}	3.45×10^{-3}	1.06×10^{-8}

Table 11: The comparison of the MAE, MRE, and MEF of the mentioned method for different values of N with [2]. (Example 4).

N	Ref. [2]			mentioned method		
	$\|E_N\|_\infty$	$\|E_N\|_R$	$\|e_N\|_\infty$	$\|E_N\|_\infty$	$\|E_N\|_R$	$\|e_N\|_\infty$
4	1.58×10^{-3}	2.68×10^{-3}	1.22×10^{-1}	1.10×10^{-4}	1.50×10^{-4}	7.51×10^{-3}
6	5.27×10^{-5}	8.23×10^{-5}	8.46×10^{-3}	3.47×10^{-6}	4.12×10^{-6}	5.10×10^{-4}
8	1.77×10^{-7}	5.17×10^{-7}	4.20×10^{-4}	7.00×10^{-8}	1.22×10^{-7}	4.10×10^{-5}

Table 12: The CPU times of the mentioned method for Examples 3 and 4 and different choices of N.

	$N=4$	$N=6$	$N=8$	$N=10$
Example 3	0.281	2.031	8.969	30.093
Example 4	0.328	1.563	9.547	15.328

6 Conclusion

In this paper, we applied an effective algorithm utilizing the MVEFPs and neural networks to solve the nonlinear LEs. We used the MVEFPs and sinh functions as the activation functions of the hidden layer and the output layer, respectively. Moreover, the network was trained using the classical optimization approach. Comparison of the NRs obtained from our proposed method, existing methods, and the exact solution demonstrates the high accuracy and effectiveness of our approach.

Acknowledgments

We express our sincere thanks to the anonymous referees for valuable suggestions that improved the paper.

Contribution statements

Parisa Rahimkhani

Conceptualization, Data curation, Investigation, Software, Writing -original draft

Reza moeti

Conceptualization, Validation, Writing -review and editing

Conflict of interest

The authors declare that they have no conflict of interest.

Data availability statement

Data will be made available on reasonable request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

The authors received no financial support for the research, authorship, and publication of this paper.

References

[1] Abdullah Malik, S., Qureshi, I., Amir, M., and Haq, I.*Numerical solution of Lienard equation using hybrid heuristic computation*, World Appl. Sci. J. 28 (2013), 636−643.

[2] Adel, W. *A fast and efficient scheme for solving a class of nonlinear LEs*, Math. Sci. 14 (2020), 167−175.

[3] Feng, Z. *On explicit exact solutions for the Lienard equation and its applications*, Phys. Lett. A. 239 (2002), 50−56.

[4] Heydari, M., Hooshmandasl, M. and Ghaini, F. *Good approximate solution for Lienard equation in a large interval using block pulse functions*, J. Math. Ext. 7 (2013), 17−32.

[5] Horadam, A.F. *New Aspects of Morgan–Voyce Polynomials*, Applications of Fibonacci Numbers, (1998), 161−176.

[6] Jianyu, L., Siwei, L., Yingjian, Q. and Yaping, H. *Numerical solution of elliptic partial differential equation using radial basis function neural networks*, Neural Network, 16 (2003), 729−734.

[7] Kaya, D. and El-Sayed, S. *A numerical implementation of the decomposition method for the Lienard equation*, Appl. Math. Comput., 171 (2005), 1095−1103.

[8] Keshavarz, E., Ordokhani, Y. and Razzaghi, M. *The Taylor wavelets method for solving the initial and boundary value problems of Bratu-type equations*, Appl. Numer. Math. 128 (2018), 205−216.

[9] Kiltu, G. and Duressa, G. *Accurate numerical method for Liénard nonlinear differential equations*, J. Taibah Univ. Sci. 13 (2019), 740−745.

[10] Kong, D. *Explicit exact solutions for the Lienard equation and its applications*, Phys. Lett. A. 196 (1995), 301−306.

[11] Kumar, D., Agarwal, P., and Singh, J. *A modified numerical scheme and convergence analysis for fractional model of Lienard's equation*, J. Comput. Appl. Math. 339 (2018), 405−413.

[12] Kumbinarasaiah, S. and Adel, W. *Hermite wavelet method for solving nonlinear Rosenau-Hyman equation*, Partial Partial Differ. Equ. Appl. Math. 4 (2021), 100062.

[13] Liénard, A. *Etude des oscillations entretenues*, Rev. Gen. Electr. 23(901-912) (1928), 946−954.

[14] Malek, A. and Beidokhti Shekari, R. *Numerical solution for high order deferential equations, using a hybrid neural network-Optimization method*, Appl. Math. Comput. 183 (2006), 260−271.

[15] Mall, S. and Chakraverty, S. *Numerical solution of nonlinear singular initial value problems of Emden-Fowler type using Chebyshev neural network method*, Neurocomputing, 149 (2015), 975−982.

[16] Masood, Z., Majeed, K., Samar, R. and Raja, M.A.Z. *Design of Mexican Hat wavelet neural networks for solving Bratu type nonlinear systems*, Neurocomputing, 221 (2017), 1−14.

[17] Matinfar, M., Bahar, S. and Ghasemi, M. *Solving the Lienard equation by differential transform method*, World J. Model. Simul. 8 (20112), 142−146.

[18] Matinfar, M., Hosseinzadeh, H. and Ghanbari, M. *A numerical implementation of the variational iteration method for the Lienard equation*, World J. Model. Simul. 4(2008), 205−210.

[19] Matinfar, M., Mahdavi, M. and Raeisy, Z. *Exact and numerical solution of Lienard's equation by the variational homotopy perturbation method*, J. Inf. Comput. Sci., 6 (2011), 73−80.

[20] Meade Jr, A.J. and Fernandez, A.A. *Solution of nonlinear ordinary differential equations by feedforward neural networks*, Math. Comput. Model. 20 (1994), 19−44.

[21] Meade Jr, A.J. and Fernandez, A.A. *The numerical solution of linear ordinary differential equations by feed forward neural networks*, Math. Comput. Model. 19 (1994), 1−25.

[22] Rahimkhani, P. *Numerical solution of nonlinear stochastic differential equations with fractional Brownian motion using fractional-order Genocchi deep neural networks*, Commun. Nonlinear Sci. Numer. Simul. 126 (2023), 107466.

[23] Rahimkhani, P. *Deep neural network for solving stochastic biological Systems*, Iran. J. Sci. Technol. Trans. A: Sci. 48 (2024), 687−696.

[24] Rahimkhani, P. and Heydari, M.H. *Fractional shifted Morgan–Voyce neural networks for solving fractal-fractional pantograph differential equations*, Chaos Solit. Fractals. 175 (2023), 114070.

[25] Rahimkhani, P. and Ordokhani, Y. *Orthonormal Bernoulli wavelets neural network method and its application in astrophysics,* Comput. Appl. Math. 40(30) (2021), 1−24.

[26] Rahimkhani, P. and Ordokhani, Y. *Chelyshkov least squares support vector regression for nonlinear stochastic differential equations by variable fractional Brownian motion*, Chaos Solit. Fractals. 163 (2022), 112570.

[27] Rahimkhani, P., Ordokhani, Y. and Sabermahani, S. *Hahn hybrid functions for solving distributed order fractional Black-Scholes European option pricing problem arising in financial market*, Math. Methods Appl. Sci. 46(6) (2023), 6558−6577.

[28] Raja, M.A.Z., Mehmood, J., Sabir, Z., Nasab, A.K. and Manzar, M.A. *Numerical solution of doubly singular nonlinear systems using neural networks-based integrated intelligent computing*, Neural. Comput. Appl. 31 (2019), 793−812.

[29] Selvaraju, N. and Abdul Samant, J. *Solution of matrix Riccati differential equationfor nonlinear singular system using neural networks*, Int. J. Comput. Appl. 29 (2010), 48−54.

[30] Shannon, A.G. and Horadam, A.F. *Some Relationships among Vieta, Morgan–Voyce and Jacobsthal Polynomials*, Applications of Fibonacci Numbers, (1999), 307−323.

[31] Shiralashetti, S.C. and Kumbinarasaiah, S. *Cardinal B-spline wavelet based numerical method for the solution of generalized Burgers-Huxley equation*, Int. J. Appl. Comput. Math. 4(73) (2018), (2018).

[32] Singh, H. *Solution of fractional Lienard equation using Chebyshev operational matrix method*, Nonlinear Sci. Lett. A. 8 (2017), 397−404.

[33] Srinivasa, K., Rezazadeh, H. and Adel, W. *Numerical investigation based on Laguerre wavelet for solving the Hunter Saxton equation*, Int. J. Appl. Comput. Math. 6 (2020), 139.

[34] Xian-Lin, Y. and Jia-Shi, T. *Exact solutions to the generalized Lienard equation and its applications*, Pramana, 71 (2008), 1231−1245.

[35] Yin, Z. and Chen, H. *On stability of periodic solutions of Lienard type equations*, Discrete Dyn. Nat. Soc. 2017(1) (2017), 1805760.