AN INNOVATIVE NEUROEVOLUTIONARY APPROACH FOR HEART BEAT ANALYSIS WITH ANN

By

Sabir Ali

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

Aug, 2023

AN INNOVATIVE NEUROEVOLUTIONARY APPROACH FOR HEART BEAT ANALYSIS WITH ANN

By

Sabir Ali

MS-Math, National University of Modern Languages, Islamabad, 2023

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

In **Mathematics**

To

FACULTY OF ENGINEERING & COMPUTER SCIENCE

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

THESIS AND DEFENSE APPROVAL FORM

The undersigned certify that they have read the following thesis, examined the defense, are satisfied with overall exam performance, and recommend the thesis to the Faculty of Engineering and Computer Sciences for acceptance.

Thesis Title: An innovative neuroevolutionary approach for heart beat analysis with ANN

Submitted By: Sabir Ali **Registration #:**22 MS/MATH/S21

Master of Science in Mathematics

Title of the Degree

Mathematics

Name of Discipline

Dr. Shahzad Khattak

Name of Research Supervisor

Signature of Research Supervisor

Dr. Sadia Riaz

Name of HOD (MATH)

Signature of HOD (MATH)

Dr. Mohammad Noman Malik

Name of Dean (FECS)

Signature of Dean (FE&CS)

Aug, 2023

AUTHOR'S DECLARATION

I Sabir Ali

Son of Sher Zamin Khan

Discipline Mathematics

Candidate of Master of Science in Mathematics at the NUML do hereby declare that the thesis An innovative neuroevolutionary approach for heart beat analysis with ANN submitted by me in partial fulfillment of MSMA degree, is my original work and has not been submitted or published earlier. I also solemnly declare that it shall not, in the future, be submitted by me for obtaining any other degree from this or any other university or institution. I also understand that if evidence of plagiarism is found in my thesis/dissertation at any stage, even after the award of a degree, the work may be canceled and the degree revoked.

Signature of Candidate

Sabir Ali

Name of Candidate

04 Aug, 2023

Date

ABSTRACT

Title: An innovative neuroevolutionary approach for heart beat analysis with ANN

The prominence of artificial neural networks (ANNs) is rising in a variety of applications. Most mathematical models have a form of differential equations (DEs), and recent research work has demonstrated that neural networks (NN) can be used to solve differential equations (DEs). In this thesis, we present a new neuroevolutionary approach called hybrid fractional particle swarm optimization (FO-PSO) to solve a differential equation (DE). Here we find an approximate solution to a 2^{nd} order non-linear ordinary differential equation (ODE), known as the Van der Pol (VdP) heartbeat model (HBM), utilizing artificial neural networks (ANNs) with feed-forward (FF), and also examine the effectiveness of our technique and approach. Fractional order particle swarm optimization (FO-PSO) is a hybrid technique for the fractional order velocity of the particle swarm optimization algorithm. For this thesis, we considered two problems: One of the problems contains a forcing term, while the other does not. Each problem has two scenarios, and each scenario has four cases. These cases arise because of some variations in parameters. We make a comparison of our proposed hybrid FO-PSO–ASA technique's results with the hybrid genetic algorithm with the interior point technique's results. 100 independent runs have been performed. In terms of mean absolute deviation, root-mean-square error, and Nash–Sutcliffe efficiency, statistical analyses demonstrate its application, efficacy, and dependability.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
ANN	-	Artificial Neural Network
NN	-	Neural Network
FFNN	-	Feed Forward Neural Network
DE	-	Differential Equation
HBM	-	Heartbeat model
PSO	-	Particle swarm optimization
FO-PSO	-	Fractional order PSO
ASA	-	Active set algorithm
RBF	-	Radial Basis Function
VdP	-	Van der Pol
GA	-	Genetic Algorithm
IPA	-	Interior Point Algorithm
FF	-	Fitness Function
AE	-	Absolute Error
STD	-	Standard Deviation
AM	-	Athematic Mean
RMSE	-	Root Mean Square Error
GRMSE	-	Global Root Mean Square Error
MAD	-	Mean Absolute Error Deviation
GMAD	-	Global Mean Absolute Error Deviation
ENSE	-	Error in Nash–Sutcliffe Efficiency
GENSE	-	Global Error in Nash–Sutcliffe Efficiency

LIST OF SYMBOLS

x	-	Fiber of heart
$\widetilde{\alpha}$	-	Pulse shape modification factor of heartbeat
v_1 , v_2	-	Asymmetric components
e	-	Duration of ventricular contraction
d	-	It is a factor that appears when a cubic term is used to
	-	replace the harmonic force.
$f(t)$		External force term
ω	-	Dupping parameter
$\tilde{x}(t)$	-	Approximate solution
ϕ, w, β,	-	The real values, which is arbitrarily bounded (weights)
m	-	Number of neurons
$f(x)$	-	log-sigmoid activation function
ε	-	Mean square error
α	-	Fractional order
T	-	Time periods during which events occurred
r	-	Truncated terms
u_t	-	Current velocity of moving object
u_{t+1}	-	Updated velocity
X_t	-	Current position of moving object
X_{t+1}	-	updated position
b	-	Best position of moving particle
g	-	Global position
ϕ_1	-	Weights of local information
ϕ_2	-	Weights of global information
R_n	-	Total number of runs
G_p	-	Total number of input values
r	-	The objective value of the experiment

ACKNOWLEDGMENT

First of all, I wish to express my gratitude and deep appreciation to Almighty Allah, who made this study possible and successful. This study would not be accomplished unless the honest espousal was extended from several sources for which I would like to express my sincere thankfulness and gratitude. Yet, there were significant contributors to my attained success and I cannot forget their input, especially my research supervisor, Dr. Shahzad Khattak, who did not leave any stone unturned to guide me during my research journey. I really appreciate our HOD, Dr. Sadia Riaz providing us with a research environment and kind support. I really want to say thanks to our respected teachers, Dr. Muhammad Rizwan, Dr. Anum Naseem, Dr. Hadia Tariq, Dr. Aisha Bibi, and other teachers, for their guidance and support.

I shall also acknowledge the extended assistance from the administration of the Department of Mathematics, who supported me all through my research experience and simplified the challenges I faced. For all whom I did not mention but shall not neglect their significant contribution, thanks for everything.

DEDICATION

This thesis work is dedicated to my parents, family, and my teachers throughout my education career who have not only loved me unconditionally but whose good examples have taught me to work hard for the things that I aspire to achieve.

CHAPTER 1

INTRODUCTION

In chapter 1, some important topics related to our research area are listed. The topics, that help and guide us to understand the phenomena and basics of our research area. These topics include artificial intelligence, neural networks, types of neural networks, differential equations, their types, and different methods to solve differential equations.

1.1 Artificial intelligence (AI)

We are currently witnessing the phenomenal progress of artificial intelligence in science and society. Recent developments in research studies and real-life achievements are possible because of artificial intelligence (AI). However, establishing a strong mathematical basis is still in its infancy. The resurgence of machine learning's popularity around 1980 is marked by a number of significant developments. The rise of artificial intelligence also had a substantial impact on a variety of mathematical disciplines. Artificial neural networks (ANN) are a biological sub-field of AI.

1.2 Brain Neuron

The brain is made of cells called neurons. The neural network (NN) of the brain is made up of connections between these neurons. In the human brain, there are around 1011 neurons and about 10,000 connections between them. An imitation of a natural neural network is an ANN, where artificial neurons are interconnected similarly to the brain neural network.

A neuron consists of an axon, dendrites, and a cell body. The dendrite receives electrochemical signals from other neurons and transmits them to the cell body. The cell's soma contains the nucleus and other chemical components necessary for the cell's survival. Axons transmit neuronal signals to other neurons. A synapse is a connection between the dendrites of two neurons or between a neuron and a muscle cell [1], as shown in Fig. 1.1.

1.3 Artificial Neural Networks (ANNs)

An imitation of the human brain is known as an artificial neural network (ANN). Despite the significant advancements in machine learning techniques over the past few years, using these new methods to solve differential equations (DEs) has proven to be quite tricky sometimes. The most encouraging outcomes so far have either entailed learning enhanced solvers or directly learning the solution [2]. The notion of directly learning a DE's solution applies to all DEs, and because it is such a basic idea, it provides us with a good starting place.

Using feed-forward (FF) neural networks' ability to approximate functions, neural network approaches can solve both ordinary and partial differential equations and produce closed-form solutions. In this version, the basic approximation element is a feed-forward neural network (FF-NN), whose parameters are changed to minimize the relevant error function.

Any optimization technique may be used to train the NN, which needs the computation of the error gradient with respect to the network parameters. In this technique, a trial solution to a DE is expressed as the sum of two sections. The boundary or initial conditions are satisfied by the first section, which contains no changeable parameters. The second section is designed to have no effect on the boundary or initial conditions and consists of an FF-NN with customizable parameters.

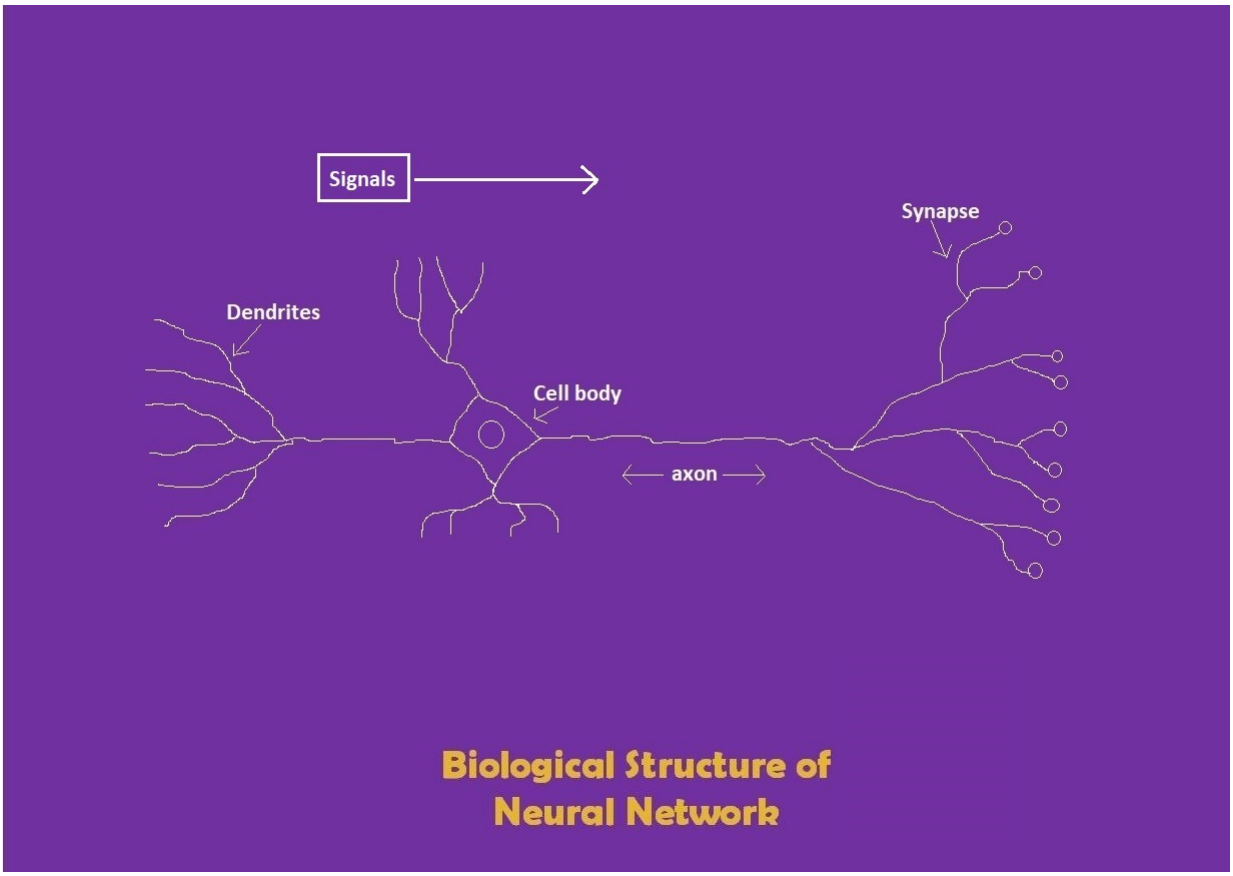

Figure 1.1: Biological neuron model

1.3.1 History of NNs

In this section, we stated the history of NN. The 1940s were the beginning of the era of neural networks. Most people think that neuro-computing began when McCulloch and Pitts published a research paper in 1943. Hebb wrote a book called "The Organization of Behavior" in 1949. In the 1940s, there were other researchers, who analyzed the challenges of neurocomputing. The era between the 1960s and 1950s marked the beginning of the "golden age" of NNs. The quiet years were the 1970s. The 1980s: Revived Enthusiasm [3, 4, 5].

1.4 Modeling ANNs Mathematically

A neuron M_i will take in a number of n inputs $S = \{t_j | j = 1, 2, ..., n\}$ shown in Fig 1.2. This one has a simple structure for an ANN. Before an input reaches the main body of a neuron, M_i,

it is weighted based on the connection strength, i.e., w_{ij} for $j = 1, 2, ..., n$. The threshold value must be met or surpassed for the neuron to create an output signal, and it has a bias term called b. On the generated weighted signal, $f(s)$ operates known as activation function is the name of this process. The i^{th} neuron's output, M_i is expressed mathematically as:

$$O_i = f\left[b + \sum_{j=1}^{n} w_{ij} t_j\right]. \tag{1.1}$$

The operation of an artificial neuron (AN) in a NN is depicted in great detail in Fig. 1.2.

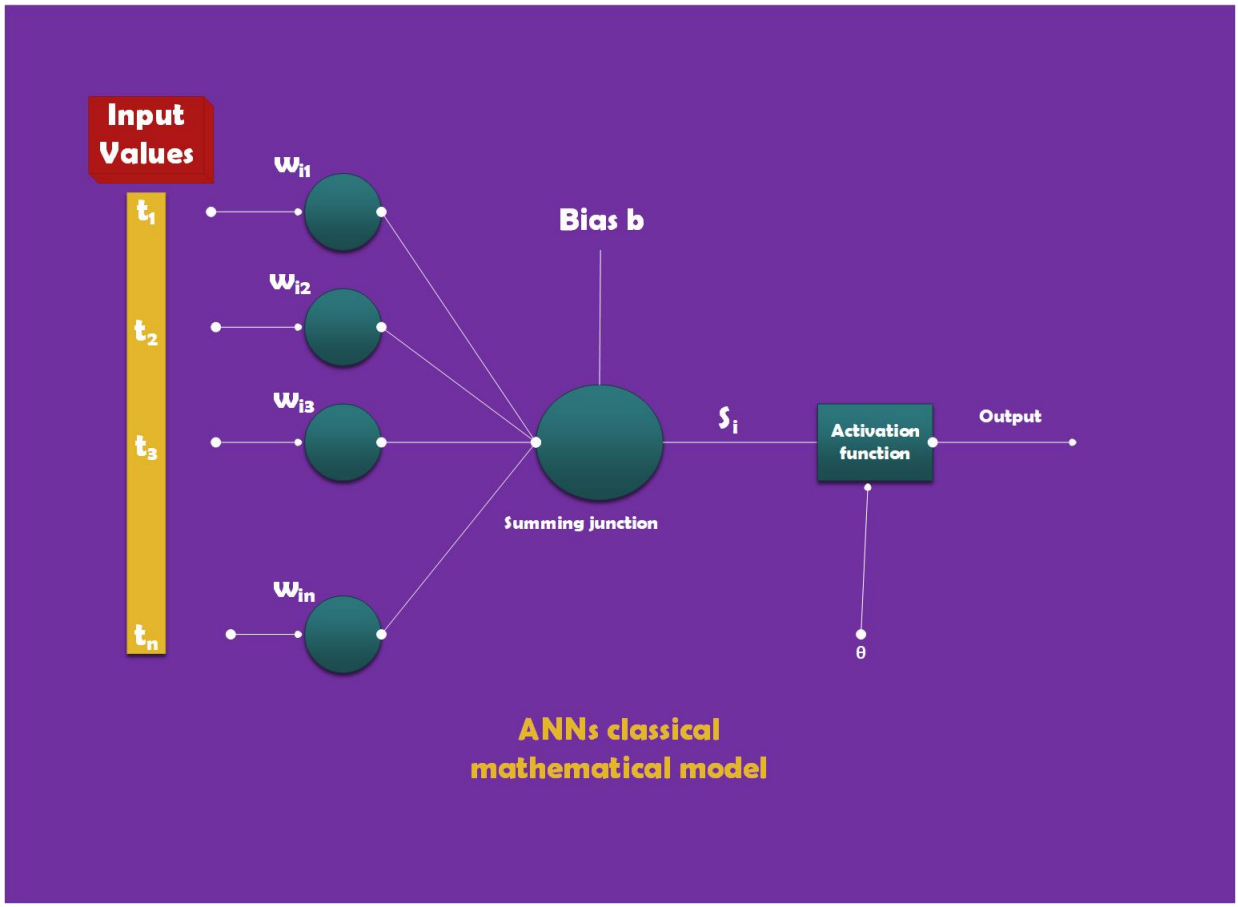

Figure 1.2: ANN Mathematical model

1.5 Activation Functions

As stated in the preceding section, the output is a function of the neuron's multiple weights w_{ij} and inputs t_j. Initially, a threshold function was proposed for the neuron output function, although sigmoid, step functions, linear, and sign are extensively utilized [6], as shown in Fig. 1.3

inputs, thresholds, neurons, and weights output may be real. Neuron's net input is mathematically written as:

$$net = \left(\theta + \sum_{j=1}^{n} w_{ij} t_j \right), \tag{1.2}$$

where θ is a threshold value added to the neurons. The neuron acts in an activation-function like manner or $f(net)$ and create a result say y, stated as:

$$y = f(net) = f\left(\theta + \sum_{j=1}^{n} w_{ij} t_j \right). \tag{1.3}$$

In the above equation, f represents the activation function. Following are some examples of activation functions.

1.5.1 Linear Activation Function

The linear activation function, which is depicted in Fig. 1.3a, is the name of the linear neuron transfer function. Mathematically, it can be represented as follows:

$$y = f(net) = f\left(\theta + \sum_{j=1}^{n} w_{ij} t_j \right) = net. \tag{1.4}$$

1.5.2 Sigmoid Activation Function

It is a nonlinear function with an S-type shape. The output is between 0 and 1. If the input value approaches infinity, then the output value converges to +1, as shown in Fig. 1.3b, and mathematically can be written as follows:

$$y = \left(\frac{1}{1 + e^{-\lambda t}} \right). \tag{1.5}$$

1.5.3 Sign Activation Function

This function is a type of piecewise function. This function defines limits to -1 and $+1$, and sometimes zero, as shown in Fig. 1.3c, and can be mathematically formulated as:

$$y = \begin{cases} +1 & \text{if} net \geq 0 \\ -1 & \text{if} net < 0. \end{cases} \tag{1.6}$$

Figure 1.3: Activation Functions

1.5.4 Step Activation Function

This one is also a type of piecewise function. This function defines the $+1$ and 0 limits shown in Fig. 1.3d and mathematically can be expressed as follows:

$$y = \begin{cases} +1 & \text{if net } \geq 0 \\ 0 & \text{if net } < 0. \end{cases} \tag{1.7}$$

1.6 Neural Network Architecture

In practical situations, a single node is insufficient, and networks with numerous nodes are routinely employed. The manner in which nodes are connected impacts how computations are performed and are a crucial early design decision for neural network developers. A neural network is essentially a data processing system made up of a large number of simple, highly interconnected processing components called neurons, as shown in Figs. 1.4-1.6. In this graph, we chose one input layer, one output layer, and two hidden layers. The word nodes stand for neurons and synaptic lengths are shown by edges [7].

1.6.1 Recurrent Neural Networks

A recurrent network can include arbitrary connections between any two nodes including connections that travel backward from output nodes to input nodes. A recurrent network can be said to have a memory because of the way that its internal state can change as input data sets are presented to it. This is very helpful, when trying to solve issues, when the answer depends on all prior inputs as well as the current ones.

The recurrent network sends data from inputs to outputs through the network while learning, as well as the other way around, and until the output numbers stay the same, continue this operation. The network is said to be in a stable or equilibrated condition at this moment. This type of NN is presented graphically in Fig. 1.5.

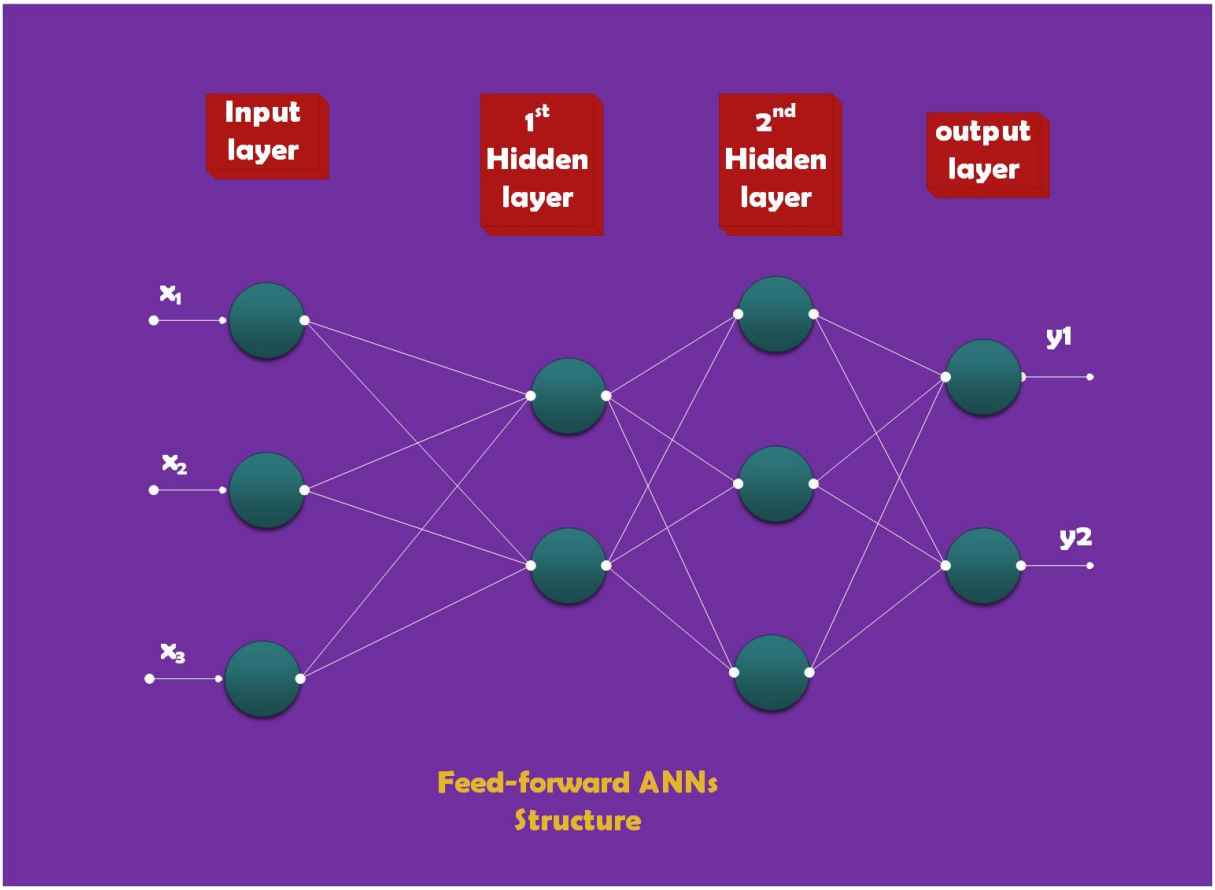

Figure 1.4: Feed Forward Neural Networks structure

1.6.2 Feed Forward Neural Networks (FF-NN)

A feed-forward neural network's (FF-NN) neurons are organized into layers. Instead of generating a series of values from a given input, FF only produces one set of output values. The simplest version or devise of ANN is FF-NN. In this type of network, we move the signals or information in one direction, from input to hidden layers and finally to the output layer [8]. This type of NN represents in Fig. 1.4.

1.6.3 Radial Basis Function Neural Network (RBF-NN)

This one is made of three layers. Considering the first layer as the input layer, the second layer as the BBF layer, and the third layer as the output layer. Graphically shown in Fig. 1.6

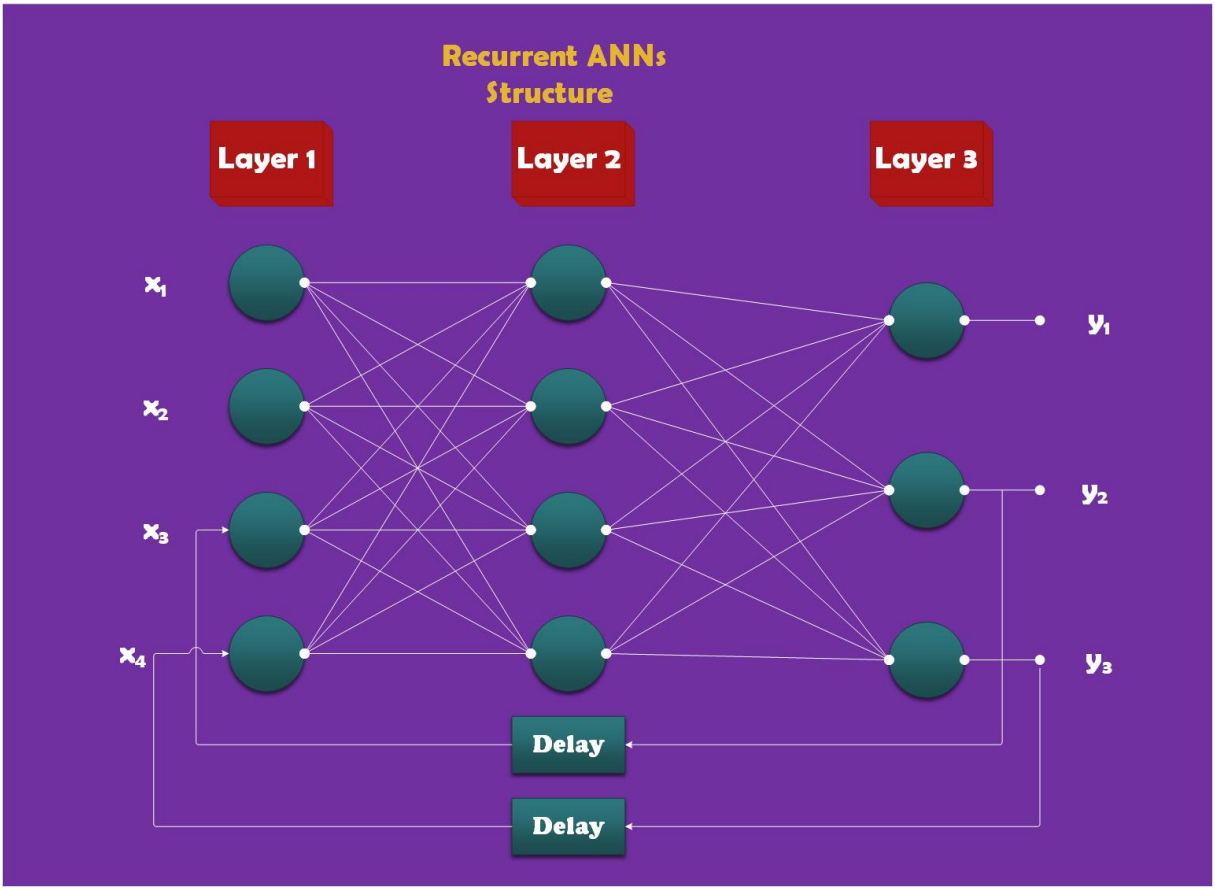

Figure 1.5: Recurrent Neural Networks structure

1.7 Paradigms of Learning

The ability of ANN to learn from a set of data and generalize is one of its most important features. There are two types of learning, that is stated below:

1.7.1 Supervised Learning

In supervised training, both inputs and results are provided. Then the network examines the inputs and evaluates the outcomes about the desired outcomes. A comparison between the network's computed output and the predicted output after correction is done to determine the inaccuracy. Once the error has been identified, network settings can be changed to enhance performance [9]. In other words, the inputs to the causal chain are assumed to come first, and the outputs are assumed to come last. Variables that serve as bridges between inputs and outputs can be incorporated into models.

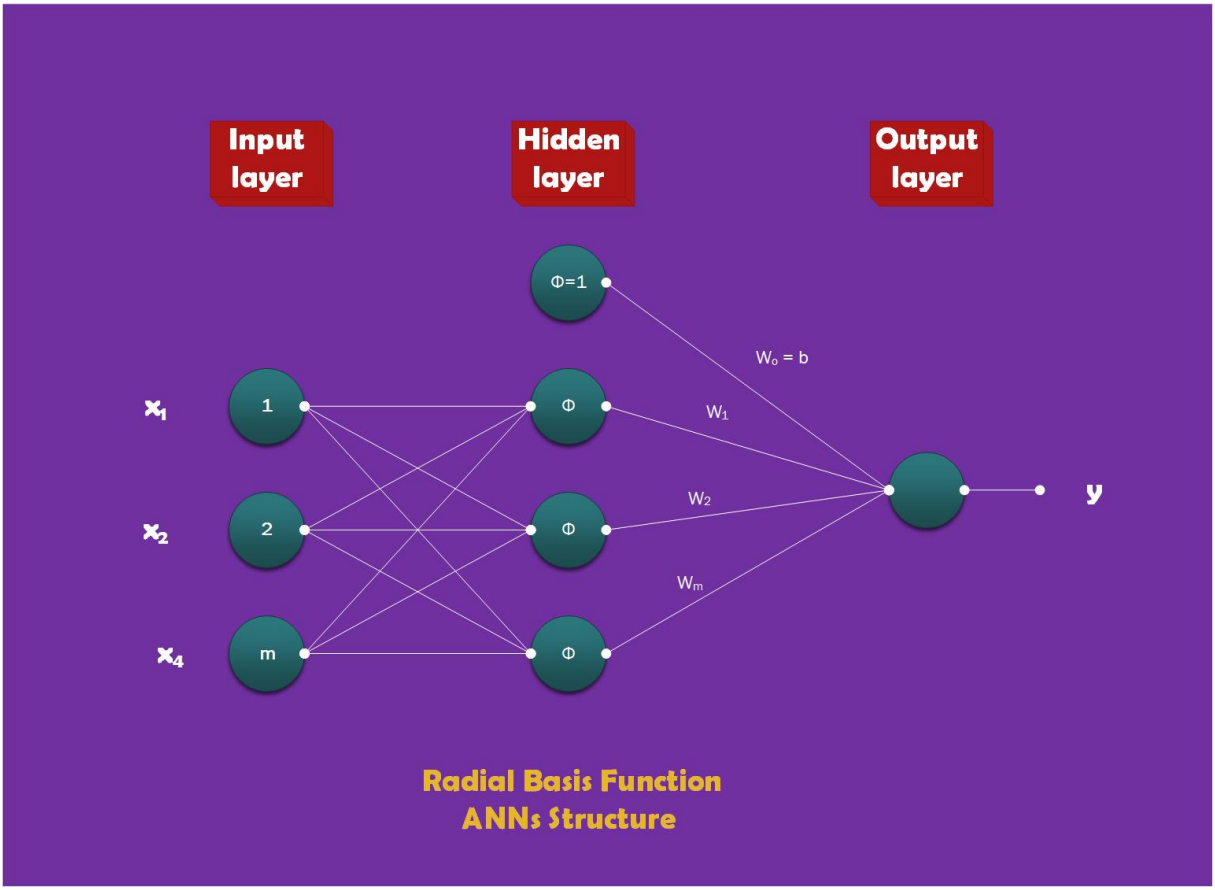

Figure 1.6: (BBF) NNs structure

1.7.2 Unsupervised Learning

The alternative to supervised training is unsupervised training. In unsupervised training, the network is given inputs but not the anticipated outputs. The following step is for the system to decide which features will be utilized to arrange the data input. Adaptability or self-organization are phrases used to characterize this process [9]. Since the goal is to make decisions that maximize the benefits rather than produce a classification, unsupervised learning generally fits within the decision problem paradigm and is seen to be more challenging than supervised learning. Unsupervised learning's objective is to unearth latent structures in unlabeled data [10]. Because the examples given to the learner are unlabeled, there is no mistake or reward signal to assess a suggested potential solution.

1.8 Differential Equation (DE)

In the mathematics field, an equation with one or more functions' derivatives is called a differential equation (DE) [11]. The differential equation (DE) can also be defined as a relationship between the derivatives of an unknown function and the function itself. Most mathematical models are formulated in the form of differential equations (DEs).

1.9 Differential Equations Classification

Differential equations (DEs) can be classified into the following categories [12]:

1.9.1 The Ordinary Differential Equation (ODE)

The ordinary differential equation (ODE) is a differential equation with just one independent variable. This type of differential equation (DEs) is also known as $2D$ differential equation (DE). ODEs have two types of variables, one is an unknown variable and the second one is a known variable. The classical form of the n^{th}-order ODE is stated below:

$$f\left(\theta, \phi, \frac{d\phi}{d\theta}, \frac{d^2\phi}{d\theta^2}, \ldots, \frac{d^n\phi}{d\theta^n}\right) = 0. \tag{1.8}$$

1.9.2 Partial Differential Equation (PDE)

Partial differential equations (PDE) are differential equations with two or more independent variables. The classical form of n^{th} order PDE is written bellow:

$$f\left(\theta_1, \theta_2, \ldots, \theta_n, \frac{\partial\phi}{\partial\theta_1}, \frac{\partial\phi}{\partial\theta_2}, \ldots, \frac{\partial\phi}{\partial\theta_n}, \frac{\partial^2\phi}{\partial\theta_1\partial\theta_2}, \ldots, \frac{\partial^2\phi}{\partial\theta_1\partial\theta_n}, \ldots\right) = 0. \tag{1.9}$$

1.9.3 Delay Differential Equation (DDE)

DDE is a sub-type of functional differential equations. In a delay differential equation, the derivative of an unknown function at a particular time is expressed in terms of the function's

values at earlier times, as opposed to the ODE in which the derivative is expressed in terms of the function's values at earlier times. Therefore, the solution of a delay differential equation necessitates information not only of the current state but also of the situation at a specific period in the past.

1.9.4 Differential algebraic equation (DAE)

A differential algebraic equation (DAE) is a generalized form of an ordinary differential equation in which an unknown function and its derivatives are involved. This type of equation emerges in the mathematical modeling of a wide range of engineering and scientific problems, chemical process control, including optimum control, in-compressible fluids, etc.

1.9.5 Stochastic differential equation (SDE)

A stochastic differential equation (SDE) is a differential equation with stochastic terms. Consequently, the solution is itself a stochastic process. Modeling many phenomena based on stochastic differential equations, such as fluctuating stock prices or physical systems sensitive to thermal fluctuations.

1.10 Types of DE Problems

Differential equations have two types of problems. These are stated below:

1.10.1 Initial Value Problems (IVPs)

In an initial value problem, the dependent variable and any potential derivatives are stated immediately or at the same value as the independent variable in the equation. Initial value problems typically include a time component.

For example:

An initial value problem would define a value of $y(t)$ at time 0 if the independent variable is time over the range [0, 1]. Physically, if someone taps a motionless body of water with a certain force, the ripple that results provides us with the original situation.

1.10.2 Boundary Value Problems (BVPs)

When the dependent variable and its potential derivatives are stated at the extreme of the independent variable, we have a boundary value problem. The auxiliary conditions for steady-state equilibrium problems are boundary conditions applied to the full closed solution domain boundary.

For example:

An equation for a boundary value problem would specify values for $y(t)$ for both $t = 0$ and $t = 1$ if the independent variable is time over the range [0, 1]. If the problem depends on both space and time, the data could be provided at a given time for all spaces rather than describing the value of the problem at a specific place for all time.

Further, boundary condition have three types [13], which are stated as follows:

Dirichlet Boundary Condition

The values of the function are given on the boundary in a Dirichlet boundary condition. For example, if one end of an iron rod is maintained at absolute zero, the value of the issue is known at that location in space. Finding the solution to such equations is known as the Dirichlet problem. A Dirichlet boundary condition is imposed on an ordinary or partial differential equation that specifies the values of a solution to take on the border of the domain.

For example: Consider the following example of a partial differential equation.

$$\begin{cases} \frac{\partial^2 \phi}{\partial \theta^2} + \frac{\partial^2 \phi}{\partial t^2} = f(x,y) \\ \phi(0,t) = A, \phi(L,t) = B, \end{cases} \tag{1.10}$$

where A and B are integers. Given that the value of the function $\phi(\theta,t)$ is stated on the boundary for $\theta = 0$ and $\theta = L$, the boundary condition in the aforementioned equation is a

Dirichlet boundary condition.

Neumann Boundary Condition

Values of the function are defined on the derivative normal to the boundary in the Neumann boundary condition. For instance, energy would be provided continuously if one iron rod had a heater at one end, but the actual temperature would not be known. Ordinary or partial differential equations are subjected to Neumann boundary conditions, which specify the derivative values of solutions to be applied to the domain's edge.

For example:

$$\begin{cases} \frac{\partial^2 \phi}{\partial \theta^2} + \frac{\partial^2 \phi}{\partial t^2} = f(x,y) \\ \frac{\partial \phi}{\partial n}(0.t) = \omega_1, \frac{\partial \phi}{\partial n}(L.t) = \omega_2. \end{cases} \tag{1.11}$$

Mixed Boundary Condition

Mixed boundary conditions, commonly referred to as the Cauchy boundary condition, are the linear combination of Dirichlet and Neumann boundary conditions. When a mixed boundary condition is applied to an ordinary or partial differential equation, it specifies the values that the differential equation must take into account at the domain's boundary as well as its normal derivative. Invoking both the Dirichlet and Neumann boundary conditions is equivalent.

For example:

$$\begin{cases} \frac{\partial^2 \phi}{\partial \theta^2} + \frac{\partial^2 \phi}{\partial t^2} = f(x,y) \\ \omega_1 \frac{\partial \phi}{\partial n}(0,t) + \omega_2 \phi(0,t) = \omega, \\ \omega_1 \frac{\partial \phi}{\partial n}(L,t) + \omega_2 \phi(L,t) = \omega_i. \end{cases} \tag{1.12}$$

1.11 Numerical Methods (NMs) for solving DEs

Numerous theorems in the field of mathematics provide assurance that solutions exist in many situations, yet there is currently no known numerical method for getting those solutions

in the explicit and closed form [14, 15]. Due to these drawbacks of analytical approaches in real-world applications, numerical methods have developed. For different types of complicated problems with no analytical solution, there exist several numerical techniques. These are stated following:

1.11.1 Shooting Method

This method transforms boundary value problems into two initial value problems by adding a sufficient number of conditions at one end and adjusting these conditions until the given conditions are satisfied at the other end. The Taylor series, Runge-Kutta, and other techniques are used to solve these two initial value problems. and the sum of the two solutions acquired by resolving initial value problems provides the necessary solution to the supplied boundary value problem.

1.11.2 Finite Difference Method

Functions are represented by their values at specific grid locations in the finite difference method (FDM), and derivatives are approximately determined by differences in these values. The domain under examination is represented by a limited subset of points for the finite difference method. These are referred to as "nodal points" of the grid. This grid is generally set up in a rectangular fashion, whether it is uniform or not. A collection of difference equations is used in place of the differential equation, and these equations can be solved directly or iteratively.

1.11.3 Finite Element Method

The finite element method is a numerical approach similar to the finite difference approach, but it is more powerful and applicable to issues in the real world with intricate boundary conditions. Functions are modeled using basis functions in the finite element method (FEM), and differential equations are resolved in their integral (weak) form. The domain under examination is divided into a finite set of elements, $\{\Omega_i\}$, in the finite element approach such that $\Omega_i \bigcap \Omega_j$ for (i not equal j), and $\bigcup \overline{\Omega_i} = \overline{\Omega}$: The function is then approximated by a low degree piece-wise

polynomial. Additionally, they are built in a way that just a few aspects are supported by them. It is simpler to describe a complicated function as a collection of simple polynomials, which is the basic justification for using an approximation solution on a group of subdomains.

1.11.4 Finite Volume Method

Differential equations are modeled and evaluated as algebraic equations using the finite volume method. The little volume that surrounds each node point on a mesh is known as a finite volume. Similar to the Finite Difference Method, the Finite Volume Method calculates values at discrete points inside mesh geometry. Using the divergence theorem, this approach transforms volume integrals in a differential equation with a divergence term into surface integrals. These terms are assessed as fluxes at the surfaces of each finite volume, and the procedure is conservative in that the flux entering one volume is the same as the flux exiting the one next to it. In comparison to the finite difference approach, the finite volume method has the benefit of not requiring a structured mesh and allowing for non-intrusive application of boundary conditions. This approach works well with irregular grids and calculations where the mesh moves to follow interfaces.

1.11.5 Spline-Based Method

A spline is typically a piecewise polynomial function that is defined in a region, with the function being a polynomial of some degree in each of the region's subregions. The differential equation is discretized in spline-based approaches utilizing approximations based on splines. For the definition of a spline, the end conditions are derived. The created algorithm approximates both the answers and their higher-order derivatives.

1.11.6 Physical Problems Associated with Differential Equations (DEs)

Mountains crumble, river bottoms move, machinery malfunctions, the environment becomes increasingly polluted, populations shift, economies change, and technology develops as the planet rotates. Therefore, every quantity that can be expressed mathematically over an extended period

of time must alter with time. Many quantities vary quickly as a function of time, relativistically speaking, including heartbeats, pendulum swings, chemical explosions [16], etc. Other examples are the natural pulsing of quartz crystals, heartbeats, and the swing of a pendulum. In this research work, we considered the heartbeat mathematical model known as the van der Pol differential equation.

Our experience suggests that the rate of change of a physical or biological quantity relative to time contains crucial information about the system when it comes to the quantitative study of any system. The majority of physical and biological models that can be analyzed mathematically are formulated with this rate of change in mind.

1.12 Design Framework of This Thesis

This thesis structure includes six chapters, as mentioned below:
Chapter 2 describes the literature review. The review model of the heartbeat problem modification is listed in Chapter 4. Chapter 5 proposes a design methodology for computing the results. Chapter 6 describes numerical simulation results and has some discussion on them. Chapter 3 reports the conclusion and represents future work.

we review the importance of artificial intelligence (AI), artificial neural networks (ANNs), and their types and the beginning of this revolution in the mathematics field. Then we presented the definition of differential equations (DE), their types, and classifications. The applications of DEs are overviewed. The heartbeat model (HBM) modification review is presented in chapter 4. Actually, HBM is the problem that we are going to solve in our thesis research. The next chapter 2 is about the literature review.

CHAPTER 2

LITERATURE REVIEW

In this chapter, the different research studies that are related to our research area are reviewed. Also, highlighted the literature related to the research topic and cite different articles. This chapter will help us improve our basic knowledge regarding our topic. At the end, we conducted our numerical simulation process.

Due to the lack of exact solutions for the nonlinear VdP oscillatory system, a wide variety of analytical and numerical solvers have been developed. A few of them are Adomian decomposition methods (ADM) [17], Laplace decomposition method [18], homotopy analysis method [19], He's parameter-expanding methods [20], and linearization method (LM) [21], etc. Each of these techniques has its own inherent value, scope, restrictions, and applications. However, aside from the stochastic solvers' well-known strength, AI-based stochastic solvers are thought to be efficient, precise, and dependable methods for solving numerous unconstrained and constrained optimization problems that arise in a number of applications in a wide range of domains [22, 23, 24, 25].

Astrophysics modeling is among the few new applications tackled by ANNs-based AI approaches [26]. Some other applications of astrophysics modeling are Bi-linear programming problems (BPP) [27], second kind Fredholm integral equations (FIE) [28], buckling analysis of a beam–column [29], an inverse kinematics problem [30], etc. The effectiveness of FF ANN as an approximation of a universal function is utilised to build stochastic numerical approaches, and typically, these networks are optimised by combining viable local and also global search techniques for solving nonlinear systems [31, 32, 33, 34, 35]. Recent potential applications of

these solvers include the solution of nonlinear, rigid VdP oscillatory systems [36, 37], control fuel ignition model derived from combustion theory [38], nanofluidics problems [39, 40], problems in fluid mechanics based on thin-film flow [41], nonlinear equations of the Troesch type arise in plasma physics [42, 43], model of longitudinal heat transfer fins [44], system based on linear volterra integral equations [45], fuzzy differential equations [46], Navier–Stokes equations [47], BVPs of pantograph-type functional ODEs [48, 49], the problem of moving singularities in nonlinear Painleve'-type equations [50], magnetohydrodynamics (MHD) studies [51, 52], models of electrical conducting solids [53], thermodynamics studies of spherical cloud model [54], nonlinear fractional order systems [55], fractional optimal control problems [56], problems arising in electromagnetic theory [57] and nonlinear. Lane–Emden-type equations [58], the above are my motivations as a researcher. It explores the robustness of stochastic solvers to design a robust, alternate, reliable, and accurate computing method for studying the VdP heartbeat dynamic model.

The PSO has been employed with effectiveness in numerous fields, including robotics [59, 60, 61, 62], sport sciences [63] and electric systems [64]. However, a common issue with the PSO and other optimization algorithms is that they might become stuck in a local optimum, even though they may be effective for a particular situation, but could fall short on another issue. Numerous authors have recommended different changes to the PSO algorithm's settings that combine fuzzy logic to get around this issue, where fuzzy logic is used to dynamically change the inertia weight w "IF THEN" [65] rules.

Recently, Pires et al. used fractional calculus to regulate the PSO's rate of convergence [66]. The authors adjust the order of the velocity derivative by rearranging the original velocity equation (4.1). Based on the work of Pires et al., this research attempts to regulate the convergence rate of an evolving version of the PSO, since the most successful PSO variations are those based on evolutionary approaches [67].

Numerous writers have investigated combining selection, differential evolution (DE) , crossover, and mutation into the PSO algorithm. The primary objective is to increase population variety by preventing particles from moving too closely together and colliding [68, 69] or to automatically adjust variables like the acceleration constants, constriction factor [70] or inertia weight [71]. The GA-PSO was created by combining genetic algorithms (GA) and the PSO [72]. This combines the benefits of a natural selection system and swarm intelligence, include GA, to boost the proportion of highly regarded agents, while reducing the quantity of agents with

low evaluation scores at each iteration stage. Similar to the previous one, the EPSO algorithm adds a selection process to the basic PSO algorithm in an evolutionary fashion, furthermore has characteristics that can adapt on their own. This approach incorporates an evolutionary programming method for tournament selection [73]. On the basis of the EPSO, a DE operator has been developed to enhance the algorithm's performance in two distinct ways. The 1^{st} one [74] eliminates particles that enter local minima by applying the DE operator to the particle's optimal position (DEPSO), the second one [75] uses it to determine the acceleration constants and ideal inertia for the standard PSO (C-PSO).

A deep literature review related to our research area (topic) states and discusses the research work that has already been done in the field of "solving differential equations using artificial neural networks" or related to our research thesis topic, i.e. an innovative neuroevolutionary approach for heartbeat analysis with ANN.

CHAPTER 3

REVIEW OF HEARTBEAT MODEL

3.1 Modeling of the heart using nonlinear Van der Pol oscillators

An overview of the background research for the Van der Pol (VdP) system, which is represented in Eq. (4.1), is given in this section. The first purpose of the VdP system was to describe relaxation oscillators in electronic circuit modeling and is often used in theoretical models of the heart's rhythm. The following nonlinear oscillator model is the heart's mathematical representation based on the VdP system [76, 77]:

$$\ddot{x} + \tilde{\alpha}\left(x^2 - 1\right)\dot{x} + \omega x = 0, \tag{3.1}$$

where the constant coefficients are ω and $\tilde{\alpha}$, represents dupping and damping respectively related parameters of system. Due to the similarity of the biological systems and VdP equation in terms of chaos, limit cycles, and synchronization. Theoretical models of heart oscillations commonly employ a VdP-based DE system [78, 17, 18, 19, 20, 21, 22]. The VdP equation adjusts its inherent frequency to the driving pacemaker frequency from outside, without any amplitude change, which is an essential component of a cardiac pacemaker. Zebrowski and Grudzinski 1^{st} presented these traditional VdP heart models (HMs) [79]. Later modifications were made to the original VdP HM and its characteristics are drastically altered, simply adding together two fixed positions, stable node at $x = -2d$ and saddle-node at $x = -d$. Accordingly, the VdP system for the HM with updated unsymmetrical voltage-related damping terms is given as follows:

$$\ddot{x} + \tilde{\alpha}\left(x^2 - \mu\right)\dot{x} + \frac{x(x+d)(x+(2\times d))}{d\times d} = 0. \tag{3.2}$$

The distance between these two fixed positions cannot be altered according to equation (3.2); equation (3.2) is consequently modified by the addition of a new parameter e, it is in charge of altering the depolarization period as:

$$\ddot{x}+\tilde{\alpha}\left(x^2-\mu\right)\dot{x}+\frac{x(x+e)(x+d)}{e\times d}=0. \tag{3.3}$$

In equation (3.3) replacing the term $\left(x^2-\mu\right)$ with $(x-v_1)(x-v_2)$, the new terms are asymmetric w.r.t x, formulated as below:

$$\ddot{x}+\tilde{\alpha}\left(x-v_2\right)\left(x-v_1\right)\dot{x}+\frac{x(x+e)(x+d)}{e\times d}=0. \tag{3.4}$$

The above equation is the homogeneous form of the VdP heartbeat model (HBM). To keep the system's self-oscillatory properties, the condition $v_1v_2<0$ must be met. Further, the equation (3.4) is formulated, and detail discussed below.

3.2 Heartbeat Model (HBM)

Due to their stability, simplicity, and single limit cycle, VdP oscillatory systems have been utilized to precisely replicate the theoretical model of cardiac functions [80, 81], including relaxation, bifurcations, chaotic behavior, and periodicity [78]. In the form of a non-linear oscillator, the heartbeat model (HBM) can be formulated as follows [82]:

$$\begin{aligned}\ddot{x}+\tilde{\alpha}\left(x-v_1\right)\left(x-v_2\right)\dot{x}+\tfrac{(x+e)(x+d)x}{e\times d}&=F(t),\\ x(0)=c_1,\qquad \dot{x}(0)&=c_2.\end{aligned} \tag{3.5}$$

The above equation is the final form of equation (3.4). Where $\tilde{\alpha}$ represents the pulse modification factor of the heartbeat, v_1 and v_2 represent asymmetric parameters. x is the length of heart fiber, d is a factor that appears when a cubic term is used to replace the harmonic force in the traditional VdP equation, e is for the period of ventricular contraction, and $F(t)$ is an external force term. The heart biological model is given in Fig. 3.1.

Equation (4.1) is the representation of the Van der Pol (VdP) heartbeat model (HBM) modified form. In this thesis, we analyzed this model using our hybrid technique fractional order particle swarm optimization (FO-PSO). We perform some variations in parameters $(\tilde{\alpha},v_1,v_2)$, that are used in equation (4.1) and choose constant values for some parameters (d,e). The terms $x(0)=c_1$ and $\dot{x}(0)=c_2$ are initial conditions. Where the values of $c_1=-0.1$ and $c_2=0.025$.

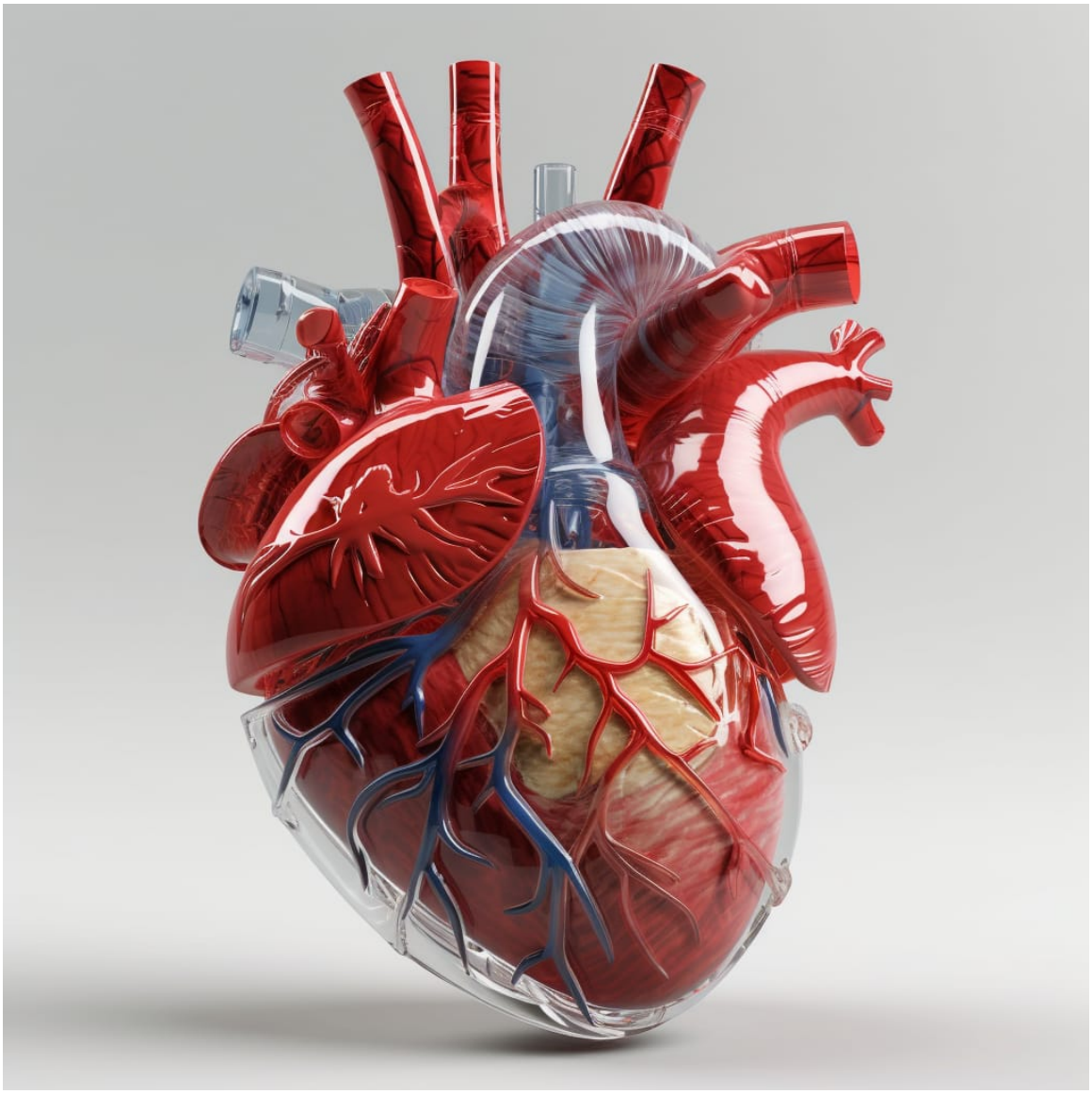

Figure 3.1: 3D Human heart model generated in mid-journey AI (an independent research lab)

3.3 Particle Swarm Optimization (PSO) Algorithm

Population-based searches may be a useful alternative when the search space is too huge to be searched thoroughly, but they cannot ensure that you will find the optimal (best) solution.

Dr. James Kennedy and Dr. Russell C. Eberhart originally introduced particle swarm optimization in 1995. Particle Swarm Optimisation (PSO), a population-based search methodology, will be covered. The PSO Algorithm and the Genetic Algorithm both have features in common, but they approach the search area in fundamentally different ways [60].

The following points are shared by both genetic algorithms and particle swarm optimizers:

1. Both methods initialize a population in the same way.

2. To establish how suitable (excellent) a proposed solution is, both employ an evaluation function.

3. Both are generational, which means they both repeatedly go through the same set of steps over a predefined period of time.

The two main operators in particle swarm are velocity update and position update. Each particle is accelerated towards its prior best position and the overall best position during each generation. Based on each particle's current velocity, distance from its prior best location, and distance from the global best position, a new velocity value is determined for each iteration. The next point of the particle in the search space is then determined using the new velocity value. Then, this procedure is repeated a predetermined number of times, or up until a minimum error is reached [67, 65]. The detailed mathematical formulation is stated in chapter 4.

3.4 Fractional Order Calculus

The letter Leibniz wrote to L'Hopital in 1695 demonstrates that the intuitive notion of FOC is as old as IOC. The IOC is being applied to a genuine or complex order. Fractional order calculus dates back to the origins of differential calculus theory. FC hasn't been used much until the last 20 years, though, because of changes in nonlinear and complex systems that have shown subtle links with FC principles. Regarding FC applications study there have been initiatives in the field

of viscoelasticity [83, 84, 85], [86, 87] chaos, fractals, electronics, diffusion, signal processing, irreversibility, modeling, percolation, physics, biology, and wave propagation. Various attempts have been made since its inception to generalize the concepts of derivative and integral to a non-integer order, α. Due to this, fractional derivatives have multiple different definitions. Various attempts have been made since its inception to generalize the concepts of derivative and integral to a non-integer order of α [83, 84, 85],. Due to this, fractional derivatives have multiple different definitions. Here, taking Laplace's definition as an example, Laplace's definition of a fractional-order derivative α in C of the signal $x(t), D^{\alpha}[x(t)]$, is a 'direct' generalization of the conventional integer-order scheme, yielding:

$$\mathscr{L}\{D^{\alpha}[x(t)]\} = s^{\alpha}X(s). \tag{3.6}$$

The Grünwald–Letnikov definition, which is based on the concept of fractional differential, is an alternate method [83]. The further formulation is stated in chapter 4.

3.5 Fractional Order Particle Swarm Optimization (FO-PSO)

A modification of the standard Particle Swarm Optimisation (PSO) algorithm that uses fractional calculus ideas is called Fractional Order Particle Swarm Optimisation (FO-PSO). PSO is a metaheuristic algorithm that is frequently employed for optimization issues. It was inspired by the social behavior of flocking birds or schooling fish.

FO-PSO uses the idea of fractional order calculus to improve the algorithm's search capabilities. Because it allows for the differentiation and integration of non-integer order, fractional calculus expands on classical calculus. This method adds memory and history to the optimization process, allowing the algorithm to more thoroughly scan the search space.

A population of particles that represent potential answers to the optimization problem is iteratively updated by FO-PSO to operate. Based on its own best position from the past and the best position discovered by the entire population, each particle modifies its position. When employing fractional order differentiation and integration operators to update the particles' position and velocity, the fractional order aspect is relevant.

FO-PSO offers a more flexible and effective search space exploration by combining fractional calculus, enabling a better trade-off between exploration and exploitation. It has been

demonstrated to be very successful in resolving challenging optimization issues in non-convex and non-linear search spaces.

FO-PSO, which stands for fractional order PSO, is an improved variant of PSO that uses fractional order calculus to enhance search capabilities and algorithm efficiency in optimization tasks. For further mathematical formulation kindly see the chapter 4 proposed research methodology.

3.6 Novelty of Our Research Thesis

After reviewing so many scholarly works, it is clear that computation intelligence (CI)-based solvers continue to be the primary concern of scientists in this technological era. These CI-based strategies are efficient for solving nonlinear models such as those given in [88, 89, 90, 91, 92] due to their proficiency and high performance. Studies have shown that they can effectively scale up to tackle large-scale problems, in contrast to conventional optimization approaches, which are more effective for small-scale problems. FO-PSO has a lot of room to grow in terms of learning methodologies, but so far little has really changed.

CHAPTER 4

AN INNOVATIVE NEUROEVOLUTIONARY APPROACH FOR HEARTBEAT ANALYSIS WITH ANN

In this chapter, we proposed a novel design methodology, also called the proposed design of soft computing. This proposed design methodology consists of a series solution for our model, a fitness function used for the optimization of our problems by minimizing the mean square error, and at the end, we presented a note on our hybrid technique FO-PSO-ASA. This process will help us conduct the numerical simulation and get our results, which are stated in the next chapter 5.

4.1 Heartbeat Model

The following equation is the general form of the heartbeat model:

$$\begin{aligned} \ddot{x} + \tilde{\alpha}\,(x - v_1)\,(x - v_2)\,\dot{x} + \tfrac{(x+e)(x+d)x}{e\times d} = F(t), \\ x(0) = c_1, \qquad \dot{x}(0) = c_2. \end{aligned} \tag{4.1}$$

4.2 Proposed Design Methodology

Even though machine learning techniques have come a long way in the past few years, it has been hard to use these new methods to solve differential equations (DEs) for a long time. So far, the most promising results have come from learning either better ways to solve problems or the solution itself. This last method is especially easy to understand. The idea of directly learning how to solve a DE works for all DEs, and because it is such a basic idea, it gives us a good place to start.

Although there are numerous approaches to learning a DE's solution, our proposed methodology for our mathematical model consists of two main parts; the 1^{st} phase contains the creation of an unsupervised ANN model to govern the system of differential equations (DE) provided in Equation 4.1. In the 2^{nd} phase, weights for these networks are optimized using soft computing methods shown in figure 4.4.

4.3 Mathematical modeling of neural networks (NNs)

The math model of neural networks for ODEs is made by using the power of approximation theory in the form of continuous mapping. Here we consider $\hat{x}(t)$ as approximate solution of our heart model (HM). The following networks are provided for the approximate solution $\hat{x}(t)$ and the derivatives of $\hat{x}(t)$:

$$\hat{x}(t) = \sum_{j=1}^{m} \varphi_j f\left(w_j t + \beta_j\right), \tag{4.2}$$

$$\hat{\dot{x}}(t) = \sum_{j=1}^{m} \varphi_j \dot{f}\left(w_j t + \beta_j\right) \tag{4.3}$$

$$\hat{\ddot{x}}(t) = \sum_{j=1}^{m} \varphi_j \ddot{f}\left(w_j t + \beta_j\right), \tag{4.4}$$

$$\cdots \cdots$$

$$\cdots \cdots$$

$$\cdots \cdots$$

$$\hat{x}^{(n)}(t) = \sum_{j=1}^{m} \varphi_j f^{(n)}\left(w_j t + \beta_j\right). \tag{4.5}$$

In above equations $\beta = [\beta_1, \beta_2, \beta_3, \ldots, \beta_m]$, $\varphi = [\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_m]$ and w=[w_1, w_2, w_3, $\ldots w_m$] are the real values which is arbitrary bounded. Updated form of the network set from Equations (4.2-4.5) using the log-sigmoid activation function $f(x) = 1/(1+e^{-x})$ and its derivatives is given as:

$$\hat{x}(t) = \sum_{j=1}^{m} \varphi_j \left(\frac{1}{1+e^{-\left(w_j(t)+\beta_j\right)}} \right), \tag{4.6}$$

$$\hat{\dot{x}}(t) = \sum_{j=1}^{m} \varphi_j w_j \left(\frac{e^{-\left(w_j(t)+\beta_j\right)}}{\left(1+e^{-\left(w_j(t)+\beta_j\right)}\right)^2} \right), \tag{4.7}$$

$$\hat{\ddot{x}}(t) = \sum_{j=1}^{m} \varphi_j w_j^2 \left(\frac{2e^{-2\left(w_j t+\beta_j\right)}}{\left(1+e^{-\left(w_j t+\beta_j\right)}\right)^3} - \frac{e^{-\left(w_j t+\beta_j\right)}}{\left(1+e^{-\left(w_j t+\beta_j\right)}\right)^2} \right), \tag{4.8}$$

$$\ldots$$

$$\ldots$$

$$\ldots$$

A mathematical representation of Equations (4.2-4.5) or (4.6-4.8) can be created using the appropriate network configurations specified in a set of Equation (4.1). Input, output, and hidden layers make up the NN architecture of a heart model based on the VdP equation, which is seen in Fig. 4.1.

4.3.1 Fitness Function (FF)

Using the mean square sense, a fitness function is created as follows in order to solve the cardiac dynamics model presented in the system (4.1):

$$\textit{minimize} \qquad \varepsilon = \varepsilon_1 + \varepsilon_2, \tag{4.9}$$

where the nonlinear VdP equation's mean square error function or ε_1, is provided as follows:

$$\varepsilon_1 = \frac{1}{N} \sum_{m=1}^{M} \left(\hat{\dot{x}} + \tilde{\alpha} \left(\hat{x}_m - v_2\right) \left(\hat{x}_m - v_1\right) \hat{x}_m + \frac{\hat{x}_m \left(\hat{x}_m + e\right) \left(\hat{x}_m + d\right)}{e \times d} \right)^2 \tag{4.10}$$

$$\text{for } N = \tfrac{1}{h}, \quad \hat{x}_m = \hat{x}\left(t_m\right), \quad \text{and} \quad t_m = mh,$$

Heart model is an initial value problem (IVP), so we are takin ε_2 is the mean square error for initial conditions. It is stated below in equation form:

$$\varepsilon_2 = \frac{1}{2}\left((c_1 + \hat{x}_0)^2 + (-c_2 + \dot{x}_0)^2\right). \tag{4.11}$$

If training weights $W = [\varphi, \omega, \beta]$ for the ANNs model are available, the solution for the heart model provided in model (4.1) can be determined such that the fitness value ε is close to zero, i.e. $\varepsilon \to 0$, then the ANN model's solution becomes closer to the equation's exact solution. i.e. $\hat{x}(t) \to x(t)$.

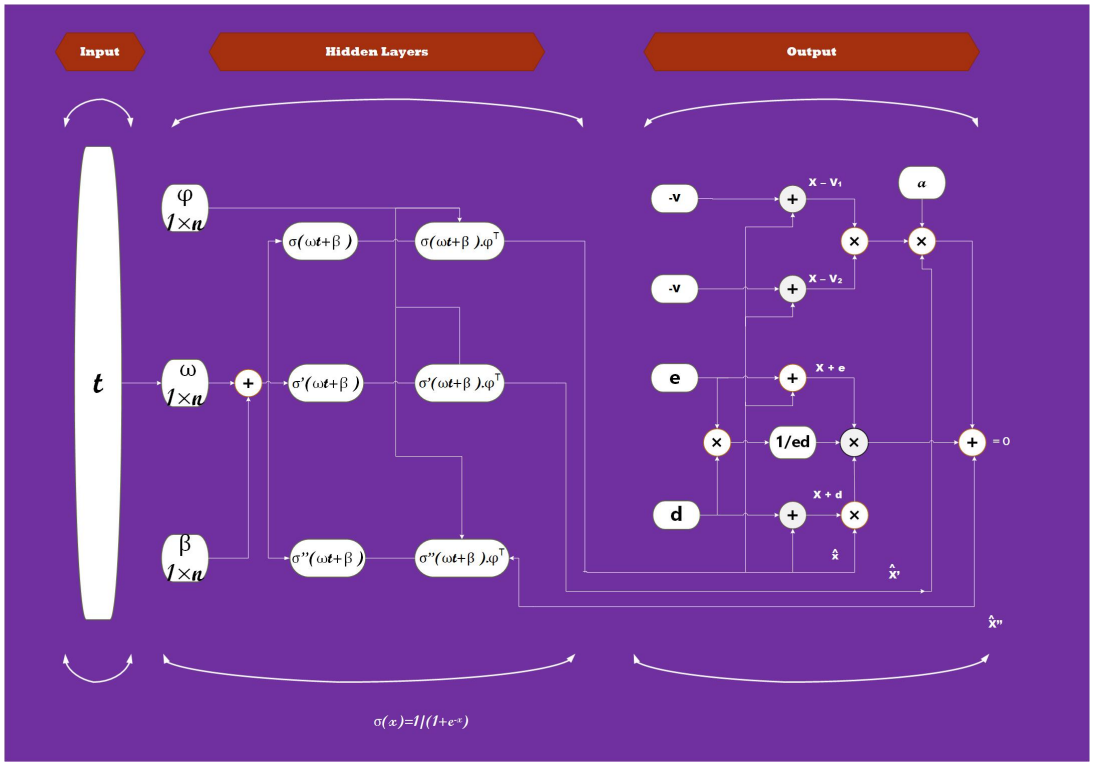

Figure 4.1: Graphical abstract of ANNs for heart model

4.3.2 Optimization procedure: FO-PSO–ASA

In order to solve the Van der Pol heart dynamics model 4.1, an evolutionary computing approach called FO-PSO-ASA, which combines fractional-order calculus (FOC) and particle swarm optimization (PSO) algorithms supported by an active set algorithm, is utilized to train the unknown adjustable parameter of ANNs.

Particle Swarm Optimization (PSO) Algorithm

The optimization technique known as Particle Swarm Optimization (PSO) was influenced by the group behaviour of fish schooling or bird flocking. It updates a swarm of particles in the

search space iteratively in an attempt to discover the best answer. The following mathematical formulation of the PSO algorithm is possible:

1. Initialization:

   - Define the N population size (particle count).
   - Set each particle's position and speed randomly at the beginning of the search space ($1 \leq i \leq N$).
   - Each particle's current location is set to the original best position (pbest).
   - Initialise the global best position (gbest) as the top particle position.

2. item Update particle position and velocity:

   Do the following for each particle ($1 \leq i \leq N$):

   - Apply the following equation to update particle velocity:

   $$u_{t+1} = u_t + \phi_1(B - X) + \phi_2(g - X), \tag{4.12}$$

   where u_t is current velocity of a particle, $u(t+1)$ is the updated velocity at time t. B is the best individual position and g is the best global position. X is the position of a particle. ϕ_1 and ϕ_2 are local and global information. The terms $\phi_1(B - X)$ and $\phi_2(g - X)$ represents local and global influence.

   - Using the equation below, adjust particle location:

   $$X_{t+1} = X_t + u_{t+1}, \tag{4.13}$$

3. Review and update the best particle positions:

   For each particle, i ($1 \leq i \leq N$), $dothefollowing$:

   4. Determine the particle fitness or objective function value in its new location.
   5. Update $pbest_i$ with the new position if it increases fitness in comparison to the previous best position ($pbest_i$).
   6. Update the global best position (gbest) with the new position if the fitness of the new position is better than the global best position (gbest).

Till a termination requirement (such as the maximum number of iterations or the target fitness level) is satisfied, repeat steps 2 and 3 as necessary. By altering the velocity and location of particles based on both their own best positions and the overall best positions discovered by the swarm, the algorithm seeks to converge to the best result.

Fractional order Calculus (FOC)

Recent studies show that fractional calculus (FC) attracted numerous scholars to use it in resolving and defining engineerin, mechanical, applied mathematics problems. By changing an unknown function $y(t)$ as in equation (4.14), Grunwald-Letnikor defined a fractional derivative including fractional coefficients as a real number $\alpha \in C$.

$$D^{\alpha}[y(t)] = \lim_{h \to 0} \left[\frac{1}{h^{\alpha}} \sum_{m=0}^{+\infty} \frac{(-1)^m \Gamma(\alpha+1)(t-mh)y}{\Gamma(m+1)\Gamma(\alpha-m+1)} \right], \tag{4.14}$$

where the gamma function is denoted by the symbol Γ. It is clarified further that in equation (4.14), the series is defined by finite terms if the derivative is of integer order. On the other side, the outcome is represented by the infinite term if α is of fractional order. Therefore, it is exciting to see that fractional operators convey a recollection of previous modifications, whereas conventional derivatives are local/instantaneous operators. Additionally, our ability to recall the past declines with time. In equation (4.15), a discrete instance derivative is defined:

$$D^{\alpha}[y(t)] = \frac{1}{T^{\alpha}} \sum_{m=0}^{r} \frac{(-1)^m \Gamma(\alpha+1)x(t-mT)}{\Gamma(m+1)\Gamma(\alpha-m+1)}, \tag{4.15}$$

where T stands for the time periods during which events occurred, while r stands for truncated terms. Due of their ability to maintain memories, fractional calculus tools are valuable in chaotic and irreversible processes.

4.3.3 Combined Form of FO-PSO algorithm

Here, we present an evolutionary technique for managing the particle swarm optimization algorithm. In order to modify the order of the velocity derivative, the initial velocity equation of a straightforward PSO is first rearranged, from equation (4.12) as shown below:

$$u_{t+1} = u_t + \phi_1(B-X) + \phi_2(g-X), \tag{4.16}$$

The movement of a particle is guided by two terms: cognitive and social. The cognitive term $\phi_1(B-X)$ represents the particle's tendency to move towards its own best position X, where ϕ_1

is the weight for personal information. while the social term $\phi_2(g-X)$ represents the particle's tendency to move towards the best position of the population g and the weight information for population is shown by ϕ_2. By equation $x_{t+1} = X_t + u_{t+1}$. The updated velocity is represented by u_{t+1} while the first term, u_t, displays the current velocity. By references [83, 84, 85], the above equation can be written as:

$$u_{t+1} - u_t = \phi_1(B-X) + \phi_2(g-X), \tag{4.17}$$

The Particle Swarm Optimization (PSO) algorithm takes into account the chaotic behavior of swarms, making fractional calculus tools useful for tracking previous swarm movements. The following expression results from setting the inertial weight in FO-PSO to $T = 1, w = 1$ using equation (4.15), and taking into consideration the work done in [93], the equation 4.17 can be written as:

$$D^{\alpha}[u_{t+1}] = \phi_1(B-X) + \phi_2(g-X). \tag{4.18}$$

For $r \geq 4$, the FO-PSO method yields identical empirical results. Additionally, the complexity of computation grows linearly and takes up $O(r)$ memory. Consequently, the 5^{th} term and subsequent terms are trimmed for speedier convergence. As a result, r is maintained at 4. When these four differential derivative terms are added, the velocity term in FO-PSO is equal to that in Eq. (4.19):

$$u_{t+1} - \alpha u_t - \frac{1}{2}\alpha u_{t-1} - \frac{1}{6}\alpha(1-\alpha)u_{t-2} - \frac{1}{24}\alpha(1-\alpha)(2-\alpha)u_{t-3} = \phi_1(B-X) + \phi_2(g-X), \tag{4.19}$$

Rearranging the equation we get the final form:

$$u_{t+1} = \alpha u_t + \frac{1}{2}\alpha u_{t-1} + \frac{1}{6}\alpha(1-\alpha)u_{t-2} + \frac{1}{24}\alpha(1-\alpha)(2-\alpha)u_{t-3} + \phi_1(B-X) + \phi_2(g-X). \tag{4.20}$$

Our new soft computing approach is based on the hybrid global search technique FO-PSO with the local search technique ASA. i.e., FO-PSO–ASA. In order to solve the Van der Pol heart model (4.1), ANNs' unknown adjustable parameter is trained using FO-PSO–ASA. PSO are regarded as one of the most reliable, robust, efficient, and accurate methods in the class of evolutionary computing algorithms that are widely applied in diverse fields of applied sciences.

As we select the hybrid form of FO-PSO, it causes an increase in the efficiency, accuracy, and reliability of our approach. Because the best velocities for our algorithm can be found in any order of velocities, In order to obtain a precise solution for linear and nonlinear systems based on

restricted and unconstrained problems, our method, FO-PSO, is a trusted optimization technique or mechanism. The flowchart of FO-PSO and illustration of ASA is shown in Fig. 4.3.

In order to swiftly get the desired results for an optimization problem, the hybridization technique with effective local search takes advantage of the real strength of FO-PSO optimization capabilities. The active set algorithm (ASA), which starts with the best FO-PSO individual, is used to quickly fine-tune the variables. In Fig. 4.3 the ASA flowchart is displayed. Many optimization issues that arise in various domains are successfully resolved by ASA, including those with hyperbolicity cone optimization, models of discrete-time infectious diseases with parameter estimation, nonlinear programming in nonconvex and many more. The above statements make ASA an attractive choice.

In order to solve the system model of heart model as provided in equation (4.1), a hybrid computing methodology called FO-PSO–ASA is utilised to identify suitable design variables of an ANNs model. This is done while taking into consideration the strengths of FO-PSO as a global method and ASA as a local, trustworthy method. These settings affect the proposed algorithm FO-PSO–ASA, and even a little adjustment could lead it to converge too soon. The parameters are set with care and through rigorous experimentation. Table 4.1 provides the pseudo-code for the design approach FO-PSO-ASA.

In short, the heart beat problem is solved through artificial neural networks trained with fractional order particle swarm optimization methodology. We formulated the proposed methodology of artificial neural networks (ANNs) while solving differential equations (our research problem). Also formulated the fitness function of differential equations. We discussed the optimization of absolute error for differential equations. Further, add the discussion of our optimization procedure for our hybrid technique of fractional order particle swarm optimization (FO-PSO) along with local search active set algorithm (ASA), i.e., FO-PSO-ASA. At the end, we formulated the equation related to FO-PSO. In the next chapter 5, we discussed the experiments, results, and discussion.

Table 4.1: FO-PSO–ASA algorithm

Start: *Fractional Order Particle Swarm Optimization Algorithms (FO-PSO)*

Inputs:

Weights W =[ϕ, ω,β]

Population P = [W_1, W_2 ,.... W_n] = $[(\phi_1, \omega_1, \beta_1), (\phi_2, \omega_2, \beta_2), \dots . (\phi_n, \omega_n, \beta_n)]^T$, for n is total number of W's in P and T is transpose operation

Output: *Best Individual W_b of FO-PSO*

Begin

//Initialization

Randomly generate $W's$ with bounded real values Set of n weights create initial population P.

//Termination Criteria (TC)

Set values for TC parameters as: Fitness limit i.e., ε ⟶ 10^{-15}

Function Tollerance i.e., TolFun ⟶ 0.

Constrained Tollerance, i.e.,TolCon ⟶ 0,

Number of generation for which fit ness not improve, i .e., StallGenLimit ⟶ ∞

// Main loop of FO-PSO

While {until TC is not fulfilled } do %

//Fitness evaluation step

Evaluate s values using equation (9) with ANN networks given in set (8) Repeat for α of Pop

// Check for TC

If TC fulfilled then end the loop otherwise continues

//Ranking step

Rank each Individual on minimum value of fitness t

//Reproducion step

Selection, crossover, mutations and Elitism operations

Elitism based on 4 with best ranking individuals of P

Call for Selection '@selectionuniform' function

Call for Crossover '@crossoverheuristic' function

Call for mutation '@mutationadaptfeasible' function

Update the P and proceed to step of fitness calculation step

End

//Storage step of FO-PSO

Store the global best individual W_b along with its, fitness execution time, generation consumed and function counts for this run of the FO-PSOs,

End FO-PSOs

Start ASA

//Initialization step

I nitialize ASA with W_b of FO-PSOs as a start points

Set the values of TC as follows

Fitness ⟶ 0,

Tolfun ⟶ 0,

TolCon ⟶ 0, and

Tolerance in optimization parametersm i .e., TolX ⟶ 0,

While {until TC is not fulfilled } do %

//Fitness evaluation step

Evaluate ε values using equation (9) with ANN networks given in set (8)

// Check for TC

If TC fulfilled then end the loop otherwise continues

// Update step

Invoking 'fminunc ' routine with algorithm 'ASA'

Update weights for each cycle update of ASA procedure.

Go to fitness evaluation step

// Storage step of FO-PSO

Store the global best individual W_b along with its, fitness execution time, generation consumed and function counts for this run of the FO-PSO--ASA,

End ASA

Repeat the procedure of FO-PSO--ASA for sufficient number of runs to generate a large data set for viable statistics

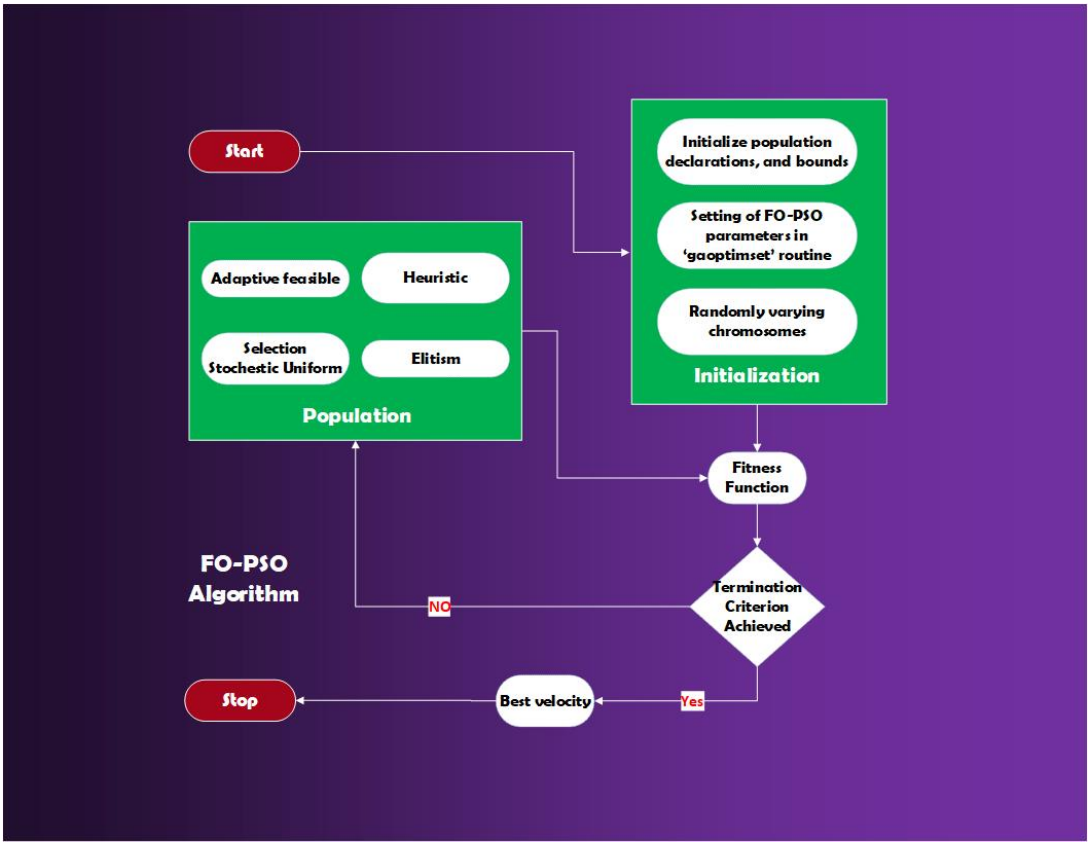

Figure 4.2: Flowchart of FO-PSO

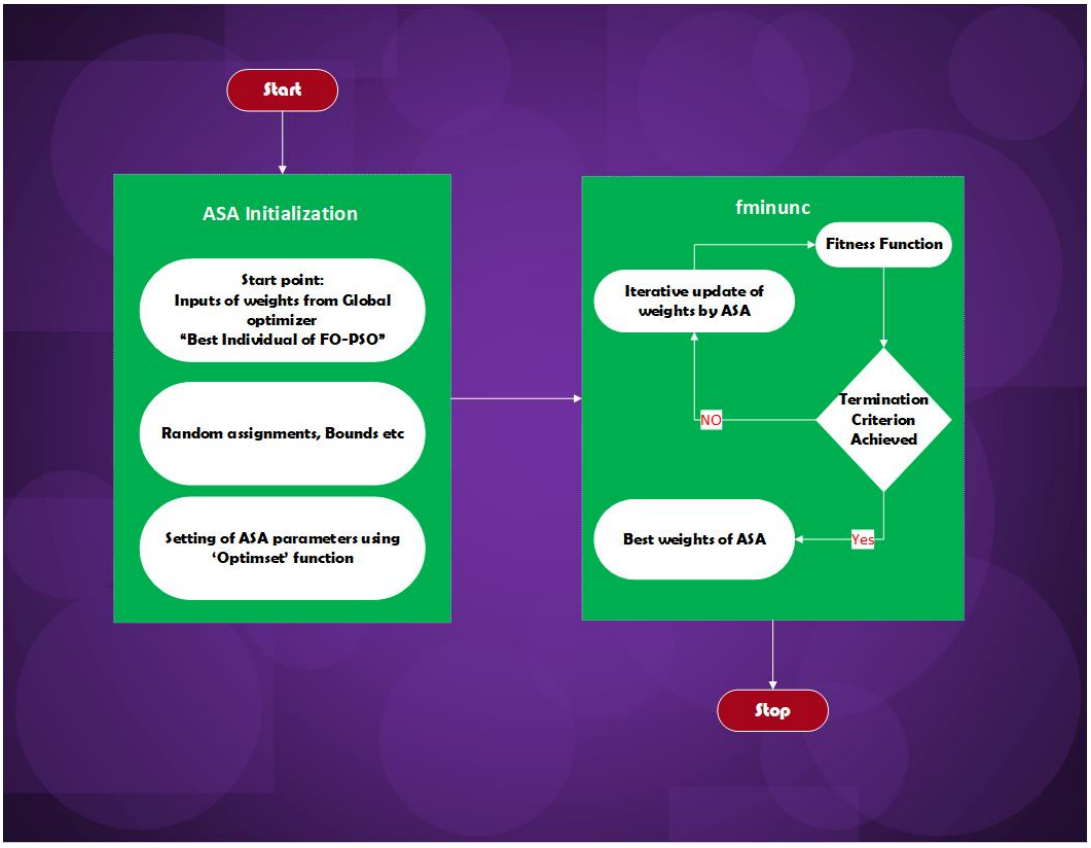

Figure 4.3: Illustration of the ASA

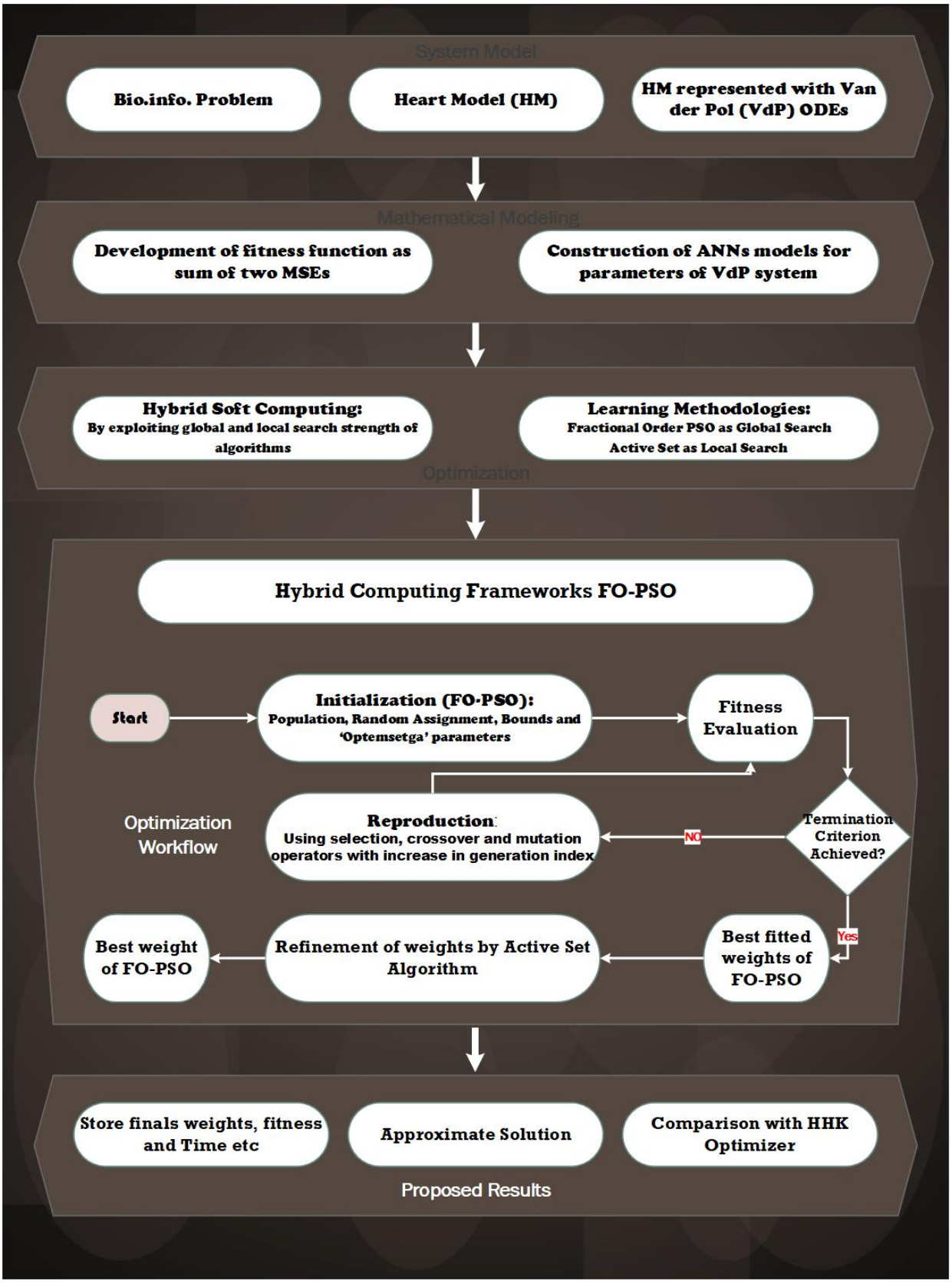

Figure 4.4: Flowchart of the designed technique for solving the VdP heartbeat model dynamics.

CHAPTER 5

NUMERICAL EXPERIMENTATION WITH DISCUSSION

Two case studies make up the results of the numerical experiments we did on the VdP equation of our proposed method. In the very first case study of the numerical experiment, we considered one problem with two scenarios, and each scenario has four cases. The second case study has the same problem: they contain two scenarios, and each scenario has four cases. Figure 5.1 depicts graphs of all of our case studies. The VdP equation is a non-linear ordinary differential equation (ODE) with initial conditions, which is stated as follows:

5.1 Problem 1; HBM with $F(t) = 0$

In this problem, we considered $F(t) = 0$ in model (4.1). In this problem, we analyzed the dynamics of HBM in the absence of an external force term, i.e., $F(t)$. In problem 1, we are considering two types of scenarios. In the 1^{st} scenario, we are making variations in $\tilde{\alpha}$, that is, the pulse shape modification factor, and in the 2^n scenario, we are varying the values of v_1 and v_2, which represent the asymmetric damping terms in the heartbeat model (4.1). We consider the other parameters to have constant values. This problem scenario is stated as follows:

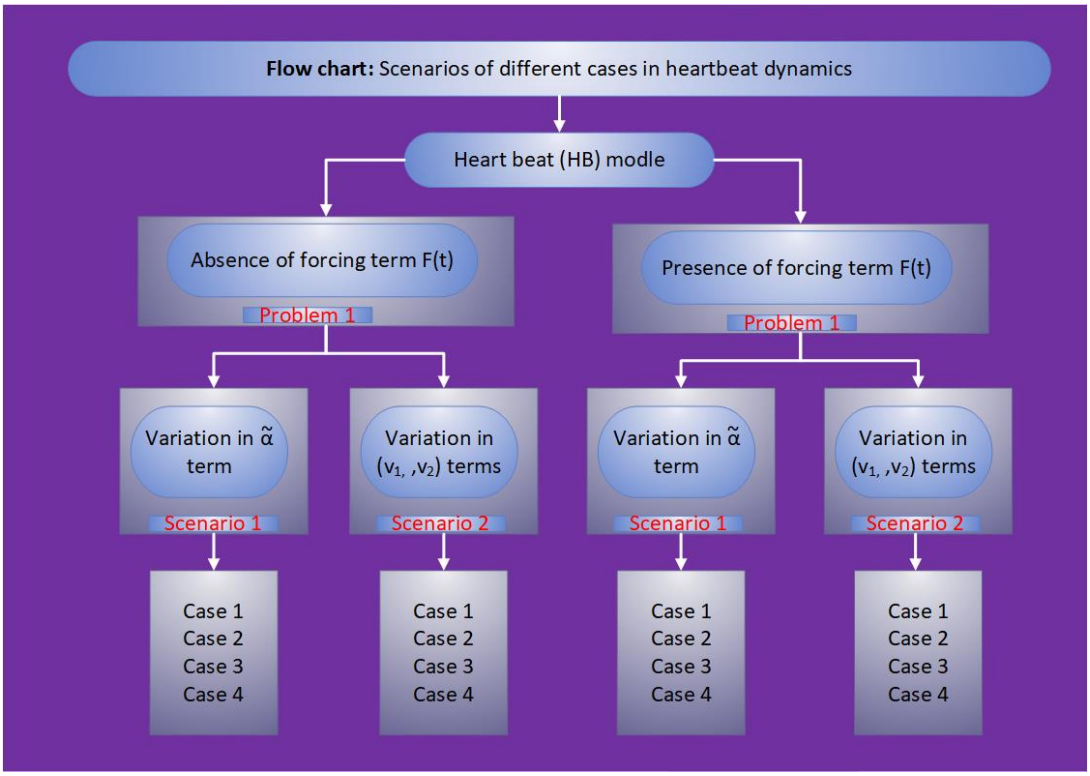

Figure 5.1: Flowchart of Problem 1,2

5.1.1 Scenario 1: variation in $\tilde{\alpha}$

In this scenario, we make a variation in the pulse shape modification factor, that is, $\tilde{\alpha}$ and consider fixed values for the remaining variables, as stated below:

Case1: $v_1 = 0.83$, $d = 3$, $v_2 = -0.83$, $e = 6$, and $\tilde{\alpha} = 3$,

Case2: $v_1 = 0.83$, $d = 3$, $v_2 = -0.83$, $e = 6$, and $\tilde{\alpha} = 2$,

Case3: $v_1 = 0.83$, $d = 3$, $v_2 = -0.83$, $e = 6$, and $\tilde{\alpha} = 1$,

Case4: $v_1 = 0.83$, $d = 3$, $v_2 = -0.83$,$e = 6$, and $\tilde{\alpha} = 0.01$.

Substituting these values in the equation (4.1), we get the form, as formulated below:

$$\begin{aligned} \ddot{x} + \tilde{\alpha}(x-0.83)(x+0.83)\dot{x} + \tfrac{(x+6)(x+3)}{18}x = 0. \\ x(0) = -0.1 \quad \text{and} \quad \dot{x}(0) = 0.025. \end{aligned} \tag{5.1}$$

In equation (5.1) $\tilde{\alpha}$ varies for values $\tilde{\alpha} = 3,2,1,0,01$. As we know that for equation 5.1 exact solution is not available. In this simulation process, we solved the four cases in scenario 1. The results of the hybrid genetic algorithm with the interior point algorithm (GA-IPA) were compared with our results to assess the accuracy level of our proposed technique. The next step

for our proposed scheme is to write the equation (5.1) in the form of a fitness function (FF) with the same domain [0,2] and an increment of $h = 0.1$. The fitness function is stated below:

$$\varepsilon = \frac{1}{N}\sum_{m=1}^{N}\Big(\hat{\dot{x}}_m + \tilde{\alpha}\,(\hat{x}_m - 0.83)\,(\hat{x}_m + 0.83)\,\hat{x}_m + \frac{\hat{x}_m\,(\hat{x}_m+6)\,(\hat{x}_m+3)}{18}\Big)^2 + \frac{1}{2}\Big((\hat{x}_0+0.1)^2 + \big(\hat{\dot{x}}_0 - 0.025\big)^2\Big). \tag{5.2}$$

Next step to optimize the equation (4.6-4.8) using our proposed algorithm FO-PSO and one set of FF-valued weights

$$\hat{x}_{c_1} = \begin{cases} \frac{15.3279}{1+e^{-(-0.3096t+4.3232)}} + \frac{5.9050}{1+e^{-(29.9998t+23.7594)}} \\ + \frac{-8.8026}{1+e^{-(-1.7310t+6.2122)}} + \frac{28.9849}{1+e^{(-26.3797t-39.8251)}} \\ + \frac{-30.1700}{1+e^{-(7.7927t-24.8558)}} + \frac{-0.0118}{1+e^{-(-51.8496t--0.8005)}} \\ + \frac{-15.5022284679284}{1+e^{-(30.0000t+28.4725)}} + \frac{3.1565}{1+e^{-(-8.9775t+28.9828)}} \\ + \frac{29.9843}{1+e^{-(-35.7825t-9.5631)}} + \frac{29.9995}{1+e^{-(6.6484t-30.0124)}}, \end{cases} \tag{5.3}$$

$$\hat{x}_{c_2} = \begin{cases} \frac{0.1339}{1+e^{-(-0.6756t+1.4881)}} + \frac{-1.5017}{1+e^{-(-0.3369t+3.0265)}} \\ + \frac{0.0075}{1+e^{-(-2.1802t-0.6363)}} + \frac{0.0688}{1+e^{-(1.1027t-2.4080)}} \\ + \frac{-0.6091}{1+e^{-(1.0731t-2.0099)}} + \frac{-0.2614}{1+e^{-(0.1408t-0.2667)}} \\ + \frac{-0.2756}{1+e^{-(-0.6184t-1.3405)}} + \frac{1.4370}{1+e^{-(-0.1136t+2.5645)}} \\ + \frac{0.4944}{1+e^{-(-1.2362t+5.0140)}} + \frac{0.4717}{1+e^{-(1.0161t--1.3526)}}, \end{cases} \tag{5.4}$$

$$\hat{x}_{c_3} = \begin{cases} \frac{-1.4091}{1+e^{-(-0.6858t-3.8223)}} + \frac{2.9797}{1+e^{-(-0.9719t-4.8931)}} \\ + \frac{2.5975}{1+e^{-(0.0782t-1.3661)}} + \frac{-1.8859}{1+e^{-(-0.3299t-3.7000)}} \\ + \frac{1.6800}{1+e^{-(0.9489t-2.3008)}} + \frac{-3.3085}{1+e^{-(0.9972t-4.7074)}} \\ + \frac{2.1867}{1+e^{-(1.3454t+4.3235)}} + \frac{-2.4459}{1+e^{-(0.9411t+3.3095)}} \\ + \frac{2.4692}{1+e^{-(-0.4934t-2.0018)}} + \frac{1.0490}{1+e^{-(1.0401t-3.1263)}}, \end{cases} \tag{5.5}$$

$$\hat{x}_{c_4} = \begin{cases} \frac{2.2978}{1+e^{-(1.2415t-2.5226)}} + \frac{1.9443}{1+e^{-(-0.7308t-1.0914)}} \\ + \frac{2.3700}{1+e^{-(-0.0995t+0.0553)}} + \frac{-0.7981}{1+e^{-(-1.8161t-3.8849)}} \\ + \frac{-2.0235}{1+e^{-(0.8352t+4.7106)}} + \frac{-1.5512}{1+e^{-(1.6637t+4.6148)}}. \\ + \frac{-3.6439}{1+e^{-(1.1124t-3.4057)}} + \frac{1.5489}{1+e^{-(1.9606t-4.6833)}} \\ + \frac{-0.2709}{1+e^{-(1.3279t-3.8548)}} + \frac{2.23711}{1+e^{-(0.6577t+1.1247)}}. \end{cases} \tag{5.6}$$

The graphical representation of our approximate solution was determined using the above equations (5.3-5.6) on a domain [0,2] with increment $h = 0.1$. shown in figure 5.3a and the

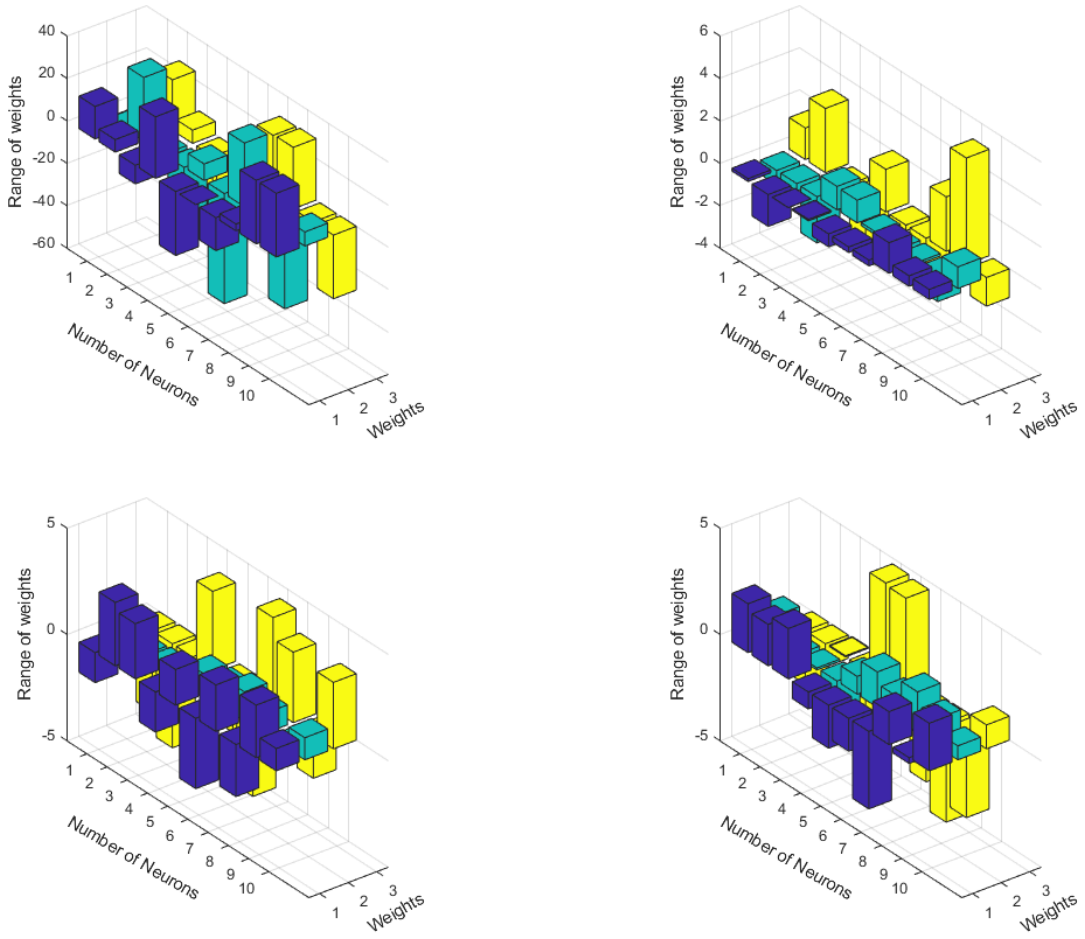

Figure 5.2: Trained weights for all the 4 cases; scenario 1, problem 1

absolute error (AE) graph is shown in figure 5.3b. It has been observed that the accuracy of our technique (FO-PSO) is $10^{-14} - 10^{-7}$ for cases 1, 2, 3, and 4, respectively; see table 5.1. For this step, we run our method (FO-PSO) 100 times in Matlab to find weights for our model. Our best-trained weights are plotted in figure 5.2. This helps us figure out how well our method converges and how accurate it is. We used standard deviation (STD) and minimum mean results for statistical analysis. Table 5.1 lists the comparison of the outcomes of the absolute error (AE) based on 100 independent runs of the hybrid GA-IPA and our proposed technique FO-PSO-ASA.

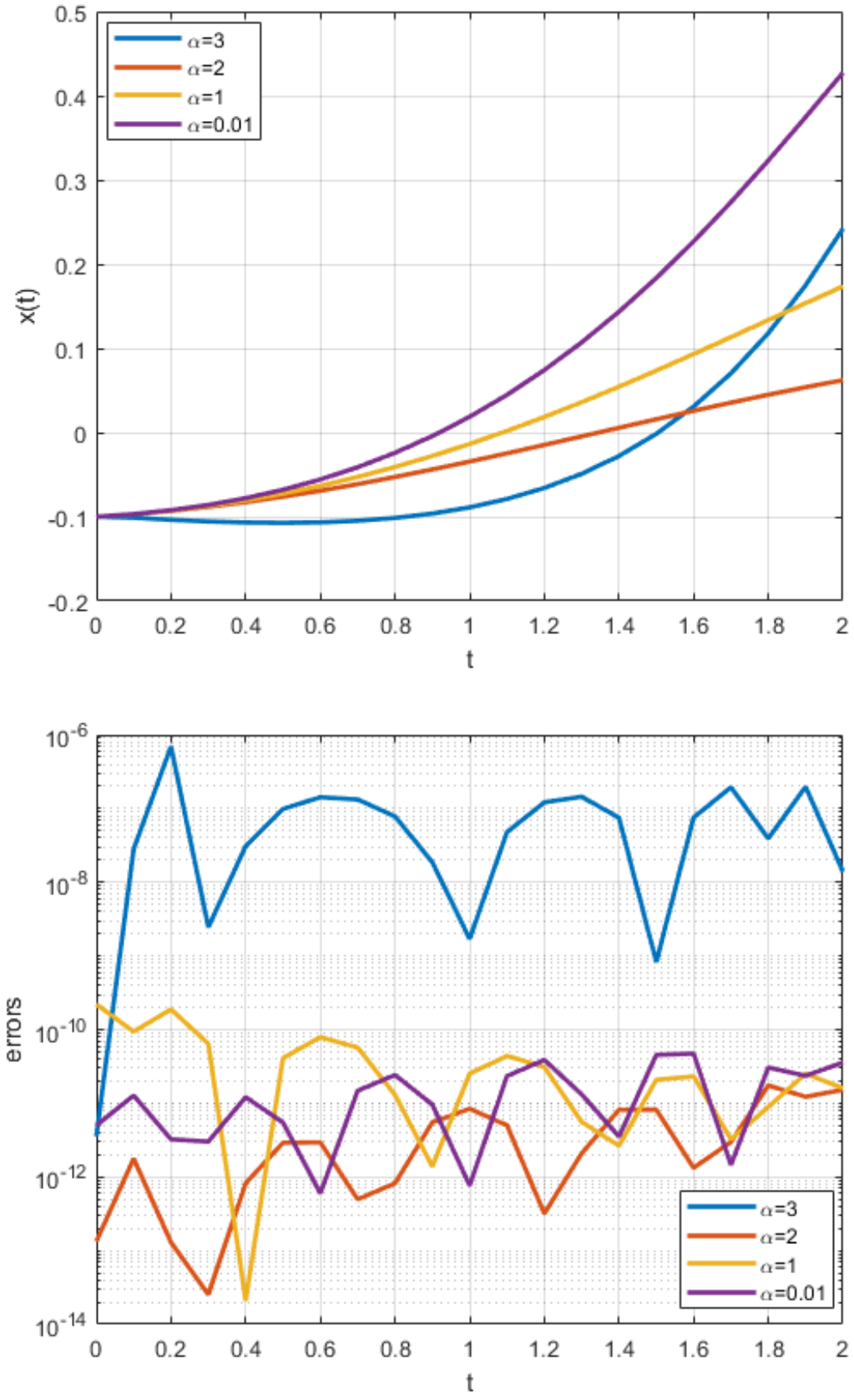

Figure 5.3: Solution and AE graphs for all the 4 cases; scenario 1, problem 1

Table 5.1: Comparison of min AE of scenario 1, problem 1 of the heart model

	Case 1		Case 2		Case 3		Case 4	
t	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)
0.0	1.98E-09	3.55E-12	8.44E-10	1.33E-13	7.76E-11	2.19E-10	1.00E-11	4.99E-12
0.1	4.19E-09	2.82E-08	7.23E-10	1.77E-12	7.81E-10	9.39E-11	8.09E-06	1.27E-11
0.2	3.29E-08	6.93E-07	1.28E-08	1.29E-13	5.49E-10	1.88E-10	3.75E-05	3.23E-12
0.3	1.31E-07	2.45E-09	9.55E-09	2.53E-14	1.63E-08	6.42E-11	9.25E-05	3.01E-12
0.4	1.14E-07	3.09E-08	1.31E-08	8.22E-13	3.90E-10	2.08E-14	1.79E-04	1.21E-11
0.5	9.84E-08	9.88E-08	2.68E-08	2.90E-12	3.51E-08	4.10E-11	3.02E-04	5.46E-12
0.6	4.57E-07	1.42E-07	1.34E-08	2.94E-12	7.45E-08	7.89E-11	4.65E-04	5.99E-13
0.7	7.04E-08	1.32E-07	8.11E-10	4.97E-13	9.36E-08	5.70E-11	6.71E-04	1.48E-11
0.8	3.60E-07	7.83E-08	2.59E-09	8.18E-13	1.19E -07	1.30E-11	9.24E-04	2.42E-11
0.9	4.73E-07	1.87E-08	1.87E-10	5.50E-12	7.15E-09	1.38E-12	1.22E-03	9.61E-12
1.0	6.36E-07	1.69E-09	1.93E-07	8.42E-12	6.51E-08	2.51E-11	1.57E-03	7.64E-13
1.1	3.08E-07	4.77E-08	1.97E-07	5.02E-12	8.95E-08	4.39E-11	1.96E-03	2.34E-11
1.2	6.62E-07	1.21E-07	3.11E-07	3.18E-13	6.34E-08	3.10E-11	2.40E-03	3.87E-11
1.3	3.03E-07	1.44E-07	3.43E-07	2.08E-12	2.93E-08	5.58E-12	2.88E-03	1.32E-11
1.4	1.30E-07	7.51E-08	1.96E-07	8.15E-12	2.19E-07	2.64E-12	3.40E-03	3.52E-12
1.5	2.40E-07	8.25E-10	1.85E-07	8.18E-12	3.66E-07	2.08E-11	3.95E-03	4.54E-11
1.6	2.98E-07	7.57E-08	3.68E-07	1.32E-12	5.43E-08	2.31E-11	4.52E-03	4.71E-11
1.7	5.18E-07	1.94E-07	5.58E-07	2.98E-12	1.94E-08	3.23E-12	5.11E-03	1.45E-12
1.8	2.25E-06	3.92E-08	5.51E-07	1.74E-11	2.97E-07	8.90E-12	5.72E-03	3.05E-11
1.9	1.81E-06	1.97E-07	3.65E-07	1.22E-11	2.70E-07	2.52E-11	6.33E-03	2.36E-11
2.0	7.62E-07	1.38E-08	4.36E-07	1.52E-11	2.54E-07	1.59E-11	6.92E-03	3.57E-11

5.1.2 Scenario 2: Variation in $v_1 v_2$

In this scenario, we are varying the asymmetric components (v_1, v_2) to analyze another dynamic of our heartbeat model.

Case1: $v_1 = 0.93$, $v_2 = -0.93$, $e = 6, d = 3$, and $\tilde{\alpha} = 2$.

Case2: $v_1 = 0.83$, $v_2 = -0.83$,$e = 6, d = 3$, and $\tilde{\alpha} = 2$.

Case3: $v_1 = 0.63$, $v_2 = -0.63$,$e = 6, d = 3$, and $\tilde{\alpha} = 2$.

Case4: $v_1 = 0.43$, $v_2 = -0.43$, $e = 6, d = 3$ and $\tilde{\alpha} = 2$.

Using above conditions, the heartbeat model 4.1 can be written as:

$$\begin{aligned} \ddot{x}+2(x-v_2)(x-v_1)\dot{x}+\tfrac{x(x+6)(x+3)}{18}=0, \\ x(0)=-0.1 \quad \text{and} \quad \dot{x}(0)=0.025. \end{aligned} \tag{5.7}$$

In this scenario, we make variations in the asymmetric damping components (v_1, v_2) in the above equation (5.7), while fixing the values of other parameters. The above equation (5.7) has no exact solution. Therefore, calculated numerical AM solutions are used as references for the results. applied in the same way as in the above scenario. But the FF for this section is noted below:

$$\begin{aligned} \varepsilon = \frac{1}{N}\sum_{m=1}^{N}\Big(\hat{\dot{x}}_m + 2\left(\hat{x}_m - v_1\right)\left(\hat{x}_m - v_2\right)\hat{x}_m \\ + \frac{\hat{x}_m\left(\hat{x}_m+6\right)\left(\hat{x}_m+3\right)}{18}\Big)^2 \\ + \frac{1}{2}\left(\left(\hat{x}_0+0.1\right)^2 + \left(\hat{\dot{x}}_0 - 0.025\right)^2\right), \end{aligned} \tag{5.8}$$

After this section, we find out the best weights for our approximate solution using the hybrid FO-PSO-ASA and then tune these weights in the series solution. The series solution for four cases is formulated below:

$$\begin{aligned} \hat{x}_{c_1} = \Bigg\{ & \frac{3.9180}{1+e^{-(3.1520t-4.3060)}} + \frac{-0.5103}{1+e^{-(3.0545t-0.2147)}} \\ & + \frac{1.8239}{1+e^{-(0.2630t-1.0223)}} + \frac{2.2897}{1+e^{-(2.6451t-2.6244)}} \\ & + \frac{2.8513}{1+e^{-(-0.1097t+1.0381)}} + \frac{-2.0137}{1+e^{-(-2.6539t+5.2125)}} \\ & + \frac{-0.3643}{1+e^{-(4.0960t+1.8291)}} + \frac{-6.5574}{1+e^{-(1.9996t-6.0663)}} \\ & + \frac{--5.7526}{1+e^{-(2.0457t-3.3746)}} + \frac{-2.0913}{1+e^{-(-2.8881t-2.5976)}}, \end{aligned} \tag{5.9}$$

$$\begin{aligned} \hat{x}_{c_2} = \Bigg\{ & \frac{-1.5586}{1+e^{-(-0.4589t-2.1229)}} + \frac{-0.0476}{1+e^{-(-0.4495t-0.2177)}} \\ & + \frac{-7.6070}{1+e^{-(1.9958t-6.0543)}} + \frac{-3.3697}{1+e^{-(-2.4299t+2.8337)}} \\ & + \frac{2.9401}{1+e^{-(3.1289t-4.4210)}} + \frac{4.9809}{1+e^{-(-0.2378t-5.0801)}} \\ & + \frac{-3.2737}{1+e^{-(-3.9565t-6.0022)}} + \frac{2.0875}{1+e^{-(-2.1929t+0.0056)}} \\ & + \frac{2.9932}{1+e^{-(1.34366t+1.1201)}} + \frac{-2.9876}{1+e^{-(2.48193t-3.6877)}}, \end{aligned} \tag{5.10}$$

$$\begin{aligned} \hat{x}_{c_3} = \Bigg\{ & \frac{-0.2666}{1+e^{-(0.5858t-0.8541)}} + \frac{1.5401}{1+e^{-(-1.3615t+2.8040)}} \\ & + \frac{-2.3245}{1+e^{-(-2.8163t-3.9490)}} + \frac{-0.3548}{1+e^{-(-3.4767t-1.8749)}} \\ & + \frac{-1.8780}{1+e^{-(-2.9173t+3.4448)}} + \frac{1.7181}{1+e^{-(3.2114t-4.7240)}} \\ & + \frac{-6.3073}{1+e^{-(2.1168t-6.3104)}} + \frac{0.6229}{1+e^{-(2.9093t-2.2264)}} \\ & + \frac{0.8699}{1+e^{-(-2.2375t-0.1870)}} + \frac{-3.0195}{1+e^{-(-2.0782t-4.9961)}}, \end{aligned} \tag{5.11}$$

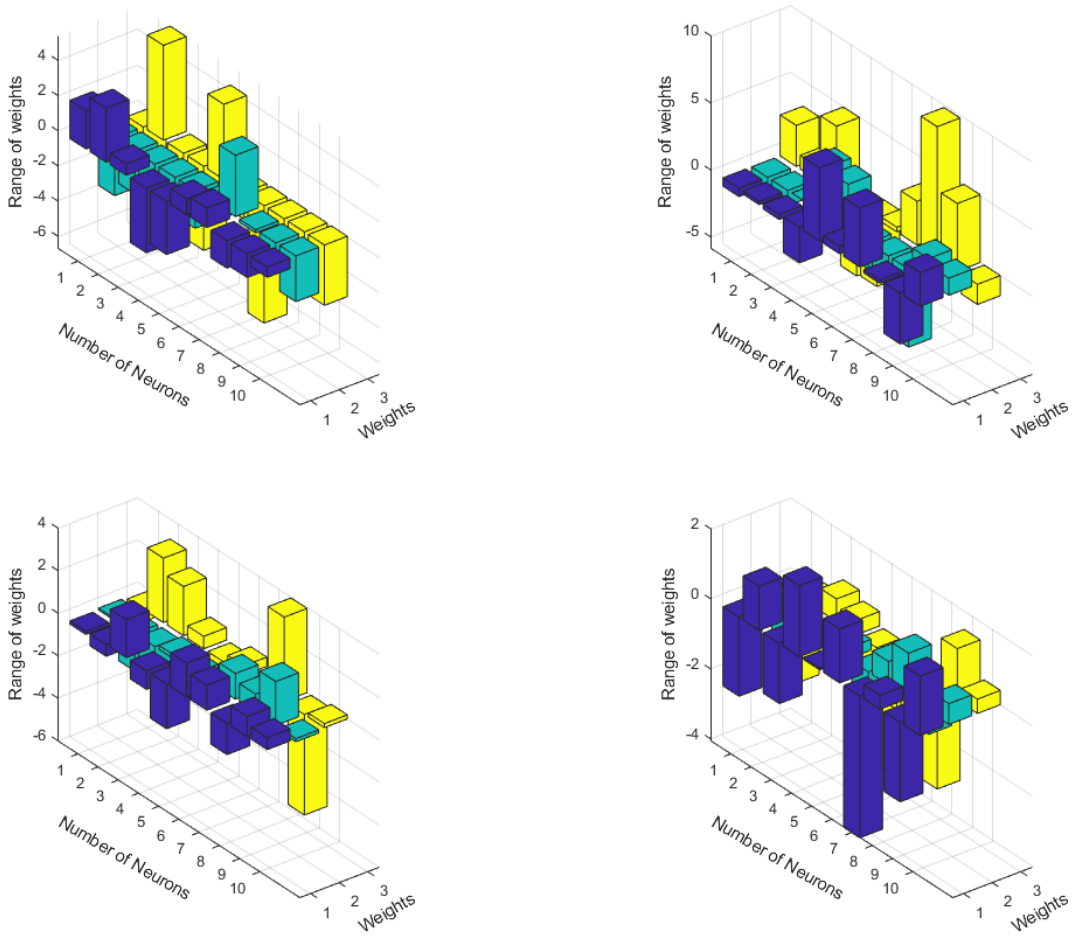

Figure 5.4: Trained weights for all the 4 cases; scenario 2, problem 1

$$\hat{x}_{c_4} = \begin{cases} \frac{-0.9901}{1+e^{-(2.9090t-0.2019)}} + \frac{-2.3957}{1+e^{-(-1.2450t-0.9180)}} \\ + \frac{0.8400}{1+e^{-(-1.1748t+2.5664)}} + \frac{-5.7366}{1+e^{-(1.4708t-5.3218)}} \\ + \frac{-0.2039}{1+e^{-(-0.0993t-0.8017)}} + \frac{0.1564}{1+e^{-(2.1576t-3.3206)}} \\ + \frac{-1.1670}{1+e^{-(0.9005t-1.6729)}} + \frac{-2.9374}{1+e^{-(0.6231t+5.1076)}} \\ + \frac{3.2784}{1+e^{-(-2.2409t+5.5770)}} + \frac{4.8947}{1+e^{-(2.3920t-3.3090)}}. \end{cases} \tag{5.12}$$

Equations (5.9-5.12) are used to get the proposed solution and absolute error (AE) for domain [0,1] with increment $h = 0.1$. The proposed solution and absolute error are graphically represented in Fig. 5.5. Our best-trained weights are plotted in Fig. 5.4. The minimum AE range is between $10^{-13} - 10^{-8}$ see the tale 5.2. The results of 100 separate runs of our proposed technique, the hybrid FO-PSO-ASA, are summarized in Table 5.2 which shows the comparison between our proposed technique's AE and the GA-IPA technique's AE. Fitness evaluation is represented in Fig. 5.6.

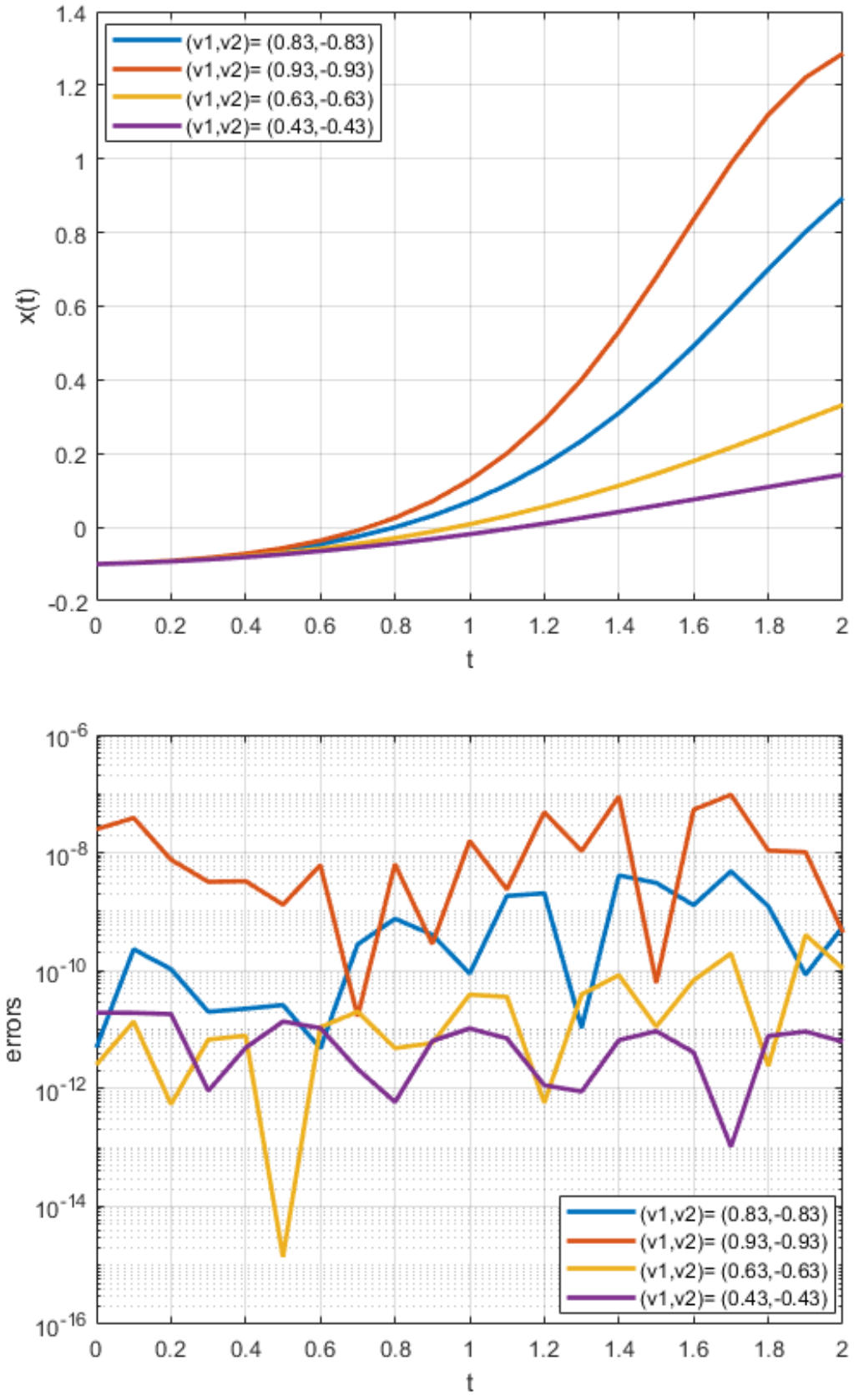

Figure 5.5: Solution and AE graphs for all the 4 cases; scenario 2, problem 1

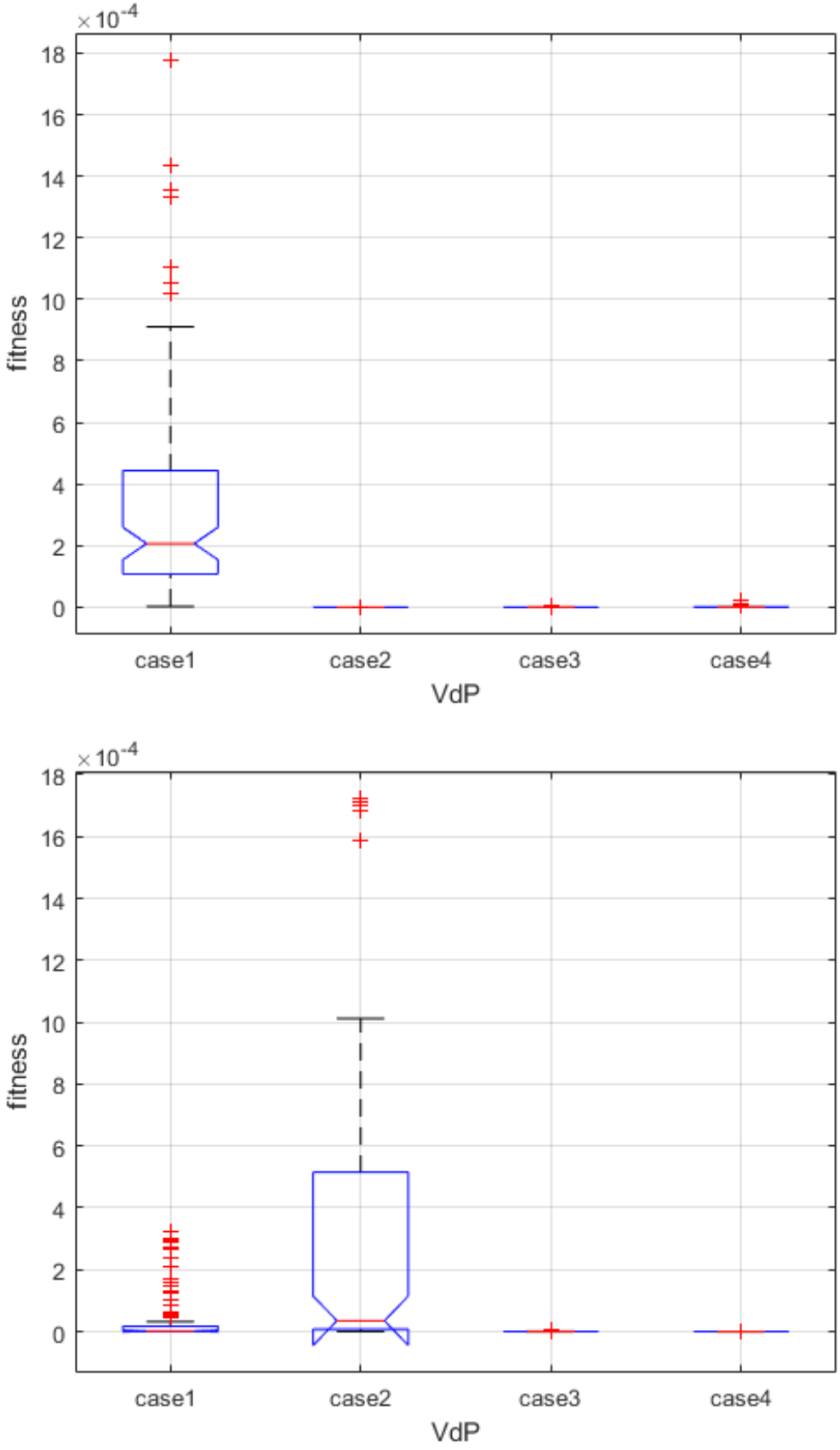

Figure 5.6: Fitness function graphs, problem 1

Table 5.2: Comparison of min AE of scenario 2, problem 1 of the heart model

	Case 1		Case 2		Case 3		Case 4	
t	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)
0.0	2.00E-09	4.95E-12	1.20E-10	2.50E-08	2.10E-11	2.53E-12	1.70E-11	1.93E-11
0.1	4.20E-09	2.30E-10	3.80E-10	3.90E-08	6.60E-10	1.37E-11	4.60E-11	1.92E-11
0.2	3.30E-08	1.05E-10	5.90E-09	7.63E-09	5.30E-09	5.42E-13	2.30E-10	1.83E-11
0.3	1.30E-07	2.00E-11	7.90E-09	3.21E-09	1.10E-08	6.76E-12	5.70E-09	9.15E-13
0.4	1.10E-07	2.26E-11	5.30E-09	3.29E-09	3.30E-08	7.87E-12	3.20E-10	4.92E-12
0.5	9.80E-08	2.60E-11	1.30E-08	1.31E-09	2.90E-08	1.38E-15	1.70E-08	1.37E-11
0.6	4.60E-07	4.78E-12	4.00E-08	6.19E-09	3.90E-08	1.09E-11	4.40E-09	1.06E-11
0.7	7.00E-08	2.79E-10	6.80E-08	1.67E-11	8.90E-09	2.01E-11	3.60E-09	2.13E-12
0.8	3.60E-07	7.59E-10	8.20E-08	6.41E-09	4.60E-08	4.84E-12	2.40E-09	5.90E-13
0.9	4.70E-07	4.07E-10	6.60E-08	2.84E-10	2.50E-08	5.90E-12	4.60E-09	6.44E-12
1.0	6.40E-07	8.92E-11	6.90E-08	1.58E-08	9.90E-08	3.91E-11	2.00E-08	1.04E-11
1.1	3.10 E-07	1.85E-09	5.10E-08	2.39E-09	2.10E-07	3.59E-11	2.90E-08	7.08E-12
1.2	6.60E-07	2.05E-09	4.70E-08	4.87E-08	1.80E-07	5.73E-13	3.30E-10	1.13E-12
1.3	3.00E-07	1.06E-11	6.40E-08	1.07E-08	7.60E-08	3.96E-11	2.00E-09	8.91E-13
1.4	1.30E-07	4.13E-09	9.40E-08	9.03E-08	3.30E-09	8.36E-11	5.50E-09	6.61E-12
1.5	2.40E-07	3.08E-09	1.20E-07	6.21E-11	3.50E-09	1.12E-11	3.30E-08	9.42E-12
1.6	3.00E-07	1.29E-09	1.10E-07	5.39E-08	5.40E-08	6.85E-11	2.40E-08	4.10E-12
1.7	5.20E-07	4.85E-09	8.50E-08	9.65E-08	8.80E-08	1.96E-10	2.10E-08	1.03E-13
1.8	2.30E-06	1.23E-09	6.80E-08	1.09E-08	2.10E-07	2.40E-12	9.50E-09	7.80E-12
1.9	1.80E-06	8.56E-11	7.50E-08	1.02E-08	1.90E-07	4.04E-10	6.00E-09	9.26E-12
2.0	7.60E-07	5.65E-10	5.10E-08	4.45E-10	1.10E-07	1.08E-10	7.30E-09	6.23E-12

5.2 Problem 2; Heartbeat (HB) dynamic Model with $F(t) \neq 0$

In problem 2, we analyzed two scenarios of the heartbeat model (HBM) (4.1). In this problem, we consider the forcing function $F(t) \neq 0$ in the model (4.1) and conduct the same procedure as we did in problem 1. In 1^{st} scenario, we vary $\tilde{\alpha}$ and take the constant value of other (v_1, v_2, e, d) parameters. The same procedure is conducted for the 2^{nd} scenario but only varies the value of v_1, v_2 in equation (4.1) and takes constant values for other $(\tilde{\alpha}, e, d)$ parameters.

While in this analysis the forcing term $F(t)$ is assumed to:

$$F(t) = A\sin(wt). \tag{5.13}$$

5.2.1 Scenario 1: Variation in $\tilde{\alpha}$

In this scenario, we analyze the dynamics of the HBM (4.1) by making variations in the pulse modification factor ($\tilde{\alpha}$). Four different cases for HBM (4.1) are listed below:

Case1: $v_1 = 0.97$, $\omega = 1.9$, $b = 2.5, e = 6$, $d = 3$, $v_2 = -1$, and $\tilde{\alpha} = 0.5$.
Case2: $v_1 = 0.97$, $\omega = 1.9$, $b = 2.5, e = 6, d = 3$, $v_2 = -1$, and $\tilde{\alpha} = 0.4$.
Case3: $v_1 = 0.97$, $\omega = 1.9$, $b = 2.5, e = 6, d = 3$, $v_2 = -1$, and $\tilde{\alpha} = 0.3$.
Case4: $v_1 = 0.97$, $\omega = 1.9$, $b = 2.5, e = 6, d = 3$, $v_2 = -1$, and $\tilde{\alpha} = 0.2$.

The HBM VdP equation (4.1) with initial condition is written bellow:

$$\begin{aligned} \ddot{x} + \tilde{\alpha}(x+1)(x-0.97)\dot{x} + \tfrac{x(x+6)(x+3)}{18} &= 2.5\sin(1.9t), \\ x(0) = -0.1 \quad \text{and} \quad \dot{x}(0) &= 0.025. \end{aligned} \tag{5.14}$$

We analyzed the dynamics of HBM by making variations in $\tilde{\alpha}$ from $0.5, 0.4, 0.3, and 0.2$ for cases 1, 2, 3, and 4, respectively. The equation (5.14) has no exact solution. This scenarios have the same pattern, which is applied to the above scenario. So, the fitness function (FF) is stated below:

$$\begin{aligned} \varepsilon = \frac{1}{N}\sum_{m=1}^{N} \Big(\dot{\hat{x}}_m + \tilde{\alpha}\left(\hat{x}_m - 1\right)\left(\hat{x}_m - 0.97\right)\Big(\dot{\hat{x}}_m \\ + \frac{\hat{x}_m\left(\hat{x}_m + 6\right)\left(\hat{x}_m + 3\right)}{18}\Big)^2 \\ + \frac{1}{2}\left(\left(\hat{x}_0 + 0.1\right)^2 + \left(\dot{\hat{x}}_0 - 0.025\right)^2\right). \end{aligned} \tag{5.15}$$

For one set of optimised weights with the same FF value as for four cases of the VdP model, the proposed solution is given below:

$$\hat{x}_{c_1} = \begin{cases} \frac{2.1726}{1+e^{-(-3.2809t-4.1302)}} + \frac{3.0586}{1+e^{-(-2.4326t+5.3891)}} \\ + \frac{0.6439}{1+e^{-(-1.4141t-1.5433)}} + \frac{-3.6723}{1+e^{-(-3.0796t-4.7784)}} \\ + \frac{-3.0205}{1+e^{-(-2.1545t+4.2737)}} + \frac{0.8840}{1+e^{-(-0.9828t-1.5020)}} \\ + \frac{1.1035}{1+e^{-(3.4842t-6.6746)}} + \frac{-1.5605}{1+e^{-(-0.1807t-1.6291)}} \\ + \frac{-1.3497}{1+e^{-(-1.0666t-1.9327)}} + \frac{-0.5944}{1+e^{-(-2.6004t-3.4302)}}, \end{cases} \tag{5.16}$$

$$\hat{x}_{c_2} = \begin{cases} \frac{-0.6180}{1+e^{-(-1.1150t+3.0545)}} + \frac{-0.2386}{1+e^{-(-1.1466t+1.0196)}} \\ + \frac{-0.3717}{1+e^{-(0.0108t+4.9489)}} + \frac{-2.6266}{1+e^{-(3.7084t-5.2430)}} \\ + \frac{5.4389}{1+e^{-(3.3932t-5.0487)}} + \frac{0.5789}{1+e^{-(-1.9580t+0.3469)}} \\ + \frac{4.3881}{1+e^{-(-1.8008t+3.2579)}} + \frac{-0.2476}{1+e^{-(-5.8220t+9.7978)}} \\ + \frac{-3.7829}{1+e^{-(1.7613t+5.0492)}} + \frac{2.4947}{1+e^{-(1.2763t-1.4992)}}, \end{cases} \tag{5.17}$$

$$\hat{x}_{c_3} = \begin{cases} \frac{-0.1436}{1+e^{-(0.0864t-0.5822)}} + \frac{-0.5525}{1+e^{-(-1.7690t+3.0262)}} \\ + \frac{1.7855}{1+e^{-(-0.9672t+2.3123)}} + \frac{-0.8736}{1+e^{-(0.1860t+0.5865)}} \\ + \frac{-2.1041}{1+e^{-(-0.3969t-0.7890)}} + \frac{1.5929}{1+e^{-(-0.5689t+0.4342)}} \\ + \frac{1.1744}{1+e^{-(1.2098t-1.3870)}} + \frac{-1.4695}{1+e^{-(0.8598t+3.9592)}} \\ + \frac{0.9422}{1+e^{-(2.0668t-4.7300)}} + \frac{0.5570}{1+e^{-(-0.1479t+0.1885)}}, \end{cases} \tag{5.18}$$

$$\hat{x}_{c_4} = \begin{cases} \frac{-2.2562}{1+e^{-(-0.4479t-2.5143)}} + \frac{1.2557}{1+e^{-(-0.8271t+0.1927)}} \\ + \frac{-1.7151}{1+e^{-(-0.4168t+0.6442)}} + \frac{2.0245}{1+e^{-(-0.1759t+0.4111)}} \\ + \frac{0.0505}{1+e^{-(0.3064t-3.5071)}} + \frac{1.5396}{1+e^{-(-0.9133t-1.3956)}} \\ + \frac{-4.0473}{1+e^{-(0.6725t-2.9291)}} + \frac{0.3933}{1+e^{-(21.2848t-2.9088)}} \\ + \frac{-2.2713}{1+e^{-(-0.6572t+01.4765)}} + \frac{1.6891}{1+e^{-(0.5921t+0.4085)}}. \end{cases} \tag{5.19}$$

We get our proposed solution using equations (5.16-5.19) for domain [0,2] with $h = 0.1$ to find the AE and our proposed approximate solution. Our A graphic representation of AE and the proposed solution are plotted in Fig. 5.8. We have witnessed the significantly high accuracy of our proposed outcomes in these cases. The weights 3D graph is shown in Fig. 5.7. The values of absolute error (AE) around $10^{-8} - 10^{-12}$ in the table 5.3 clearly indicated the convergence and accuracy of the outcomes of our technique. Moreover, 100 independent runs of FO-PSO-ASA yielded the statistical performance indicators inserted in table 5.5. In addition, minor values of the standard deviation (STD) are seen in every case. The comparison of our hybrid technique FO-PSO-ASA and another hybrid technique GA-IPA is inserted in table 5.3.

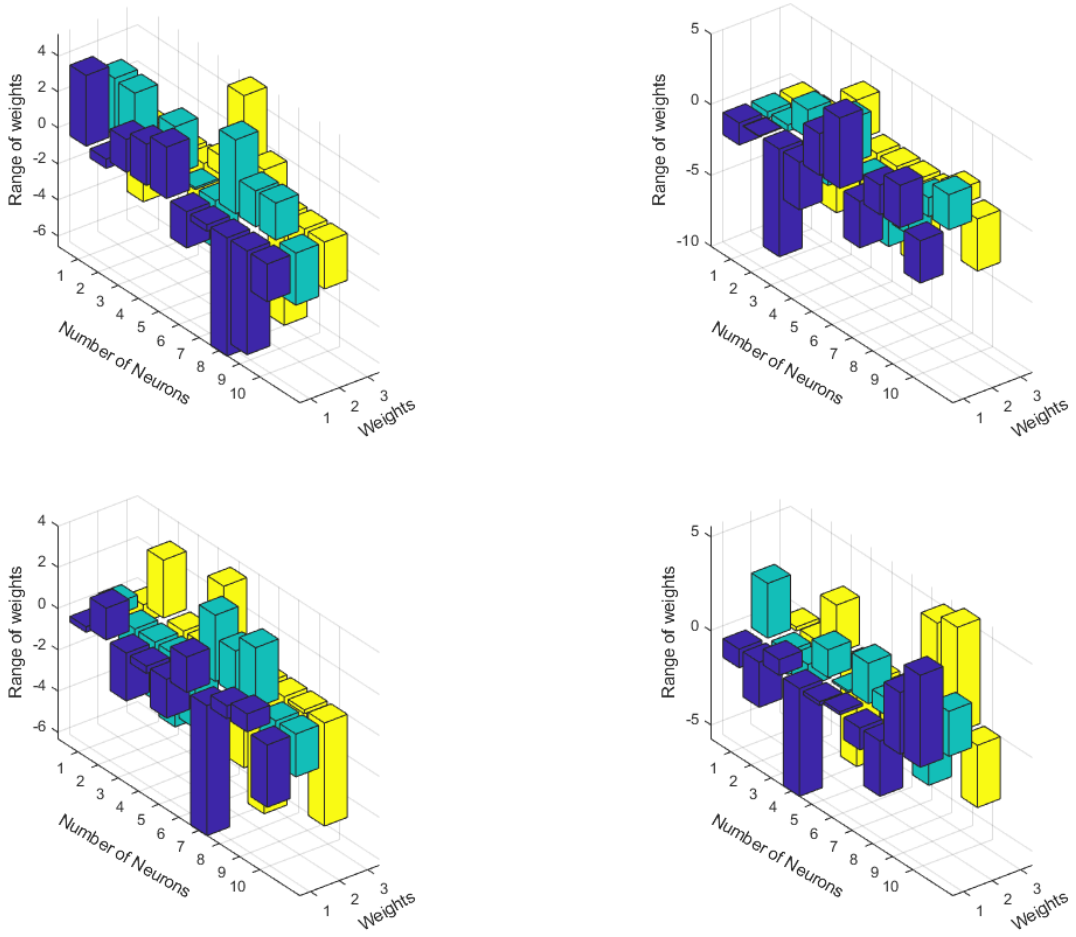

Figure 5.7: Trained weights for all the 4 cases; scenario 1, problem 2

5.2.2 Scenario 2: Variation in $v_1 v_2$

In this scenario of problem 2, we analyze the dynamics of the heartbeat model (HBM) (4.1) by making variations in the pulse modification factor (v_1, v_2) and keeping constant the values of other terms $(e, \tilde{\alpha}, d$. Four different cases for HBM (4.1) are listed below:

Case1: b = 2.5 $v_1 = 0.97$,$e = 6$, $d = 3$, $\omega = 1.9$, $v_2 = -1$, and $\tilde{\alpha} = 0.5$.

Case2: b = 2.5 $v_1 = 0.87$, $e = 6$, $d = 3$, $\omega = 1.9$, $v_2 = -3$, and $\tilde{\alpha} = 0.5$.

Case3: b = 2.5 $v_1 = 0.67$, $e = 6$, $d = 3$, $\omega = 1.9$, $v_2 = -4$, and $\tilde{\alpha} = 0.5$.

Case4: b = 2.5 $v_1 = 0.47$, $e = 6$, $d = 3$, $\omega = 1.9$, $v_2 = -5$, and $\tilde{\alpha} = 0.5$.

A derived Van der Pol (VdP) differential equation (DE) with initial conditions (ICs) in

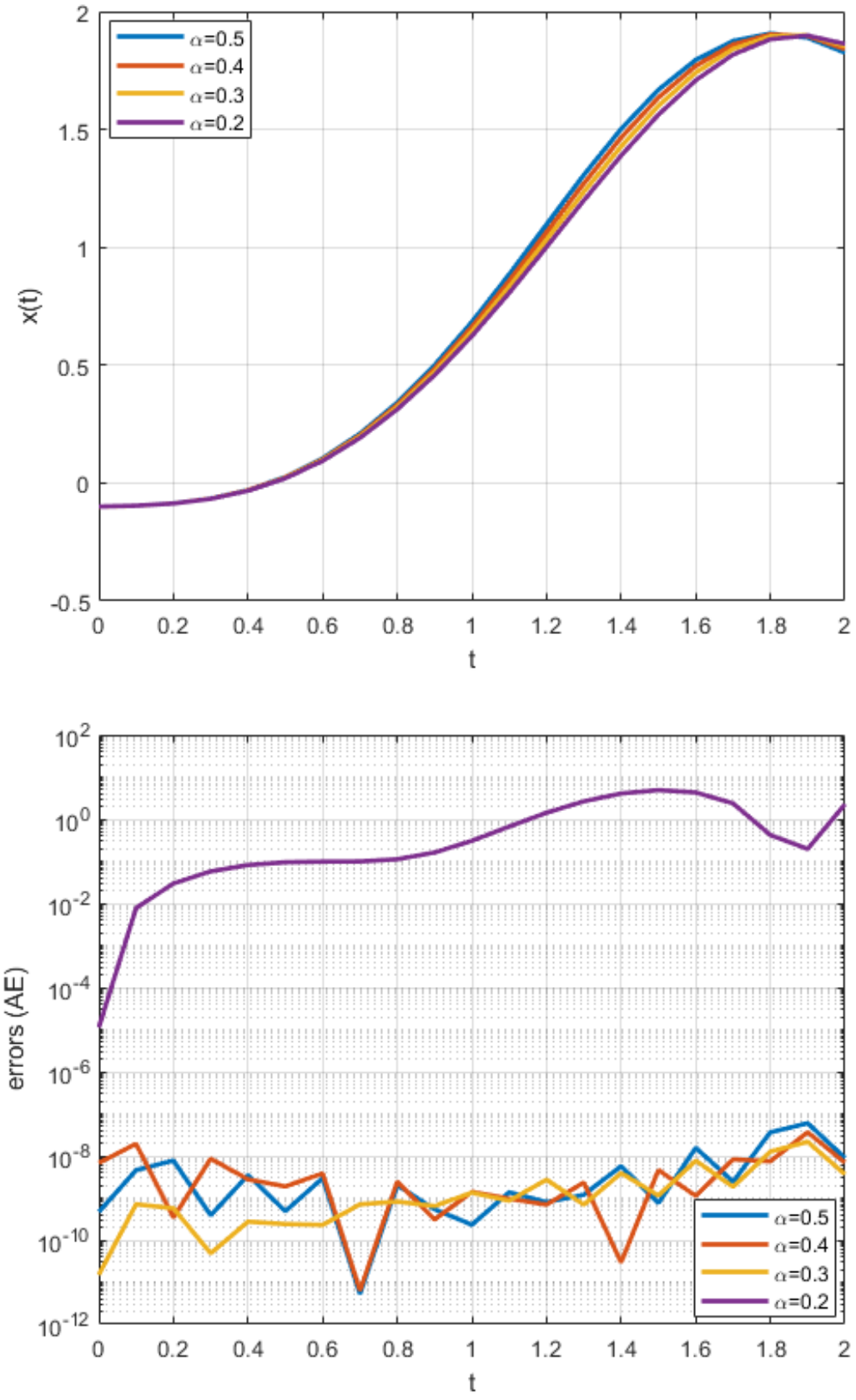

Figure 5.8: Solution and AE graphs for all the 4 cases; scenario 1, problem 2

light of the above circumstances or cases can be written in the form below:

$$\ddot{x} + .5(x - v_2)(x - v_1)\dot{x} + \frac{x(x+6)(x+3)}{18} = 2.5\sin(1.9 \times t), \quad x(0) = -0.1 \quad \text{and} \quad \dot{x}(0) = 0.025. \tag{5.20}$$

Table 5.3: Comparison of min AE of scenario 1, problem 2 of the heart model

	Case 1		Case 2		Case 3		Case 4	
t	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)
0.0	2.00E-09	4.65E-10	1.20E-10	6.84E-09	2.10E-11	1.50E-11	1.70E-11	1.11E-05
0.1	4.20E-09	4.58E-09	3.80E-10	1.92E-08	6.60E-10	7.10E-10	4.60E-11	0.007
0.2	3.30E-08	7.72E-09	5.90E-09	3.42E-10	5.30E-09	5.68E-10	2.30E-10	0.029
0.3	1.30E-07	3.87E-10	7.90E-09	8.59E-09	1.10E-08	4.85E-11	5.70E-09	0.057
0.4	1.10E-07	3.50E-09	5.30E-09	2.77E-09	3.30E-08	2.73E-10	3.20E-10	0.081
0.5	9.80E-08	4.84E-10	1.30E-08	1.87E-09	2.90E-08	2.40E-10	1.70E-08	0.094
0.6	4.60E-07	2.96E-09	4.00E-08	3.84E-09	3.90E-08	2.29E-10	4.40E-09	0.097
0.7	7.00E-08	5.28E-12	6.80E-08	6.31E-12	8.90E-09	7.06E-10	3.60E-09	0.098
0.8	3.60E-07	1.98E-09	8.20E-08	2.43E-09	4.60E-08	8.21E-10	2.40E-09	0.111
0.9	4.70E-07	5.41E-10	6.60E-08	3.07E-10	2.50E-08	6.42E-10	4.60E-09	0.160
1.0	6.40E-07	2.30E-10	6.90E-08	1.39E-09	9.90E-08	1.33E-09	2.00E-08	0.304
1.1	3.10 E-07	1.37E-09	5.10E-08	9.61E-10	2.10E-07	8.58E-10	2.90E-08	0.664
1.2	6.60E-07	7.97E-10	4.70E-08	6.97E-10	1.80E-07	2.72E-09	3.30E-10	1.415
1.3	3.00E-07	1.19E-09	6.40E-08	2.32E-09	7.60E-08	6.92E-10	2.00E-09	2.642
1.4	1.30E-07	5.69E-09	9.40E-08	2.95E-11	3.30E-09	3.91E-09	5.50E-09	4.058
1.5	2.40E-07	7.60E-10	1.20E-07	4.60E-09	3.50E-09	1.14E-09	3.30E-08	4.881
1.6	3.00E-07	1.55E-08	1.10E-07	1.12E-09	5.40E-08	7.71E-09	2.40E-08	4.297
1.7	5.20E-07	2.44E-09	8.50E-08	8.27E-09	8.80E-08	1.84E-09	2.10E-08	2.364
1.8	2.30E-06	3.62E-08	6.80E-08	7.47E-09	2.10E-07	1.26E-08	9.50E-09	0.418
1.9	1.80E-06	5.97E-08	7.50E-08	3.63E-08	1.90E-07	2.16E-08	6.00E-09	0.195
2.0	7.60E-07	8.96E-09	5.10E-08	6.89E-09	1.10E-07	3.64E-09	7.30E-09	2.255

In equation (5.20) we vary the values of asymmetric components v_1 , v_2 from (97, -1) (87,-3) (67, -4) (47, -5), for cases 1, 2 ,3 , 4 respectively, while keeping the values of $\tilde{\alpha}, d, and e$ constant. The exact solution of the above equation is not possible. For this scenario 2, we applied the same procedure as above in problem 2 scenario 1. Also using the same procedure, we derived the fitness function (FF) for scenario 2 of problem 2, which is stated below:

$$\varepsilon = \frac{1}{N}\sum_{m=1}^{N}\Big(\dot{\hat{x}}_m + \tilde{\alpha}\left(\hat{x}_m - 1\right)\left(\hat{x}_m - 0.97\right)\Big(\dot{\hat{x}}_m + \frac{\hat{x}_m\left(\hat{x}_m + 6\right)\left(\hat{x}_m + 3\right)}{18}\Big)^2 + \frac{1}{2}\Big(\left(\hat{x}_0 + 0.1\right)^2 + \left(\dot{\hat{x}}_0 - 0.025\right)^2\Big). \tag{5.21}$$

The above equation (5.21) is the fitness function (FF) for the Van der Pol (VdP) heartbeat model (4.1) under the case studies of scenario 2 of problem 2. Using the trained weights of this case study, we generate the equations stated below:

$$\hat{x}_{c_1} = \begin{cases} \frac{-0.7699}{1+e^{-(0.4062t-3.0756)}} + \frac{-1.7545}{1+e^{-(-2.7929t+2.8618)}} \\ + \frac{1.2037}{1+e^{-(-2.3263t+4.7310)}} + \frac{0.4745}{1+e^{-(-2.9802t+0.3020)}} \\ + \frac{-3.3257}{1+e^{-(-3.4729t-4.4040)}} + \frac{2.9330}{1+e^{-(3.3611t-4.5447)}} \\ + \frac{1.1171}{1+e^{-(-3.0159t+4.1018)}} + \frac{-1.0560}{1+e^{-(0.2411t+2.6528)}} \\ + \frac{4.1400}{1+e^{-(-0.1895t-4.9949)}} + \frac{-7.4376}{1+e^{-(2.4223t-7.7011)}}, \end{cases} \tag{5.22}$$

$$\hat{x}_{c_2} = \begin{cases} \frac{1.8118}{1+e^{-(3.9979t-4.6522)}} + \frac{-8.2619}{1+e^{-(1.9613t-6.5601)}} \\ + \frac{0.0196}{1+e^{-(-5.1951t+0.6021)}} + \frac{-1.7990}{1+e^{-(1.2795t-1.2709)}} \\ + \frac{-2.0077}{1+e^{-(0.4887t+1.4769)}} + \frac{2.3604}{1+e^{-(-0.5356t-2.4571)}} \\ + \frac{2.6796}{1+e^{-(2.6065t+3.0080)}} + \frac{-0.9617}{1+e^{-(0.1821t+3.6132)}} \\ + \frac{2.0357}{1+e^{-(3.0991t-2.8617)}} + \frac{-0.3740}{1+e^{-(3-2.1198t-4.8614)}}, \end{cases} \tag{5.23}$$

$$\hat{x}_{c_3} = \begin{cases} \frac{4.4180}{1+e^{-(-1.5198t-5.4495)}} + \frac{-1.6323}{1+e^{-(-3.3597t+3.1037)}} \\ + \frac{0.4365}{1+e^{-(-1.3097t+1.5445)}} + \frac{-2.9833}{1+e^{-(2.8321t+1.9532)}} \\ + \frac{3.3277}{1+e^{-(4.6426t+3.8022)}} + \frac{-0.5749}{1+e^{-(1.1840t-1.6403)}} \\ + \frac{-5.4185}{1+e^{-(2.0064t-7.4368)}} + \frac{-2.3747}{1+e^{-(1.9764t-5.7124)}} \\ + \frac{0.7107}{1+e^{-(4.1447t+1.0848)}} + \frac{1.3305}{1+e^{-(4.0857t-4.5109)}}, \end{cases} \tag{5.24}$$

$$\hat{x}_{c_4} = \begin{cases} \frac{-2.03418}{1+e^{-(1.8127t+1.0480)}} + \frac{1.9961}{1+e^{-(-14.2577t-30.2714)}} \\ + \frac{-3.4863}{1+e^{-(41.6965t+10.3994)}} + \frac{3.0349}{1+e^{-(1.6507t-3.9701)}} \\ + \frac{11.7822}{1+e^{-(-20.4480t-76.0752)}} + \frac{25.2156}{1+e^{-(-4.1055t-27.8810)}} \\ + \frac{-2.3653}{1+e^{-(-2.6490t-2.1974)}} + \frac{2.3798}{1+e^{-(-1.6103t+1.0441)}} \\ + \frac{-0.0247}{1+e^{-(-132.8579t+0.0174)}} + \frac{3.3249}{1+e^{-(103.4426t+49.3694)}}. \end{cases} \tag{5.25}$$

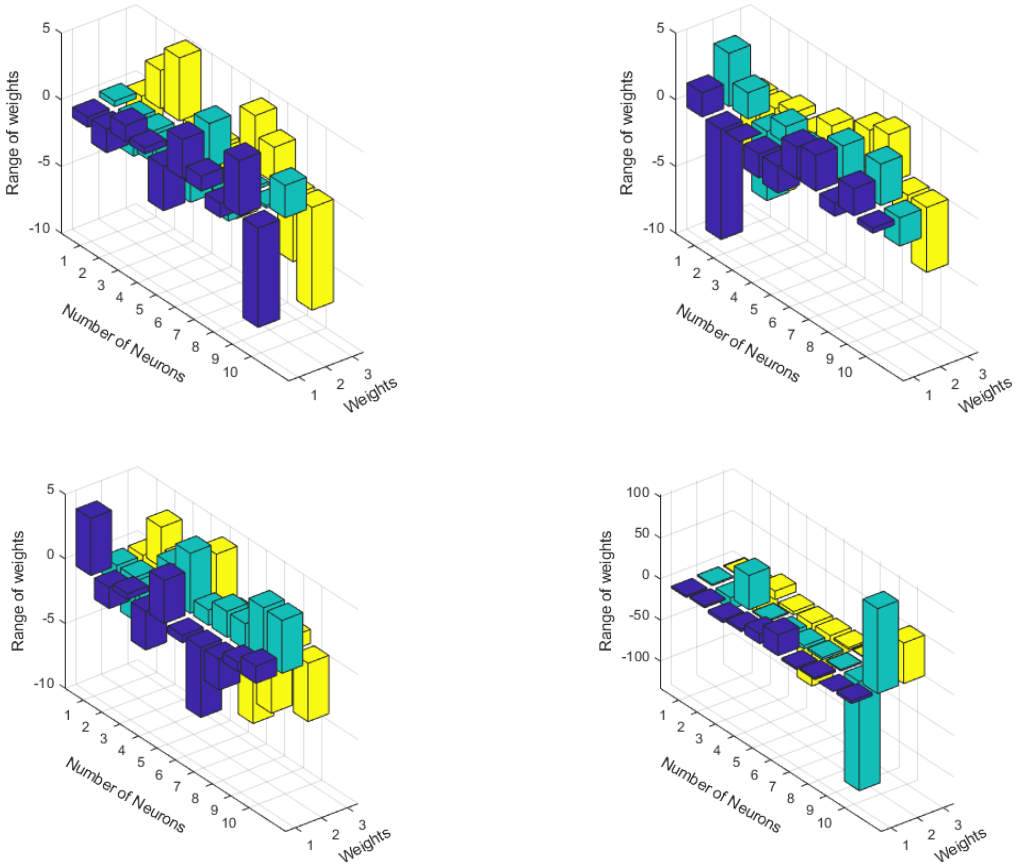

Figure 5.9: Trained weights for all the 4 cases; scenario 2, problem 2

Each solution for cases 1, 2, 3, and 4 in equation (5.22-5.25) contains thirty weights that we generated using Matlab. The trained weights are plotted in 3D graphs in Fig. 5.9. The minimum absolute errors (AE) for our solutions are represented in table 5.4. using domain [0,2] with increment $h = 0.1$. The proposed solution and AE graphs are shown in Fig. 5.11. Our AE is in the range of $10^{-8} - 10^{-13}$ The fitness values of the proposed solution are represented in table 5.4 These results are the results of the hybrid technique FO-PSO–ASA. Our results indicated that our technique has more accuracy and convergence.

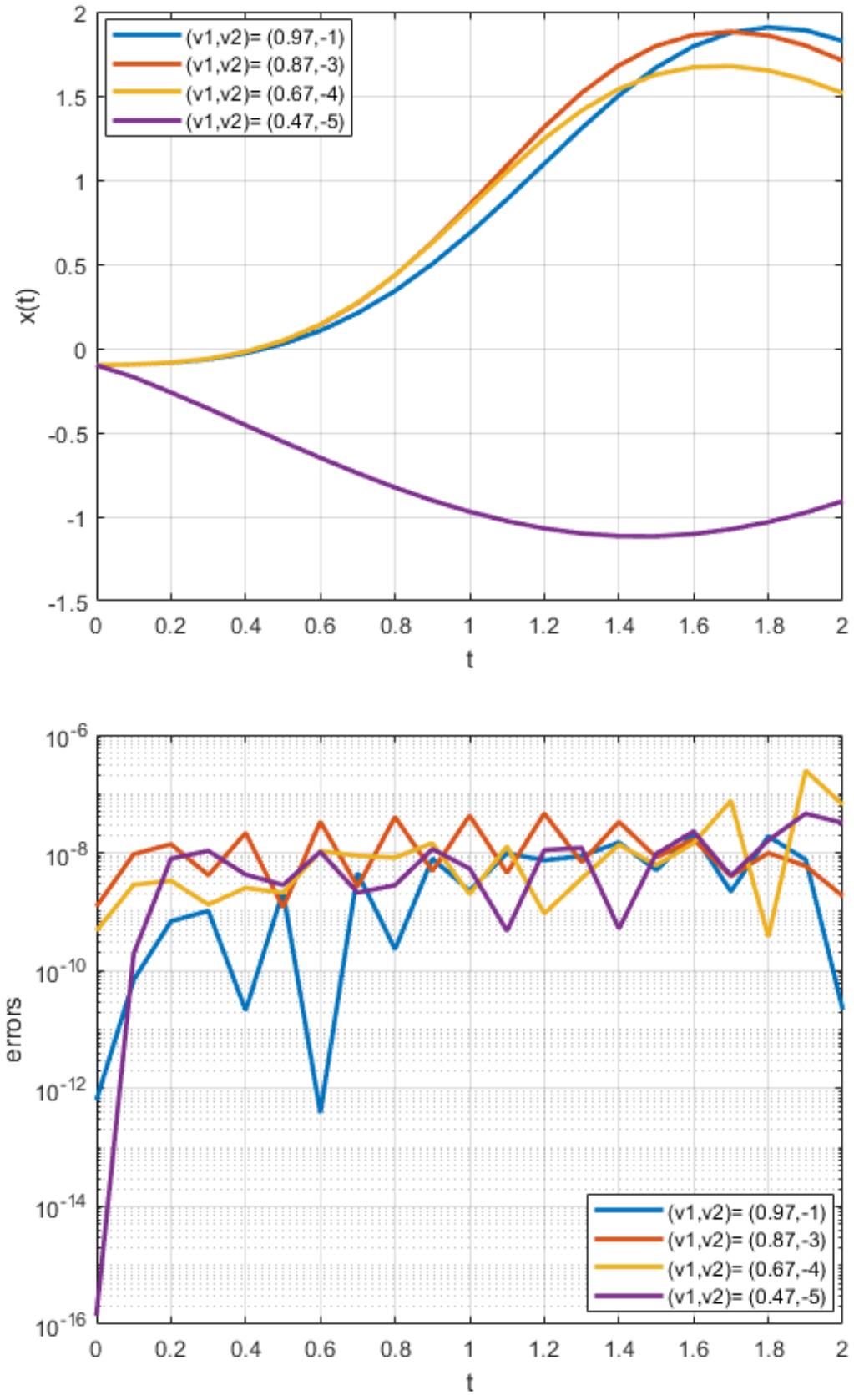

Figure 5.10: Solution and AE graphs for all the 4 cases; scenario 2, problem 2

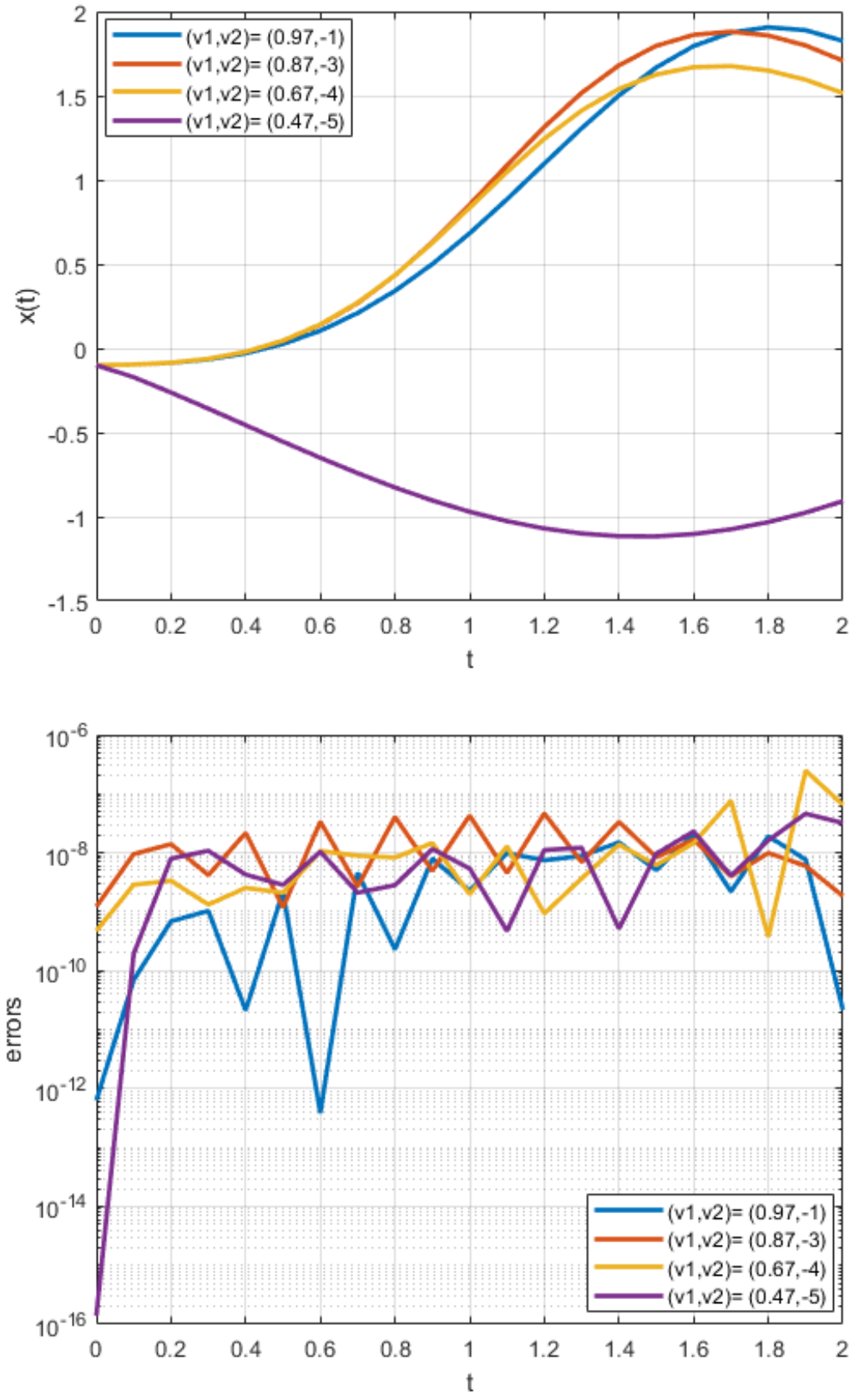

Figure 5.11: Solution and AE graphs for all the 4 cases; scenario 2, problem 2

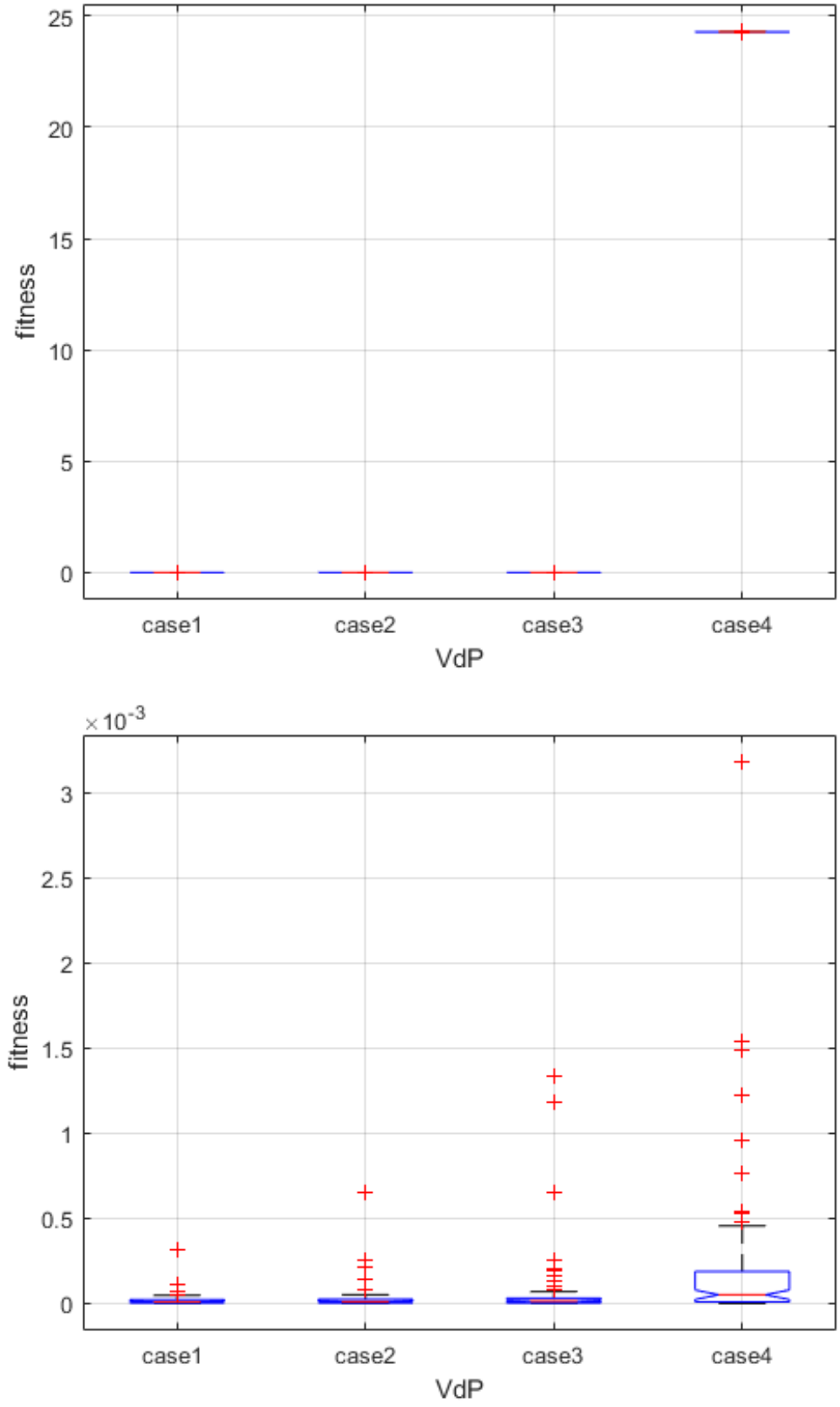

Figure 5.12: Fitness function graphs, problem 2

Table 5.4: Comparison of min AE of scenario 2, problem 2 of the heart model

	Case 1		Case 2		Case 3		Case 4	
t	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)	Min (GA)	Min(FO-PSO)
0.0	2.00E-08	6.32E-13	4.00E-10	1.22E-09	2.10 E-10	4.73E-10	5.10E-10	1.40E-16
0.1	3.00E-07	6.95E-11	1.50E-09	9.49E-09	2.90E-08	2.87E-09	1.40E-07	1.93E-10
0.2	9.80E-07	6.87E-10	4.70E-08	1.40E-08	1.20E-08	3.35E-09	2.50E-07	7.98E-09
0.3	4.60E-07	1.04E-09	3.30E-08	4.18E-09	4.60E-08	1.32E-09	1.70E-07	1.08E-08
0.4	8.40E-08	2.11E-11	3.40E-08	2.17E-08	1.20E-07	2.53E-09	1.90E-07	4.26E-09
0.5	6.40E-07	2.20E-09	4.60E-08	1.18E-09	4.30E-08	2.10E-09	7.70E-08	2.84E-09
0.6	1.10E-06	3.85E-13	1.20E-07	3.37E-08	1.70E-08	1.07E-08	1.30E-08	1.04E-08
0.7	1.90E-07	4.58E-09	1.00E-08	2.68E-09	1.30E-08	8.93E-09	4.90E-07	2.10E-09
0.8	2.40E-07	2.30E-10	1.80E-07	4.05E-08	3.40E-08	8.30E-09	7.40E-07	2.82E-09
0.9	3.10E-08	7.97E-09	3.30E-08	4.93E-09	1.40E-07	1.46E-08	1.10E-07	1.15E-08
1.0	8.50E-08	2.27E-09	2.10E-08	4.27E-08	1.50E-07	1.98E-09	8.50E-08	5.43E-09
1.1	8.10E-07	9.82E-09	4.70E-08	4.56E-09	2.10 E-07	1.29E-08	3.00E-07	4.71E-10
1.2	1.10E-06	7.46E-09	7.60E-08	4.64E-08	3.80E-08	9.29E-10	3.00E-07	1.11E-08
1.3	4.80E-07	8.83E-09	1.00E-08	6.94E-09	1.30E-08	3.67E-09	5.50E-01	1.21E-08
1.4	1.10E-07	1.50E-08	1.20E-08	3.34E-08	8.70E-08	1.38E-08	8.10E-07	5.15E-10
1.5	3.60E-07	5.10E-09	5.90E-08	8.42E-09	9.60E-08	6.18E-09	2.20E-07	9.66E-09
1.6	3.60E-07	2.06E-08	9.20E-08	1.68E-08	8.50E-08	1.46E-08	1.80E-07	2.30E-08
1.7	9.10E-07	2.20E-09	1.60E-08	4.00E-09	1.70E-07	7.63E-08	9.70E-07	4.10E-09
1.8	1.00E-06	1.90E-08	6.60E-09	9.95E-09	9.70E-08	3.76E-10	1.10E-06	1.62E-08
1.9	1.10E-06	7.67E-09	1.80E-09	5.99E-09	5.90E-09	2.48E-07	4.90E-07	4.61E-08
2.0	9.30E-08	2.18E-11	1.60E-07	1.84E-09	2.80E-08	6.50E-08	1.30E-06	3.20E-08

5.3 Analyses Comparative to Performance Indices

In this analysis, we study the comparison of the proposed technique, hybrid PO-PSO-ASA, in order to evaluate its convergence and accuracy. There are three performance indicators included to measure performance: root mean square error (RMSE), mean absolute error deviation (MAD), and error in Nash-Sutcliffe efficiency (ENSE). Mathematical formations for these RMSE, MAD, RMSE, and NSE are given below:

$$MAD = \frac{1}{N}\sum_{m=1}^{N}\left(\left|\hat{x}\left(t_i\right) - x\left(t_i\right)\right|\right), \tag{5.26}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{m=1}^{N}\left(\hat{x}\left(t_i\right) - x\left(t_i\right)\right)^2}, \tag{5.27}$$

$$NSE = 1 - \left(\frac{\sum_{i=1}^{N}\left(\hat{x}\left(t_i\right) - x\left(t_i\right)\right)^2}{\sum_{i=1}^{N}\left(x\left(t_i\right) - \frac{1}{N}\sum_{i=1}^{N}\left(x\left(t_i\right)\right)\right)^2}\right). \tag{5.28}$$

$$ENSE = \left|1 - NSE\right|. \tag{5.29}$$

For 100 independent runs, fitness, RMSE, ENSE, and MAD values for performance measures are computed, and the efficacy and consistency of the suggested computing approach are evaluated. See Figures 5.6 and 5.12 for semi-log scale depictions of MAD, RMSE, fitness, and ENSE, respectively, for 100 runs.

Observations reveal that MAD, RMSE, and ENSE values fluctuate directly with low-to-high fitness levels. MAD, ENSE, and RMSE data indicate that these variances are minimal for problem-1, scenario-1, and C_1. Moreover, in the instance in question, the value of these indicators is relatively diminished. In addition, the technique's dependability is assessed by calculating the proportion of converged runs based on the fitness, MAD, RMSE, and ENSE values. For each example, the converged runs (Cr) successfully for a hundred different simulations are calculated, and our outputs for both issues are given in table 5.5. These computations are performed according to the following conditions: $(C_{rFIT}) \leqslant E-07$, $(C_{rMAD}) \leqslant E-05$, $(C_{rRMSE}) \leqslant E-05$, and $(C_{rENSE}) \leqslant E-07$. It is intriguing to note that our scheme's average rate of convergence is about 100%.

The performance of our technique is evaluated by characterizing its efficacy using global metrics, such as global MAD, worldwide RMSE, global fitness, and global ENSE. The formulas for these factors are as follows:

$$Gb_{MAD} = \frac{1}{R_n}\sum_{r=1}^{R_n}\left(\frac{1}{G_p}\sum_{i=1}^{G_p}\left|\hat{x}_r\left(t_i\right) - x\left(t_i\right)\right|\right), \tag{5.30}$$

$$Gb_{RMSE} = \frac{1}{R_n}\sum_{r=1}^{R_n}\left(\sqrt{\frac{1}{G_p}\sum_{i=1}^{G_p}\left(\hat{x}\left(t_i\right) - x\left(t_i\right)\right)^2}\right), \tag{5.31}$$

$$Gb_{ENSE} = \frac{1}{R_n}\sum_{r=1}^{R_n}\left(\frac{\sum_{i=1}^{N}\left(\hat{x}\left(t_i\right) - x\left(t_i\right)\right)^2}{\sum_{i=1}^{N}\left(\frac{1}{N}\sum_{i=1}^{N}\left(x\left(t_i\right)\right) - x\left(t_i\right)\right)^2}\right), \tag{5.32}$$

$$Gb_{FIT} = \frac{1}{R_n} \sum_{r=1}^{R_n} \varepsilon_r, \tag{5.33}$$

where G_p represents the total no. of input values, where R_n is the total no. of runs, r is the objective value of the experiment, r^{th}, and $\hat{x}(t)$ and $x_r(t)$ are the standard solutions for the same number of runs. In this investigation, the inputs $t \in [0,2]$ with a step size of 0.1 are used, i.e., G_p is 21 whereas R_n is 100. In addition, the lower global performance indicator values for the majority of cases demonstrate the accuracy and consistency of FO-PSO.

The suggested algorithm is subjected to a computational complexity analysis (CCA) because of the average duration required for the computation of undetermined parameters of artificial neural networks by FO-PSO, the average number of populations, and function evaluation creation. The no. of function evaluations, mean population creation, and approximate time of 2045 seconds, respectively, for problem 1 The values for problem 2 are about 2111 seconds. All calculations and evaluations for this study were performed on a Dell Latitude E7450 laptop with an Intel(R) Core(TM) i5-5300U CPU @ 2.30GHz, 8.00 GB of RAM, 64-bit operating system, $\times$ 64-based processor, Microsoft Windows 10 Pro, and MATLAB R2021a.

Table 5.5: Study comparing several heart model modifications based on the values of global performance indices

Problem	Scenario	Cases	FIT		MAD		RMSE		ENSE	
			Mean	STD	Mean	STD	Mean	STD	Mean	STD
P1	S1	C_1	6.48E-09	7.22E-08	9.23E-07	9.98.E-07	6.65E-06	1.05E-05	9.96E-08	1.32E-06
		C_2	1.21E-09	9.96E-10	9.87E-07	4.21E-06	8.20E-06	7.56E-06	1.01E-08	1.09E-08
		C_3	8.03E-10	1.06.E-09	7.26E-07	8.26E-07	4.89E-06	1.24E-06	3.26E-09	8.27E-09
		C_4	4.23E-11	9.45E-11	6.85E-05	7.86E-07	9.43E-05	7.38E-07	9.97.E-05	2.51E-06
	S2	C_1	5.86E-09	7.19E-08	4.53E-05	9.11E-06	2.71E-06	4.55E-05	3.24E-07	9.07E-08
		C_2	9.21E-10	9.72E-10	9.12E-07	9.90E-07	5.12E-06	2.34E-06	1.03.E-08	1.10E-08
		C_3	3.11E-10	8.46E-09	5.93E-06	8.66E-07	8.36E-07	1.29E-06	2.06E-09	4.28E-09
		C_4	6.56E-11	2.44E-09	6.92E-07	3.42E-07	7.92E-07	9.83E-07	3.43E-10	7.82E-10
P2	S1	C_1	1.01E-06	4.69E-06	9.67E-06	5.18E-05	9.26E-05	1.32E-04	2.61E-05	9.98E-06
		C_2	7.32.E-07	8.79E-06	2.01.E-05	8.09E-06	2.21E-05	8.89E-05	7.56E-06	1.02E-04
		C_3	1.45E-07	9.68E-08	1.04E-06	9.01E-06	3.44E-06	7.84E-06	1.22E-08	2.36E-07
		C_4	1.64E-05	1.09E-05	6.21.E-04	2.30E-06	3.90E-05	8.32E-06	5.06E-09	6.77E-07
	S2	C_1	1.01E-06	6.88E-06	7.11E-05	3.81E-05	1.61E-04	8.36E-04	1.23E-05	6.80E-04
		C_2	1.76E-07	7.46.E-07	2.43E-06	7.11E-05	5.36E-06	6.01E-06	2.34E-08	8.32E-07
		C_3	9.17.E-09	1.01.E-08	7.56E-07	9.60E-07	9.89E-07	7.76E-07	1.30E-09	3.96E-10
		C_4	2.23E-08	6.67E-08	3.32E-06	8.21E-06	6.56E-06	6.99E-06	5.87E-10	4.01E-08

In short, we discussed the mathematical structure of neural networks while solving our problem (the heartbeat model). Here we consider two main problems. In problem 1, we considered the external force term $F(t) = 0$, and for problem 1, we considered two main scenarios, each scenario having four different cases. In the first scenario, we vary pulse modification factors ($\tilde{\alpha}$ and in the second scenario, we vary asymmetric terms for the modification of damping terms ($v_1 and v_2$). In problem 2, we considered the forcing term, i.e., $F(t) = 2.5sin(1.9t)$, and also considered two scenarios. Each scenario has four cases, and then the same procedure is followed as in problem 1. We also inserted graphs of trained weights for each case, graphs of solutions for each scenario, fitness value graphs, and four results containing tables for each scenario. In the tables, we show the absolute error (AE) comparison between our hybrid technique (FO-PSO–ASA) and hybrid genetic algorithm (GA-IPA) to assess the accuracy of our proposed technique. Also, we show the statistical analysis of our proposed methodology.

CHAPTER 6

CONCLUSION

Recent studies about the development of different ANN methods to solve differential equations (DEs) shows that the ANNs methodology is a very convenient methodology. Each technique of ANN represents a novel way to solve DEs. In order to take advantage of these methods' characteristics, this may be very different from more conventional numerical techniques. This chapter of our thesis contains conclusions on the basis of our numerical experiments, which are listed below:

A new approach inspired biologically by fractional order velocity is designed to analyze the Van der Pol (VdP) equation-based heartbeat model using the robustness of series solution and feed-forward (FF) artificial neural networks (ANNs). Then carried the optimization process uses hybrid search techniques, that is a combination of the global search technique fractional order particle swarm optimization (FO-PSO) and local search technique active search algorithm (ASA), which can be written as FO-PSO–ASA.

Our approximate solution of the Van der Pol (VdP) heartbeat model is seen to be in excellent accord with the cited hybrid genetic algorithm (GA) as a global and interior point technique, written as GA-IPA for each case of problems 1 and 2. The suggested hybrid algorithm's performance is evaluated using the standard deviation (STD) and mean values of statistical markers, and the absolute error (AE) based results are analyzed, The AE is almost very close to zero for all cases of problems 1 and 2.

Analyzing the performance of our proposed technique or algorithm is evaluated using the standard deviation (STD) and mean values of statistical indicators, and results based on AE

are almost consistently found to be close to zero for both VdP heartbeat models, demonstrating the scheme's consistency in accuracy.

The suggested approach is computationally analyzed using efficiency operators based on mean execution time values, the maximum number of function evaluations, and the number of iterations the optimization algorithm uses to determine the design parameter of ANN models of VdP systems, and results indicate that these parameters do not differ noticeably, which alleviates the algorithm's constant action in each case.

Convergence and accuracy of the proposed algorithm are proven by reaching ideal values of several performance metrics based on ENSE, RMSE, and MAD, as well as their global equivalents GbENSE, GbRMSE, and GbMAD, in both VdP-based heart dynamics investigations.

In short, in this research study, we discussed our findings and assess how accurate, robust, efficient, and convergent our proposed hybrid technique (FO-PSO-ASA) is. We also discussed the applications of this methodology in different fields of study. we also presented our contribution to this research work and also discussed our future work.

6.1 Contribution

The following are some of the study's numerous contributions that show the dynamics of the HB model:

An FFNN-based integrated technique is created to model the proposed problem. Set the order of velocity $0 < \alpha \leq 1$ of moving particles in a global search technique i.e. particle swarm optimization. The combination is FO-PSO. To optimize FO-PSO, the mean squared error (MSE) is minimized as the fitness objective function.

To validate the efficacy and robustness of the proposed algorithm, the optimized solutions of FO-PSO are compared with the numerical results of a genetic algorithm hybrid with an interior point algorithm (GAs-IPA) approach. The statistical and quantitative analyses are established to validate FO-PSO's performance.

6.2 Future Work

A fast-growing field of study is the application of artificial neural networks (ANNs) to solve real-world problems containing differential equations. Despite the recent considerable advancements, there are still numerous chances for future work in this area. Here are some potential future research topics that could be covered in my Ph.D. thesis or research articles on employing ANN approaches to solve differential equations:

Improving accuracy: Achieving high levels of accuracy is one of the main challenges when utilizing ANNs to solve differential equations. The design of more effective computational algorithms, the application of new activation functions, the incorporation of regularisation techniques, and other methods of improving accuracy could all be explored in future studies.

Handling complex systems: The creation of techniques for employing ANNs to resolve complex systems of differential equations is a significant topic of future research. This can entail creating new architectures that can process several equations concurrently or looking into how to include extra data, like boundary conditions or beginning values.

Dealing with high-dimensional data: High-dimensional data is a common component of differential equations, which presents difficulties for conventional ANN structures. The usage of neural networks could be one of the new methods for processing high-dimensional data that are investigated in future research.

Dealing with fractional-order differential equations: The differential equations having fractional order are one of the latest research projects. We can consider it in our future work. We can apply ANN techniques to solve fractional-order partial differential equations (PDEs).

Applications of ANN in biomathematics: This research area is the most prominent and novel one. We can consider ANN as a tool for evaluating and optimizing the spread of different health diseases. A study on applying ANN techniques to solve differential equations can improve the development of a novel new method for solving challenging mathematical issues by examining these and other areas of research.

In short, The following are prospective directions for future research in the application of artificial intelligence algorithms to complex biomedical engineering challenges. Robustness, efficiency, accuracy, and convergence can be enhanced for the VdP heart model by adding either recently established optimization mechanisms based on complex software packages or the availability of powerful computer hardware platforms. Design methodology can also be utilized

to produce dependable answers for mathematical models of treatment and diagnostic issues for HIV infection, brain tumors, breast cancer research, etc.

Appendices

APPENDIX A

SERIES SOLUTION OF OUR PROPOSED TECHNIQUE

Our derived series solution for VdP heartbeat model using propose algorithm FO-PSO–ASA for problem 1 is written below:

A.1 Scenario 1, All Cases, Problem 1

$$\hat{x}_{c_1} = \begin{cases} \frac{15.3279}{1+e^{-(-0.3096t+4.3232)}} + \frac{5.9050}{1+e^{-(29.9998t+23.7594)}} \\ + \frac{-8.8026}{1+e^{-(-1.7310t+6.2122)}} + \frac{28.9849}{1+e^{(-26.3797t-39.8251)}} \\ + \frac{-30.1700}{1+e^{-(7.7927t-24.8558)}} + \frac{-0.0118}{1+e^{-(-51.8496t--0.8005)}} \\ + \frac{-15.5022284679284}{1+e^{-(30.0000t+28.4725)}} + \frac{3.1565}{1+e^{-(-8.9775t+28.9828)}} \\ + \frac{29.9843}{1+e^{-(-35.7825t-9.5631)}} + \frac{29.9995}{1+e^{-(6.6484t-30.0124)}}, \end{cases} \tag{A.1}$$

$$\hat{x}_{c_2} = \begin{cases} \frac{0.1339}{1+e^{-(-0.6756t+1.4881)}} + \frac{-1.5017}{1+e^{-(-0.3369t+3.0265)}} \\ + \frac{0.0075}{1+e^{-(-2.1802t-0.6363)}} + \frac{0.0688}{1+e^{-(1.1027t-2.4080)}} \\ + \frac{-0.6091}{1+e^{-(1.0731t-2.0099)}} + \frac{-0.2614}{1+e^{-(0.1408t-0.2667)}} \\ + \frac{-0.2756}{1+e^{-(-0.6184t-1.3405)}} + \frac{1.4370}{1+e^{-(-0.1136t+2.5645)}} \\ + \frac{0.4944}{1+e^{-(-1.2362t+5.0140)}} + \frac{0.4717}{1+e^{-(1.0161t--1.3526)}}, \end{cases} \tag{A.2}$$

$$\hat{x}_{c_3} = \begin{cases} \frac{-1.4091}{1+e^{-(-0.6858t-3.8223)}} + \frac{2.9797}{1+e^{-(-0.9719t-4.8931)}} \\ + \frac{2.5975}{1+e^{-(0.0782t-1.3661)}} + \frac{-1.8859}{1+e^{-(-0.3299t-3.7000)}} \\ + \frac{1.6800}{1+e^{-(0.9489t-2.3008)}} + \frac{-3.3085}{1+e^{-(0.9972t-4.7074)}} \\ + \frac{2.1867}{1+e^{-(1.3454t+4.3235)}} + \frac{-2.4459}{1+e^{-(0.9411t+3.3095)}} \\ + \frac{2.4692}{1+e^{-(-0.4934t-2.0018)}} + \frac{1.0490}{1+e^{-(1.0401t-3.1263)}}, \end{cases} \tag{A.3}$$

$$\hat{x}_{c_4} = \begin{cases} \frac{2.2978}{1+e^{-(1.2415t-2.5226)}} + \frac{1.9443}{1+e^{-(-0.7308t-1.0914)}} \\ + \frac{2.3700}{1+e^{-(-0.0995t+0.0553)}} + \frac{-0.7981}{1+e^{-(-1.8161t-3.8849)}} \\ + \frac{-2.0235}{1+e^{-(0.8352t+4.7106)}} + \frac{-1.5512}{1+e^{-(1.6637t+4.6148)}} \\ + \frac{-3.6439}{1+e^{-(1.1124t-3.4057)}} + \frac{1.5489}{1+e^{-(1.9606t-4.6833)}} \\ + \frac{-0.2709}{1+e^{-(1.3279t-3.8548)}} + \frac{2.23711}{1+e^{-(0.6577t+1.1247)}}. \end{cases} \tag{A.4}$$

A.2 Scenario 2, All Cases, Problem 1

$$\hat{x}_{c_1} = \begin{cases} \frac{3.9180}{1+e^{-(3.1520t-4.3060)}} + \frac{-0.5103}{1+e^{-(3.0545t-0.2147)}} \\ + \frac{1.8239}{1+e^{-(0.2630t-1.0223)}} + \frac{2.2897}{1+e^{-(2.6451t-2.6244)}} \\ + \frac{2.8513}{1+e^{-(-0.1097t+1.0381)}} + \frac{-2.0137}{1+e^{-(-2.6539t+5.2125)}} \\ + \frac{-0.3643}{1+e^{-(4.0960t+1.8291)}} + \frac{-6.5574}{1+e^{-(1.9996t-6.0663)}} \\ + \frac{--5.7526}{1+e^{-(2.0457t-3.3746)}} + \frac{-2.0913}{1+e^{-(-2.8881t-2.5976)}}, \end{cases} \tag{A.5}$$

$$\hat{x}_{c_2} = \begin{cases} \frac{-1.5586}{1+e^{-(-0.4589t-2.1229)}} + \frac{-0.0476}{1+e^{-(-0.4495t-0.2177)}} \\ + \frac{-7.6070}{1+e^{-(1.9958t-6.0543)}} + \frac{-3.3697}{1+e^{-(-2.4299t+2.8337)}} \\ + \frac{2.9401}{1+e^{-(3.1289t-4.4210)}} + \frac{4.9809}{1+e^{-(-0.2378t-5.0801)}} \\ + \frac{-3.2737}{1+e^{-(-3.9565t-6.0022)}} + \frac{2.0875}{1+e^{-(-2.1929t+0.0056)}} \\ + \frac{2.9932}{1+e^{-(1.34366t+1.1201)}} + \frac{-2.9876}{1+e^{-(2.48193t-3.6877)}}, \end{cases} \tag{A.6}$$

$$\hat{x}_{c_3} = \begin{cases} \frac{-0.2666}{1+e^{-(0.5858t-0.8541)}} + \frac{1.5401}{1+e^{-(-1.3615t+2.8040)}} \\ + \frac{-2.3245}{1+e^{-(-2.8163t-3.9490)}} + \frac{-0.3548}{1+e^{-(-3.4767t-1.8749)}} \\ + \frac{-1.8780}{1+e^{-(-2.9173t+3.4448)}} + \frac{1.7181}{1+e^{-(3.2114t-4.7240)}} \\ + \frac{-6.3073}{1+e^{-(2.1168t-6.3104)}} + \frac{0.6229}{1+e^{-(2.9093t-2.2264)}} \\ + \frac{0.8699}{1+e^{-(-2.2375t-0.1870)}} + \frac{-3.0195}{1+e^{-(-2.0782t-4.9961)}}, \end{cases} \tag{A.7}$$

$$\hat{x}_{c_4} = \begin{cases} \frac{-0.9901}{1+e^{-(2.9090t-0.2019)}} + \frac{-2.3957}{1+e^{-(-1.2450t-0.9180)}} \\ + \frac{0.8400}{1+e^{-(-1.1748t+2.5664)}} + \frac{-5.7366}{1+e^{-(1.4708t-5.3218)}} \\ + \frac{-0.2039}{1+e^{-(-0.0993t-0.8017)}} + \frac{0.1564}{1+e^{-(2.1576t-3.3206)}} \\ + \frac{-1.1670}{1+e^{-(0.9005t-1.6729)}} + \frac{-2.9374}{1+e^{-(0.6231t+5.1076)}} \\ + \frac{3.2784}{1+e^{-(-2.2409t+5.5770)}} + \frac{4.8947}{1+e^{-(2.3920t-3.3090)}}. \end{cases} \tag{A.8}$$

Our derived series solution for VdP heartbeat model using propose algorithm FO-PSO–ASA for problem 2 is written below:

A.3 Scenario 1, All Cases, Problem 2

$$\hat{x}_{c_1} = \begin{cases} \frac{2.1726}{1+e^{-(-3.2809t-4.1302)}} + \frac{3.0586}{1+e^{-(-2.4326t+5.3891)}} \\ + \frac{0.6439}{1+e^{-(-1.4141t-1.5433)}} + \frac{-3.6723}{1+e^{-(-3.0796t-4.7784)}} \\ + \frac{-3.0205}{1+e^{-(-2.1545t+4.2737)}} + \frac{0.8840}{1+e^{-(-0.9828t-1.5020)}} \\ + \frac{1.1035}{1+e^{-(3.4842t-6.6746)}} + \frac{-1.5605}{1+e^{-(-0.1807t-1.6291)}} \\ + \frac{-1.3497}{1+e^{-(-1.0666t-1.9327)}} + \frac{-0.5944}{1+e^{-(-2.6004t-3.4302)}}, \end{cases} \tag{A.9}$$

$$\hat{x}_{c_2} = \begin{cases} \frac{-0.6180}{1+e^{-(-1.1150t+3.0545)}} + \frac{-0.2386}{1+e^{-(-1.1466t+1.0196)}} \\ + \frac{-0.3717}{1+e^{-(0.0108t+4.9489)}} + \frac{-2.6266}{1+e^{-(3.7084t-5.2430)}} \\ + \frac{5.4389}{1+e^{-(3.3932t-5.0487)}} + \frac{0.5789}{1+e^{-(-1.9580t+0.3469)}} \\ + \frac{4.3881}{1+e^{-(-1.8008t+3.2579)}} + \frac{-0.2476}{1+e^{-(-5.8220t+9.7978)}} \\ + \frac{-3.7829}{1+e^{-(1.7613t+5.0492)}} + \frac{2.4947}{1+e^{-(1.2763t-1.4992)}}, \end{cases} \tag{A.10}$$

$$\hat{x}_{c_3} = \begin{cases} \frac{-0.1436}{1+e^{-(0.0864t-0.5822)}} + \frac{-0.5525}{1+e^{-(-1.7690t+3.0262)}} \\ + \frac{1.7855}{1+e^{-(-0.9672t+2.3123)}} + \frac{-0.8736}{1+e^{-(0.1860t+0.5865)}} \\ + \frac{-2.1041}{1+e^{-(-0.3969t-0.7890)}} + \frac{1.5929}{1+e^{-(-0.5689t+0.4342)}} \\ + \frac{1.1744}{1+e^{-(1.2098t-1.3870)}} + \frac{-1.4695}{1+e^{-(0.8598t+3.9592)}} \\ + \frac{0.9422}{1+e^{-(2.0668t-4.7300)}} + \frac{0.5570}{1+e^{-(-0.1479t+0.1885)}}, \end{cases} \tag{A.11}$$

$$\hat{x}_{c_4} = \begin{cases} \frac{-2.2562}{1+e^{-(-0.4479t-2.5143)}} + \frac{1.2557}{1+e^{-(-0.8271t+0.1927)}} \\ + \frac{-1.7151}{1+e^{-(-0.4168t+0.6442)}} + \frac{2.0245}{1+e^{-(-0.1759t+0.4111)}} \\ + \frac{0.0505}{1+e^{-(0.3064t-3.5071)}} + \frac{1.5396}{1+e^{-(-0.9133t-1.3956)}} \\ + \frac{-4.0473}{1+e^{-(0.6725t-2.9291)}} + \frac{0.3933}{1+e^{-(21.2848t-2.9088)}} \\ + \frac{-2.2713}{1+e^{-(-0.6572t+01.4765)}} + \frac{1.6891}{1+e^{-(0.5921t+0.4085)}}. \end{cases} \tag{A.12}$$

A.4 Scenario 2, All Cases, Problem 2

$$\hat{x}_{c_1} = \begin{cases} \frac{-0.7699}{1+e^{-(0.4062t-3.0756)}} + \frac{-1.7545}{1+e^{-(-2.7929t+2.8618)}} \\ + \frac{1.2037}{1+e^{-(-2.3263t+4.7310)}} + \frac{0.4745}{1+e^{-(-2.9802t+0.3020)}} \\ + \frac{-3.3257}{1+e^{-(-3.4729t-4.4040)}} + \frac{2.9330}{1+e^{-(3.3611t-4.5447)}} \\ + \frac{1.1171}{1+e^{-(-3.0159t+4.1018)}} + \frac{-1.0560}{1+e^{-(0.2411t+2.6528)}} \\ + \frac{4.1400}{1+e^{-(-0.1895t-4.9949)}} + \frac{-7.4376}{1+e^{-(2.4223t-7.7011)}}, \end{cases} \tag{A.13}$$

$$\hat{x}_{c_2} = \begin{cases} \frac{1.8118}{1+e^{-(3.9979t-4.6522)}} + \frac{-8.2619}{1+e^{-(1.9613t-6.5601)}} \\ + \frac{0.0196}{1+e^{-(-5.1951t+0.6021)}} + \frac{-1.7990}{1+e^{-(1.2795t-1.2709)}} \\ + \frac{-2.0077}{1+e^{-(0.4887t+1.4769)}} + \frac{2.3604}{1+e^{-(-0.5356t-2.4571)}} \\ + \frac{2.6796}{1+e^{-(2.6065t+3.0080)}} + \frac{-0.9617}{1+e^{-(0.1821t+3.6132)}} \\ + \frac{2.0357}{1+e^{-(3.0991t-2.8617)}} + \frac{-0.3740}{1+e^{-(3-2.1198t-4.8614)}}, \end{cases} \tag{A.14}$$

$$\hat{x}_{c_3} = \begin{cases} \frac{4.4180}{1+e^{-(-1.5198t-5.4495)}} + \frac{-1.6323}{1+e^{-(-3.3597t+3.1037)}} \\ + \frac{0.4365}{1+e^{-(-1.3097t+1.5445)}} + \frac{-2.9833}{1+e^{-(2.8321t+1.9532)}} \\ + \frac{3.3277}{1+e^{-(4.6426t+3.8022)}} + \frac{-0.5749}{1+e^{-(1.1840t-1.6403)}} \\ + \frac{-5.4185}{1+e^{-(2.0064t-7.4368)}} + \frac{-2.3747}{1+e^{-(1.9764t-5.7124)}} \\ + \frac{0.7107}{1+e^{-(4.1447t+1.0848)}} + \frac{1.3305}{1+e^{-(4.0857t-4.5109)}}, \end{cases} \tag{A.15}$$

$$\hat{x}_{c_4} = \begin{cases} \frac{-2.03418}{1+e^{-(1.8127t+1.0480)}} + \frac{1.9961}{1+e^{-(-14.2577t-30.2714)}} \\ + \frac{-3.4863}{1+e^{-(41.6965t+10.3994)}} + \frac{3.0349}{1+e^{-(1.6507t-3.9701)}} \\ + \frac{11.7822}{1+e^{-(-20.4480t-76.0752)}} + \frac{25.2156}{1+e^{-(-4.1055t-27.8810)}} \\ + \frac{-2.3653}{1+e^{-(-2.6490t-2.1974)}} + \frac{2.3798}{1+e^{-(-1.6103t+1.0441)}} \\ + \frac{-0.0247}{1+e^{-(-132.8579t+0.0174)}} + \frac{3.3249}{1+e^{-(103.4426t+49.3694)}}. \end{cases} \tag{A.16}$$

BIBLIOGRAPHY

[1] E. M. Petriu, K. Watanabe, and T. H. Yeap, "Applications of Random-Pulse Machine Concept to Neural Network Design," *IEEE transactions on instrumentation and measurement*, vol. 45, no. 2, pp. 665–669, 1996. 2

[2] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial Neural Networks for Solving Ordinary and Partial Differential Equations," *IEEE transactions on neural networks*, vol. 9, no. 5, pp. 987–1000, 1998. 2

[3] N. Yadav, A. Yadav, M. Kumar *et al.*, *An Introduction to Neural Network Methods for Differential Equations*. Springer, 2015, vol. 1. 3

[4] P. beim Graben and J. Wright, "From Mcculloch–Pitts Neurons Toward Biology," *Bulletin of mathematical biology*, vol. 73, no. 2, pp. 261–265, 2011. 3

[5] D. O. Hebb, "The First Stage of Perception: Growth of the Assembly," *The Organization of Behavior*, vol. 4, pp. 60–78, 1949. 3

[6] B. Karlik and A. V. Olgac, "Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, 2011. 4

[7] K. Suzuki, *Artificial Neural Networks: Architectures and Applications*. BoD–Books on Demand, 2013. 7

[8] G. Bebis and M. Georgiopoulos, "Feed-Forward Neural Networks," *IEEE Potentials*, vol. 13, no. 4, pp. 27–31, 1994. 8

[9] K. Lee, D. Booth, and P. Alam, "A Comparison of Supervised and Unsupervised Neural Networks in Predicting Bankruptcy of Korean Firms," *Expert Systems with Applications*, vol. 29, no. 1, pp. 1–16, 2005. 9, 10

[10] N. Kasabov, "Evolving Fuzzy Neural Networks for Supervised/UnsuperVised Online Knowledge-Based Learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 6, pp. 902–918, 2001. 10

[11] M. Braun and M. Golubitsky, *Differential Equations and their Applications*. Springer, 1983, vol. 1. 11

[12] L. L. Littlejohn, "On the Classification of Differential Equations having Orthogonal Polynomial Solutions," *Annali di matematica pura ed applicata*, vol. 138, no. 1, pp. 35–53, 1984. 11

[13] D. Zwillinger and V. Dobrushkin, *Handbook of Differential Equations*. Chapman and Hall/CRC, 1998. 13

[14] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2016. 15

[15] R. Bulirsch, J. Stoer, and J. Stoer, *Introduction to Numerical Analysis*. Springer, 2002, vol. 3. 15

[16] K. Zhukovsky, "Operational Solution for Some Types of Second Order Differential Equations and for Relevant Physical Problems," *Journal of Mathematical Analysis and Applications*, vol. 446, no. 1, pp. 628–647, 2017. 17

[17] A. Kimiaeifar, A. Saidi, G. Bagheri, M. Rahimpour, and D. Domairry, "Analytical Solution for Van der Pol–Duffing Oscillators," *Chaos, Solitons & Fractals*, vol. 42, no. 5, pp. 2660–2666, 2009. 18, 21

[18] Y. Khan, M. Madani, A. Yildirim, M. Abdou, and N. Faraz, "A New Approach to Van der Pol's Oscillator Problem," *Zeitschrift für Naturforschung A*, vol. 66, no. 10-11, pp. 620–624, 2011. 18, 21

[19] A. Kimiaeifar, A. Saidi, G. Bagheri, M. Rahimpour, and D. Domairry, "Analytical Solution for Van der Pol–Duffing Oscillators," *Chaos, Solitons & Fractals*, vol. 42, no. 5, pp. 2660–2666, 2009. 18, 21

[20] A. Kimiaeifar, A. Saidi, A. Sohouli, and D. Ganji, "Analysis of Modified Van der Pol's Oscillator using He's Parameter-Expanding Methods," *Current Applied Physics*, vol. 10, no. 1, pp. 279–283, 2010. 18, 21

[21] S. S. Motsa and P. Sibanda, "A Note on the Solutions of the Van der Pol and Duffing Equations Using a Linearisation Method," *Mathematical Problems in Engineering*, vol. 2012, 2012. 18, 21

[22] N. Yadav, A. Yadav, and J. H. Kim, "Numerical Solution of Unsteady Advection Dispersion Equation Arising in Contaminant Transport Through Porous Media Using Neural Networks," *Computers & Mathematics with Applications*, vol. 72, no. 4, pp. 1021–1030, 2016. 18, 21

[23] N. Yadav, A. Yadav, M. Kumar, and J. H. Kim, "An Efficient Algorithm Based on Artificial Neural Networks and Particle Swarm Optimization for Solution of Nonlinear Troesch's Problem," *Neural Computing and Applications*, vol. 28, no. 1, pp. 171–178, 2017. 18

[24] M. Kumar and N. Yadav, "Multilayer Perceptrons and Radial Basis Function Neural Network Methods for the Solution of Differential Equations: a Survey," *Computers & Mathematics with Applications*, vol. 62, no. 10, pp. 3796–3811, 2011. 18

[25] M. Kumar, Neha and Yadav, "Numerical Solution of Bratu's Problem Using Multilayer Perceptron Neural Network Method," *National Academy Science Letters*, vol. 38, no. 5, pp. 425–428, 2015. 18

[26] M. Khalid, M. Sultana, and F. Zaidi, "Numerical Solution of Sixth-Order Differential Equations Arising in Astrophysics by Neural Network," *Int J Comput Appl*, vol. 107, no. 6, pp. 1–6, 2014. 18

[27] M. Khalid, M. Zaidi, and F. Sultana, "Numerical Solution of Sixth-Order Differential Equations Arising in Astrophysics by Neural Network," *Int J Comput Appl*, vol. 107, no. 6, pp. 1–6, 2014. 18

[28] S. Effati and R. Buzhabadi, "A Neural Network Approach for Solving Fredholm Integral Equations of the Second Kind," *Neural Computing and Applications*, vol. 21, no. 5, pp. 843–852, 2012. 18

[29] M. Kumar and N. Yadav, "Buckling Analysis of a Beam–Column Using Multilayer Perceptron Neural Network Technique," *Journal of the Franklin Institute*, vol. 350, no. 10, pp. 3188–3204, 2013. 18

[30] S. Momani, Z. S. Abo-Hammour, and O. M. Alsmadi, "Solution of Inverse Kinematics Problem Using Genetic Algorithms," *Applied Mathematics & Information Sciences*, vol. 10, no. 1, p. 225, 2016. 18

[31] K. S. McFall, "Automated Design Parameter Selection for Neural Networks Solving Coupled Partial Differential Equations with Discontinuities," *Journal of the Franklin Institute*, vol. 350, no. 2, pp. 300–317, 2013. 18

[32] J. C. Chedjou and K. Kyamakya, "A universal Concept Based on Cellular Neural Networks for Ultrafast and Flexible Solving of Differential Equations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 4, pp. 749–762, 2014. 18

[33] S. Mall and S. Chakraverty, "Application of Legendre Neural Network for Solving Ordinary Differential Equations," *Applied Soft Computing*, vol. 43, pp. 347–356, 2016. 18

[34] S. Chakraverty and S. Mall, "Regression-Based Weight Generation Algorithm in Neural Network for Solution of Initial and Boundary Value Problems," *Neural Computing and Applications*, vol. 25, no. 3, pp. 585–594, 2014. 18

[35] S. Mall and S. Chakraverty, "Numerical Solution of Nonlinear Singular Initial Value Problems of Emden–Fowler Type Using Chebyshev Neural Network Method," *Neurocomputing*, vol. 149, pp. 975–982, 2015. 18

[36] J. A. Khan, M. A. Z. Raja, M. I. Syam, S. A. K. Tanoli, and S. E. Awan, "Design and Application of Nature Inspired Computing Approach for Nonlinear Stiff Oscillatory Problems," *Neural Computing and Applications*, vol. 26, no. 7, pp. 1763–1780, 2015. 19

[37] S. Mall and S. Chakraverty, "Hermite Functional Link Neural Network for Solving the Van der Pol–Duffing Oscillator Equation," *Neural computation*, vol. 28, no. 8, pp. 1574–1598, 2016. 19

[38] M. A. Z. Raja, "Solution of the One-Dimensional Bratu Equation Arising in the Fuel Ignition Model Using ann Optimised with PSO and SQP," *Connection Science*, vol. 26, no. 3, pp. 195–214, 2014. 19

[39] "Design of Bio-Inspired Computing Technique for Nanofluidics Based on Nonlinear JEFFERY–HAMEL FLOW EQUATIONS, author=Raja, Muhammad Asif Zahoor and

Khan, Mohmmad Abdul Rehman and Mahmood, Tariq and Farooq, Umair and Chaudhary, Naveed Ishtiaq, journal=Canadian Journal of Physics, volume=94, number=5, pages=474–489, year=2016, publisher=NRC Research Press." 19

[40] M. A. Z. Raja, U. Farooq, N. I. Chaudhary, and A. M. Wazwaz, "Stochastic Numerical Solver for Nanofluidic Problems Containing Multi-Walled Carbon Nanotubes," *Applied Soft Computing*, vol. 38, pp. 561–586, 2016. 19

[41] M. A. Z. Raja, J. A. Khan, and T. Haroon, "Stochastic Numerical Treatment for Thin Film Flow of Third Grade Fluid Using Unsupervised Neural Networks," *Journal of the Taiwan Institute of Chemical Engineers*, vol. 48, pp. 26–39, 2015. 19

[42] Z. Abo-Hammour, O. Abu Arqub, S. Momani, and N. Shawagfeh, "Optimization Solution of Troesch's and Bratu's Problems of Ordinary Type Using Novel Continuous Genetic Algorithm," *Discrete Dynamics in Nature and Society*, vol. 2014, 2014. 19

[43] M. A. Z. Raja, "Stochastic Numerical Treatment for Solving Troesch's Problem," *Information Sciences*, vol. 279, pp. 860–873, 2014. 19

[44] A. Sadollah, Y. Choi, J. H. Kim *et al.*, "Metaheuristic Algorithms for Approximate Solution to Ordinary Differential Equations of Longitudinal Fins having Various Profiles," *Applied Soft Computing*, vol. 33, pp. 360–379, 2015. 19

[45] S. Effati and M. H. N. Skandari, "Optimal Control Approach for Solving Linear Volterra Integral Equations," *International Journal of Intelligent Systems and Applications*, vol. 4, no. 4, p. 40, 2012. 19

[46] S. Effati and M. Pakdaman, "Artificial Neural Network Approach for Solving Fuzzy Differential Equations," *Information Sciences*, vol. 180, no. 8, pp. 1434–1457, 2010. 19

[47] E. Sohrab and M. Pakdaman, "Artificial Neural Network Approach for Solving Fuzzy Differential Equations," *Information Sciences*, vol. 180, no. 8, pp. 1434–1457, 2010. 19

[48] M. A. Z. Raja, I. Ahmad, I. Khan, M. I. Syam, and A. M. Wazwaz, "Neuro-Heuristic Computational Intelligence for Solving Nonlinear Pantograph Systems," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 4, pp. 464–484, 2017. 19

[49] M. A. Z. Raja, "Numerical Treatment for Boundary Value Problems of Pantograph Functional Differential Equation Using Computational Intelligence Algorithms," *Applied Soft Computing*, vol. 24, pp. 806–821, 2014. 19

[50] M. A. Z. Raja, J. A. Khan, S. M. Shah, R. Samar, and D. Behloul, "Comparison of Three Unsupervised Neural Network Models for First Painlevé Transcendent," *Neural Computing and applications*, vol. 26, no. 5, pp. 1055–1071, 2015. 19

[51] M. Raja, R. Samar, T. Haroon, and S. Shah, "Unsupervised Neural Network Model Optimized with Evolutionary Computations for Solving Variants of Nonlinear MHD Jeffery-Hamel Problem," *Applied Mathematics and Mechanics*, vol. 36, no. 12, pp. 1611–1638, 2015. 19

[52] M. A. Z. Raja and R. Samar, "Numerical Treatment for Nonlinear MHD Jeffery–Hamel Problem Using Neural Networks Optimized with Interior Point Algorithm," *Neurocomputing*, vol. 124, pp. 178–193, 2014. 19

[53] M. A. Z. Raja, R. Samar, E. S. Alaidarous, and E. Shivanian, "Bio-Inspired Computing Platform for Reliable Solution of Bratu-Type Equations Arising in the Modeling of Electrically Conducting Solids," *Applied Mathematical Modelling*, vol. 40, no. 11-12, pp. 5964–5977, 2016. 19

[54] I. Ahmad, M. A. Z. Raja, M. Bilal, and F. Ashraf, "Neural Network Methods to Solve the Lane–Emden Type Equations Arising in Thermodynamic Studies of the Spherical Gas Cloud Model," *Neural Computing and Applications*, vol. 28, no. 1, pp. 929–944, 2017. 19

[55] M. A. Z. Raja, M. A. Manzar, and R. Samar, "An Efficient Computational Intelligence Approach for Solving Fractional Order Riccati Equations Using ANN and SQP," *Applied Mathematical Modelling*, vol. 39, no. 10-11, pp. 3075–3093, 2015. 19

[56] J. Sabouri, S. Effati, and M. Pakdaman, "A Neural Network Approach for Solving a Class of Fractional Optimal Control Problems," *Neural Processing Letters*, vol. 45, no. 1, pp. 59–74, 2017. 19

[57] J. A. Khan, M. A. Z. Raja, M. M. Rashidi, M. I. Syam, and A. M. Wazwaz, "Nature-Inspired Computing Approach for Solving Non-Linear Singular Emden–Fowler Problem Arising in Electromagnetic Theory," *Connection Science*, vol. 27, no. 4, pp. 377–396, 2015. 19

[58] S. Mall and S. Chakraverty, "Chebyshev Neural Network Based Model for Solving Lane–Emden Type Equations," *Applied Mathematics and Computation*, vol. 247, pp. 100–114, 2014. 19

[59] J. Tang, J. Zhu, and Z. Sun, "A Novel Path Planning Approach Based on Appart and Particle Swarm Optimization," in *International Symposium on Neural Networks*. Springer, 2005, pp. 253–258. 19

[60] E. Pires, P. Oliveira, J. Tenreiro Machado, and J. B. Cunha, "Particle Swarm Optimization Versus Genetic Algorithm in Manipulator Trajectory Planning," in *CONTROLO 2006-7th Portuguese Conference on Automatic Control*, 2006, pp. 1–7. 19, 24

[61] M. S. Couceiro, R. Mendes, N. F. Ferreira, and J. T. Machado, "Control Optimization of a Robotic Bird," *EWOMS*, vol. 9, pp. 4–6, 2009. 19

[62] M. S. Couceiro, J. M. A. Luz, C. M. Figueiredo, and N. F. Ferreira, "Modeling and Control of Biologically Inspired Flying Robots," *Robotica*, vol. 30, no. 1, pp. 107–121, 2012. 19

[63] M. S. Couceiro, J. M. A. Luz, C. M. Figueiredo, N. F. Ferreira, and G. Dias, "Parameter Estimation for a Mathematical Model of the Golf Putting," in *Proceedings of WACI-Workshop Applications of Computational Intelligence. ISEC. IPC. Coimbra*, vol. 2, 2010, pp. 1–8. 19

[64] M. Alrashidi and M. El-Hawary, "A Survey of Particle Swarm Optimization Applications in Power System Operations," *Electric Power components and systems*, vol. 34, no. 12, pp. 1349–1357, 2006. 19

[65] Y. Shi and R. C. Eberhart, "Fuzzy Adaptive Particle Swarm Optimization," in *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, vol. 1. IEEE, 2001, pp. 101–106. 19, 24

[66] E. Solteiro Pires, J. Tenreiro Machado, P. de Moura Oliveira, J. Boaventura Cunha, and L. Mendes, "Particle Swarm Optimization with Fractional-Order Velocity," *Nonlinear Dynamics*, vol. 61, no. 1, pp. 295–301, 2010. 19

[67] Y. Del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Sys-

tems," *IEEE Transactions on evolutionary computation*, vol. 12, no. 2, pp. 171–195, 2008. 19, 24

[68] T. M. Blackwell and P. Bentley, "Don't Push Me! Collision-Avoiding Swarms," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2. IEEE, 2002, pp. 1691–1696. 19

[69] T. Krink, J. S. VesterstrOm, and J. Riget, "Particle Swarm Optimisation with Spatial Particle Extension," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2. IEEE, 2002, pp. 1474–1479. 19

[70] N. Fonseca and V. Miranda, "New Evolutionary Particle Swarm Algorithm (epso) Applied to Voltage/var Control," 2002. 19

[71] M. Lovbjerg and T. Krink, "Extending Particle Swarm Optimisers with Self-Organized Criticality," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2. IEEE, 2002, pp. 1588–1593. 19

[72] C.-F. Juang, "A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 2, pp. 997–1006, 2004. 19

[73] P. J. Angeline, "Using Selection to Improve Particle Swarm Optimization," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)*. IEEE, 1998, pp. 84–89. 20

[74] W.-J. Zhang and X.-F. Xie, "Depso: Hybrid Particle Swarm with Differential Evolution Operator," in *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, vol. 4. IEEE, 2003, pp. 3816–3821. 20

[75] S. Kannan, S. M. R. Slochanal, P. Subbaraj, and N. P. Padhy, "Application of Particle Swarm Optimization Technique and its Variants to Generation Expansion Planning Problem," *Electric Power Systems Research*, vol. 70, no. 3, pp. 203–210, 2004. 20

[76] B. Van Der Pol and J. Van Der Mark, "Lxxii. the Heartbeat Considered as a Relaxation Oscillation, and an Electrical Model of the Heart," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 6, no. 38, pp. 763–775, 1928. 21

[77] Z.-M. Ge and M.-Y. Hsu, "Chaos in a Generalized Van der Pol System and in its Fractional Order System," *Chaos, Solitons & Fractals*, vol. 33, no. 5, pp. 1711–1745, 2007. 21

[78] A. M. dos Santos, S. R. Lopes, and R. R. L. Viana, "Rhythm Synchronization and Chaotic Modulation of Coupled Van der Pol Oscillators in a Model for the Heartbeat," *Physica A: Statistical Mechanics and its Applications*, vol. 338, no. 3-4, pp. 335–355, 2004. 21, 22

[79] K. Grudziński and J. J. Żebrowski, "Modeling Cardiac Pacemakers with Relaxation Oscillators," *Physica A: statistical Mechanics and its Applications*, vol. 336, no. 1-2, pp. 153–162, 2004. 21

[80] Y. A. Aman, "Bifurcation Analysis and Lyapunov Exponent Spectrum for a Modified Van Der Pol Heart Model," Tezpur University, Tech. Rep., 2022. 22

[81] K. Hall, D. J. Christini, M. Tremblay, J. J. Collins, L. Glass, and J. Billette, "Dynamic Control of Cardiac Alternans," *Physical Review Letters*, vol. 78, no. 23, p. 4518, 1997. 22

[82] B. B. Ferreira, A. S. de Paula, and M. A. Savi, "Chaos Control Applied to Heart Rhythm Dynamics," *Chaos, Solitons & Fractals*, vol. 44, no. 8, pp. 587–599, 2011. 22

[83] E. Solteiro Pires, J. Tenreiro Machado, P. de Moura Oliveira, J. Boaventura Cunha, and L. Mendes, "Particle Swarm Optimization with Fractional-Order Velocity," *Nonlinear Dynamics*, vol. 61, no. 1, pp. 295–301, 2010. 25, 33

[84] W. Waseem, M. Sulaiman, A. Alhindi, and H. Alhakami, "A Soft Computing Approach Based on Fractional Order DPSO Algorithm Designed to Solve the Corneal Model for Eye Surgery," *IEEE Access*, vol. 8, pp. 61 576–61 592, 2020. 25, 33

[85] D. Acharya, B. Sarkar, and D. Bharti, "A Fractional Order Particle Swarm Optimization for Tuning Fractional Order pid Controller for Magnetic Levitation Plant," in *2020 First IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA)*. IEEE, 2020, pp. 1–6. 25, 33

[86] G. Gatti, "Statics and Dynamics of a Nonlinear Oscillator with Quasi-Zero Stiffness Behaviour for Large Deflections," *Communications in Nonlinear Science and Numerical Simulations*, vol. 83, p. 105143, Apr. 2020. 25

[87] M. Kumar and N. Yadav, "Multilayer Perceptrons and Radial Basis Function Neural Network Methods for the Solution of Differential Equations: A Survey," *Computers and Mathematics with Applications*, vol. 62, no. 10, pp. 3796–3811, 2011. 25

[88] M. Shoaib, K. S. Nisar, M. A. Z. Raja, Y. Tariq, R. Tabassum, and A. Rafiq, "Knacks of Neuro-Computing to Study the Unsteady Squeezed Flow of mhd Carbon Nanotube with Entropy Generation," *International Communications in Heat and Mass Transfer*, vol. 135, p. 106140, 2022. 26

[89] M. A. Zahoor Raja, M. Shoaib, R. Tabassum, M. I. Khan, C. Jagannatha, and C. Gali, "Performance Analysis of Backpropagated Networks for Entropy Optimized Mixed Convection Nanofluid with Second-Order Slip Over a Stretching Surface," *Waves in Random and Complex Media*, pp. 1–23, 2022. 26

[90] M. A. Z. Raja, M. Shoaib, R. Tabassum, M. I. Khan, C. Jagannatha, C. Gali, and S. Elattar, "A Stochastic Intelligent Approach for Entropy Optimized Mixed Convective Second-Order Slip Flow Over a Movable Surface," *Archive of Applied Mechanics*, vol. 92, no. 8, pp. 2435–2454, 2022. 26

[91] M. Shoaib, R. Tabassum, M. I. Khan, M. A. Z. Raja, S. Elattar, S. Qayyum, and A. M. Galal, "Intelligent Computing for Entropy Generation in Jeffrey Nanofluid Through Radiated Flow," *Waves in Random and Complex Media*, vol. 33, no. 2, pp. 461–488, 2023. 26

[92] M. Shoaib, K. S. Nisar, M. A. Z. Raja, M. Saad, R. Tabassum, A. Rafiq, and I. Ullah, "Intelligent Networks Knacks for Numerical Treatment of Three-Dimensional Darcy–Forchheimer Williamson Nanofluid Model Past a Stretching Surface," *Waves in Random and Complex Media*, pp. 1–29, 2022. 26

[93] E. Solteiro Pires, J. Tenreiro Machado, P. de Moura Oliveira, J. Boaventura Cunha, and L. Mendes, "Particle Swarm Optimization with Fractional-Order Velocity," *Nonlinear Dynamics*, vol. 61, no. 1, pp. 295–301, 2010. 33