



A Novel ANN-ARMA Scheme Enhanced by Metaheuristic Algorithms for Dynamical Systems and Time Series Modeling and Identification

Zahia Nabi^{1,2}, Mohammed Assam Ouali^{1,2*}, Mohamed Ladjal^{1,2}, Hamza Bennacer^{2,3}

¹ LASS, Laboratory of Analysis of Signals and Systems, University of M'sila, M'sila 28000, Algeria

² Department of Electronics, Faculty of Technology, University of M'sila, M'sila 28000, Algeria

³ Elaboration and Physico-Mechanical and Metallurgical Characterization of Materials Laboratory, ECP3M, University of Mostaganem, Mostaganem 27000, Algeria

Corresponding Author Email: mohamedassam.ouali@univ-msila.dz

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ria.380320>

ABSTRACT

Received: 9 September 2023

Revised: 2 February 2024

Accepted: 21 March 2024

Available online: 21 June 2024

Keywords:

dynamical systems, times series, modeling and identification, metaheuristics algorithms, AR, ARMA.

This paper presents a new scheme for dynamical systems and time series modeling and identification. It is based on artificial neural networks (ANN) and metaheuristic algorithms. This scheme combines the strength of ANN with the dexterity of metaheuristic algorithms. This fusion is renowned for its ability to detect complex patterns, which considerably improves accuracy, computational efficiency, and robustness. The proposed scheme deals with the curve fitting and addresses ANN's local minima problem. This approach introduces the identification concept using a fresh novel identification element, referred to as the error model. The proposed framework encompasses a parallel interconnection of two models. The principal sub-model is the elementary model, characterized by standard specifications and a lower resolution, designed for the data being examined. In order to address the resolution limitation and achieve heightened precision, a second sub-model, named the error model, is introduced. This error model captures the disparities between the primary model and considered data. The parameters of the proposed scheme are adjusted using metaheuristic algorithms. This technique is tested across many benchmark data sets to determine its efficacy. A comparative study along with benchmark approaches will be provided. Extensive computer studies show that the suggested strategy considerably increases convergence and resolution.

1. INTRODUCTION

Dynamical systems have an extended variety of applications in diverse engineering sections, such as communication, biology, sociology, physiology, meteorology, economics, neuroscience, epidemiology, model based control design and pattern recognition. A dynamical system is a set of laws or differential equations from mathematics and physics, which describe the interactions between the states of particular systems and their evolution over the time [1]. However modeling and identification of those systems become a principal problem in engineering and science [2] and this is due to the fact that these systems operate using historical operations and investigations, which means that the current output is a function of past outputs, or past inputs, or both, contrary to static systems which are described by algebraic equations, which are straight and assimilated readily.

Different methods for modeling and identification of both linear and nonlinear dynamical systems have been outlined in the literature. Some of those strategies are mathematical methods based on the theory of differential equations which fail in many cases to model these systems and obtain their mathematical model because of the complicity of some plants due to the unknown system parameters, and the others are

computational intelligence techniques based on artificial intelligence, which are the most widely adopted by researchers in recent decades [3]. These methods include the concepts of Neural Networks, Radial Basis Function networks [4, 5], Fuzzy Logic [6, 7], Neuro-Fuzzy Systems [8], Machine learning [9] and Deep learning methods [10].

Inspired by the functioning biological nervous systems function in the human brain, the artificial neural network is a highly efficient computational system. Three layers constitute an artificial neural network (ANN): input, hidden, and output layers. Each neuron is connected to other neurons, and each link between these neurons is associated with a weight that holds information about the input signal. Each neuron has an internal state, known as the activation function. The signals that come out generated by combining the input signals and the activation rule are able to be transmitted to additional units [11, 12]. The flexibility, learning capabilities and symbolic reasoning make ANNs the most used in several branches, namely engineering, economics, medicine, military, navy, optimization, prediction, forecasting, control of complex systems, modeling, identification and control of dynamical systems. A possible advantage of using artificial neural networks (ANNs) in modeling is their ability to improve the accuracy and usability of complex natural systems with a large

number of inputs. This prompted many researchers to adopt it in their studies as a modeling tool for dynamical systems instead of statistical modeling techniques [13-17], for the reason that the combination with other techniques can be an effective way to improve the Modeling performance and gives the best accuracies compared to the other technique used separately.

Several ANN hybrid methods are discussed in literature. Loussifi et al. [18] have provided a novel hybrid intelligent neural network model for nonlinear dynamical systems identification, which uses wavelet Multi-resolution analysis (MRA) as activation functions for the ANN structure.

Using improved particle swarm optimization, Cavuslu et al. [19] provide the hardware implementation of ANN with learning abilities on field programmable gate arrays (FPGA) for dynamic system identification.

Singh et al. [20] developed a novel method based on ANN structure and learning algorithm for identification and control of a nonlinear system. Jovanović [21] in his work presented a ANN approach for dynamical system Modeling and identification trained and tested by using the responses recorded in a real frame during earthquakes. A novel neural network estimator was constructed for nonlinear systems identification and control in the research [22] by Gautam. For the purpose of identifying nonlinear dynamical systems utilizing the back-propagation algorithm, Patra et al. [23] have proposed an alternative ANN structure called functional link ANN (FLANN). A new neural networks approach called a singularity-free approach for dynamical systems identification and control was developed by Zheng et al. [24].

Achieving a higher performance for any ANN-based technique depends on the algorithm used for its training and the iterative updating of its weights in order to minimize the error function, which is defined as the desired and target output and to overcome the entrapment in local minimums and slow convergence rate. The effectiveness of the Metaheuristic algorithms lies in their ability to improve neural network models to solve large and complex problems precisely [25]. They are currently the state of the art for a variety of optimization problems, especially for problems that are very complex and have a high dimensionality.

In this paper, we propose a new structural method based on ANN and metaheuristic algorithms to address common difficulties in the modeling and identification of dynamical systems and time series. The change of states across time characterizes dynamical systems, and precisely modeling their behavior is critical for understanding and forecasting their dynamics. Traditional mathematical models may be insufficient for complex, nonlinear systems, and there is a growing interest in using machine-learning approaches, specifically ANN, for dynamical system modeling.

Dynamical systems frequently demonstrate nonlinear behavior, which poses a challenge in accurately representing their dynamics using conventional linear models. ANNs possess the capability to capture nonlinear relationships. However, the task of designing a network architecture that accurately models the dynamics of a system is not straightforward. Moreover, it is worth noting that Dynamical systems often encounter constraints in terms of the data available for training purposes. Furthermore, the process of obtaining additional data can prove to be resource-intensive or even impractical in certain scenarios. The development of ANN models that exhibit data efficiency and strong generalization capabilities despite limited data availability is a

significant challenge. Moreover, it is worth noting that dynamical systems have the potential to exhibit sensitivity to both initial conditions and external perturbations. The ANN model must possess sufficient robustness to effectively manage uncertainties and variations in the system parameters. ANNs are commonly regarded as "black-box" models, which poses challenges in interpreting the acquired representations. The development of techniques aimed at enhancing the interpretability of ANN models for dynamical systems is crucial in order to gain a deeper understanding of the underlying dynamics.

The computational cost and time required for training complex neural network models pose challenges for applications with limited resources. The proposed structure addresses the aforementioned issue.

The issue of local minima is a concern in the training of ANNs, especially when using gradient-based optimization algorithms. Local minima are locations in the loss landscape where the gradient is zero, and the algorithm may converge prematurely, preventing the network from finding the global minimum of loss function. In this study, metaheuristic algorithms were used to solve this problem. These algorithms accomplish the adaptive tuning of ANN parameters. Due to their general effectiveness, these algorithms are widely used in many different fields across various domains. Parallel processing within the population yields the best answer.

The proposed approach introduced the notion of modeling using the error module. It consists of an association of two sub-ANN models. The initial one is the primary model, which is a low-resolution representation of the ordinary model for the dynamical system or time series under investigation. The second sub-ANN model termed error model reflects the error modeling between the primary model outcome and the output of the real system or time series under consideration in order to resolve the resolution quality constraint and get a model with greater resolution. The effectiveness of this method is assessed through testing on the three nonlinear dynamical systems as stated by Narendra and Parthasarathy in references [16, 17] and benchmark time series. Intensive computer experiments improved the convergence and resolution of the proposed approach.

The rest of the paper is structured as follows: Section 2 introduces a brief description of ANNARMA and the metaheuristic algorithms. In Section 3 we explain the proposed technique, Section 4 includes several experimental instances and different validation tests to verify the effectiveness of the proposed method. Finally, the conclusion, which summarizes the entire paper, is given in Section 5.

2. PRELIMINARIES

This section presents a brief explanation of the concept of ANN-ARMA and the metaheuristic algorithms used in building our proposed model.

2.1 ANN-ARMA Concept

Artificial neural networks are a field of artificial intelligence that attempts to emulate the functioning of the human neurological system in order to resolve complicated issues. ANN is made up of a large number of nodes known as artificial neurons. A weight and bias are assigned to each neuron, which indicates the information the network employs to solve a

problem. Neurons are connected to one another through communication interactions. Inputs that a neuron receives determine its internal state or activation functions. A neuron's activity is often sent as a signal to multiple other neurons. ANN changes the weights and biases of each neuron during training to minimize the difference between the expected and actual output using an optimization approach of gradient descent, which assesses the amount and direction of weight changes based on the error.

Autoregressive (AR) models and Moving Average (MA) models are commonly used in time series analysis. Autoregressive models predict the next value in a series based on the previous values, while the moving average models predict the next value based on the average of the previous values.

Artificial neural networks can also be used for time series analysis, including both autoregressive and moving average models. In an autoregressive ANN, the input to the network is a time series sequence, and the network uses previous values to predict the next value in the sequence. In a moving average ANN, the input to the network is a moving window of the time series sequence, and the network predicts the next value based on the average of the previous values in the window. A popular combination of these two approaches is the Autoregressive Moving Average (ARMA) model, which combines the strengths of both methods. In ARMA model, the network uses both the previous values and the moving average of previous values to predict the next value in the time series sequence. Autoregressive, moving average, and ARMA models can all be implemented using ANNs for time series analysis. The choice of which model to use depends on the particular characteristics of the time series data and objectives of analysis. For ARMA model, output is modelled as a linear difference equation between current and past inputs and past outputs as described in the equation that follows:

$$y(t) = \sum_{i=1}^n a_i x(t-i) + \sum_{j=1}^n b_j u(t-j) \quad (1)$$

where, $x(t)$ and $u(t)$ are inputs and outputs, a_i and b_j are the ARMA parameters.

Adding these outputs to a neural network as inputs is equivalent to changing its structure into a recurrent neural network. The objective of this hybrid structure (ANNARMA) is to combine the advantages of both and to obtain a more reliable modeling result.

In order to minimize the error between the model's output and real data output, optimization algorithms are used to update the model's parameters. Creating an appropriate fitness function, also known as an objective function, is essential for the effectiveness of the system identification and it is formulated to determine the control parameter values that best satisfy the desired goal. Usually, the control parameters must be selected within certain restrictive limits. In this work, Mean Square Error (MSE) criterion function was used which is described in the equation that follows:

$$MSE = \frac{\sum_{k=1}^N (y_k - \hat{y}_k)^2}{N} \quad (2)$$

where, y_k and \hat{y}_k are the actual measurement and its estimate, respectively, and N is the length of the data.

2.2 Metaheuristic algorithms

In this section, we provide a synopsis of the IWO, PSO, ICA and CMA-ES metaheuristics algorithms.

2.2.1 Invasive weeds optimization

The population-based optimization approach known as "invasive weed optimization" (IWO) was first proposed by Mehrabian and Lucas in 2006, as stated in the study [26] and takes inspiration for solving continuous optimization problems from how invasive weeds operate in the natural world.

A significant threat to agricultural crops, weeds is distinguished by their strength, rapid adaption, and ability for propagation in the environment. Weeds invade fields by dispersing their seeds through the air. These seeds occupy the available spaces and grow into flowering weeds using the available resources. New weeds following the same process are randomly dispersed in the field and develop into the flowering weeds and the process continues.

The IWO algorithm is outlined below:

- Generate randomly a population of n weeds.
- Generation of the seeds population.
- Evaluation of the fitness of each seed and rank them according to their fitness. The seeds now are called flowering weeds.
- Production of new seeds by the previous flowering weeds according to their rank. The number of seeds produced by a weed varies between S_{min} and S_{max} increasing linearly from the lowest ranked weed to the highest ranked weed.
- Generation of the seeds using the normally dispersed arbitrary numbers with mean equivalent to the location of the generating weeds. The standard deviations are varied according to the following equation:

$$\theta_{iter} = \frac{(iter_{max} - iter)^n}{iter_{max}^n} (\theta_i - \theta_f) + \theta_f \quad (3)$$

where, $iter_{max}$ the maximum number of iterations, θ_i and θ_f are both the initial and the last standard deviations, and n is the nonlinear modulation index.

- Evaluation of the fitness of newly generated seeds that become flowering weeds. They are then ranked with their parents according to their fitness.
- Elimination of weeds of lower fitness in order to attain the maximum number of weeds allowed in the colony (P_{max}).
- The survived weeds can produce new seeds according to their rank and this process continues until the stopping criterion is reached. The stopping criterion is usually considered to be the maximum number of iterations or a certain limit value of fitness.

2.2.2 Particle swarm optimization

Particle Swarm Optimization (PSO) is a forceful metaheuristic optimization algorithm established by Kennedy and Eberhart in 1995 [27]. It was relying upon the comportment of flocking birds and schooling fish observed in nature. This algorithm works with this concept: a flock of birds is randomly initialized in the search area, where each bird is named a "particle". After a specific number of iterations, these birds (particles) locate the optimal global position. For each iteration, every particle is able to alter its velocity vector depending on its momentum and the effect of its best position as well as the

best position of the most qualified individual. The particle then travels to a newly calculated location, and its fitness may be assessed using the optimization problem's objective function. The particle's previously visited best position is marked as its personal best position (P_{best}). The global best position (g_{best}) is the location of the best person in the swarm. These two equations are used to assign the particle's velocity and new location at each step:

$$V_{(t+1)} = V_{(t)} + C_1 r_1 (pb_{(t)} - X_{(t)}) + C_2 r_2 (pg_{(t)} - X_{(t)}) \quad (4)$$

$$X_{(t+1)} = X_{(t)} + V_{(t+1)} \quad (5)$$

where, C_1 and C_2 are the acceleration coefficients, and r_1, r_2 are random variables in a range of $[0, 1]$.

The PSO algorithm is described as follows:

- Generate randomly a population swarm of N particles.
- Initialization of the parameters of PSO (C_1 and C_2).
- Initialize particle with random position (x) and velocity (v).
- For each iteration, the following steps are repeated until satisfying the stopping criterion:
 - ✓ Solve the target problem.
 - ✓ Calculate the objective function.
 - ✓ Update the g_{best} and p_{best} values.
 - ✓ Update particle position (x) and velocity (v) according to the velocity and position updating Eqs. (4) and (5).
- Stopping conditions are satisfied: g_{best} and optimal solution has been found.

2.2.3 Imperialist competitive algorithm

Imperialist Competitive Algorithm (ICA) is a socio-political metaheuristic algorithm proposed in 2007 by Atashpaz-Gargari and Lucas [28]. It was formed by the historical colonization process and the rivalry between empires for more colonies. The algorithm starts with a random initial population (country). The most powerful nation shall serve as the empire's imperialist, with the rest forming colonies. To assess the general intent of an empire, a linear combination of the imperialist's desired result and the average of the objective values of the empire's colonies is employed. The most vulnerable empire may be found after assessing all of the empires. Then all other empires compete to seize the weakest colony of the weakest empire.

The ICA algorithm is detailed as follows:

- Initialization of the algorithm.
- Generation of a collection of arbitrary solutions within the optimization problem's search space and create initial empires. Generation of random countries and determination of their power by the cost function.
- The countries having the cheapest function value turn Imperialists, seize control of other countries (or colonies), and establish the initial Empires.
- Assimilation induces each empire's colonies to move closer to the imperialist state in the space of seeking optimization.
- The revolution guides to sudden and random changes in the position of certain countries in the research space.
- Throughout assimilation and revolution, a colony may attain a better position and have the opportunity to take

over the entire empire and replace the present imperialist state of the empire.

- For the imperialist competition, all empires compete to win the game and take control of colonies of other empires. Depending on their might, all empires have an opportunity of acquiring one or more colonies of the weaker empire at each stage of the algorithm.
- Until a stop condition is satisfied, the algorithm continues to progress through the above mentioned steps (Assimilation, Revolution, Competition).

2.2.4 Covariance matrix adaptation-evolution strategy

Covariance Matrix Adaptation Evolution Strategy (CMA-ES), as cited in the research, is a powerful version of evolution strategy algorithm for solving continuous optimization problems, introduced by Hansen and Ostermeier in 2001 [29]. CMA-ES is based on Evolution Strategy (ES) which is a type of evolution algorithm (EA). The mathematical and statistical model employed in the construction of CMA-ES is quite intriguing and sets it apart from all other evolutionary algorithms and metaheuristics. It consists in finding a vector of parameters x that maximizes an objective or a fitness function $f(x)$. The mean and covariance of exploration distribution are ordinary variables and are not conditioned by the agent's current state.

Dynamic modifications to the search distribution are possible through the covariance matrix's adaptation mechanism, which sets CMA-ES unique. CMA-ES's unique characteristic significantly simplifies complex, high-dimensional objective function optimization. It has been successfully applied in many fields, including engineering, robotics, and other fields where conventional optimization techniques may falter, such as neural network architecture optimization and parameter tuning.

The CMA-ES is clarified below:

- Initialize the CMA-ES parameters.
- Generate a population of candidate solutions.
- Evaluate the fitness values for every individual in the population.
- Select parents from the fittest individuals, based on their fitness values, to establish the next generation's parent population.
- For each iteration, the following steps are repeated until satisfying the stopping criterion:
 - ✓ Update the Covariance Matrix:

$$C^{t+1} = (1 - c_1 - c_\mu)C^t + c_1 + P_c^{t+1}(P_c^{t+1})^T + c_\mu \sum_{i=1}^Y W_{Ri} \frac{X_i - m^t}{\sigma^t} \left(\frac{X_i - m^t}{\sigma^t} \right)^T \quad (6)$$

where, c_1 and c_μ are learning rate parameters, W_{Ri} is the weight for the R^{ith} highest point, and P_c is the evolution path.

$$P_c^{t+1} = (1 - C_c)P_c^t + \sqrt{\frac{C_c(2 - C_c)}{\sum_{i=1}^Y W_i^2}} \frac{m^{t+1} - m^t}{\sigma^t} \quad (7)$$

- ✓ Update the Step Size:

$$\sigma^{t+1} = \sigma^t \exp\left(\frac{c_\sigma P_\sigma^{t+1} - X_d}{d_\sigma X_d}\right) \quad (8)$$

where, c_σ is the learning rate, d_σ is the damping rate, and P_σ is the evolution path.

- ✓ Generate Sample Population for generation $t + 1$.
- ✓ Update the mean for generation $t + 1$:

$$m^{t+1} = \sum_{i=1}^{\gamma} W_{Ri} X_i \quad (9)$$

- ✓ Update the best-ever solution.
- ✓ Stopping conditions are satisfied: Results.

3. PROPOSED ANN SCHEME

In this section, the proposed ANN method for dynamical systems and time series modeling and identification will be discussed. The proposed technique comprises three stages, the first is the identification of the primary model, the second is the identification of the error process, and finally the design of the final model, which consists of a parallel interconnection between the first two steps.

3.1 Parameters update ANN

The neural network optimization algorithm employed in this paper is a feedforward neural network. Figure 1 shows the ANN configuration throughout used in this work.

Weights (w_i) are the parameters in a neural network's hidden layers that modify the input data, and Biases (b_n) are the constants added to the product of features and weights. These parameters determine the parameters of the ANN model to be trained by optimization methods. They are applied in order to offset the result.

3.2 Primary model identification

During this stage, the input-output dataset (U_k, Y_k) is utilized to establish the primary ANN model (Y_{PM}) for the given dynamical system or time series (Figure 2). The ANN primary model is designed using an ANN-autoregressive moving average model (ANN-ARMA) that clearly strives to anticipate the current output based on the sum of previous outputs and inputs. The primary ANN-model's structure is mainly on online adaptation of the feed forward neural network's parameters. The parameters optimization bloc (Figure 2) which can be either IWO, PSO, ICA or CMA-ES algorithms, will adjust the parameters of the primary model such that the error E_k between the process output Y_k and the primary model output \hat{Y}_k attains its lowest value.

3.3 Error process identification

For this second stage, it will be the same as the first step, but the focus will be on identifying the error of the first stage (E_k). This error results from a parallel connection between the relevant dynamical system process or time series (y_k) and the primary model output (\hat{y}_k). The error E_k is precisely defined by:

$$E_k = y_k - \hat{y}_k. \quad (10)$$

After having obtained the error process E_k , we proceed to

its modeling by a second ANN model. This model is called ANN error model (Y_{EM}). The error E_k can be considered as a time series, thus it makes sense to use an autoregressive model (AR) when designing its model, which strives to predict the new output based on the previous results. The structure of this stage is illustrated in the Figure 3.

The structure of the ANN error model is mainly on online adaptation of the feed forward neural network's parameters. The parameters optimization bloc (Figure 3), which can be either IWO, PSO, ICA or CMA-ES algorithms, will adjust the parameters of the error model such that the error E_{1k} between the error process output E_k and the error model output \hat{E}_k attains its lowest value.

3.4 Final model design

Ultimately, the primary model and the error model will be interconnected in parallel, resulting in the final ANN model depicted in Figure 4. This interconnection was made in order to reduce the Modeling error and obtain a net final model.

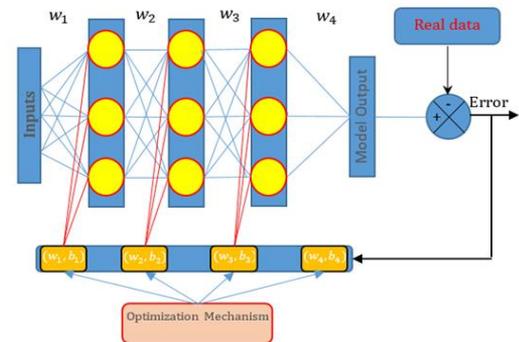


Figure 1. ANN Structure

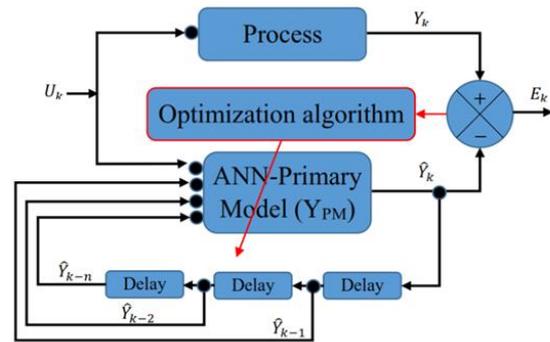


Figure 2. ANN-Primary Model

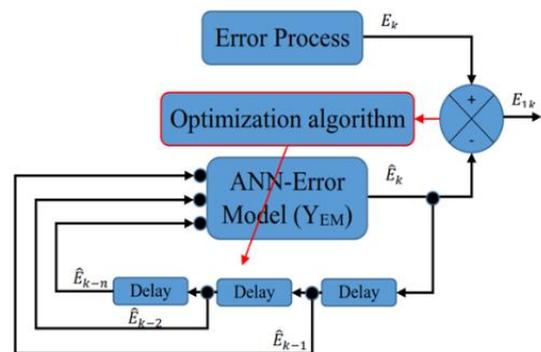


Figure 3. ANN-Error Model

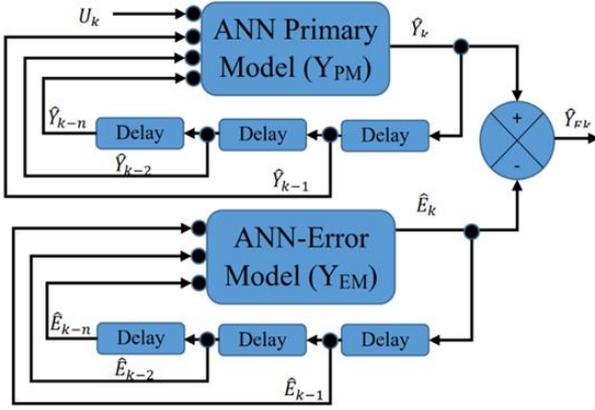


Figure 4. ANN-Final Model

4. RESULTS AND DISCUSSION

In this section, we present and discuss the simulation results of the proposed method for modeling and identification of dynamical systems. For this purpose, the three nonlinear dynamical systems described below will be used for testing the ability of the proposed approach [16, 17]:

- System 1:

$$y_p(k+1) = f[y_p(k), y_p(k-1), \dots, y_p(k-n+1)] + \sum_{i=0}^{m-1} \beta_i u(k-1). \quad (11)$$

- System 2:

$$y_p(k+1) = \sum_{i=0}^{n-1} \alpha_i y_p(k-1) + g[u(k), u(k-1), \dots, u(k-m+1)] \quad (12)$$

- System 3:

$$y_p(k+1) = f[y_p(k), y_p(k-1), \dots, y_p(k-n+1)] + g[u(k), u(k-1), \dots, u(k-m+1)] \quad (13)$$

Following an extensive comparative analysis, we have identified that the most effective optimization algorithm among the four utilized is the IWO algorithm. This will be showcased in the comparative study section. Subsequently, we will present the simulation results of our technique based on the IWO algorithm. The weights and the bias are parts of the proposed ANN model that can be tuned. Below is a list of the various parameters of the IWO algorithm:

- ✓ The Initial and final population size are 10 and 25, respectively.
- ✓ The Minimum and maximum number of seeds are 0 and 5, respectively.
- ✓ The Initial and final values of the standard deviation are 1.5 and -1.5, respectively.

4.1 Modeling and identification of system I

For this system, we consider the particular case governed by the following differential equation:

$$y_p(k+1) = f[y_p(k), y_p(k-1)] + u(k); \quad (14)$$

with:

$$f[y_p, y_p(k-1)] = \frac{y_p(k)y_p(k-1)[y_p(k) + 2.5]}{1 + y_p^2(k) + y_p^2(k-1)}. \quad (15)$$

$$u(k) = \sin\left(\frac{2\pi k}{25}\right) \quad (16)$$

where, f is the part of Eq. (14) to be identified using the primary model, and u is the input signal.

The simulation results of system I are presented in Figure 5, where,

- ✓ Figure 5 (a): Superposition between the system I output and the primary model output.
- ✓ Figure 5 (b): Zoomed segments of Figure 5 (a).
- ✓ Figure 5 (c): Superposition between the modeling error and the model of modeling error.
- ✓ Figure 5 (d): Superposition between the system I output and the final model output.

4.2 Modeling and identification of system II

For the second nonlinear dynamical system, the process to be determined is provided by the following difference equation:

$$y_p(k+1) = 0.3y_p(k) + 0.6y_p(k-1) + f[u(k)] \quad (17)$$

where, the following form represents the unknown function that has to be found:

$$f(u) = 0.6 \sin(\pi u) + 0.3 \sin(3\pi u) + 0.1 \sin(5\pi u) \quad (18)$$

The input signal u is chosen to be sinusoidal as follows:

$$u(k) = \sin\left(\frac{2\pi k}{250}\right) \quad (19)$$

The same parameters of the IWO optimization algorithm as those used for system I are used to simulate the second system. Modeling and identification, the simulation results are presented in Figure 6.

4.3 Modeling and identification of system III

For this system, we consider the particular case described by the following difference equation:

$$y_p(k+1) = f(y_p(k), u(k)) = \frac{y_p(k)}{1 + y_p(k)^2} + u(k)^3 \quad (20)$$

$$u(k) = \sin\left(\frac{2\pi k}{25}\right) + \sin\left(\frac{2\pi k}{10}\right) \quad (21)$$

where, $u(k)$ is the input signal. We simulated this case with the same parameters of the IWO optimization algorithm as with system I and system II. The simulation results are shown in Figure 7.

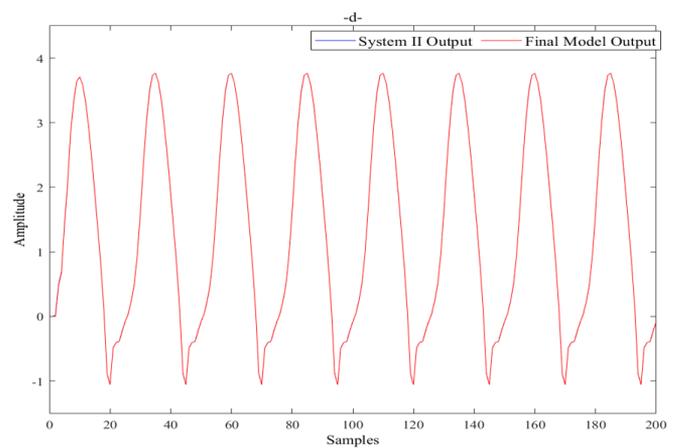
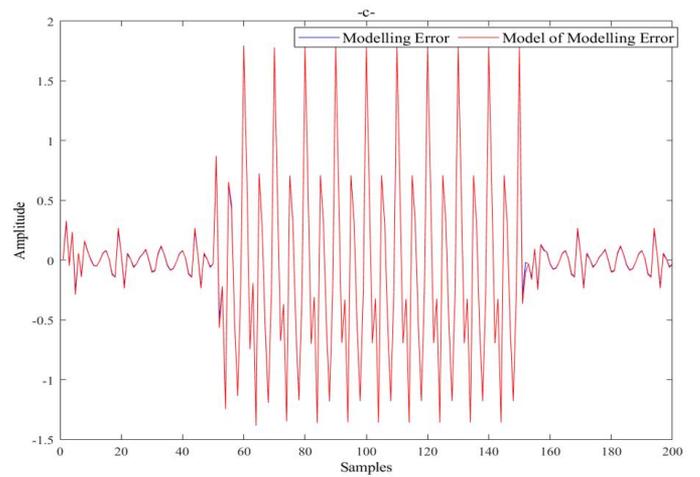
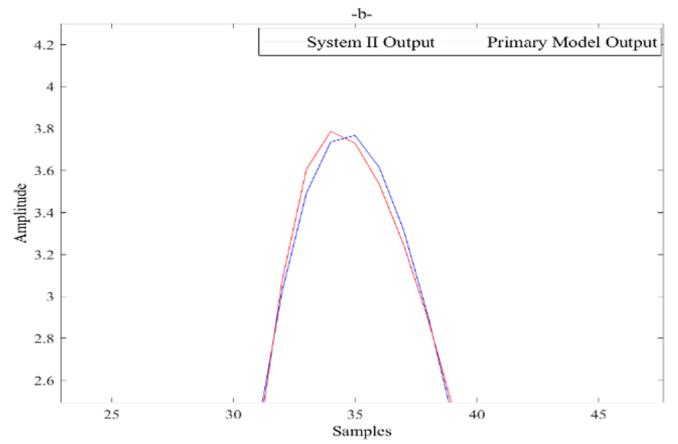
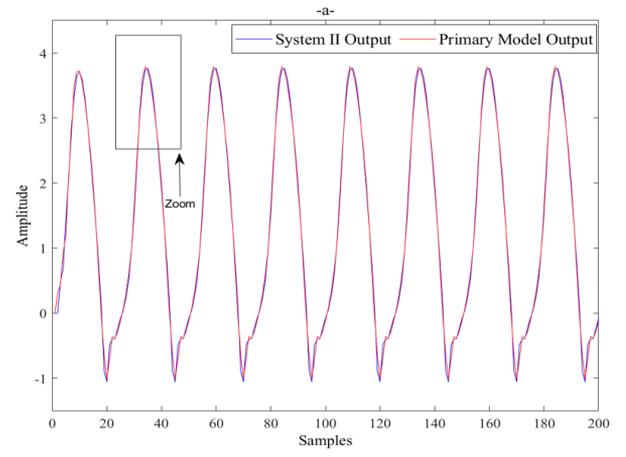
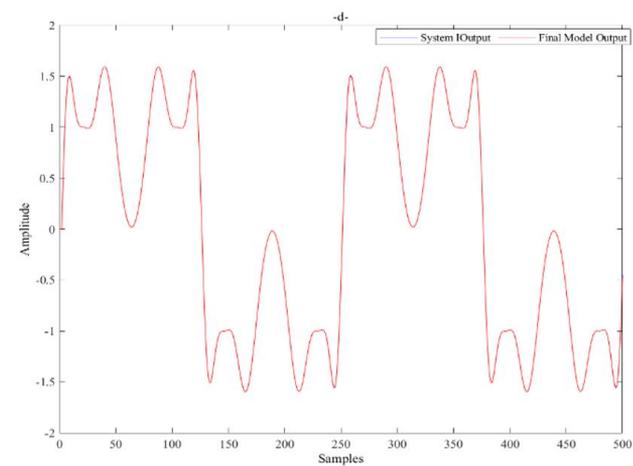
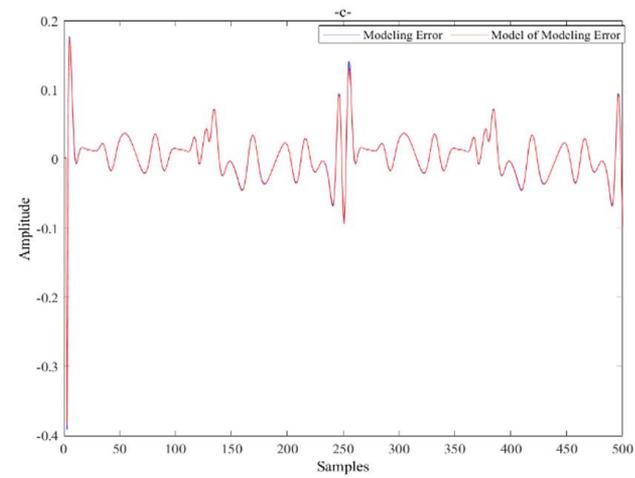
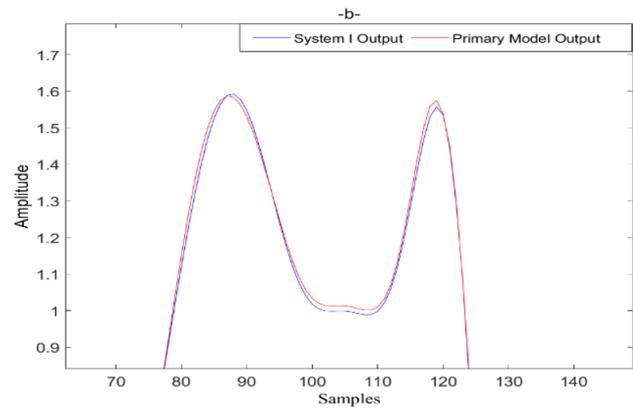
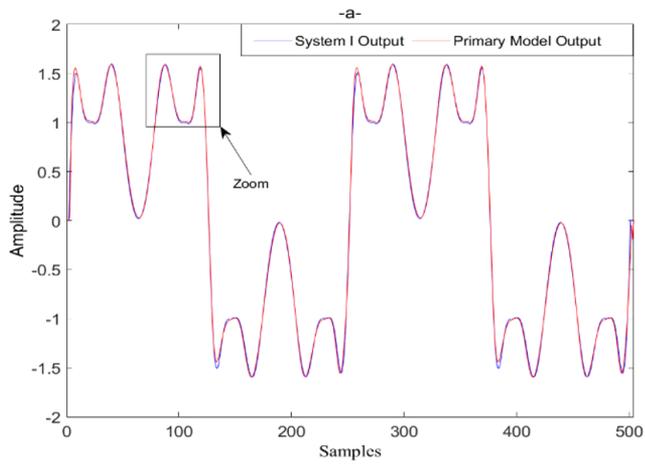


Figure 5. ANN-based IWO model for system I

Figure 6. ANN-based IWO model for system II

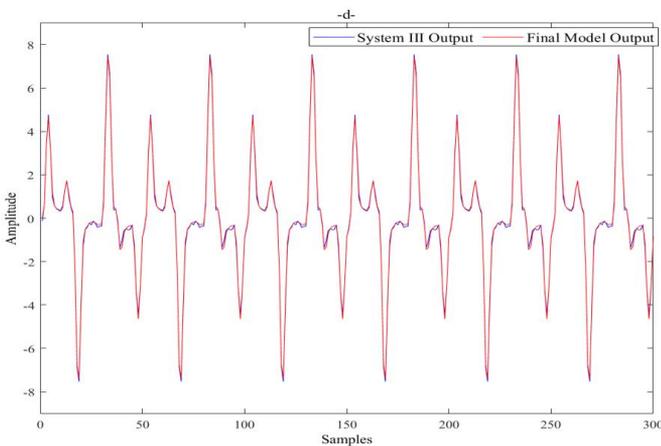
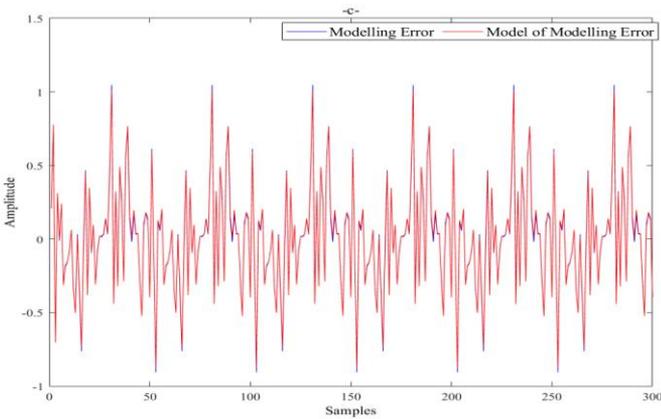
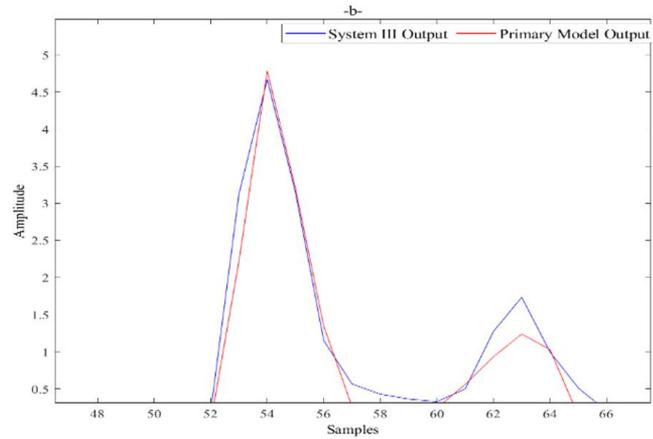
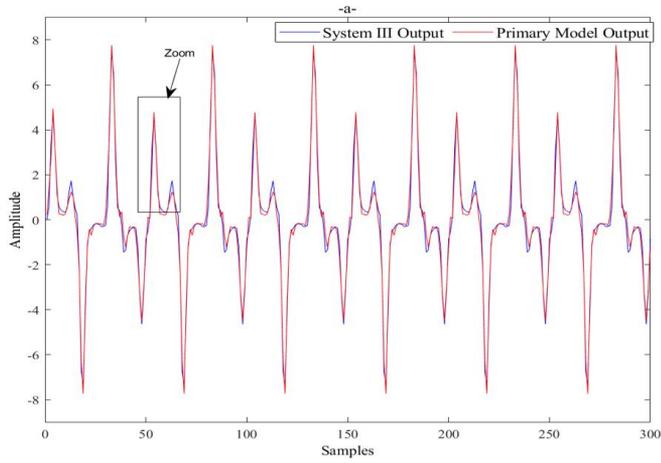


Figure 7. ANN-based IWO model for system III

A visual examination of all of these Figures 5-7 reveals that the ultimate model is greatly superior to the primary model. This observation verifies the usefulness of the error model-based identification idea.

4.4 Validation and generalization tests

To ensure both efficiency and robustness in our approach, validation tests have been conducted. A concise description of these validation tests is provided in this section.

4.4.1 Generalization test

The generalization process follows these steps: First, the primary model is validated using new input data u_2 , resulting in a new error. Next, this error is utilized in the error identification step. Finally, the final model is generated, representing a concurrent interconnection involving two models (primary model and error model). The outcomes of the validation process are illustrated in Figure 8 where:

- ✓ Figure 8 (a): represents the primary model output with the input data u_1 defined as follows:

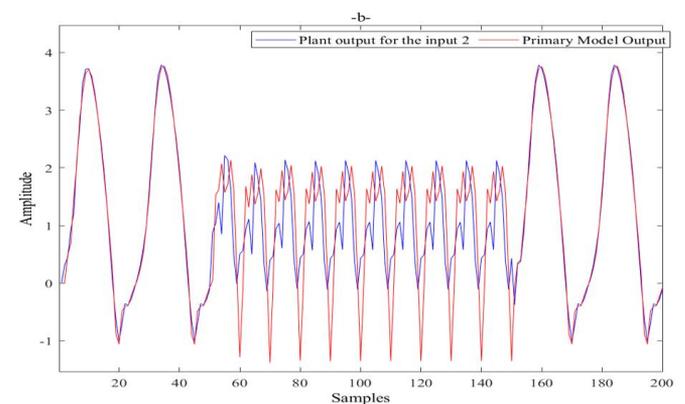
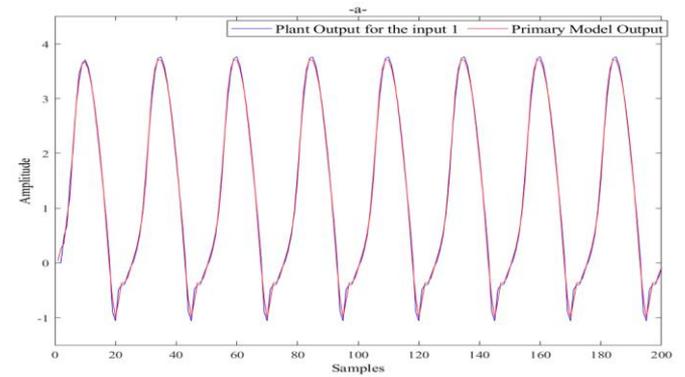
$$u_1(k) = \sin\left(\frac{2\pi k}{25}\right) \quad (22)$$

- ✓ Figure 8 (b): represents the primary model output with the new input data u_2 given by the following equation:

$$u_2(k) = \sin\left(\frac{2\pi k}{25}\right) \text{ for } 1 \leq k \leq 50 \text{ and } 150 \leq k \leq 200 \quad (23)$$

$$u_2(k) = \sin\left(\frac{2\pi k}{10}\right) + \sin\left(\frac{2\pi k}{5}\right) \text{ for } 50 \leq k \leq 150 \quad (24)$$

- ✓ Figure 8 (c): represents the error process model.
- ✓ Figure 8 (d): represents the final model output.



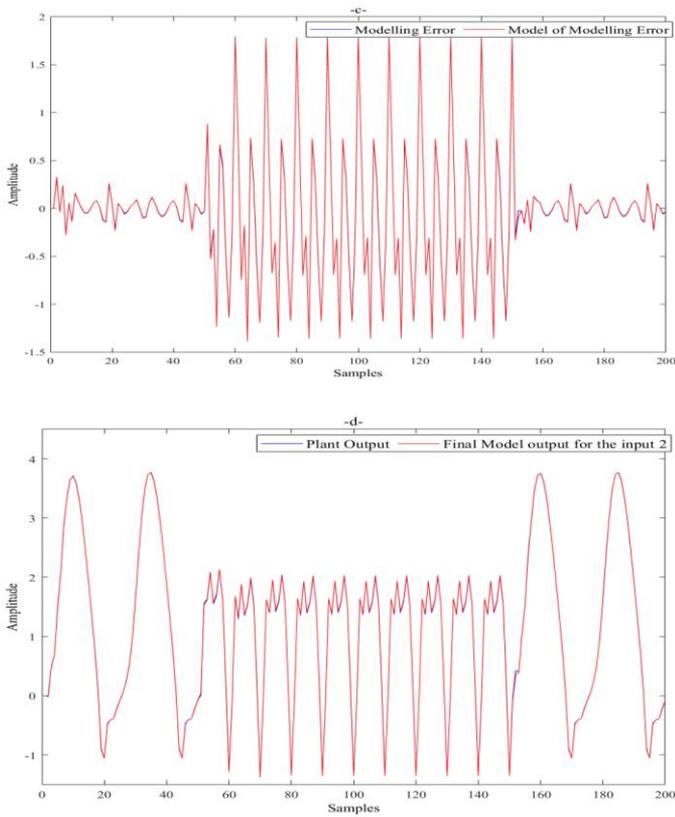


Figure 8. Generalization test

The generalization test for models is a crucial step in the evaluation of machine learning and statistical models. It assesses how well a trained model can perform on unseen or new data. Based on a visual examination of Figure 8 (d), it can be verified that our model demonstrates satisfactory performance on applying a new input, which confirms the effectiveness of our proposed approach.

4.5 Validation test

By conducting validation tests, we can assess the model's reliability, generalization ability, and suitability for real-world applications. These tests are crucial in ensuring that the model is not only accurate on the data it was trained on but also effective in making predictions on new, unseen data.

4.5.1 Modeling and identification of ECG signal

An ECG signal is a type of time series data which illustrates the heart's electrical activity over a particular period of time. In a time series, data points are recorded in chronological order at regular intervals. In the case of an ECG, the time series consists of a sequence of voltage measurements taken at successive time points during the cardiac cycle. In this section, our approach is applied to the identification of two types of ECG signals: Real ECG signals acquired from the ECG PhysioNet database [30] and synthetic ECG signal [31].

(1). Real ECG signal

In the following, we explore the implementation of the suggested approach on real ECG data. To conduct this study, we obtained the real ECG signal 100.dat dataset from the MIT-BIH normal sinus rhythm database [30], where it was recorded at a sampling rate of 360Hz with a resolution of 11 bits per sample. The outcome of applying the proposed method to the real ECG signal is visualized in Figure 9.

(2). Synthetic ECG signal

The same precepts and procedures used in the previous sections will be applied to model the synthetic ECG signal data [31]. The result gained is shown in Figure 10.

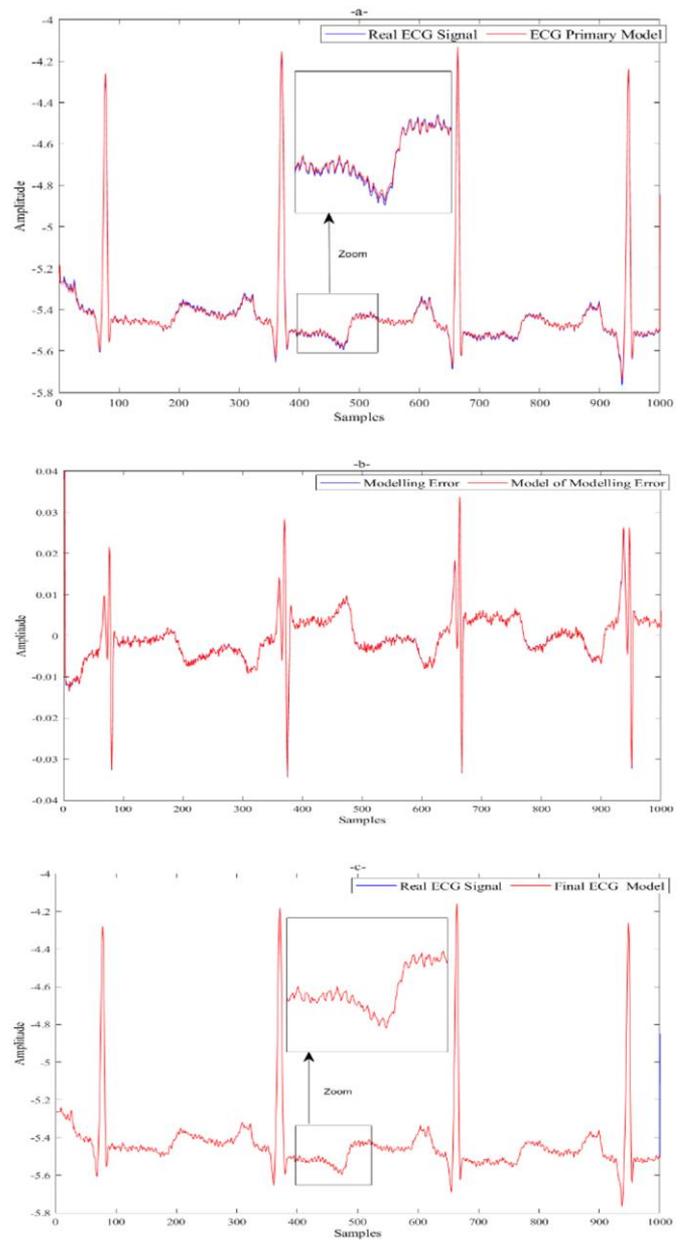
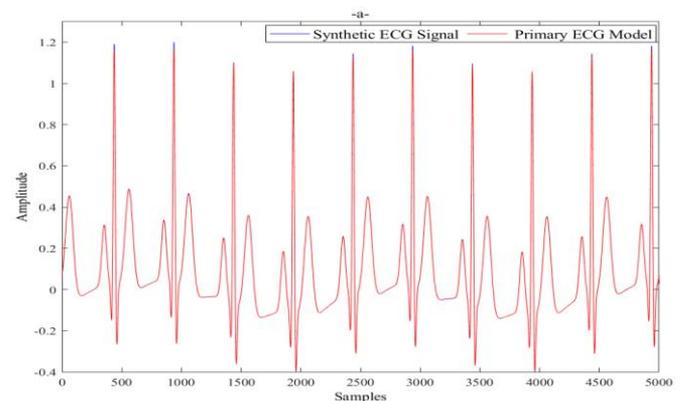


Figure 9. ANN based IWO model for real ECG signal: a) Real ECG signal vs Primary ECG model, b) the Modeling error vs Model of modeling error, c) the Real ECG signal vs the Final ECG model



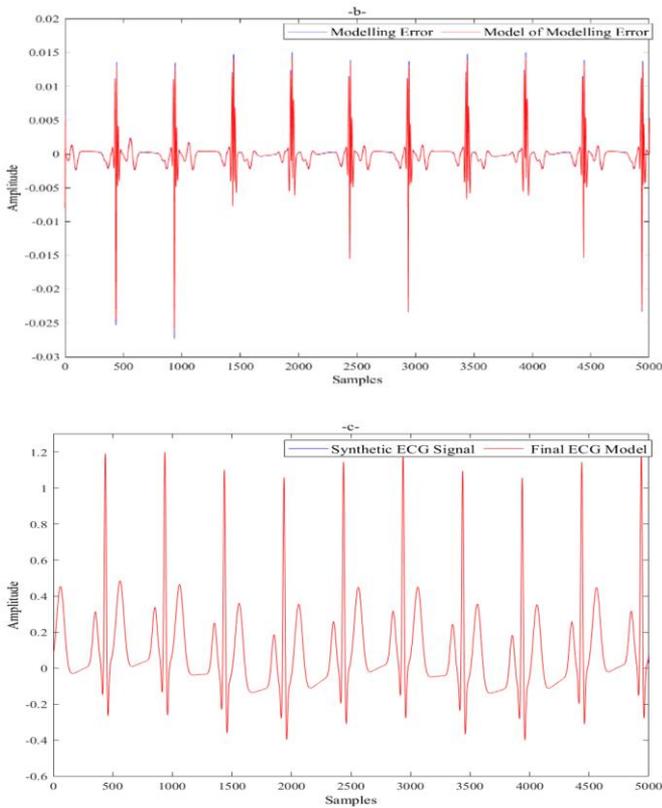


Figure 10. ANN based IWO model for synthetic ECG signal: a) synthetic ECG signal vs Primary ECG model, b) Modeling error vs Model of Modeling error, c) synthetic ECG signal vs Final ECG model

4.5.2 Mackey-Glass time series modelling and identification

Remember that a time series is only a set of data points organized temporally. Our method is also used to additional data from a time series. Time often serves as the independent variable in a time series, and future forecasting is the main goal. We take into consideration a time series produced by the Mackey-Glass equation for this purpose. Figure 11 shows the simulation's results.

4.6 Comparative study

In this section, a comparison is conducted to demonstrate the efficacy of the IWO optimization algorithm in contrast to other optimization techniques. To accomplish this, we have selected three algorithms, namely PSO, ICA, and ES-CMA, as described in section.1.2. The specific parameters for each optimization algorithm are chosen as follows: PSO algorithm parameters (the acceleration constants $C_1 = 1.5$, $C_2 = 2.5$ and the coefficient of inertia $\omega = 0.48$), ICA algorithm parameters ($\alpha = 1$, $\beta = 1.5$, μ (revolution rate) =0.1), ES-CMA algorithm parameters (λ (Population size) =140, μ (Number of Parents) =40). Figures 12-14 depict the modeling and identification results of the three dynamical systems mentioned earlier, employing the proposed modelling method based on the four-optimization algorithms (IWO, PSO, ICA, and ES-CMA).

After analysing all the figures, it is evident that the IWO algorithm exhibits superior performance when compared to the PSO, ICA, and ES-CMA algorithms. The results indicate that the IWO algorithm outperform the other optimization techniques in terms of Modeling and identification of the dynamical systems under consideration.

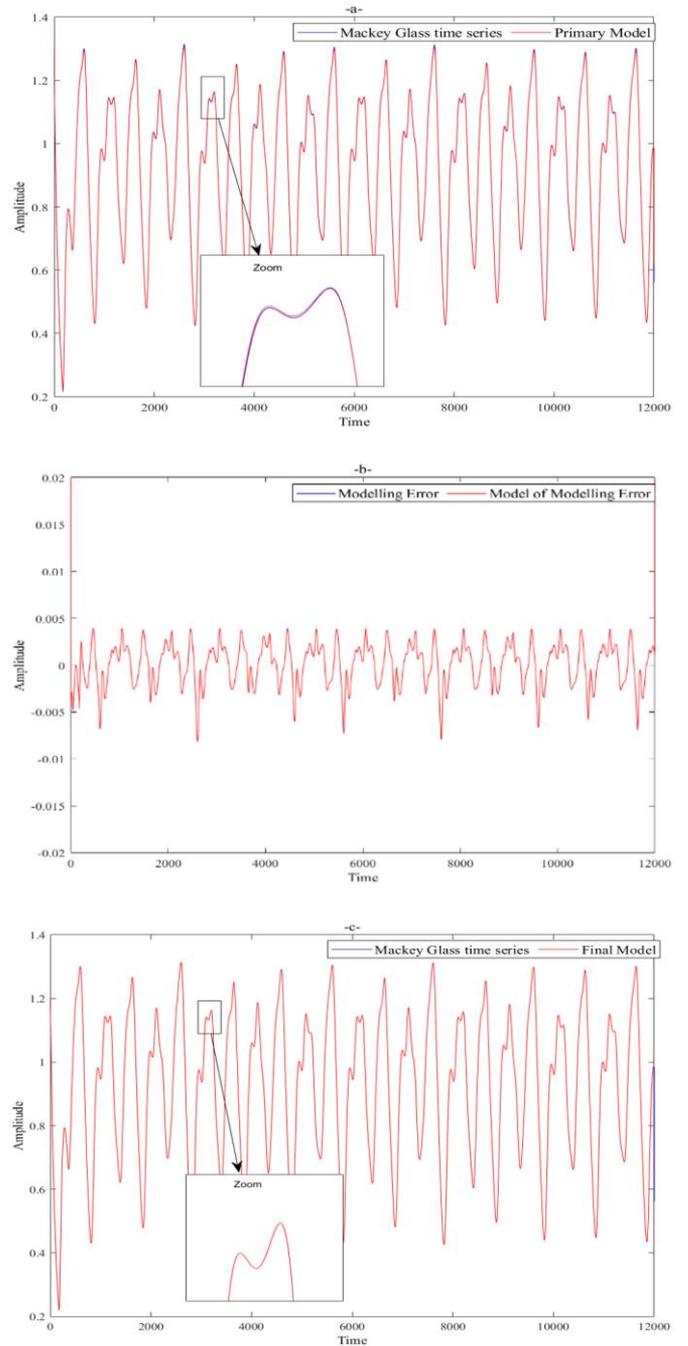
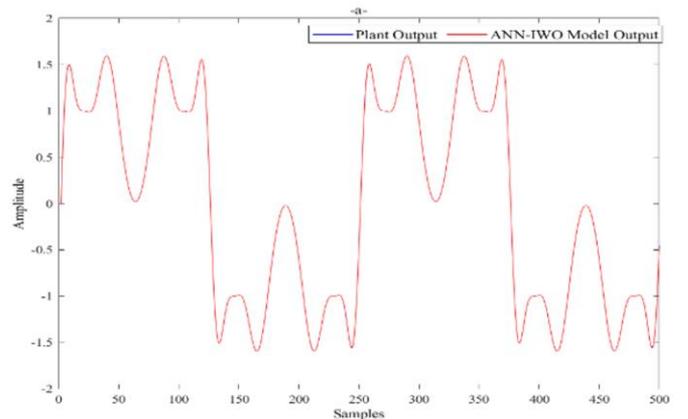


Figure 11. ANN based IWO model for Mackey Glass time series: a) Mackey Glass time series vs Primary model, b) Modeling error vs Model of Modeling error, c) Mackey Glass time series vs Final model



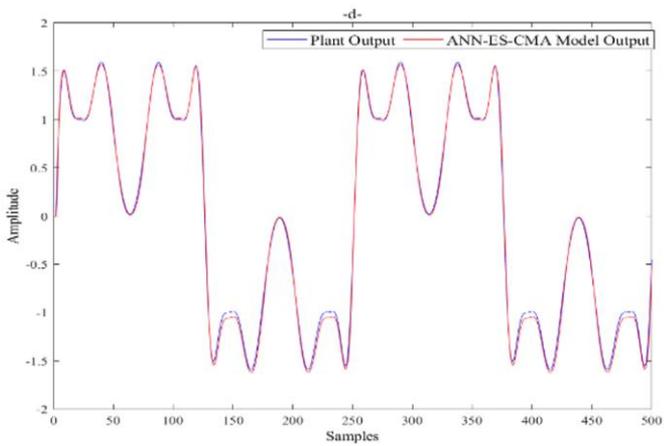
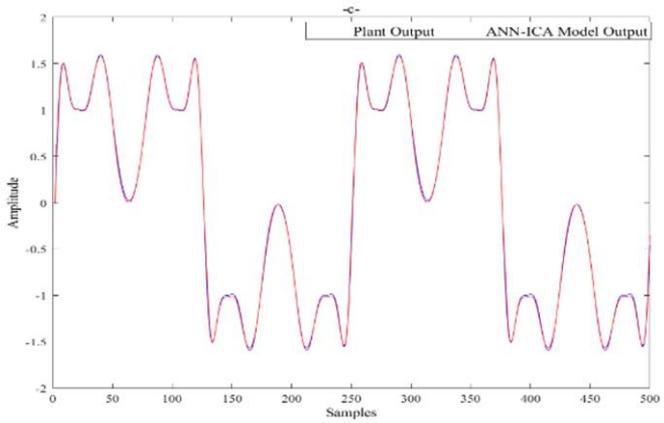
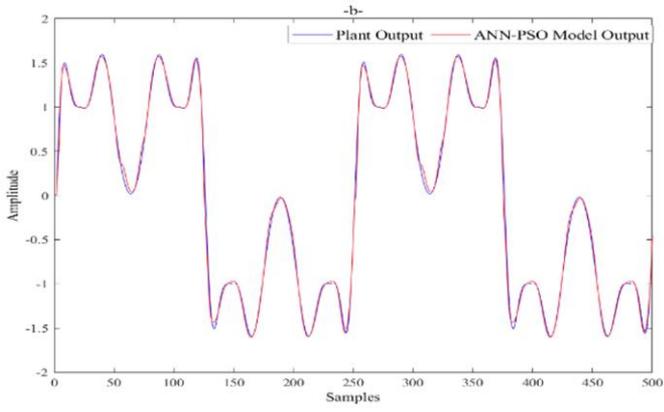


Figure 12. System I based ANN model: (a) IWO, (b) PSO, (c) ICA, (d) ES-CMA

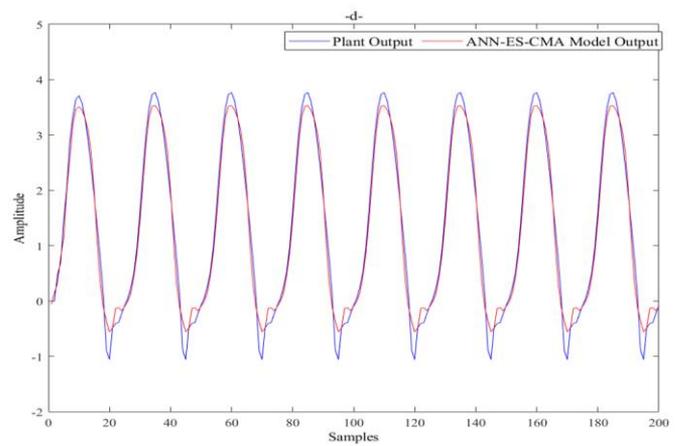
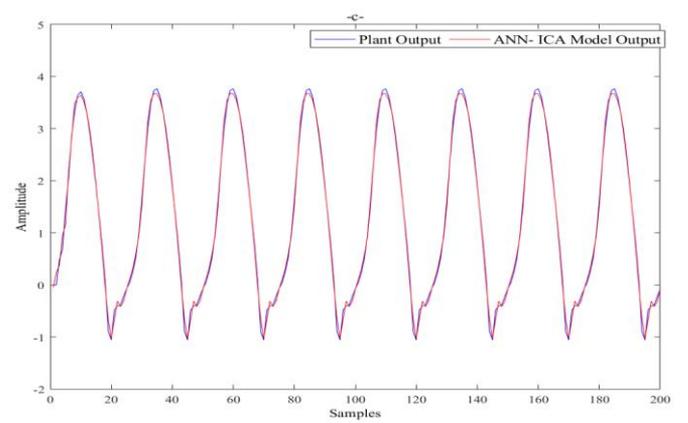
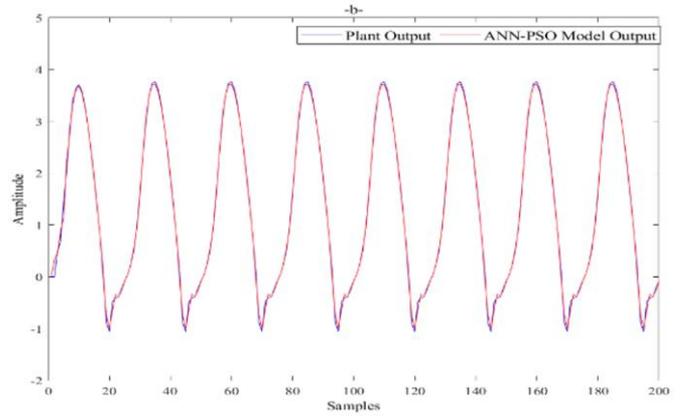
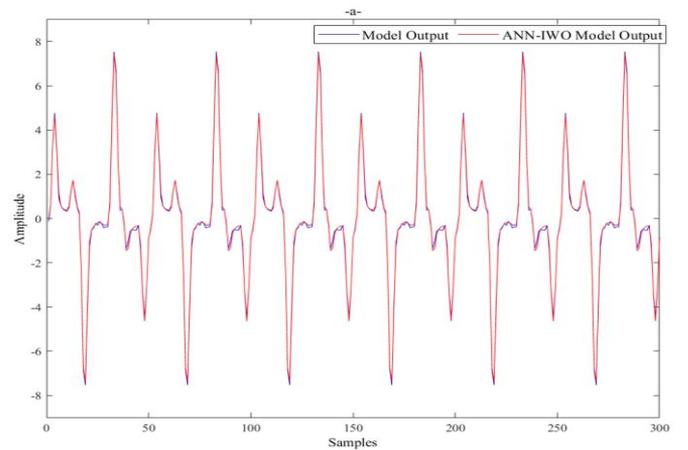
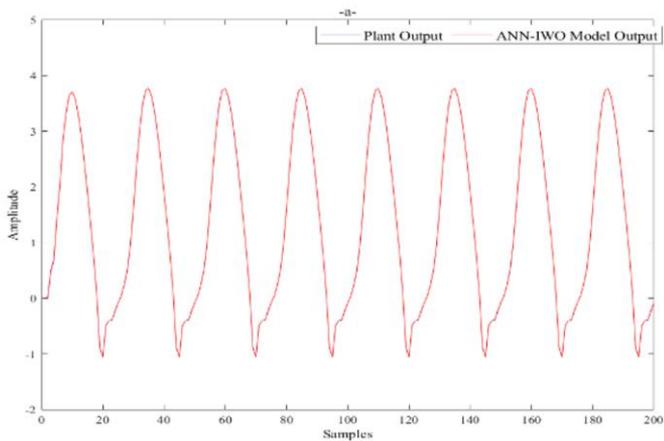


Figure 13. System II based ANN model: (a) IWO, (b) PSO, (c) ICA, (d) ES-CMA



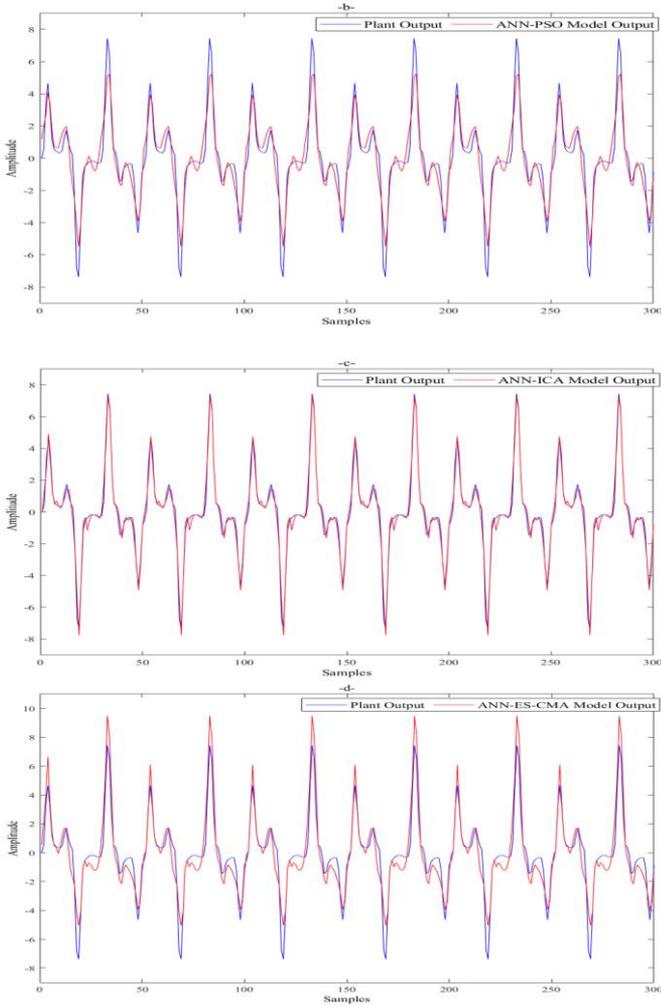


Figure 14. System III based ANN model: (a) IWO, (b) PSO, (c) ICA, (d) ES-CMA

Currently, we proceed with a numerical evaluation of the method's performance using a fitness function known as the mean square error (MSE). To ensure reliability, we conducted 20 independent trials of our method and for each optimization algorithm. Table 1 presents statistical performance measures, including the worst and best values of the fitness function.

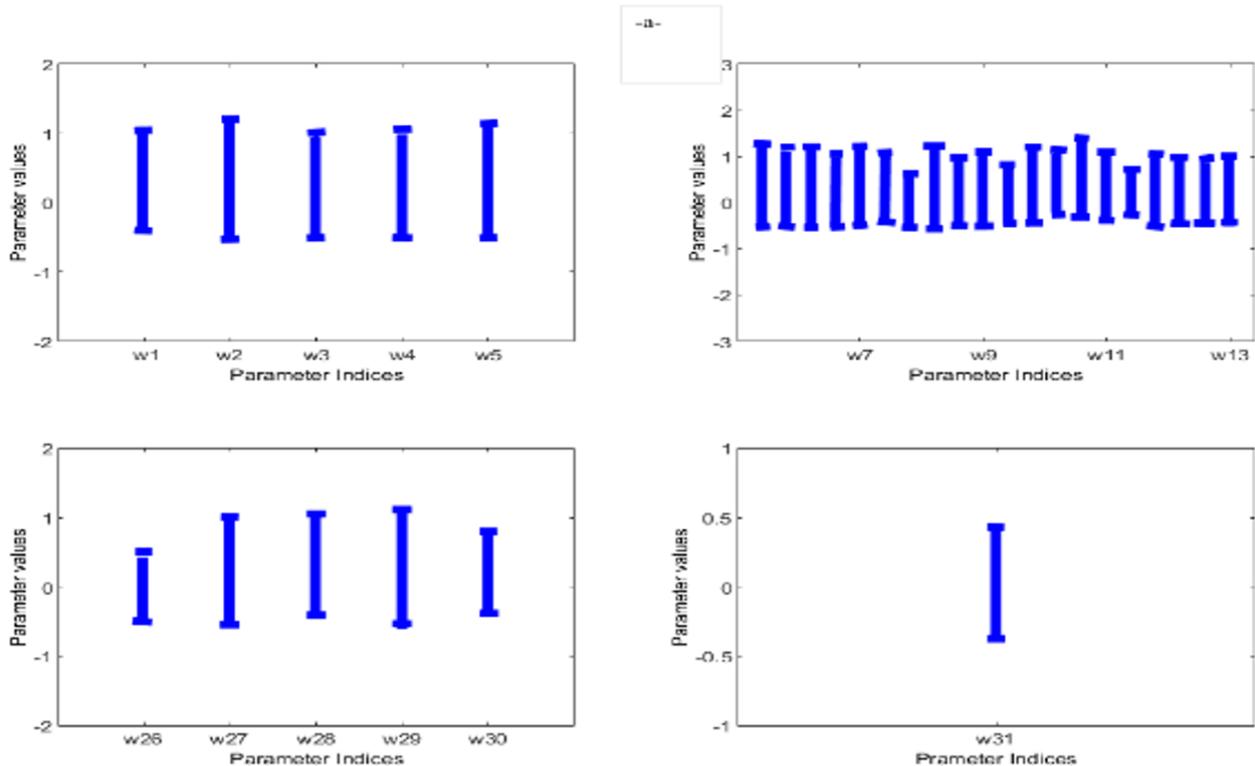
Table 1. The fitness function results for 20 independent trials

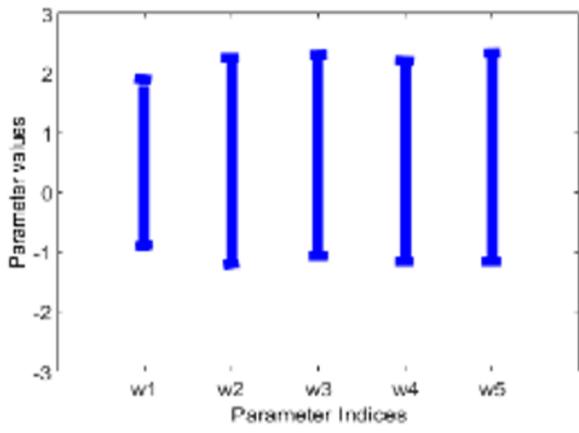
Optimization Algorithm	Best Value	Worst Value
IWO	9.1127e-9	8.1002e-7
PSO	8.2201e-6	5.9317e-4
ICA	4.2138e-6	1.2139e-5
ES-CMA	4.2138e-6	1.7293e-5

Based on Table 1 and out of the techniques discussed, the IWO (Invasive Weed Optimization) algorithm demonstrated superior performance by achieving the best value across the 20 independent trials.

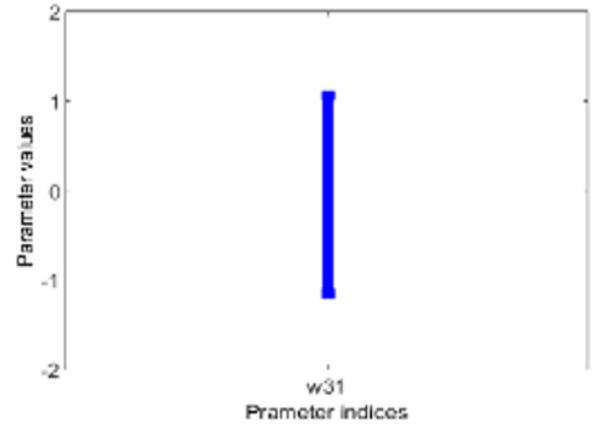
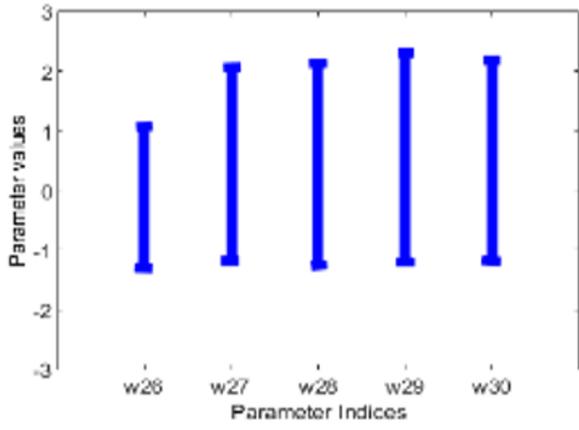
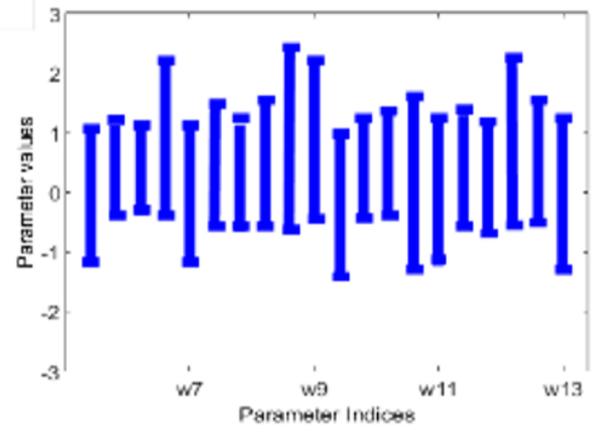
Moreover, to conduct comprehensive statistical investigations, we incorporated error bars for parameter optimization. This graphical technique represents the variability of the estimated parameters on graphs, providing an indication of the uncertainty associated with estimates and offering a general understanding of the parameter's values accuracy. The primary model's error bars parameters and the error process Modeling parameters can be observed in Figures 15 and 16, respectively.

A simple visual inspection of these figures, indicating that the error bars widths associated with the IWO method are the smallest when compared to the error bars of the PSO method, the ICA method, and CMA methods. This finding suggests that the IWO method exhibits greater precision and consistency in parameter optimization.

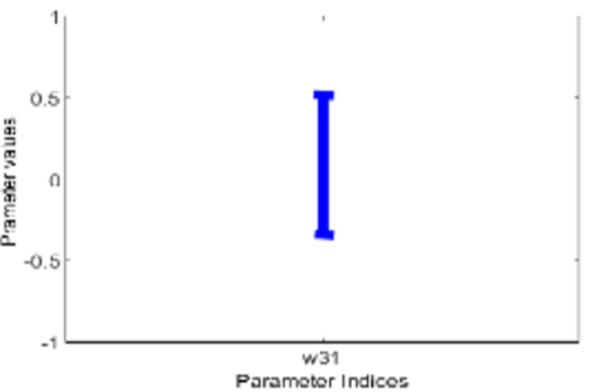
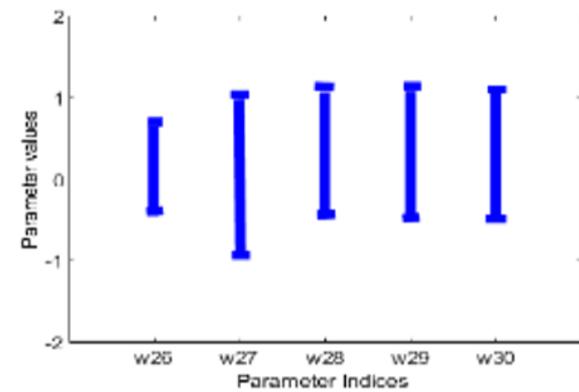
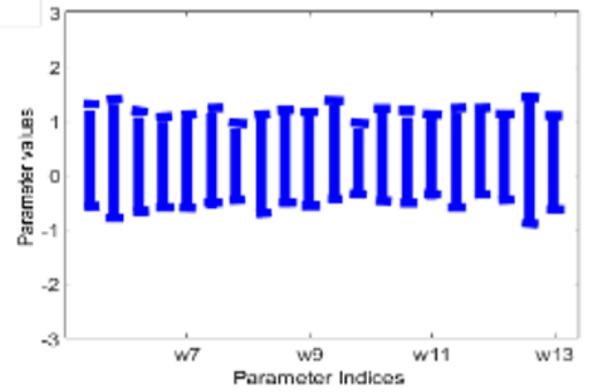
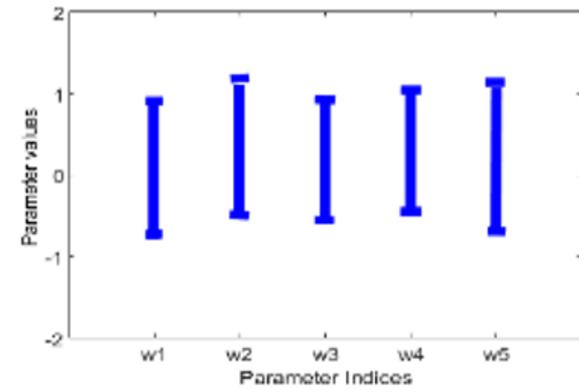




-b-



-c-



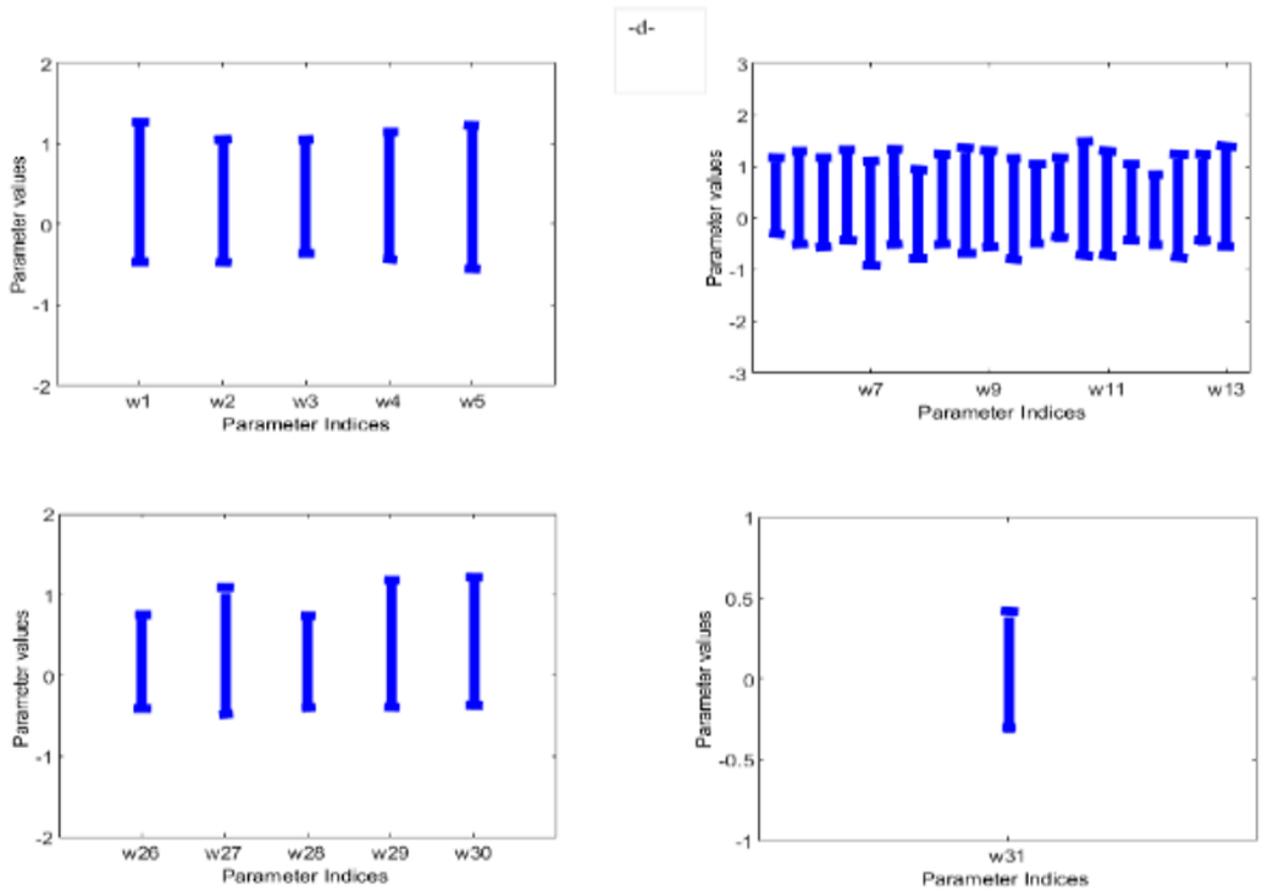
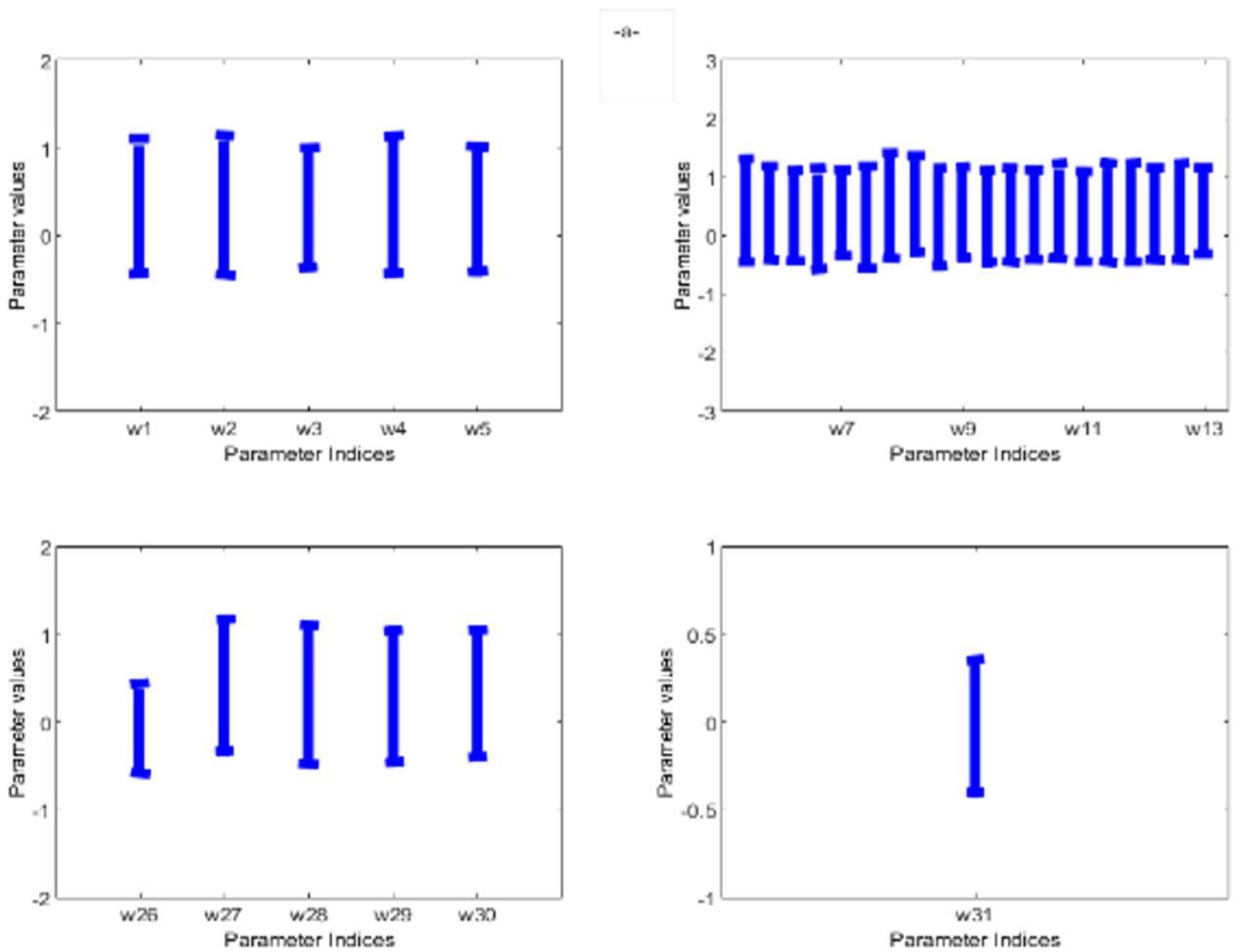
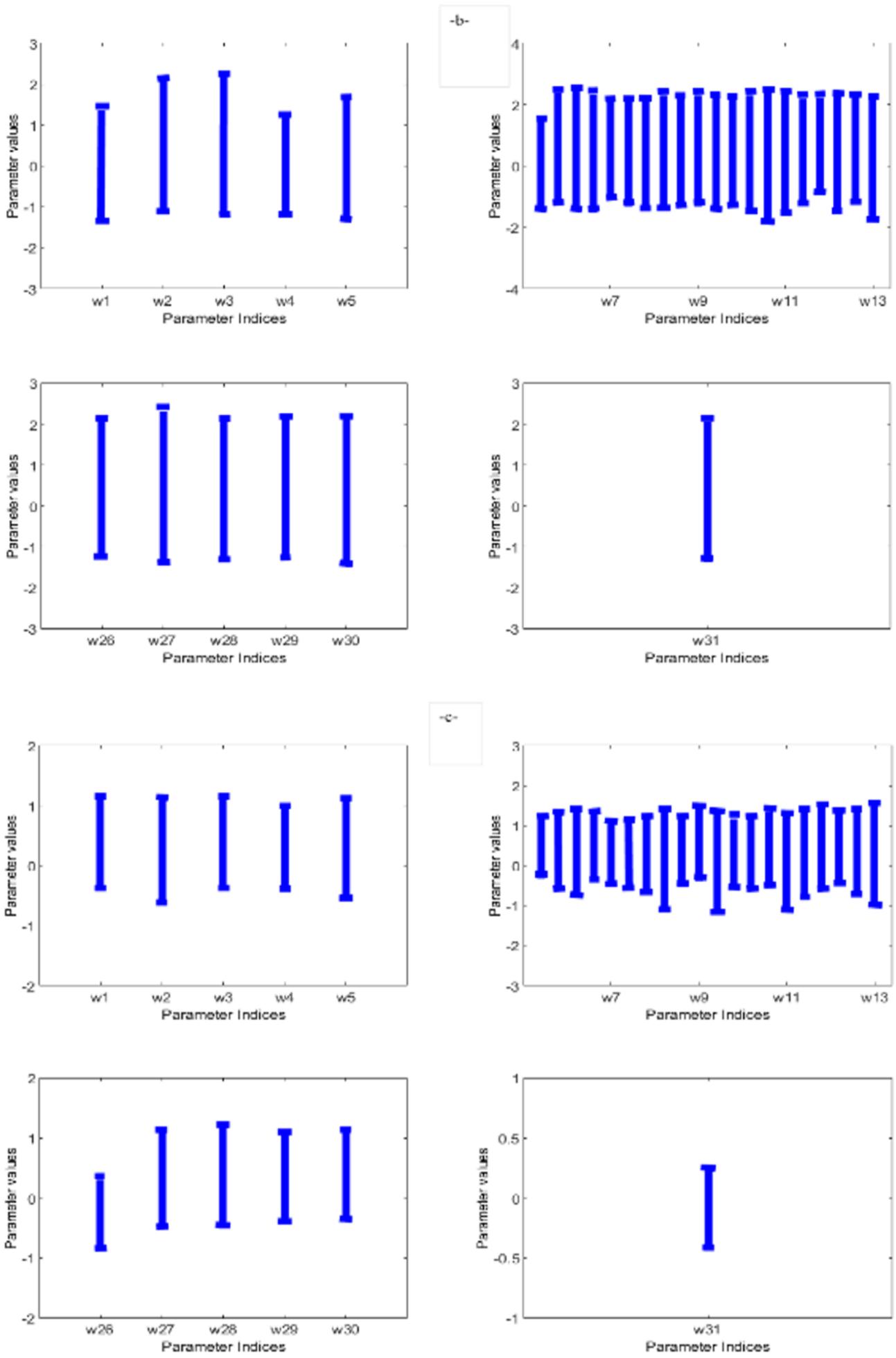


Figure 15. Primary model parameters error bars: (a) IWO, (b) PSO, (c) ICA, (d) ES-CMA





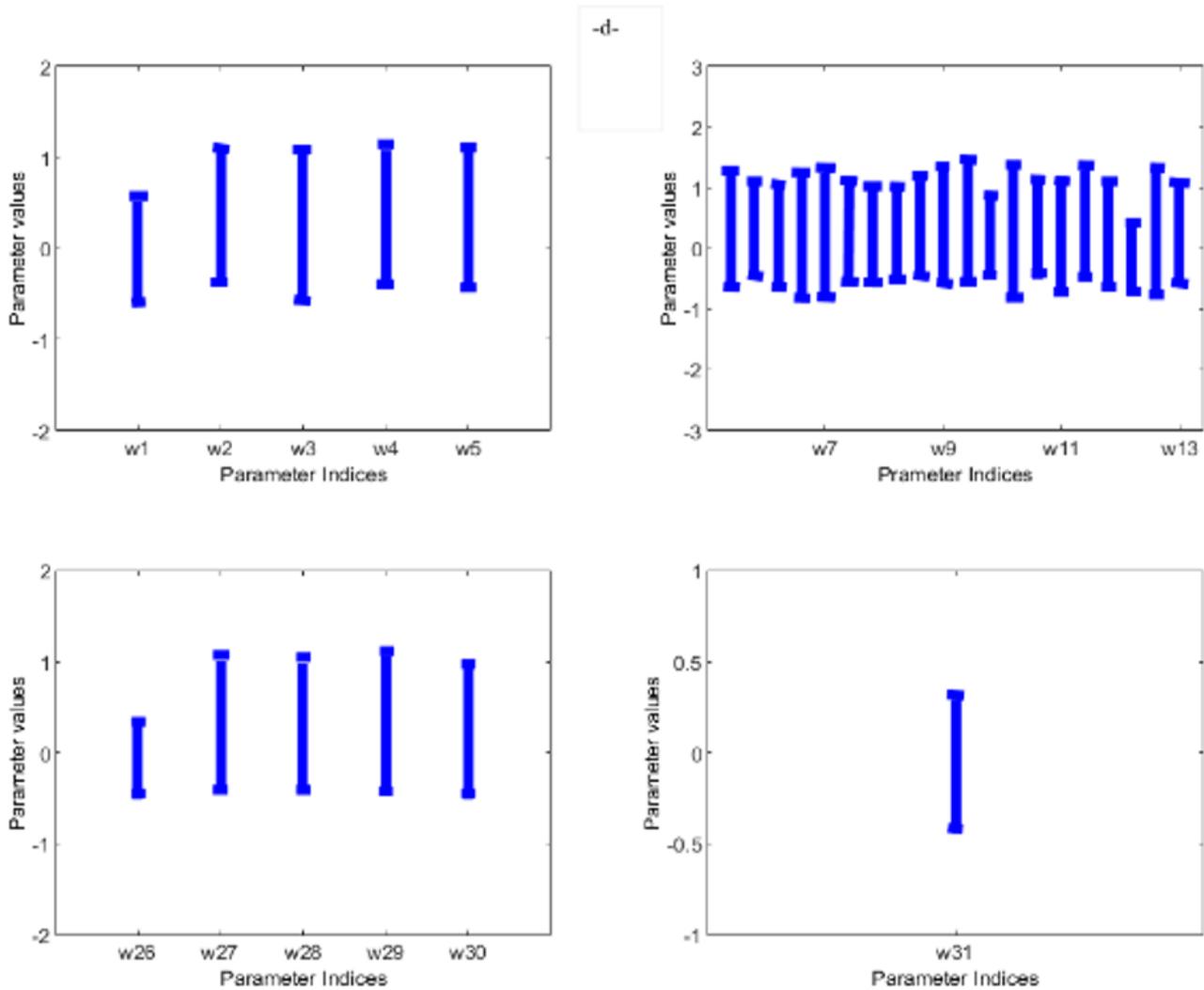


Figure 16. Error process parameters error bars: (a) IWO, (b) PSO, (c) ICA, (d) ES-CMA

4.7 Discussion

Our approach, which combines ANN and IWO methods and incorporates an error model, improves the modeling and identification of dynamical systems and time series significantly. By combining ANN and IWO, we increase efficiency and produce better solutions in situations where traditional methods may fail. This collaboration between ANN and IWO not only improves the ability to solve complex problems, but also enables us to deal with difficult scenarios more effectively. Our findings have a significant impact because they provide better predictions and optimization strategies for real-world tasks in fields such as finance and engineering. Our adaptable method is a valuable tool for control and decision-making in dynamical systems, making it applicable across industries and allowing us to address complex challenges effectively.

5. CONCLUSION

In this research paper, we introduced a novel strategy to address common challenges in Modeling and identification of dynamical systems and times series. Our approach involves combining hybrid Artificial Neural Network Autoregressive Moving Average (ANNARMA) with metaheuristics

algorithms. Through this integration, we aim to tackle the classical problems that arise in this domain effectively. The presented approach introduces an innovative identification module known as the “error model.” This module serves as a valuable supplement to the primary model, enhancing its overall quality and leading to a more precise fit. As a result, the proposed approach yields a higher resolution model with improved accuracy. To achieve optimization in ANN identification, various metaheuristic algorithms, such as ICA, PSO, CMA-ES, and IWO, have been applied. These algorithms play a crucial role in refining the ANN identification process and enhancing its efficiency. The effectiveness of the proposed method is validated through simulation results and comparative studies. The outcome of these comparisons demonstrates that IWO method outperformed the other metaheuristic algorithms utilized in this study, providing the best optimization results. The superiority of IWO further reinforces the credibility and efficiency of the proposed approach in Modeling and identification of dynamical systems.

REFERENCES

- [1] Van Geert, P., Steenbeek, H. (2005). Explaining after by before: Basic aspects of a dynamic systems approach to

- the study of development. *Developmental Review*, 25(3-4): 408-442. <https://doi.org/10.1016/j.dr.2005.10.003>
- [2] Niven, R.K., Mohammad-Djafari, A., Cordier, L., Abel, M., Quade, M. (2020). Bayesian identification of dynamical systems. *Multidisciplinary Digital Publishing Institute Proceedings*, 33(1): 33. <https://doi.org/10.3390/proceedings2019033033>
- [3] Quaranta, G., Lacarbonara, W., Masri, S.F. (2020). A review on computational intelligence for identification of nonlinear dynamical systems. *Nonlinear Dynamics*, 99(2): 1709-1761. <https://doi.org/10.1007/s11071-019-05430-7>
- [4] Qiao, J.F., Han, H.G. (2012). Identification and modeling of nonlinear dynamical systems using a novel self-organizing RBF-based approach. *Automatica*, 48(8): 1729-1734. <https://doi.org/10.1016/j.automatica.2012.05.034>
- [5] Yu, H., Xie, T., Paszyczynski, S., Wilamowski, B.M. (2011). Advantages of radial basis function networks for dynamic system design. *IEEE Transactions on Industrial Electronics*, 58(12): 5438-5450. <https://doi.org/10.1109/TIE.2011.2164773>
- [6] Jinfeng, L., Fucai, L., Yaxue, R. (2020). Fuzzy identification of nonlinear dynamic system based on input variable selection and particle swarm optimization parameter optimization. *IEEE Access*, 8: 220557-220569. <https://doi.org/10.1109/ACCESS.2020.3043275>
- [7] Rajasekaran, S., Pai, G.V. (2017). *Neural networks, fuzzy systems and evolutionary algorithms: Synthesis and applications*. PHI Learning Pvt. Ltd.
- [8] Mira, J. (2001). *Connectionist models of neurons, learning processes, and artificial intelligence*. 6th International Work-Conference on Artificial and Natural Neural Networks, IWANN 2001 Granada, Spain, Proceedings. Springer Science & Business Media, Vol. 2084.
- [9] Biagetti, G., Crippa, P., Falaschetti, L., Turchetti, C. (2019). A machine learning approach to the identification of dynamical nonlinear systems. In 2019 27th European Signal Processing Conference (EUSIPCO), Coruna, Spain, IEEE, pp. 1-5. <https://doi.org/10.23919/EUSIPCO.2019.8902539>
- [10] Rajendra, P., Brahmajirao, V. (2020). Modeling of dynamical systems through deep learning. *Biophysical Reviews*, 12(6): 1311-1320. <https://doi.org/10.1007/s12551-020-00776-4>
- [11] Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A., Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11). <https://doi.org/10.1016/j.heliyon.2018.e00938>
- [12] Elbrächter, D., Perekrestenko, D., Grohs, P., Bölcskei, H. (2021). Deep neural network approximation theory. *IEEE Transactions on Information Theory*, 67(5): 2581-2623. <https://doi.org/10.1109/TIT.2021.3062161>
- [13] Chen, S.H., Jakeman, A.J., Norton, J.P. (2008). Artificial intelligence techniques: An introduction to their use for modelling environmental systems. *Mathematics and Computers in Simulation*, 78(2-3): 379-400. <https://doi.org/10.1016/j.matcom.2008.01.028>
- [14] Chen, S., Billings, S. A. (1992). Neural networks for nonlinear dynamic system modelling and identification. *International Journal of Control*, 56(2): 319-346. <https://doi.org/10.1080/00207179208934317>
- [15] Huang, C.C., Loh, C.H. (2001). Nonlinear identification of dynamic systems using neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 16(1): 28-41. <https://doi.org/10.1111/0885-9507.00211>
- [16] Narendra, K.S., Parthasarathy, K. (1992). Neural networks and dynamical systems. *International Journal of Approximate Reasoning*, 6(2): 109-131. [https://doi.org/10.1016/0888-613X\(92\)90014-Q](https://doi.org/10.1016/0888-613X(92)90014-Q)
- [17] Narendra, K.S., Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1): 4-27. <https://doi.org/10.1109/72.80202>
- [18] Loussifi, H., Nouri, K., Braiek, N.B. (2016). A new efficient hybrid intelligent method for nonlinear dynamical systems identification: The wavelet kernel fuzzy neural network. *Communications in Nonlinear Science and Numerical Simulation*, 32: 10-30. <https://doi.org/10.1016/j.cnsns.2015.08.010>
- [19] Cavuslu, M.A., Karakuzu, C., Karakaya, F. (2012). Neural identification of dynamic systems on FPGA with improved PSO learning. *Applied Soft Computing*, 12(9): 2707-2718. <https://doi.org/10.1016/j.asoc.2012.03.022>
- [20] Singh, M., Srivastava, S., Gupta, J.R.P., Handmandlu, M. (2007). Identification and control of a nonlinear system using neural networks by extracting the system dynamics. *IETE Journal of Research*, 53(1): 43-50. <https://doi.org/10.1080/03772063.2007.10876120>
- [21] Jovanović, O. (1997). Identification of dynamic system using neural network. *The Scientific Journal FACTA UNIVERSITATIS Series: Architecture and Civil Engineering*, 31(1997): 525-532.
- [22] Gautam, P. (2016). System identification of nonlinear Inverted Pendulum using artificial neural network. In 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, IEEE, pp. 1-5. <https://doi.org/10.1109/ICRAIE.2016.7939522>
- [23] Patra, J.C., Pal, R.N., Chatterji, B.N., Panda, G. (1999). Identification of nonlinear dynamic systems using functional link artificial neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(2): 254-262. <https://doi.org/10.1109/3477.752797>
- [24] Zheng, D.D., Pan, Y., Guo, K., Yu, H. (2019). Identification and control of nonlinear systems using neural networks: A singularity-free approach. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9): 2696-2706. <https://doi.org/10.1109/TNNLS.2018.2886135>
- [25] Chong, H.Y., Yap, H.J., Tan, S.C., Yap, K.S., Wong, S.Y. (2021). Advances of metaheuristic algorithms in training neural networks for industrial applications. *Soft Computing*, 25(16): 11209-11233. <https://doi.org/10.1007/s00500-021-05886-z>
- [26] Mehrabian, A.R., Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics*, 1(4): 355-366. <https://doi.org/10.1016/j.ecoinf.2006.07.003>
- [27] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, WA, Australia, IEEE, 4: 1942-1948. <https://doi.org/10.1109/ICNN.1995.488968>
- [28] Atashpaz-Gargari, E., Lucas, C. (2007). Imperialist competitive algorithm: An algorithm for optimization

- inspired by imperialistic competition. In 2007 IEEE Congress on Evolutionary Computation, Singapore, IEEE, pp. 4661-4667. <https://doi.org/10.1109/CEC.2007.4425083>
- [29] Hansen, N., Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, MIT Press, 9(2): 159-195. <https://doi.org/10.1162/106365601750190398>
- [30] MIT-BIH Arrhythmia Database. (2005). <https://www.physionet.org/physiobank/database/mitdb/>.
- [31] McSharry, P.E., Clifford, G.D., Tarassenko, L., Smith, L.A. (2003). A dynamical model for generating synthetic electrocardiogram signals. *IEEE Transactions on Biomedical Engineering*, 50(3): 289-294. <https://doi.org/10.1109/TBME.2003.808805>