mathematics

MDPI

Article

Morlet Wavelet Neural Network Investigations to Present the Numerical Investigations of the Prediction Differential Model

Zulqurnain Sabir [1], Adnène Arbi [2,3,*], Atef F. Hashem [4,5] and Mohamed A Abdelkawy [4,5]

[1] Department of Computer Science and Mathematics, Lebanese American University, Beirut 1401, Lebanon; zulqurnain_maths@hu.edu.pk
[2] Laboratory of Engineering Mathematics (LR01ES13), Tunisia Polytechnic School, University of Carthage, Tunis 2078, Tunisia
[3] Department of Advanced Sciences and Technologies at National School of Advanced Sciences and Technologies of Borj Cedria, University of Carthage, Hammam-Chott 1164, Tunisia
[4] Department of Mathematics and Statistics, College of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 13318, Saudi Arabia; affaragalla@imamu.edu.sa (A.F.H.); maohamed@imamu.edu.sa (M.A.A.)
[5] Department of Mathematics and Information Science, Faculty of Science, Beni-Suef University, Beni-Suef 62514, Egypt
* Correspondence: adnen.arbi@enseignant.eduent.tn

Abstract: In this study, a design of Morlet wavelet neural networks (MWNNs) is presented to solve the prediction differential model (PDM) by applying the global approximation capability of a genetic algorithm (GA) and local quick interior-point algorithm scheme (IPAS), i.e., MWNN-GAIPAS. The famous and historical PDM is known as a variant of the functional differential system that works as theopposite of the delay differential models. A fitness function is constructed by using the mean square error and optimized through the GA-IPAS for solving the PDM. Three PDM examples have been presented numerically to check the authenticity of the MWNN-GAIPAS. For the perfection of the designed MWNN-GAIPAS, the comparability of the obtained outputs and exact results is performed. Moreover, the neuron analysis is performed by taking 3, 10, and 20 neurons. The statistical observations have been performed to authenticate the reliability of the MWNN-GAIPAS for solving the PDM.

Keywords: Morlet wavelet kernel; prediction differential system; genetic algorithm; delay differential system; interior-point algorithm scheme

MSC: 68T07; 60G25; 68W50; 34K40; 93C43; 90C51

check for updates

Citation: Sabir, Z.; Arbi, A.; Hashem, A.F.; Abdelkawy, M.A. Morlet Wavelet Neural Network Investigations to Present the Numerical Investigations of the Prediction Differential Model. *Mathematics* **2023**, *11*, 4480. https://doi.org/10.3390/math11214480

Academic Editors: Rodolfo Haber, Krzysztof Ejsmont, Aamer Bilal Asghar and Yong Wang

Received: 3 October 2023
Revised: 22 October 2023
Accepted: 24 October 2023
Published: 29 October 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

The study of the prediction differential model (PDM) is considered very significant for researchers due to its various applications in climate forecasting, biological systems, stock markets, transport, astrophysics, engineering, etc. The sense of delay differential model (DDM) is presented due to the historical system, which has been applied to design the form of PDM. The idea of DDM, introduced by Newton and Leibnitz, was presented a few centuries ago and has widely been applied in many applications of engineering, economic systems, population dynamics, communication, and transport networks [1–5]. Many researchers applied different techniques to solve the DDM. For example, Bildik et al. [6] proposedsolving the DDM using the optimal perturbation iterative scheme. Rahimkhani et al. [7] presented an approach to solve the fractional form of DDM. Sabir et al. [8] presented a new multi-singular nonlinear system with delayed factors. Aziz et al. [9] used a Haar wavelet approach to solve the partial form of DDM. Frazier [10] implemented a wavelet Galerkin method to solve the DDM of the second kind. Tomasiello [11] solved a famous class of the historical DDM by applying the fuzzy transform method. Vaid [12] implemented

the trigonometric B-spline approach to solve the second kind of singularly perturbed-based DDM. Hashemi et al. [13] solved the fractional pantograph delay system by applying an efficient computational approach. Adel et al. [14] discussed the solutions of a pantograph singular DDM using the Bernoulli collocation scheme. Erdogan et al. [15] worked to solve the perturbed singularly DDM using a well-known finite difference approach. The DDM is a second-order differential model, which is given as [16]:

$$\begin{cases} y''(t) = f(t,\, y(t),\, y(t-\gamma_1)), & \gamma_1 > 0,\ \ c \le t \le b, \\ y(t) = \theta(t), & \sigma \le t \le c, \qquad 0 \le \gamma_1 \le |c-\sigma|, \\ y'(c) = w, \end{cases} \tag{1}$$

where γ_1 and $\theta(t)$ indicate the delayed factor and initial conditions. The delayed form $y(t-\gamma_1)$ is shownin the above model, which is to subtract in time t, i.e., $c \le t \le b$. σ, is a small constant and w is value derivative of y. The prediction form of the DDM is achieved by adding some terms in t, i.e., $y(t+\gamma_1)$, with prediction term γ_1. The literature form of the mathematical PDM is given as [17,18]:

$$\begin{cases} y''(t) = f(t,\, y(t),\, y(t+\gamma_1)), & \gamma_1 > 0,\ \ c \le t \le b, \\ y(t) = \theta(t), & \sigma \le t \le c, \qquad 0 \le \gamma_1 \le |\sigma-c|, \\ y'(c) = w, \end{cases} \tag{2}$$

The above mathematical PDM shown in Equation (2) has been designed recently and has never been solved by functioning the universal approximation ability of the Morlet wavelet neural network (MWNN) together with the global and local search optimizations of a genetic algorithm (GA) and interior-point algorithm scheme (IPAS), i.e., MWNN-GAIPAS. The numerical investigations have been performed by using the MWNN-GAIPAS by taking 3, 10, and 20 neurons. Recently, stochastic computing solvers have been used to exploit the Hall effects on boundary layer flow [19], the nonlinear doubly singular model [20], large-scale continuous multi-objective optimization [21], the heat flux model [22], functional differential singular systems [23], the detection and contact tracing of coronavirus [24], control autoregressive systems [25], the Thomas–Fermi system [26], the Ree–Eyring dissipative fluid flow system [27], stiff nonlinear models [28], fractional multi-singular differential models [29], a mosquito dispersal system in a heterogeneous atmosphere [30], eye surgery models [31], and dynamical models based on the HIV prevention system [32]. These above performances of stochastic solvers authenticate their worth in terms of robustness, convergence, and precision. Based on the above applications, the authors are inspired to present the solutions of the PDM by using the universal approximation ability of MWNN together with the optimization procedures of GAIPAS. A few noticeable, prominent, and salient measures of the current study aresummarized as follows:

- A layer structure of MWNNs is designed, and optimization is performed through an integrated neuro-evolution-based heuristic with IPAS to solve the PDM numerically;
- The analysis with 3, 10, and 20 neurons is presented to interpret the stability and accuracy of the designed approach for solving the PDM;
- The proposed MWNN-GAIPAS is executed for three different examples based on PDM, and a comparison is performed with the exact solutions to validate the accuracyof the proposed MWNN-GAIPAS;
- Statistical investigations through different performances of fitness: "root mean square error (R.MSE)", "variance account for (VAF)", "Theil's inequality coefficients (TIC)", and semi-inter quartile range (S.I.R) further authenticate the MWNN-GAIPAS for solving all examples of the PDM;
- The complexity performances of the MWNN-GAIPAS based on 3, 10, and 20 neurons with the use of different statistical operators are examined for all of the examples of the PDM;

- The proposed MWNN-GAIPAS provides reasonable and accurate results in the training span. Furthermore, smooth processes of implementation, constancy, and expendability are other obvious advantages.

The organization of the paper is as follows: Section 2 provides the details of the design of MWNN-GAIPAS. Performance procedures are given in Section 3. Results are provided in Section 4. Conclusions, along with upcoming reports of the research, are provided in the final Section.

2. Methodology: MWNN-GAIPAS

The proposed methodology based on the MWNN-GAIPAS to solve each example of the PDM is separated into two phases.

- An error-based merit function is presented to construct the MWNNs;
- For the optimization of the merit function, the hybrid form of GAIPAS is described for the decision variables of MWNNs.

2.1. MWNN Modeling

The ability of the NNs to usethe MW function is used to present stable, steady, and reliable outcomes in many areas. The PDM mathematical form given in Equation (2) is stated with feed-forward NNs, including the derivatives in input, hidden, and output layers as:

$$\begin{aligned} \hat{y}(t) &= \sum_{k=1}^{s} q_k v(w_k t + m_k), \\ \hat{y}^{(\mathrm{n})} &= \sum_{k=1}^{s} q_k v^{(\mathrm{n})}(w_k t + m_k). \end{aligned} \tag{3}$$

In the above network, s represents the neurons,$\boldsymbol{W} = [\boldsymbol{q}, \boldsymbol{w}, \boldsymbol{m}]$ is the unknown weight vector, i.e., $\boldsymbol{q} = [q_1, q_2, \ldots, q_s]$, $\boldsymbol{w} = [w_1, w_2, \ldots, w_s]$ *and* $\boldsymbol{m} = [m_1, m_2, \ldots, m_s]$. The MWNN has not been implemented before to present the numerical solutions of PDM. Morlet wavelet (MW) is one of the forms of continuous wavelet, which is frequently applied in the process of time-frequency, signal processing, as well as image processing. This function is categorized through its aptitude to examine the frequency based on a signal using various scales. It is also defined as a function of complex values based on both imaginary and real mechanisms. In conventional neural networks, the neurons are characteristically stimulated by applying different functions, e.g., ReLU, log-sigmoid, or hyperbolic tangent. However, an architecture based on the neural network is defined, where the above functions can be replaced with the MW functions. The MW function is mathematically given as [33]:

$$v(t) = \cos\left(\frac{4}{3}t\right) e^{\left(-\frac{1}{2}t^2\right)} \tag{4}$$

Equation (3) takes the form of:

$$\begin{aligned} \hat{y}(t) &= \sum_{k=1}^{s} q_k \cos\left(\tfrac{4}{3}(w_k t + m_k)\right) e^{-\frac{1}{2}(w_k t + m_k)^2}, \\ \hat{y}'(t) &= \sum_{k=1}^{s} -q_k w_k e^{-\frac{1}{2}(w_k \tau + m_k)^2} \begin{pmatrix} \sin\left\{\tfrac{4}{3}(w_k \tau + m_k)\right\} \\ +\tfrac{4}{3}(w_k \tau + m_k)\cos\left\{\tfrac{4}{3}(w_k \tau + m_k)\right\} \end{pmatrix}, \\ \hat{y}''(t) &= \sum_{i=1}^{s} -q_k w_k^2 e^{-\frac{1}{2}(w_k t + m_k)^2} \begin{pmatrix} 3.0625\cos\left\{\tfrac{4}{3}(w_k t + m_k)\right\} \\ +\tfrac{7}{2}(w_k t + m_k)\sin\left\{\tfrac{4}{3}(w_k t + m_k)\right\} \\ +\left\{-1 + (w_k t + m_k)^2\right\}\cos\left\{\tfrac{4}{3}(w_k t + m_k)\right\} \end{pmatrix}, \end{aligned} \tag{5}$$

A merit function E is defined as:

$$E = E_1 + E_2 \tag{6}$$

where E_1 and E_2 are the unsupervised error based on the differential model and boundary conditions of system (2), shown as:

$$E_1 = \frac{1}{N}\sum_{m=1}^{N}(y''_m - f(t_m, y_m, y(t_m + \gamma_1)), \quad 0 \le t_m \le 1, \tag{7}$$

where $Nh = 1,$ and $t_m = mh$.

$$E_2 = \frac{1}{2}(\hat{y}_0 - \theta(t_m))^2 + \frac{1}{2}(\hat{y}'_N - w)^2. \tag{8}$$

2.2. Optimization Process: GAIPAS

The optimization through MWNN is accomplished to solve all the examples of the PDM using the computing hybrid construction of GA and IPAS, i.e., GAIPAS.

*Genetic Algorithm*is a reliable global search method that is applied to unconstrained, nonlinear systems using its important operators called selection, elitism, crossover, and mutation. Recently, GAs have been used in extensive applications in heart disease diagnosis models [34], economic and environmental multi-objective-based optimization of a household level of renewable energy [35], power and heat economic dispatch models [36], electromagnetic detection satellite scheduling models [37], prediction of air blasts [38], data sciences [39], fraud transactions of Ethereum smart contracts [40], and monorail vehicle-based dynamical systems [41]. These recent potential citations inspired the authors to use the global search GA process to achieve the decision variables of MWNNs for solving the PDM.

*The interior-point algorithm scheme*is a well-known local search mechanism implemented for convex optimization systems. IPAS works to solve the optimization problems of both types—constrained and unconstrained. Recently, IPAS has been used in image restoration [42], nested-constraint resource allocation problems [43], power system state estimation [44], risk-averse PDE-constrained optimization problems [45], and monotone weighted linear complementarity problems [46].

The hybridization is performed to regulate the sluggishness of GA with the IPAS through the optimization procedure. Structure of MWNN-GAIPAS for solving the PDM is shown in Figure 1. The detail of the hybridization of GAIPAS is provided in Algorithm 1.

Algorithm 1. The optimization-based MWNN-GAIPAS is given in the pseudo-code for solving the PDM.

"GA" start

(i) **Inputs:** Select the chromosome selected with equal entries of the system as: $\boldsymbol{W} = [\boldsymbol{q}, \boldsymbol{w}, \boldsymbol{m}]$

(ii) **Population:** Chromosomes are given as:

$$\boldsymbol{q} = [q_1, q_2, \ldots, q_s],\ \boldsymbol{w} = [w_1, w_2, \ldots, w_s], \boldsymbol{m} = [m_1, m_2, m_3, \ldots, m_k].$$

(iii) **Output:** Best GA values are denoted as: $\boldsymbol{W_{BGA}}$

(iv) **Initialization:** Form a vector of weights "$\boldsymbol{W}$" to signify the 'chromosome'. This set is related tothe initial population. Set the values of 'Generation' and 'declaration' based on [gaoptimset] and [GA].

(v) **Fitness designs:** Adjust$\boldsymbol{E}$, which is fitness in population for each vector.

(vi) **Ending process:** Stop if

- [Fitness=10^{-19}], [Iterations =120], [PopulationSize240], [TolCon/Fun=10^{-21}], [StallLimit=135],
- Setas default the other values.

Goto **[storage]** to achieve the stopping standards.

(vii) **Rank:** For the fitness, rank the weight vectors in *Population.*

Algorithm 1. *Cont.*

(viii) **Generation:** {selectionuniform}, {mutationadaptfeasible}, {crossoverheuristic}.

(ix) **Generation: Storage:** To obtain the best W_{BGA}, counts of function, Generations, and E.

End of GA
IPAS initiates

(i) W_{BGA} is an initial point.

(ii) **Output:** Optimal GAIPAS denoted as W_{GAIPAS}.

(iii) **Initialize:** Take the best W_{BGA}, Bounded restraints, assignments, iterations, and the other points.

(iv) **Terminate:** Practice terminates when any of the condition obtains [$E = 10^{-18}$], [MaxFunEvals= 275000], [Iterations = 650], [TolX/Con=10^{-21}], [TolFun =10^{-22}].

(v) **Fitness:** Calculate$E$, and W.

(vi) **Modifications:** Use "fmincon" usingIPAS. Adjust 'W' for IPAS.

(vii) **Accumulate:**Adjust 'W_{GAIPAS}', 'E', 'count of function', epochs, and 'time'.

IPAS procedure ends

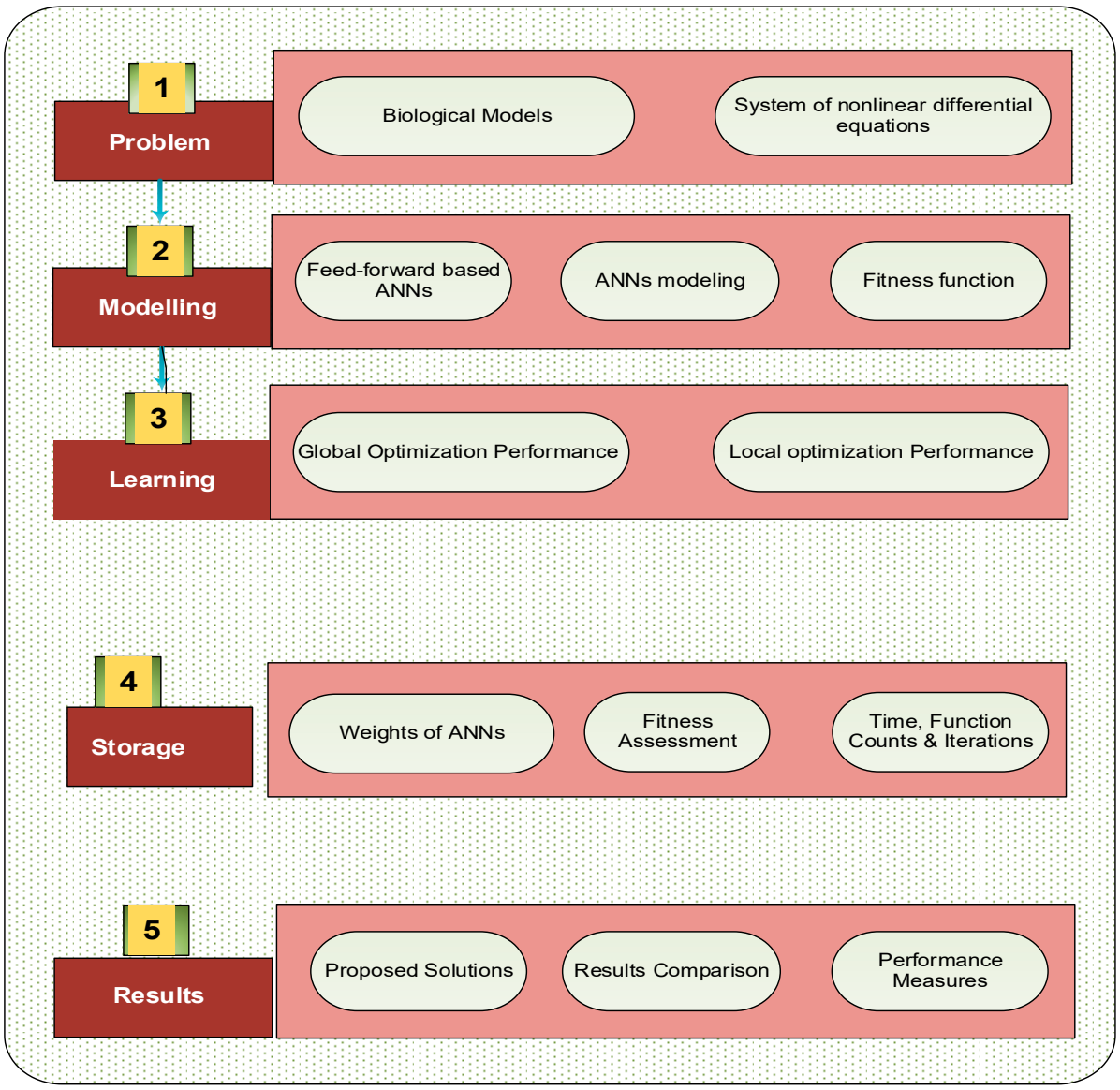


Figure 1. Structure of MWNN-GAIPAS for solving the PDM.

3. Statistical Performances

To check the consistency and reliability of the MWNN-GAIPAS, particularly in the framework of statistical modeling or data analysis, a number of statistical measures and schemes have been applied. The choice of metrics is dependent on the precise scheme's features and the goals of the obtained analysis. Some of the common statistical measures used in this study are:

Root mean square error (R.MSE): An extensively applied metric to assess the accuracy based on the predictive system, mainly in the forecasting of time series and analysis of regression. It trials the error's average magnitude or residuals based on the predicted performances along with the actual observed measures using the dataset. The mathematical form of the RMSE is given as:

$$RMSE = \sqrt{\frac{1}{s}\sum_{i=1}^{s}(y_i - \hat{y}_i)^2}, \tag{9}$$

where s is the total data points used in the dataset, y_i and $\hat{y}_i$ represent the actual and proposed values, while the symbol of summation is used to sum the values for i from 1 to s.

VAF: One of the metrics thatis applied to assess the presentation of a regression. It provides the percentage of the total variance in the real performances, which is reported through the variance in the proposed measures. The mathematical VAF is shown as follows:

$$\begin{cases} VAF = \left(1 - \frac{\mathrm{var}(y_i - \hat{y}_i)}{\mathrm{var}(y_i)}\right) \times 100, \\ \mathrm{E-VAF} = [|100 - \mathrm{VAF}|], \end{cases} \tag{10}$$

TIC: Also one of the statistical operators used to check the reliability of the MWNN-GAIPAS for solving the PDM, which is mathematically provided as:

$$TIC = \frac{\sqrt{\frac{1}{s}\sum_{i=1}^{s}(y_i - \hat{y}_i)^2}}{\left(\sqrt{\frac{1}{s}\sum_{i=1}^{s} y_i^2} + \sqrt{\frac{1}{n}\sum_{i=1}^{s} \hat{y}_i^2}\right)}, \tag{11}$$

S. IR: Also used to measure the statistical dispersal, which is associated withthe interquartile range. The mathematical S. IR form is presented as:

$$\begin{cases} \mathrm{S.I\,R} = -\frac{1}{2} \times (q_1 - q_3), \\ q_1 \text{ and } q_3 = 1^{st} \text{ and } 3^{rd} \text{ quartiles.} \end{cases} \tag{12}$$

4. Simulations of the Results

The comprehensive form of the solutions based on three examples of the PDM ispresented in this section.

Example 1. *Consider*

$$\begin{cases} 2y''(t) - y(t+\pi) + y(t) = 0, \\ y(0) = 1, y'(0) = 1. \end{cases} \tag{13}$$

The exact form of the Equation (13) is$y(t) = 1 + \sin(t)$*, while the fitness function is shown as:*

$$E = \frac{1}{N}\sum_{i=1}^{N}\left(2\hat{y}_i'' + \hat{y}_i - \hat{y}(t_i + \pi)\right)^2 + \frac{1}{2}\left((\hat{y}_0 - 1)^2 + \left(\hat{y}_0' - 1\right)^2\right). \tag{14}$$

Example 2. *Consider the trigonometric PDM-based problem given as:*

$$\begin{cases} y''(t) - y'(t+1) + y(t+1) + y(t) + \cos(1+t) - \sin(1+t) = 0, \\ y(0) = 0, \quad y'(0) = 1. \end{cases} \tag{15}$$

The exact form of the above model (15) is $y(t) = \sin(t)$, *and the merit function is given as:*

$$E = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i'' - \hat{y}'(t_i+1) + \hat{y}(t_i+1) + \hat{y}(t_i) + \cos(1+t_i) - \sin(1+t_i)\right)^2 + \frac{1}{2}\Big((\hat{y}_0)^2 + (\hat{y}_0' - 1)^2\Big). \tag{16}$$

Example 3. *Consider the PDM-based equation given as:*

$$\begin{cases} \hat{y}''(t) + y(t+1) - y(t) - 2t = 0, \\ y(1) = 0, \quad y(0) = 2. \end{cases} \tag{17}$$

The exact form of the above model (17) is $y(t) = t^2 - 3t + 2$, and the merit function is given as:

$$E = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i'' - \hat{y}(t_i) + \hat{y}(1+t_i) - 2t_i\right)^2 + \frac{1}{2}\Big((\hat{y}_N - 2)^2 + (\hat{y}_0)^2\Big). \tag{18}$$

The prediction terms are $y'(t+1)$, $y(t+\pi)$, and $y(t+1)$ in the above examples. The optimization of each example is performed by using the MWNN-GAIPAS for 40 independent executions in order to assess the parameters. The best weight set is accessible to authenticate the proposed outcomes of the PDM are given in Equations (19)–(21), (22)–(24), and (25)–(27) for 3, 10, and 20 neurons. The estimated results using 3, 10, and 20 neurons are given as:

$$\begin{aligned} \hat{y}_{E-\mathrm{I}}(t) = {} & 19.387\cos\Big(\tfrac{4}{3}(0.6186t + 3.6688)\Big)e^{-0.5(0.6186t+3.6688)^2} \\ & +5.1798\cos\Big(\tfrac{4}{3}(-1.2053t - 4.2393)\Big)e^{-0.5(-1.205t-4.23)^2} \\ & +2.0001\cos\Big(\tfrac{4}{3}(-0.3514t + 0.5519)\Big)e^{-0.5(-0.351t+0.55)^2}, \end{aligned} \tag{19}$$

$$\begin{aligned} \hat{y}_{E-\mathrm{II}}(t) = {} & -20\cos\Big(\tfrac{4}{3}(-1.1897t - 3.3343)\Big)e^{-0.5(-1.1897t-3.3343)^2} \\ & +20.0\cos\Big(\tfrac{4}{3}(20.000t + 4.8592)\Big)e^{-0.5(20.000t+4.8592)^2} \\ & +2.00\cos\Big(\tfrac{4}{3}(5.3245t + 6.4450)\Big)e^{-0.5(5.3245t+6.4450)^2}, \end{aligned} \tag{20}$$

$$\begin{aligned} \hat{y}_{E-\mathrm{III}}(t) = {} & 19.9986\cos\Big(\tfrac{4}{3}(1.8152t + 3.9123)\Big)e^{-\frac{1}{2}(1.8152t+3.9123)^2} \\ & -16.3478\cos\Big(\tfrac{4}{3}(0.5487t + 2.0246)\Big)e^{-\frac{1}{2}(0.5487t+2.0246)^2} \\ & -0.18690\cos\Big(\tfrac{4}{3}(-1.0269t + 1.458)\Big)e^{-\frac{1}{2}(-1.0269t+1.458)^2}, \end{aligned} \tag{21}$$

$$\begin{aligned} \hat{y}_{E-\mathrm{I}}(t) = {} & 0.3846\cos\Big(\tfrac{4}{3}(-0.2461t+0.456)\Big)e^{-0.5(-0.2461t+0.456)^2} \\ & -0.5671\cos\Big(\tfrac{4}{3}(-0.0886t + 2.668)\Big)e^{-0.5(-0.0886t+2.668)^2} \\ & -2.1220\cos\Big(\tfrac{4}{3}(0.3888t + 0.6070)\Big)e^{-0.5(0.3888t+0.6070)^2} \\ & +.... - 0.6948\cos\Big(\tfrac{4}{3}(0.3344t - 1.6760)\Big)e^{-0.5(0.3344t-1.6760)^2}, \end{aligned} \tag{22}$$

$$\begin{aligned} \hat{y}_{E-\mathrm{II}}(t) = {} & -19.98\cos\Big(\tfrac{4}{3}(-19.996t - 5.153)\Big)e^{-\frac{1}{2}(-19.996t-5.153)^2} \\ & +19.9970\cos\Big(\tfrac{4}{3}(3.6362t - 7.2460)\Big)e^{-\frac{1}{2}(3.6362t-7.246)^2} \\ & -1.80890\cos\Big(\tfrac{4}{3}(1.3754t + 2.2409)\Big)e^{-\frac{1}{2}(1.3754t+2.240)^2} \\ & +.... + 3.09940\cos\Big(\tfrac{4}{3}(7.0618t + 8.1960)\Big)e^{-\frac{1}{2}(7.0618t+8.196)^2}, \end{aligned} \tag{23}$$

$$\begin{aligned}\hat{y}_{E-\mathrm{III}}(t) = {} & 1.3604\cos\left(\tfrac{4}{3}(-2.9655t-6.807)\right)e^{-\frac{1}{2}(-2.9655t-6.807)^2} \\ & +1.6470\cos\left(\tfrac{4}{3}(-3.027t+5.7125)\right)e^{-\frac{1}{2}(-3.027t+5.7125)^2} \\ & -18.224\cos\left(\tfrac{4}{3}(1.7829t-4.7960)\right)e^{-\frac{1}{2}(1.7829t-4.7960)^2} \\ & +\ldots+6.4136\cos\left(\tfrac{4}{3}(-2.9551t-3.147)\right)e^{-\frac{1}{2}(-2.9551t-3.147)^2},\end{aligned} \tag{24}$$

$$\begin{aligned}\hat{y}_{E-\mathrm{I}}(t) = {} & -0.4821\cos\left(\tfrac{4}{3}(-0.5832t-0.908)\right)e^{-0.5(-0.5832t-0.908)^2} \\ & -1.27390\cos\left(\tfrac{4}{3}(-1.114t+0.1298)\right)e^{-0.5(-1.114t+0.1298)^2} \\ & +1.11890\cos\left(\tfrac{4}{3}(1.1290t-0.3187)\right)e^{-0.5(1.1290t-0.3187)^2} \\ & +\ldots.-0.1105\cos\left(\tfrac{4}{3}(-0.7800t+1.3431)\right)e^{-0.5(-0.7800t+1.3431)^2},\end{aligned} \tag{25}$$

$$\begin{aligned}\hat{y}_{E-\mathrm{II}}(t) = {} & -9.24\cos\left(\tfrac{4}{3}(10.2434t+7.2396)\right)e^{-0.5(10.2434t+7.2396)^2} \\ & +19.991\cos\left(\tfrac{4}{3}(19.991t+4.9252)\right)e^{-0.5(19.991t+4.9252)^2} \\ & -19.992\cos\left(\tfrac{4}{3}(19.9923t+4.908)\right)e^{-0.5(19.9923t+4.908)^2} \\ & +\ldots.+10.897\cos\left(\tfrac{4}{3}(1.1228t-6.9925)\right)e^{-0.5(1.1228t-6.9925)^2},\end{aligned} \tag{26}$$

$$\begin{aligned}\hat{y}_{E-\mathrm{III}}(t) = {} & -1.366\cos\left(\tfrac{4}{3}(4.1848t-6.807)\right)e^{-0.5(4.1848t-6.807)^2} \\ & +0.4530\cos\left(\tfrac{4}{3}(0.0045t+3.020)\right)e^{-0.5(0.0045t+3.020)^2} \\ & -1.4459\cos\left(\tfrac{4}{3}(0.0093t-4.305)\right)e^{-0.5(0.0093t-4.305)^2} \\ & +\ldots.+19.191\cos\left(\tfrac{4}{3}(1.0855t+3.0155)\right)e^{-0.5(1.0855t+3.015)^2}.\end{aligned} \tag{27}$$

The performances of the optimization to solve the model are presented through MWNN-GAIPAS for the trained values set using forty independent executions based on 3, 10, and 20 neurons. Figure 2 is drawn using the 3, 10, and 20 neurons based on Equations (19)–(27) to find the optimal weights.

For the comparison, the obtained outcomes have been compared with the exact solutions to solve each example of the PDM using 3, 10, and 20 neurons. These comparison plots are drawn in Figure 3,and onecan observe that the optimal results intersect with the exact outcomes for each example of the PDM by considering 3, 10, and 20 neurons, which shows the precision of MWNN-GAIPAS.

The performances using the designed MWNN-GAASAS for each example of PDM with 3, 10, and 20 neurons are tabulated in Table 1, Table 2, and Table 3, respectively. The Minimum (Min), Median (Med), Mean, S.I.R, and standard deviation (SD) values for Examples 1, 2, and 3were found to be good measures for each example of the PDM. These small, calculated values based on these statistics gages for each example of the PDM based on 3, 10, and 20 neurons show the accuracy of the designed MWNN-GAIPAS.

The plots of absolute error (AE) for each example of the PDM for 3, 10, and 20 neurons are shown in Figure 4a, Figure 4b, and Figure 4c, respectively. One can observe that the AE values using three neurons for Examples 1, 2, and 3 lie at 10^{-7}–10^{-9}, 10^{-5}–$10^{-6}$and 10^{-4}–10^{-6}. The AE for 10 neurons for Examples 1, 2, and 3 lies at 10^{-7}–10^{-9}, 10^{-6}–$10^{-8,}$ and 10^{-07}–10^{-9}. For 20 neurons, the AE for Examples 1, 2, and 3 lies at around 10^{-07}–10^{-09}, 10^{-6}–10^{-07}, and 10^{-07}–10^{-10}, respectively. It is clear in the AE that each example of the PDM is found in good ranges for considering 3,10, and 20 neurons. The R.MSE, Fitness (FIT), TIC, and EVAF are measured in Figure 5a–c for 3, 10, and 20 neurons. Figure 5a presents that the best FIT for Examples 1 to 3is calculated as10^{-09}–10^{-10}, 10^{-07}–10^{-09}, and 10^{-5}–10^{-6}, respectively. The optimal R.MSE is found around 10^{-6}–10^{-07}, while the R.MSE for the other two examples lie at around 10^{-5}–10^{-6}. The best EVAF values are found around 10^{-11}–10^{-12}, while the R.MSE for the other two examples lies at around 10^{-09} to 10^{-10}. The TIC best values for all the Examples based on three neurons are calculated around

10^{-09} to 10^{-10}. The performance based on 10 neurons is presented in Figure 5b. The plots in this figure indicate that the FIT and EVAF best measures for each example of the PDM lie at 10^{-10} to 10^{-12}, while the R.MSE and TIC best values lie at around 10^{-6}–10^{-8} and 10^{-10}–10^{-12}, respectively. The performance based on 20 neurons is presented in Figure 5c. The plots in this figure indicate that the best values of these statistical operators for Example 1 to 3 lie at around 10^{-10}–10^{-12}, 10^{-6}–10^{-8}, 10^{-12}–10^{-14}, and 10^{-10}–10^{-12}. These obtained outcomes verify the competent trend for solving PDM based on 3, 10, and 20 neurons.

The statistical performance to solve each example of the PDM are drawn in Figures 6–9 for 3, 10, and 20 neurons. Figure 6 presents the FIT measures for 40trials to present the solutions of each example of the PDM using 3, 10, and 20 neurons. One can observe that the best runs are found around $10^{-01}$to 10^{-04}, 10^{-02} to $10^{-8,}$ and $10^{-02}$to 10^{-12} for solving each Example of the PDM using 3, 10, and 20 neurons. Figure 7 shows the statistical investigations through R.MSE using MWNN-GAIPAS for each example of the PDM taking 3, 10, and 20 neurons.It is seen that the best runs are found around 10^{-01}–10^{-3}, 10^{-02}–$10^{-6,}$ and 10^{-02}–10^{-8} for solving each example of the PDM using 3, 10, and 20 neurons. Figure 8 shows the statistical investigations through EVAF using MWNN-GAIPAS for each example of the PDM taking 3, 10, and 20 neurons.It is seen that the best runs are found around 10^{-01}–10^{-5}, 10^{-02}–$10^{-08,}$ and 10^{-02}–10^{-15} for solving each example of the PDM using 3, 10, and 20 neurons. Figure 9 represents the statistical investigations through TIC using MWNN-GAIPAS for each example of the PDM taking 3, 10, and 20 neurons. It is seen that the best runs are found around 10^{-02}–10^{-07}, 10^{-04}–$10^{-10,}$ and 10^{-04}–10^{-12} for solving each example of the PDM using 3, 10, and 20 neurons. It is easy to understandthat by taking three numbers of neurons, the method performs quicker in the comparison of 10 and 20 neurons, but one can obtain more reliable solutions by taking a larger neuron.

Table 1. Statistical measures using MWNN-GAAPAS for each example of the PDM using three neurons.

	Mode	$\hat{y}(t)$										
		0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
E-I	Min	4.14×10^{-9}	2.07×10^{-8}	2.85×10^{-7}	4.46×10^{-7}	4.86×10^{-7}	5.58×10^{-7}	7.48×10^{-7}	1.00×10^{-6}	1.20×10^{-6}	1.30×10^{-6}	1.34×10^{-6}
	Mean	3.75×10^{-1}	4.13×10^{-1}	4.54×10^{-1}	4.98×10^{-1}	5.41×10^{-1}	5.83×10^{-1}	6.22×10^{-1}	6.59×10^{-1}	6.92×10^{-1}	7.23×10^{-1}	7.49×10^{-1}
	SD	4.46×10^{-1}	5.1×10^{-1}	5.57×10^{-1}	6.7×10^{-1}	6.53×10^{-1}	6.95×10^{-1}	7.35×10^{-1}	7.72×10^{-1}	8.05×10^{-1}	8.34×10^{-1}	8.60×10^{-1}
	Med	3.93×10^{-2}	2.13×10^{-2}	1.21×10^{-2}	2.48×10^{-2}	4.61×10^{-2}	6.65×10^{-2}	8.59×10^{-2}	1.05×10^{-1}	1.23×10^{-1}	1.41×10^{-1}	1.59×10^{-1}
	S.IR	4.38×10^{-1}	4.84×10^{-1}	5.38×10^{-1}	5.88×10^{-1}	6.35×10^{-1}	6.79×10^{-1}	7.21×10^{-1}	7.60×10^{-1}	7.97×10^{-1}	8.30×10^{-1}	8.59×10^{-1}
E-II	Min	5.20×10^{-7}	7.23×10^{-7}	4.89×10^{-6}	1.14×10^{-5}	1.59×10^{-5}	1.68×10^{-5}	1.60×10^{-5}	1.72×10^{-5}	2.20×10^{-5}	2.71×10^{-5}	2.77×10^{-5}
	Mean	6.72×10^{-2}	1.31×10^{-1}	2.03×10^{-1}	2.76×10^{-1}	3.50×10^{-1}	4.22×10^{-1}	4.91×10^{-1}	5.57×10^{-1}	6.18×10^{-1}	6.74×10^{-1}	7.23×10^{-1}
	SD	3.19×10^{-2}	5.19×10^{-2}	7.16×10^{-2}	9.52×10^{-2}	1.19×10^{-1}	1.43×10^{-1}	1.66×10^{-1}	1.89×10^{-1}	2.10×10^{-1}	2.29×10^{-1}	2.46×10^{-1}
	Med	7.70×10^{-2}	1.57×10^{-1}	2.33×10^{-1}	3.6×10^{-1}	3.89×10^{-1}	4.79×10^{-1}	5.65×10^{-1}	6.44×10^{-1}	7.17×10^{-1}	7.83×10^{-1}	8.41×10^{-1}
	S.IR	1.55×10^{-2}	2.86×10^{-2}	1.99×10^{-2}	1.31×10^{-2}	1.46×10^{-2}	2.09×10^{-2}	2.71×10^{-2}	3.30×10^{-2}	3.83×10^{-2}	4.31×10^{-2}	4.70×10^{-2}
E-III	Min	1.98×10^{-5}	8.21×10^{-6}	8.50×10^{-6}	5.50×10^{-6}	7.92×10^{-6}	2.12×10^{-5}	1.52×10^{-5}	1.19×10^{-5}	2.95×10^{-5}	5.74×10^{-5}	5.59×10^{-5}
	Mean	1.25×10^{-1}	1.08×10^{-1}	9.21×10^{-1}	7.77×10^{-1}	6.43×10^{-1}	5.19×10^{-1}	4.03×10^{-1}	2.95×10^{-1}	1.90×10^{-1}	1.28×10^{-1}	1.50×10^{-1}
	SD	9.57×10^{-1}	8.20×10^{-1}	6.93×10^{-1}	5.72×10^{-1}	4.60×10^{-1}	3.60×10^{-1}	2.75×10^{-1}	2.15×10^{-1}	1.98×10^{-1}	2.18×10^{-1}	1.06×10^{-1}
	Med	1.91×10^{-1}	1.65×10^{-1}	1.40×10^{-1}	1.17×10^{-1}	9.51×10^{-1}	7.42×10^{-1}	5.37×10^{-1}	3.29×10^{-1}	1.37×10^{-1}	8.53×10^{-2}	1.65×10^{-1}
	S.IR	9.91×10^{-1}	8.47×10^{-1}	7.10×10^{-1}	5.77×10^{-1}	4.52×10^{-1}	3.39×10^{-1}	2.31×10^{-1}	1.34×10^{-1}	5.59×10^{-2}	6.92×10^{-2}	7.89×10^{-2}

Table 2. The performances based MWNN-GAAPAS for each example of the PDM using 10 neurons.

	Mode	$\hat{y}(t)$										
		0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
E-I	Min	1.20×10^{-9}	4.45×10^{-9}	3.18×10^{-8}	6.57×10^{-8}	8.26×10^{-8}	7.97×10^{-8}	7.48×10^{-8}	8.81×10^{-8}	1.21×10^{-7}	1.48×10^{-7}	1.47×10^{-7}
	Mean	4.26×10^{-1}	4.78×10^{-1}	5.34×10^{-1}	5.88×10^{-1}	6.39×10^{-1}	6.88×10^{-1}	7.33×10^{-1}	7.75×10^{-1}	8.14×10^{-1}	8.48×10^{-1}	8.79×10^{-1}
	SD	4.46×10^{-1}	4.97×10^{-1}	5.50×10^{-1}	6.02×10^{-1}	6.51×10^{-1}	6.98×10^{-1}	7.42×10^{-1}	7.83×10^{-1}	8.20×10^{-1}	8.54×10^{-1}	8.83×10^{-1}
	Med	2.54×10^{-1}	2.99×10^{-1}	3.68×10^{-1}	4.31×10^{-1}	4.80×10^{-1}	5.27×10^{-1}	5.71×10^{-1}	6.13×10^{-1}	6.53×10^{-1}	6.89×10^{-1}	7.22×10^{-1}
	S.IR	4.66×10^{-1}	5.23×10^{-1}	5.79×10^{-1}	6.33×10^{-1}	6.82×10^{-1}	7.27×10^{-1}	7.68×10^{-1}	8.06×10^{-1}	8.40×10^{-1}	8.77×10^{-1}	9.05×10^{-1}
E-II	Min	1.15×10^{-8}	2.04×10^{-8}	6.16×10^{-8}	1.79×10^{-7}	2.65×10^{-7}	2.88×10^{-7}	2.64×10^{-7}	2.38×10^{-7}	2.55×10^{-7}	3.30×10^{-7}	4.24×10^{-7}
	Mean	4.16×10^{-8}	8.85×10^{-8}	1.47×10^{-1}	2.06×10^{-1}	2.63×10^{-1}	3.19×10^{-1}	3.72×10^{-1}	4.23×10^{-1}	4.70×10^{-1}	5.12×10^{-1}	5.48×10^{-1}
	SD	6.10×10^{-8}	7.67×10^{-8}	1.03×10^{-1}	1.36×10^{-1}	1.73×10^{-1}	2.10×10^{-1}	2.46×10^{-1}	2.79×10^{-1}	3.11×10^{-1}	3.40×10^{-1}	3.64×10^{-1}
	Med	2.32×10^{-3}	7.26×10^{-8}	1.73×10^{-1}	2.69×10^{-1}	3.58×10^{-1}	4.48×10^{-1}	5.15×10^{-1}	5.86×10^{-1}	6.52×10^{-1}	7.12×10^{-1}	7.61×10^{-1}
	S.IR	3.70×10^{-8}	7.45×10^{-8}	1.11×10^{-1}	1.50×10^{-1}	1.93×10^{-1}	2.36×10^{-1}	2.79×10^{-1}	3.18×10^{-1}	3.53×10^{-1}	3.84×10^{-1}	4.08×10^{-1}
E-III	Min	5.56×10^{-9}	1.91×10^{-8}	4.54×10^{-9}	2.45×10^{-8}	1.44×10^{-8}	2.28×10^{-9}	9.65×10^{-9}	1.85×10^{-8}	3.85×10^{-9}	6.66×10^{-9}	3.87×10^{-9}
	Mean	2.30×10^{-1}	2.50×10^{-1}	2.13×10^{-1}	1.80×10^{-1}	1.50×10^{-1}	1.24×10^{-1}	9.84×10^{-8}	7.42×10^{-8}	5.08×10^{-8}	2.80×10^{-8}	3.14×10^{-8}
	SD	5.50×10^{-1}	5.48×10^{-1}	4.69×10^{-1}	3.97×10^{-1}	3.28×10^{-1}	2.63×10^{-1}	2.00×10^{-1}	1.40×10^{-1}	8.43×10^{-8}	4.14×10^{-8}	4.69×10^{-8}
	Med	7.86×10^{-4}	7.97×10^{-4}	4.52×10^{-4}	6.83×10^{-4}	1.40×10^{-3}	2.06×10^{-3}	3.38×10^{-3}	4.62×10^{-3}	5.79×10^{-3}	7.18×10^{-3}	8.47×10^{-3}
	S.IR	9.30×10^{-3}	6.24×10^{-3}	3.54×10^{-3}	4.98×10^{-3}	6.69×10^{-3}	1.19×10^{-8}	1.87×10^{-8}	2.58×10^{-8}	3.32×10^{-8}	2.27×10^{-8}	2.45×10^{-8}

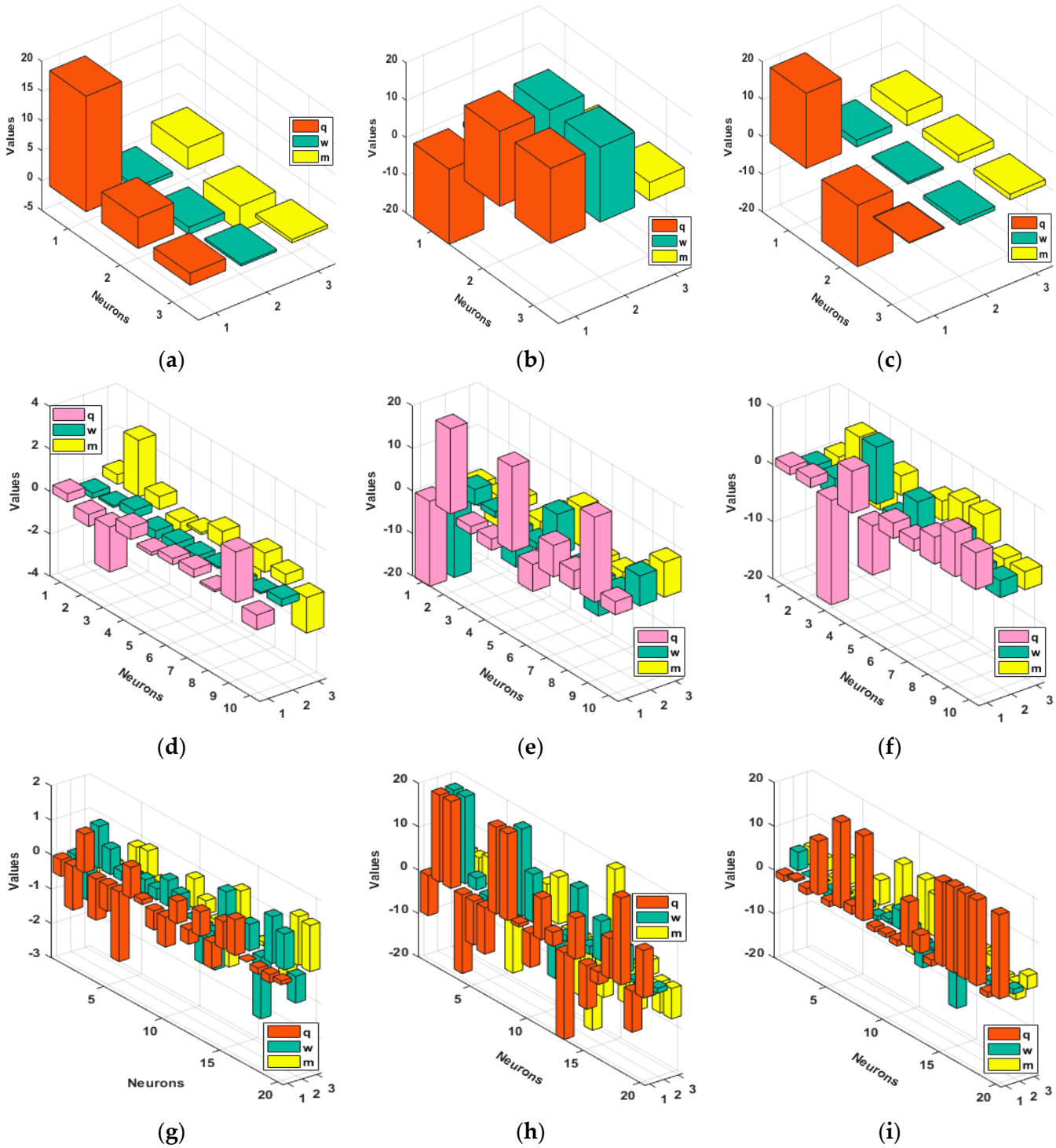


Figure 2. Best weights of MWNN-GAIPAS for each example of the PDM using 3, 10, and 20 neurons. (**a**) Optimal weights for Example 1 (3 neurons). (**b**) Optimal weights for Example 2 (3 neurons). (**c**) Optimal weights for Example 3 (3 neurons). (**d**) Best weights for Example 1 for 10 neurons. (**e**) Best weights for Example 2 for 10 neurons. (**f**) Best weights for Example 3 for 10 neurons. (**g**) Best weights for Example 1 for 20 neurons. (**h**) Best weights for Example 2 for 20 neurons. (**i**) Best weights for Example 3 for 20 neurons.

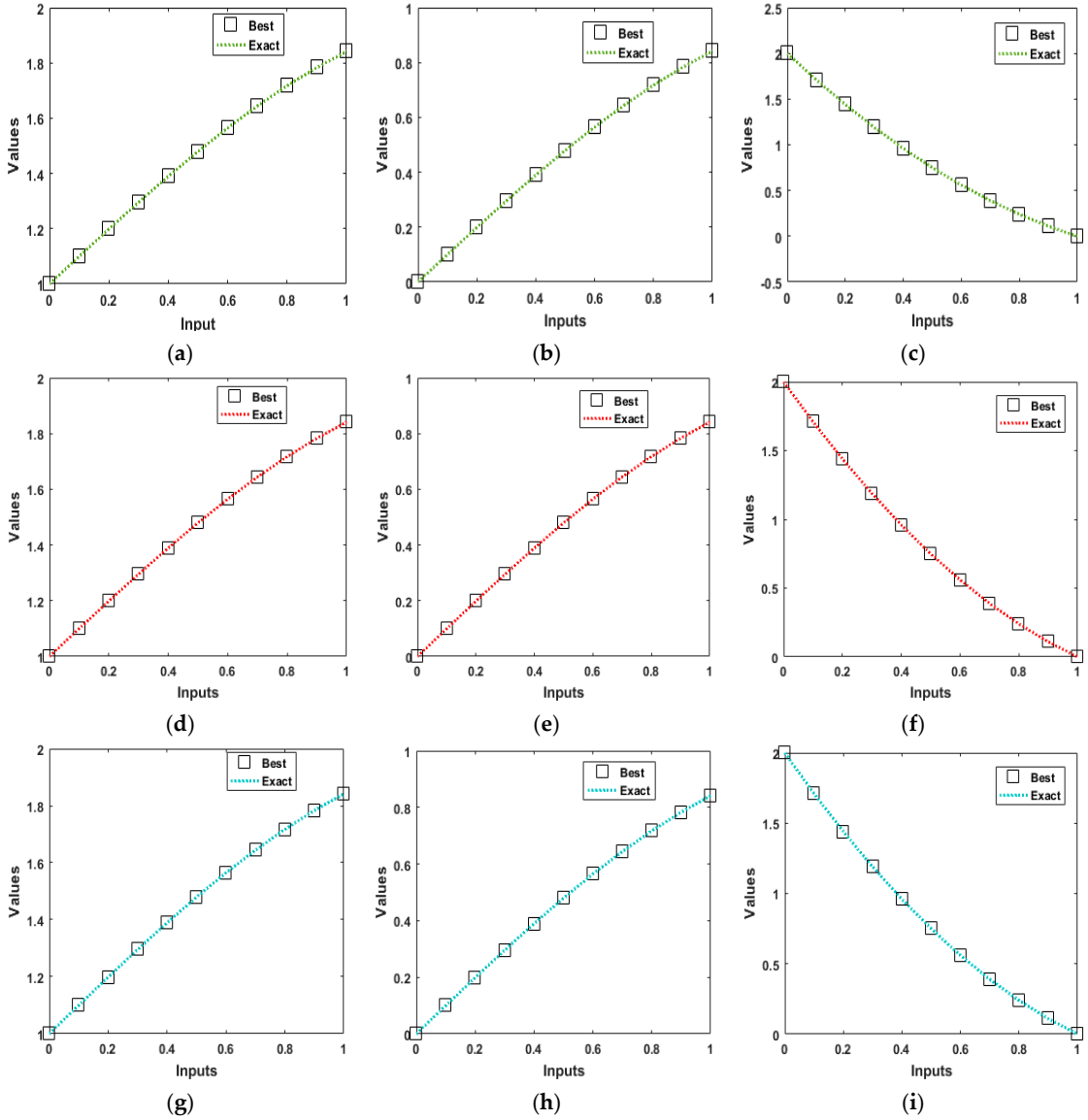
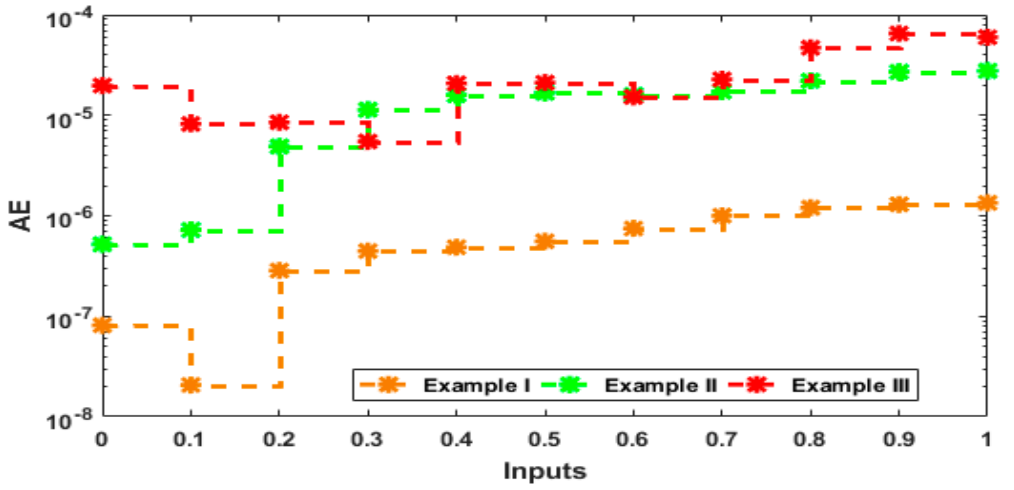


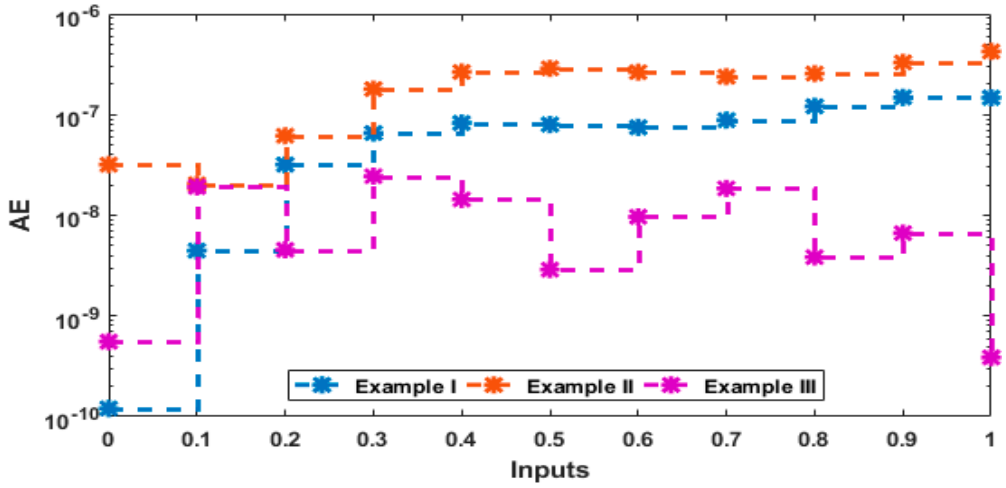
Figure 3. Comparison of the best and exact solutions based on MWNN-GAIPAS for solving each example of the PDM using 3, 10, and 20 neurons. (**a**) Result of Example 1 (3 neurons). (**b**) Result of Example 2 (3 neurons). (**c**) Result of Example 3 (3 neurons). (**d**) Result of Example 1 (10 neurons). (**e**) Result of Example 2 (10 neurons). (**f**) Result of Example 3 (10 neurons). (**g**) Result of Example 1 (20 neurons). (**h**) Result of Example 2 (20 neurons). (**i**) Result of Example 3 (20 neurons).

Table 3. The performances based MWNN-GAAPAS for each example of the PDM using 20 neurons.

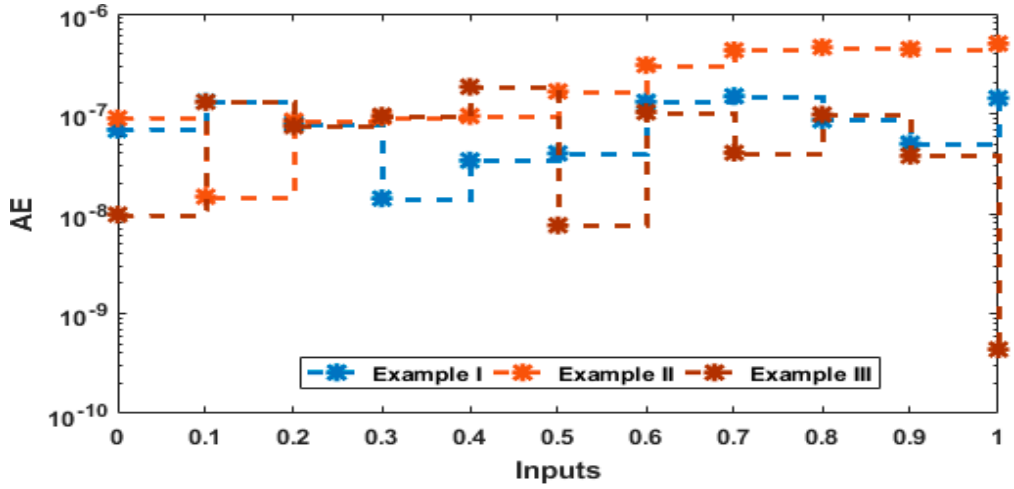
	Mode	$\hat{y}(t)$										
		0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
E-I	Min	4.22×10^{-9}	1.67×10^{-8}	7.93×10^{-8}	1.42×10^{-8}	3.39×10^{-8}	4.01×10^{-8}	1.32×10^{-7}	1.50×10^{-7}	8.64×10^{-8}	5.00×10^{-8}	1.43×10^{-7}
	Mean	3.47×10^{-1}	3.89×10^{-1}	4.43×10^{-1}	4.94×10^{-1}	5.44×10^{-1}	5.92×10^{-1}	6.37×10^{-1}	6.79×10^{-1}	7.19×10^{-1}	7.55×10^{-1}	7.87×10^{-1}
	SD	4.24×10^{-1}	4.62×10^{-1}	5.07×10^{-1}	5.53×10^{-1}	5.97×10^{-1}	6.41×10^{-1}	6.83×10^{-1}	7.23×10^{-1}	7.61×10^{-1}	7.95×10^{-1}	8.25×10^{-1}
	Med	8.71×10^{-4}	1.89×10^{-2}	4.71×10^{-2}	7.45×10^{-2}	1.02×10^{-1}	1.28×10^{-1}	1.54×10^{-1}	1.79×10^{-1}	2.03×10^{-1}	2.26×10^{-1}	2.49×10^{-1}
	S.IR	4.23×10^{-1}	4.62×10^{-1}	5.14×10^{-1}	5.62×10^{-1}	6.04×10^{-1}	6.48×10^{-1}	6.92×10^{-1}	7.32×10^{-1}	7.69×10^{-1}	7.99×10^{-1}	8.23×10^{-1}
E-II	Min	2.31×10^{-9}	1.48×10^{-8}	8.52×10^{-8}	9.16×10^{-8}	9.46×10^{-8}	1.67×10^{-7}	3.06×10^{-7}	4.31×10^{-7}	4.63×10^{-7}	4.43×10^{-7}	5.02×10^{-7}
	Mean	1.15×10^{-2}	5.33×10^{-2}	1.04×10^{-1}	1.54×10^{-1}	2.03×10^{-1}	2.50×10^{-1}	2.95×10^{-1}	3.37×10^{-1}	3.75×10^{-1}	4.10×10^{-1}	4.41×10^{-1}
	SD	2.85×10^{-2}	5.47×10^{-2}	9.77×10^{-2}	1.43×10^{-1}	1.88×10^{-1}	2.31×10^{-1}	2.72×10^{-1}	3.11×10^{-1}	3.47×10^{-1}	3.79×10^{-1}	4.08×10^{-1}
	Med	4.78×10^{-5}	6.73×10^{-2}	1.50×10^{-1}	2.29×10^{-1}	3.16×10^{-1}	3.89×10^{-1}	4.54×10^{-1}	5.16×10^{-1}	5.72×10^{-1}	6.25×10^{-1}	6.78×10^{-1}
	S.IR	9.75×10^{-4}	4.99×10^{-2}	9.93×10^{-2}	1.48×10^{-1}	1.95×10^{-1}	2.40×10^{-1}	2.82×10^{-1}	3.22×10^{-1}	3.59×10^{-1}	3.92×10^{-1}	4.21×10^{-1}
E-III	Min	2.23×10^{-9}	6.07×10^{-8}	7.51×10^{-8}	9.56×10^{-8}	1.82×10^{-7}	7.64×10^{-9}	2.99×10^{-8}	4.07×10^{-8}	5.42×10^{-8}	1.76×10^{-9}	4.38×10^{-9}
	Mean	6.58×10^{-1}	5.99×10^{-1}	5.08×10^{-1}	4.24×10^{-1}	3.46×10^{-1}	2.74×10^{-1}	2.06×10^{-1}	1.42×10^{-1}	8.09×10^{-2}	3.31×10^{-2}	4.71×10^{-2}
	SD	8.23×10^{-1}	7.13×10^{-1}	6.04×10^{-1}	5.02×10^{-1}	4.06×10^{-1}	3.17×10^{-1}	2.35×10^{-1}	1.58×10^{-1}	8.82×10^{-2}	3.78×10^{-2}	7.41×10^{-2}
	Med	2.43×10^{-2}	5.77×10^{-2}	4.54×10^{-2}	3.42×10^{-2}	3.23×10^{-2}	3.22×10^{-2}	3.50×10^{-2}	3.96×10^{-2}	4.82×10^{-2}	1.77×10^{-2}	4.90×10^{-3}
	S.IR	7.97×10^{-1}	7.09×10^{-1}	6.07×10^{-1}	5.11×10^{-1}	4.21×10^{-1}	3.36×10^{-1}	2.47×10^{-1}	1.64×10^{-1}	7.70×10^{-2}	2.73×10^{-2}	2.96×10^{-2}



(**a**)



(**b**)



(**c**)

Figure 4. AE for solving each example of the PDM using 3, 10, and 20 neurons. (**a**) AE values of Examples 1 to 3 for three neurons. (**b**) AE values of Examples 1 to 3 for 10 neurons. (**c**) AE values of Examples 1 to 3 for 20 neurons.

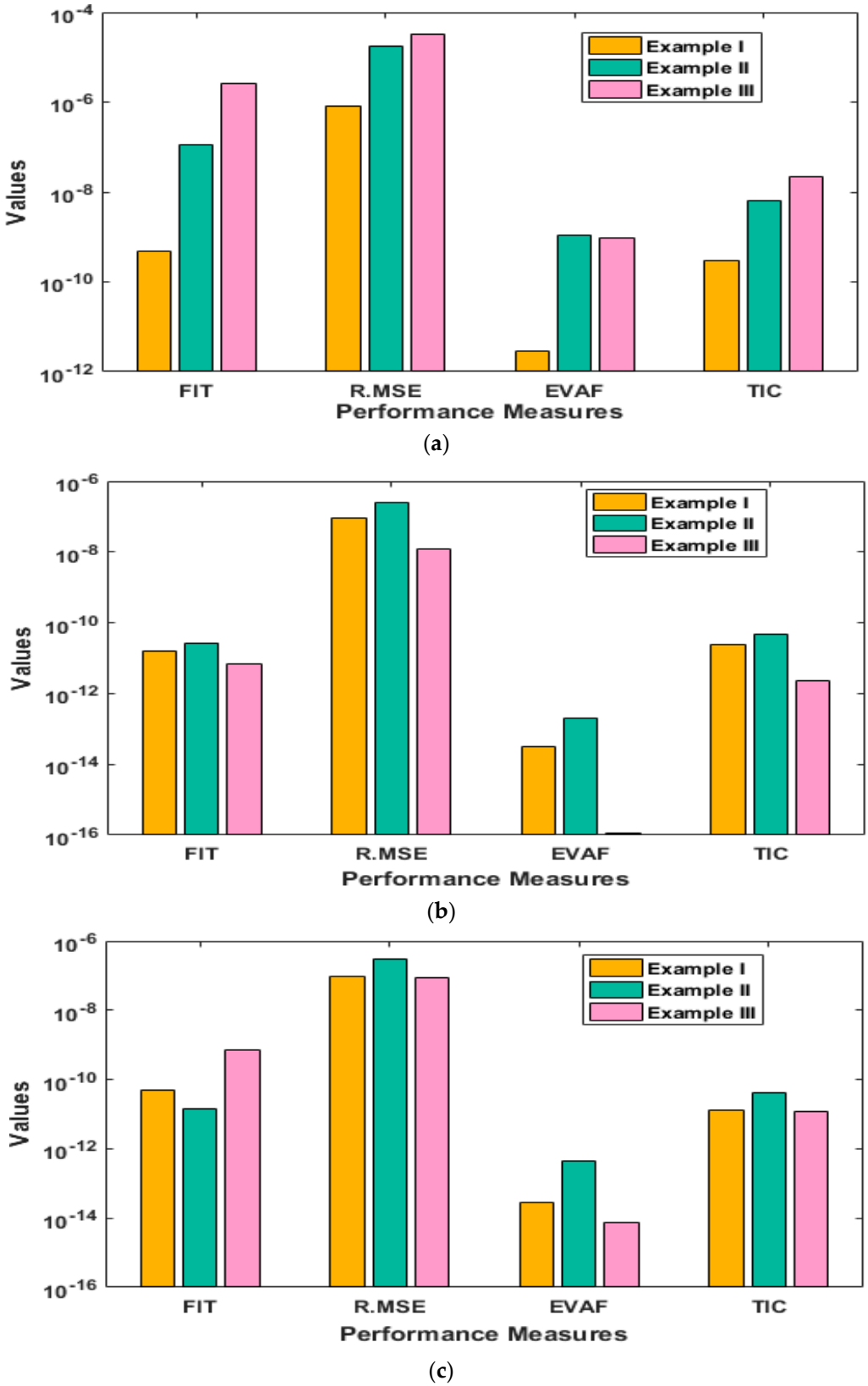
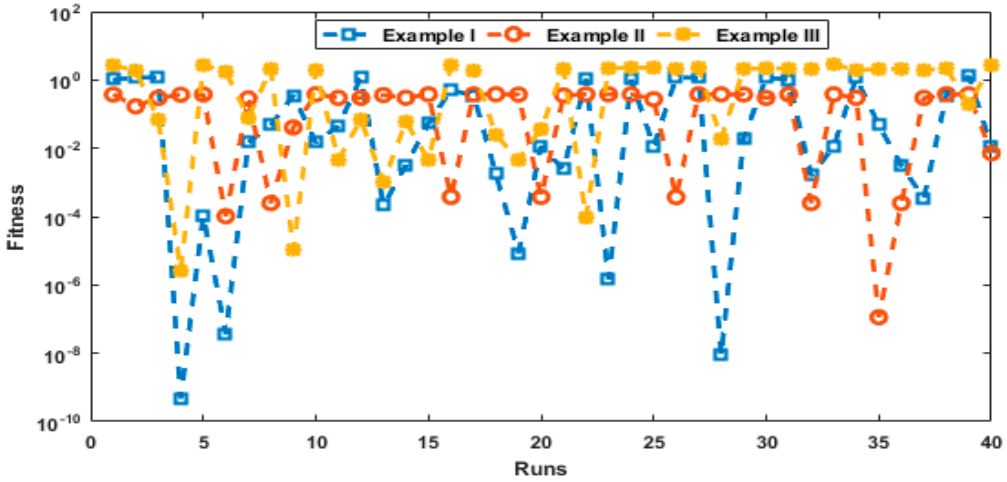
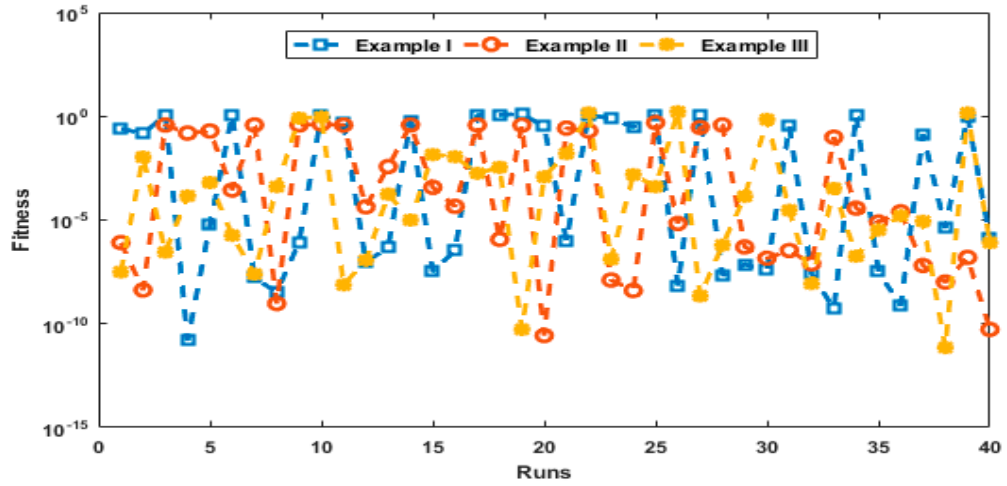


Figure 5. Performance values for solving each example of the PDM based on 3, 10, and 20 neurons. (**a**) Performance values for each example of the PDM for three neurons. (**b**) Performance values for each example of the PDM for 10 neurons. (**c**) Performance values for each example of the PDM for 20 neurons.



(**a**)



(**b**)

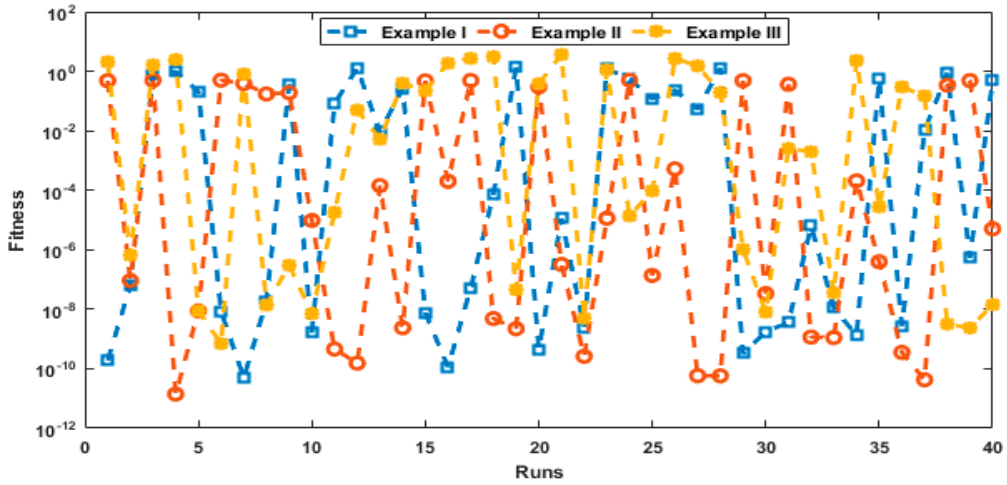
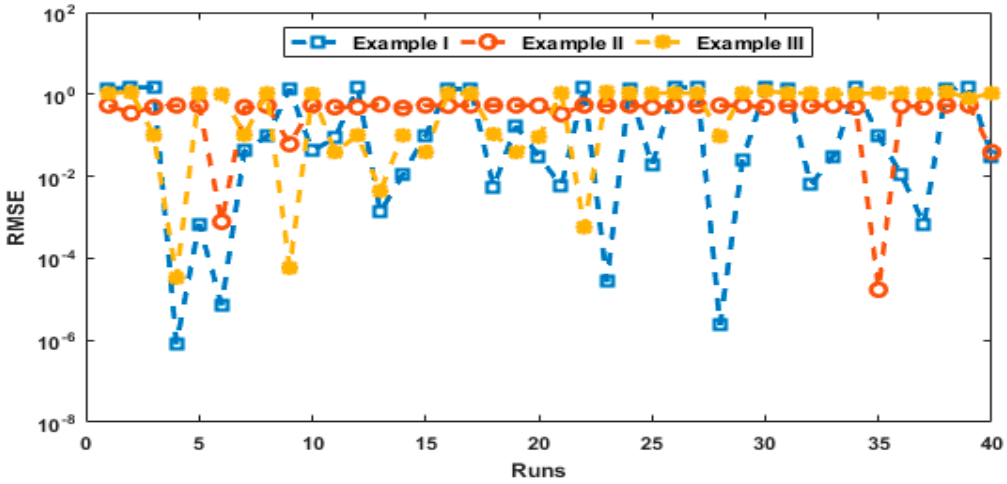
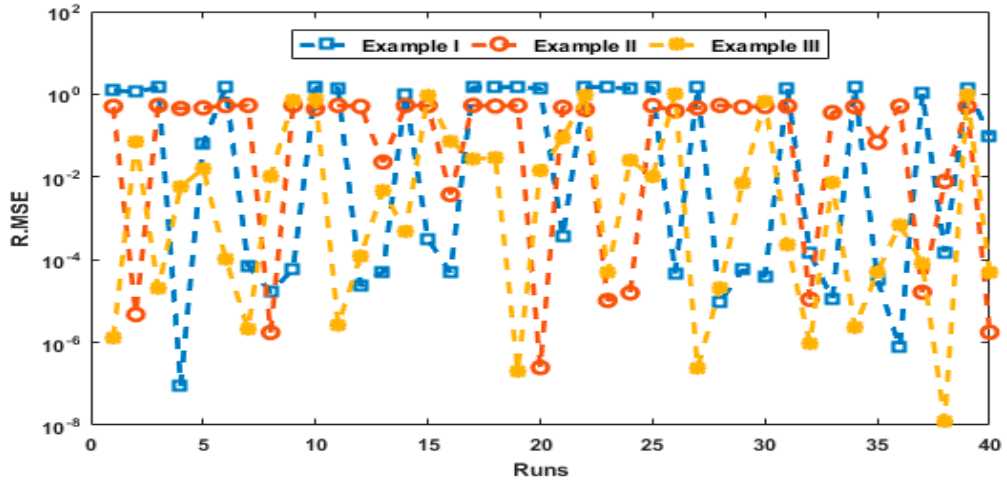


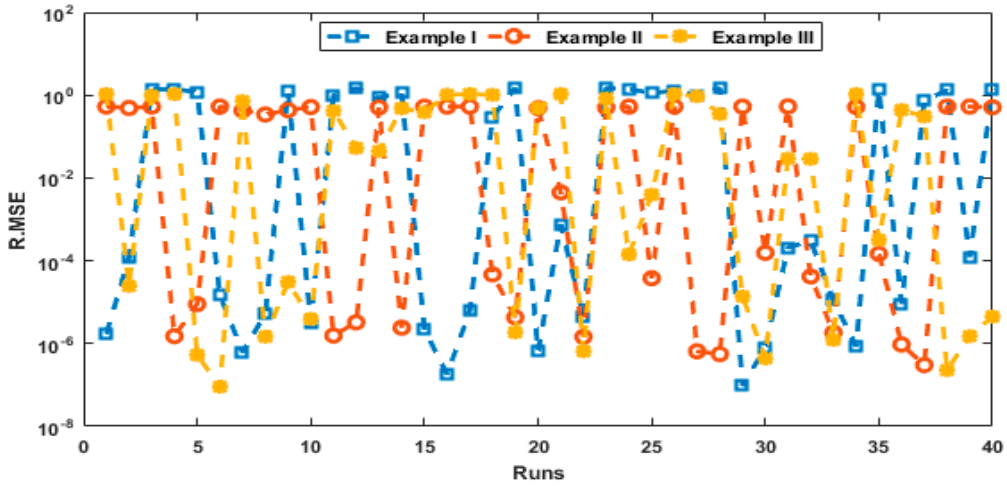
Figure 6. Statisticalinvestigations-based fitness for Examples1 to 3 by taking 3, 10, and 20 neurons. (**a**) Plots of Fitness convergence for each example of the PDM using three neurons. (**b**) Plots of Fitness convergence for each example of the PDM using 10 neurons. (**c**) Plots of Fitness convergence for each example of the PDM using 20 neurons.



(**a**)

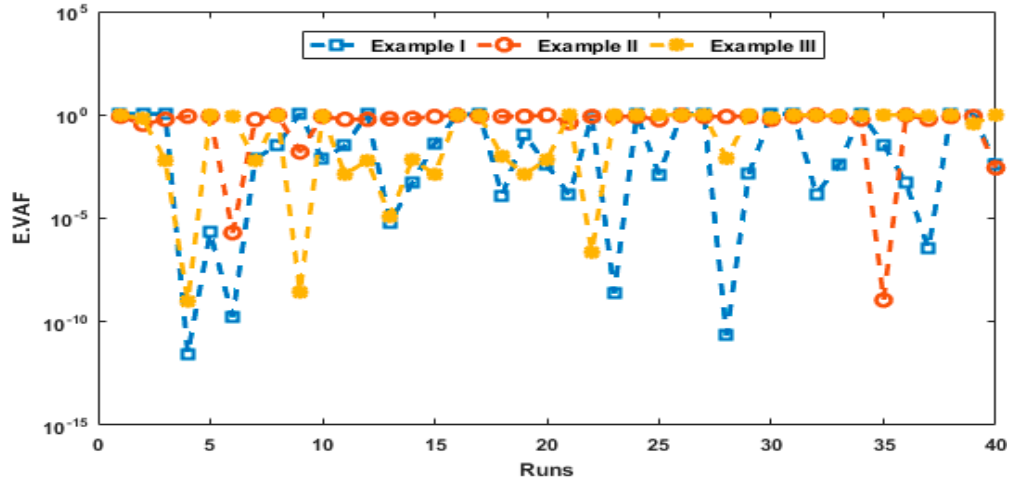


(**b**)

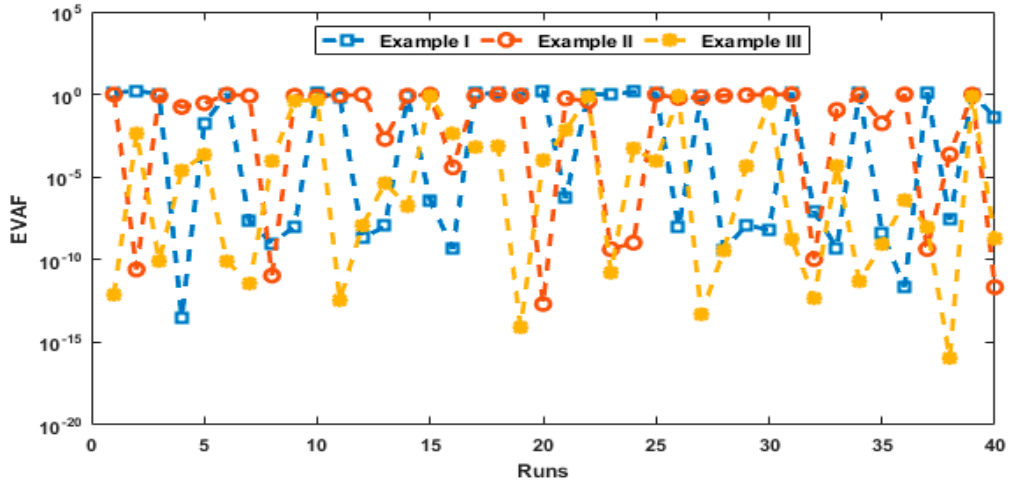


(**c**)

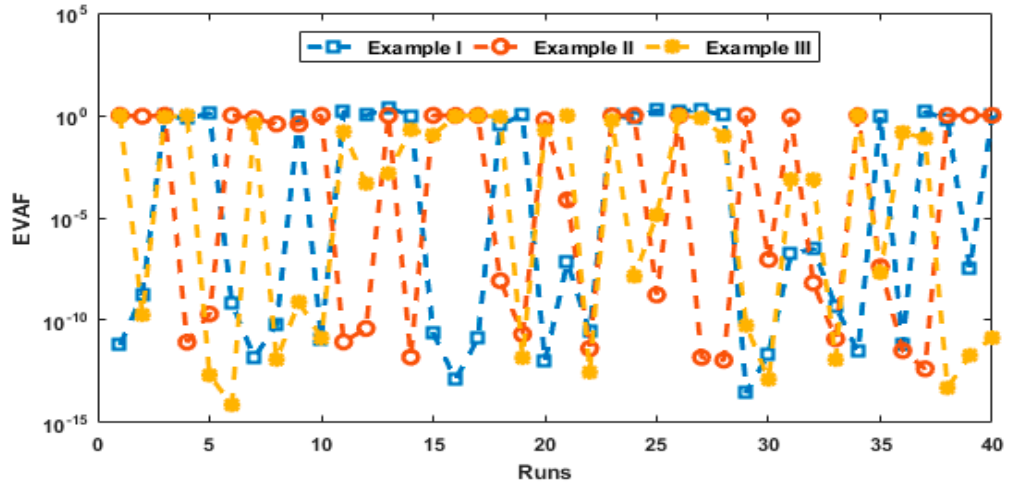
Figure 7. Statistical investigations of R.MSE for Examples1 to 3 by taking 3, 10, and 20 neurons. (**a**) Plots of R.MSE convergence for each example of the PDM using three neurons. (**b**) Plots of R.MSE convergence for each example of the PDM using 10 neurons. (**c**) Plots of R.MSE convergence for each example of the PDM using 20 neurons.



(**a**)



(**b**)



(**c**)

Figure 8. Statistical investigations of EVAF for each example by taking 3, 10, and 20 neurons. (**a**) Plots of EVAF convergence for three neurons. (**b**) Plots of EVAF convergence for 10 neurons. (**c**) Plots of EVAF convergence for 20 neurons.

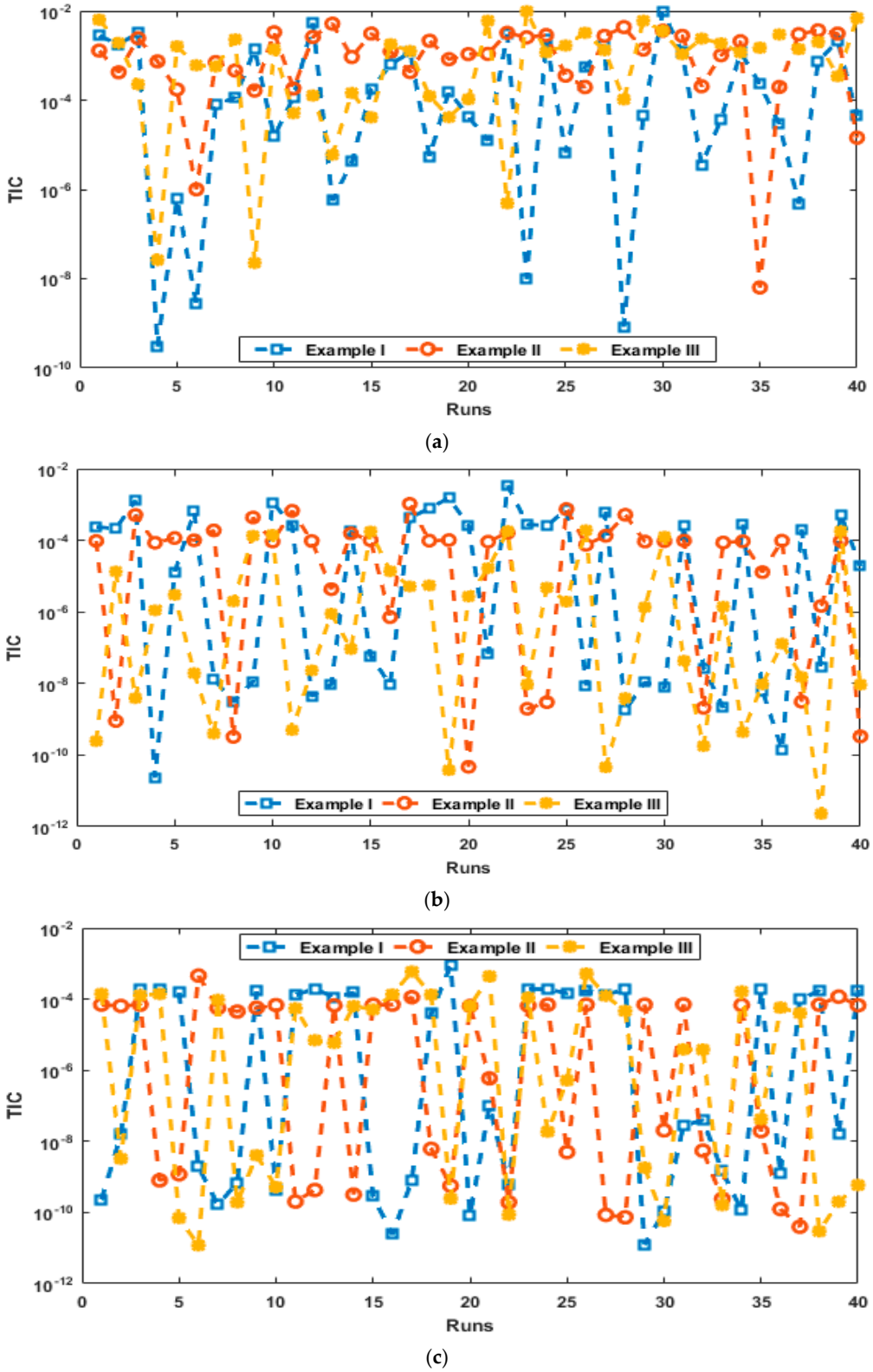


Figure 9. Statistical investigations-based TIC operator for Examples1 to 3 by taking 3, 10, and 20 neurons. (**a**) Plots of TIC convergence for each example of the PDM using three neurons. (**b**) Plots of TIC convergence for each example of the PDM using 10 neurons. (**c**) Plots of TIC convergence for each example of the PDM using 20 neurons.

5. Conclusions

The current investigations provide the design of a Morlet wavelet neural network using the hybridization process of global and local search schemes GA-IPAS to solve the prediction model. This model based on prediction is a kind of functional differential equation, which works as theopposite of the historical delay differential models. The analysis based on 3, 10, and 20 neurons is also presented to solve three different examples of the prediction differential model. The overlapping of the obtained results through the proposed methodology with the exact results shows the exactness of each example of the model based on3, 10, and 20 neurons. The AE for Examples 1 to 3 is found in good and negligible measures for 3, 10, and 20 neurons. It is observed that the proposed MWNN-GAIPAS has been implemented accurately, efficiently, and viably for different numbers of neurons to solve the model. Furthermore, the statistical soundings based on 40runs for solving the prediction system in terms of statistical Min, Med, standard deviation, mean, and SI. Range operators have also been performed. These operators authenticate the accuracy and trustworthiness of MWNN-GAIPAS, which is updated further by using the investigations based on the TIC, EVAF, and R.MSE operators to solve each example of the prediction differential system. It is also noticed that the small optimum values of these operators are further used to justify the accuracy and precision of MWNN-GAIPAS.

In upcoming studies, the MWNN-GAIPAS can be executed to solve biological systems, fluid dynamic nonlinear equations, and higher-order singular differential models [46–57].

Author Contributions: Conceptualization, A.A.; Software, Z.S. and M.A.A.; Validation, Z.S. and M.A.A.; Formal analysis, Z.S.; Investigation, A.A. and A.F.H.; Resources, A.F.H.; Data curation, A.F.H.; Writing—original draft, Z.S., A.A. and M.A.A.; Supervision, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-RP23088).

Data Availability Statement: Not Application.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Niculescu, S.I. *Delay Effects on Stability: A Robust Control Approach*; Springer Science & Business Media: Berlin, Germany, 2001; Volume 269.
2. Luo, D.; Tian, M.; Zhu, Q. Some results on finite-time stability of stochastic fractional-order delay differential equations. *Chaos Solitons Fractals* **2022**, *158*, 111996. [CrossRef]
3. Almarri, B.; Janaki, S.; Ganesan, V.; Ali, A.H.; Nonlaopon, K.; Bazighifan, O. Novel Oscillation Theorems and Symmetric Properties of Nonlinear Delay Differential Equations of Fourth-Order with a Middle Term. *Symmetry* **2022**, *14*, 585. [CrossRef]
4. Sabermahani, S.; Ordokhani, Y. General Lagrange-hybrid functions and numerical solution of differential equations containing piecewise constant delays with bibliometric analysis. *Appl. Math. Comput.* **2020**, *395*, 125847. [CrossRef]
5. Rihan, F.A.; Alsakaji, H.J. Stochastic delay differential equations of three-species prey-predator system with cooperation among prey species. *Discret. Contin. Dyn. Syst. S* **2022**, *15*, 245–263. [CrossRef]
6. Bildik, N.; Deniz, S. A new efficient method for solving delay differential equations and a comparison with other methods. *Eur. Phys. J. Plus* **2017**, *132*, 51. [CrossRef]
7. Rahimkhani, P.; Ordokhani, Y.; Babolian, E. A new operational matrix based on Bernoulli wavelets for solving fractional delay differential equations. *Numer. Algorithms* **2016**, *74*, 223–245. [CrossRef]
8. Sabir, Z.; Günerhan, H.; Guirao, J.L.G. On a New Model Based on Third-Order Nonlinear Multisingular Functional Differential Equations. *Math. Probl. Eng.* **2020**, *2020*, 1683961. [CrossRef]
9. Aziz, I.; Amin, R. Numerical solution of a class of delay differential and delay partial differential equations via Haar wavelet. *Appl. Math. Model.* **2016**, *40*, 10286–10299. [CrossRef]
10. Frazier, M.W. Background: Complex Numbers and Linear Algebra. In *An Introduction to Wavelets through Linear Algebra*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 7–100.
11. Tomasiello, S. An alternative use of fuzzy transform with application to a class of delay differential equations. *Int. J. Comput. Math.* **2016**, *94*, 1719–1726. [CrossRef]
12. Vaid, M.K.; Arora, G. Solution of Second Order Singular Perturbed Delay Differential Equation Using Trigonometric B-Spline. *Int. J. Math. Eng. Manag. Sci.* **2019**, *4*, 349–360. [CrossRef]

13. Hashemi, M.S.; Atangana, A.; Hajikhah, S. Solving fractional pantograph delay equations by an effective compu-tational method. *Math. Comput. Simul.* **2020**, *177*, 295–305. [CrossRef]
14. Adel, W.; Sabir, Z. Solving a new design of nonlinear second-order Lane–Emden pantograph delay differential model via Bernoulli collocation method. *Eur. Phys. J. Plus* **2020**, *135*, 427. [CrossRef]
15. Erdogan, F.; Sakar, M.G.; Saldır, O. A finite difference method on layer-adapted mesh for singularly perturbed delay differential equations. *Appl. Math. Nonlinear Sci.* **2020**, *5*, 425–436. [CrossRef]
16. Seong, H.Y.; Majid, Z.A. Solving second order delay differential equations using direct two-point block method. *Ain Shams Eng. J.* **2017**, *8*, 59–66. [CrossRef]
17. Sabir, Z.; Guirao, J.L.G.; Saeed, T.; Erdoğan, F. Design of a Novel Second-Order Prediction Differential Model Solved by Using Adams and Explicit Runge–Kutta Numerical Methods. *Math. Probl. Eng.* **2020**, *2020*, 9704968. [CrossRef]
18. Sabir, Z.; Raja, M.A.Z.; Wahab, H.A.; Shoaib, M.; Aguilar, J.G. Integrated neuro-evolution heuristic with sequential quadratic programming for second-order predic-tion differential models. *Numer. Methods Partial. Differ. Equ.* **2020**.
19. Raja, M.A.Z.; Shoaib, M.; Hussain, S.; Nisar, K.S.; Islam, S. Computational intelligence of Levenberg-Marquardt backpropagation neural networks to study thermal radiation and Hall effects on boundary layer flow past a stretching sheet. *Int. Commun. Heat Mass Transf.* **2021**, *130*, 105799. [CrossRef]
20. Tian, Y.; Chen, H.; Ma, H.; Zhang, X.; Tan, K.C.; Jin, Y. Integrating conjugate gradients into evolutionary algo-rithms for large-scale continuous multi-objective optimization. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 1801–1817. [CrossRef]
21. Sabir, Z.; Amin, F.; Pohl, D.; Guirao, J.L. Intelligence computing approach for solving second order system of Emden–Fowler model. *J. Intell. Fuzzy Syst.* **2020**, *38*, 7391–7406. [CrossRef]
22. Awais, M.; Rehman, H.; Raja, M.A.Z.; Awan, S.E.; Ali, A.; Shoaib, M.; Malik, M.Y. Hall effect on MHD Jeffrey fluid flow with Cattaneo–Christov heat flux model: An application of stochastic neural computing. *Complex Intell. Syst.* **2022**, *8*, 5177–5201. [CrossRef]
23. Sabir, Z.; Wahab, H.A.; Umar, M.; Erdoğan, F. Stochastic numerical approach for solving second order nonlinear singular functional differential equation. *Appl. Math. Comput.* **2019**, *363*, 124605. [CrossRef]
24. Wahid, M.A.; Bukhari, S.H.R.; Daud, A.; Awan, S.E.; Raja, M.A.Z. COVICT: An IoT based architecture for COVID-19 detection and contact tracing. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *14*, 7381–7398. [CrossRef] [PubMed]
25. Chaudhary, N.I.; Raja, M.A.Z.; Khan, Z.A.; Mehmood, A.; Shah, S.M. Design of fractional hierarchical gradient de-scent algorithm for parameter estimation of nonlinear control autoregressive systems. *Chaos Solitons Fractals* **2022**, *157*, 111913. [CrossRef]
26. Sabir, Z.; Manzar, M.A.; Raja, M.A.Z.; Sheraz, M.; Wazwaz, A.M. Neuro-heuristics for nonlinear singular Thomas-Fermi systems. *Appl. Soft Comput.* **2018**, *65*, 152–169. [CrossRef]
27. Shoaib, M.; Kausar, M.; Nisar, K.S.; Raja, M.A.Z.; Zeb, M.; Morsy, A. The design of intelligent networks for entropy generation in Ree-Eyring dissipative fluid flow system along quartic autocatalysis chemical reactions. *Int. Commun. Heat Mass Transf.* **2022**, *133*, 105971. [CrossRef]
28. Raja, M.A.Z.; Sabir, Z.; Mehmood, N.; Al-Aidarous, E.S.; Khan, J.A. Design of stochastic solvers based on genetic algorithms for solving nonlinear equations. *Neural Comput. Appl.* **2014**, *26*, 1–23. [CrossRef]
29. Sabir, Z.; Baleanu, D.; Raja, M.A.Z.; Guirao, J.L.G. Design of Neuro-Swarming Heuristic Solver for Multi-Pantograph Singular Delay Differential Equation. *Fractals* **2021**, *29*, 2140022. [CrossRef]
30. Umar, M.; Amin, F.; Ali, M.R. Neuro-swarm intelligence to study mosquito dispersal system in a heterogeneous atmosphere. *Evol. Syst.* 2023. [CrossRef]
31. Umar, M.; Amin, F.; Wahab, H.A.; Baleanu, D. Unsupervised constrained neural network modeling of boundary value corneal model for eye surgery. *Appl. Soft Comput.* **2019**, *85*, 105826. [CrossRef]
32. Umar, M.; Amin, F.; Al-Mdallal, Q.; Ali, M.R. A stochastic computing procedure to solve the dynamics of preven-tion in HIV system. *Biomed. Signal Process. Control.* **2022**, *78*, 103888. [CrossRef]
33. Łuczak, D. Mechanical vibrations analysis in direct drive using CWT with complex Morlet wavelet. *Power Electron. Drives* **2023**, *8*, 65–73. [CrossRef]
34. Reddy, G.T.; Reddy, M.P.K.; Lakshmanna, K.; Rajput, D.S.; Kaluri, R.; Srivastava, G. Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis. *Evol. Intell.* **2020**, *13*, 185–196. [CrossRef]
35. Mayer, M.J.; Szilágyi, A.; Gróf, G. Environmental and economic multi-objective optimization of a household level hybrid renewable energy system by genetic algorithm. *Appl. Energy* **2020**, *269*, 115058. [CrossRef]
36. Zou, D.; Li, S.; Kong, X.; Ouyang, H.; Li, Z. Solving the combined heat and power economic dispatch problems by an improved genetic algorithm and a new constraint handling strategy. *Appl. Energy* **2019**, *237*, 646–670. [CrossRef]
37. Song, Y.; Wei, L.; Yang, Q.; Wu, J.; Xing, L.; Chen, Y. RL-GA: A Reinforcement Learning-based Genetic Algorithm for Electromag-netic Detection Satellite Scheduling Problem. *Swarm Evol. Comput.* **2023**, *77*, 101236. [CrossRef]
38. Jahed Armaghani, D.; Hasanipanah, M.; Mahdiyar, A.; Abd Majid, M.Z.; Bakhshandeh Amnieh, H.; Tahir, M.M.D. Airblast prediction through a hybrid genetic algorithm-ANN model. *Neural Comput. Appl.* **2018**, *29*, 619–629. [CrossRef]
39. Sohail, A. Genetic Algorithms in the Fields of Artificial Intelligence and Data Sciences. *Ann. Data Sci.* **2021**, *10*, 1007–1018. [CrossRef]
40. Aziz, R.M.; Mahto, R.; Goel, K.; Das, A.; Kumar, P.; Saxena, A. Modified Genetic Algorithm with Deep Learning for Fraud Transactions of Ethereum Smart Contract. *Appl. Sci.* **2023**, *13*, 697. [CrossRef]

41. Jiang, Y.; Wu, P.; Zeng, J.; Zhang, Y.; Zhang, Y.; Wang, S. Multi-parameter and multi-objective optimisation of articulated monorail vehicle system dynamics using genetic algorithm. *Veh. Syst. Dyn.* **2019**, *58*, 74–91. [CrossRef]
42. Bertocchi, C.; Chouzenoux, E.; Corbineau, M.-C.; Pesquet, J.-C.; Prato, M. Deep unfolding of a proximal interior point method for image restoration. *Inverse Probl.* **2020**, *36*, 034005. [CrossRef]
43. Wright, S.E.; Lim, S. Solving nested-constraint resource allocation problems with an interior point method. *Oper. Res. Lett.* **2020**, *48*, 297–303. [CrossRef]
44. Pesteh, S.; Moayyed, H.; Miranda, V. Favorable properties of Interior Point Method and Generalized Correntropy in power system State Estimation. *Electr. Power Syst. Res.* **2020**, *178*, 106035. [CrossRef]
45. Garreis, S.; Surowiec, T.M.; Ulbrich, M. An interior-point approach for solving risk-averse PDE-constrained opti-mization problems with coherent risk measures. *SIAM J. Optim.* **2021**, *31*, 1–29. [CrossRef]
46. Asadi, S.; Darvay, Z.; Lesaja, G.; Mahdavi-Amiri, N.; Potra, F. A Full-Newton Step Interior-Point Method for Mon-otone Weighted Linear Complementarity Problems. *J. Optim. Theory Appl.* **2020**, *186*, 864–878. [CrossRef]
47. El Hayek, P.; Boueri, M.; Nasr, L.; Aoun, C.; Sayad, E.; Jallad, K. Cholera Infection Risks and Cholera Vaccine Safety in Pregnan-cy. *Infect. Dis. Obstet. Gynecol.* **2023**, *2023*, 4563797. [CrossRef] [PubMed]
48. Tian, M.; El Khoury, R.; Alshater, M.M. The nonlinear and negative tail dependence and risk spillovers between foreign ex-change and stock markets in emerging economies. *J. Int. Financ. Mark. Inst. Money* **2023**, *82*, 101712. [CrossRef]
49. Issa, J.S. A nonlinear absorber for the reflection of travelling waves in bars. *Nonlinear Dyn.* **2022**, *108*, 3279–3295. [CrossRef]
50. Kassis, M.T.; Tannir, D.; Toukhtarian, R.; Khazaka, R. Moments-based sensitivity analysis of x-parameters with re-spect to linear and nonlinear circuit components. In Proceedings of the 2019 IEEE 28th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS), Montreal, QC, Canada, 6–9 October 2019; pp. 1–3.
51. Younes, G.A.; El Khatib, N. Mathematical modeling of atherogenesis: Atheroprotective role of HDL. *J. Theor. Biol.* **2021**, *529*, 110855. [CrossRef]
52. Kanbar, F.; Null, R.T.; Klingenberg, C. Well-Balanced Central Scheme for the System of MHD Equations with Gravitational Source Term. *Commun. Comput. Phys.* **2022**, *32*, 878–898. [CrossRef]
53. Habre, S.S. Qualitative aspects of differential equations in an inquiry-oriented course. *Int. J. Math. Educ. Sci. Technol.* **2021**, *54*, 351–364. [CrossRef]
54. Touma, R.; Saleh, M. Well-balanced central schemes for pollutants transport in shallow water equations. *Math. Comput. Simul.* **2021**, *190*, 1275–1293. [CrossRef]
55. Younes, G.A.; El Khatib, N. Mathematical modeling of inflammatory processes of atherosclerosis. *Math. Model. Nat. Phenom.* **2022**, *17*, 5. [CrossRef]
56. Younes, Y.; Hallit, S.; Obeid, S. Premenstrual dysphoric disorder and childhood maltreatment, adulthood stressful life events and depression among Lebanese university students: A structural equation modeling approach. *BMC Psychiatry* **2021**, *21*, 548. [CrossRef]
57. Habre, S. Inquiry-oriented differential equations: A guided journey of learning. *Teach. Math. Its Appl. Int. J. IMA* **2020**, *39*, 201–212. [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.