

Article

Feasibility of Six Metaheuristic Solutions for Estimating Induction Motor Reactance

Halil Gör 

Department of Electrical and Electronics Engineering, Faculty of Engineering, Hakkari University, 30000 Hakkari, Turkey; halilgor@hakkari.edu.tr

Abstract: Industry is the primary application for induction machines. As such, it is essential to calculate the induction devices' electrical properties accurately. With DC testing, no-load rotor tests, and locked rotor tests, one may empirically evaluate the electrical variables of induction motors. These tests are expensive and difficult to conduct, however. The information supplied by machine makers can also be used to accurately approximate the equivalent variables of the circuits in induction machines. This article has successfully predicted motor reactance (X_m) for both double- and single-cage models using artificial neural networks (ANN). Although ANNs have been investigated in the literature, the ANN structures were trained to use unmemorized training. Besides ANN, six other approaches have been suggested to address this issue: heap-based optimization (HBO), leagues championship algorithm (LCA), multi-verse optimization (MVO), osprey optimization algorithm (OOA), cuckoo optimization algorithm (COA), and sooty tern optimization algorithm (STOA). The efficaciousness of the suggested approaches was compared with each other. Regarding the obtained outcomes, the suggested MVO- multi-layer perceptron (MLP) technique performed better than the other five methods regarding reactance prediction, with R^2 of 0.99598 and 0.9962, and RMSE of 20.31492 and 20.80626 in the testing and training phases, respectively. For the projected model, the suggested ANNs have produced great results. The novelty lies in the mentioned methods' ability to tackle the complexities and challenges associated with induction motor reactance optimization, providing innovative approaches to finding optimal or near-optimal solutions. As researchers continue to explore and refine these techniques, their impact on motor design and efficiency will likely grow, driving advancements in electrical engineering.

Keywords: metaheuristic; artificial neural network; appraisal; induction motor; reactance

MSC: 68U01; 68T35; 68T07



Citation: Gör, H. Feasibility of Six Metaheuristic Solutions for Estimating Induction Motor Reactance. *Mathematics* **2024**, *12*, 483. <https://doi.org/10.3390/math12030483>

Received: 6 January 2024
Revised: 30 January 2024
Accepted: 31 January 2024
Published: 2 February 2024



Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Induction devices are suitable for harsh operating settings due to their robust and uncomplicated design. Because of their adaptable, affordable, and dependable architecture, direct grid supply, and single or multi-phase manufacturing capacity, tiny or big powers are widely utilized in daily life and various industries [1]. Among the most favored basic varieties of induction machines are squirrel-cage asynchronous machines. Compared to comparable permanent magnet machines, squirrel-cage induction machines are easier to assemble, less expensive, and require no driver [2]. Understanding the comparable circuit model and its variables is crucial to operating squirrel-cage induction machines correctly and consistently. To the extent that one intends to use the machine in real-world scenarios, examine and test it in lab or simulation settings, and completely optimize it, these characteristics must be understood accurately or very near to reality [3]. Maximum torque, current data, and no-load and locked-rotor tests or full-load tests are utilized to calculate the equivalent circuit variables of squirrel-cage machines [4]. But running these tests is not always safe and feasible, especially when using large, powerful equipment. Machine manufacturers provide information about machine specs in their catalogs [5].

However, the manufacturer's data does not provide the machine's corresponding circuit variables. It is possible to approximate the device's equivalent circuit parameters by using information provided by the manufacturer. Recent studies on asynchronous machine parameter estimation have seen a sharp rise. The literature presents a few important and thorough investigations of methods based on calculating the machine parameters and performance of asynchronous machines [6–8]. A novel approach to determining the double-cage method parameters of squirrel-cage induction motors was presented in a study. The machine impedance was measured at several places to approximate the motor characteristics [9]. Through the use of machine labels and manufacturer catalog information, a new, simple, and non-repetitive methodology for deriving equivalent circuit characteristics of an asynchronous machine was described in another study. A slip function was used to simulate the variation in the rotor parameters [10].

The motor variables' accuracy utilized by the control algorithm determines how successful the controller is, which is why parameter determination of the induction motor is a significant topic in the literature on electric drives [11]. Squirrel-cage induction motor models are often constructed using single- or double-cage versions. These methods' variables can be acquired in two different models [12]:

- Using information regarding the maximum torque, current, and comprehensive load test results [13].
- Making use of the findings from the locked rotor and no-load evaluations.

Accurate forecasting of the behavior and induction motors' efficiency has challenged designers. Efficiency forecasting has become more accurate when electromagnetic field problems are solved numerically [14]. The finite element method is one of the most effective and widely utilized machine analysis and design tools [11,15]. However, the numerical approach's primary deficiency is the high processing time and resource requirement [16]. The finite element method can be used during machine design to examine performance and optimize the motor design iteratively.

Moreover, the complete computing procedure may be repeated while creating a special machine that satisfies various constraints and requirements [17]. Consequently, developing a model replicating finite element data and determining the necessity and adequate machine efficiency more quickly while retaining high precision would be an invaluable and crucial design instrument [18]. Artificial neural networks (ANNs) fit such a method well [19].

As previously mentioned, an induction motor's reactance is a crucial factor influencing its efficiency and performance. It represents the resistance to the alternating current flow provided by the motor's windings. In this way, the ANNs can be applied to several parts of the study of induction motors, such as the motor's reactance prediction and modeling. ANNs are robust computational models modeled after the human brain's composition and operation [20]. They are made up of artificial neurons that are networked together and can learn from data to anticipate the future or carry out tasks. Recently, various techniques for determining the characteristics of induction motors have been developed by researchers [12,21,22]. One of these techniques was used to identify the characteristics of induction motors operating in the electromechanical mode using the transient stator current.

Regarding the squirrel-cage model of induction motors presented in Ref. [23], new parameters are determined by estimating the mechanical speed and instantaneous electrical power using a free acceleration test. Authors have recently proposed a two-step method to detect the characteristics of an induction motor using a nonlinear method that incorporates magnetic saturation [24]. In Ref. [25], the genetic algorithm (GA) presents a unique method for predicting the electrical parameters of an equipollent circuit for a three-stage induction motor. In [26], the topic of variable determination for induction motors is covered. The proposed algorithm for an artificial bee colony (ABC) is compared with the most recent approaches. The shuffling frog-leaping algorithm—initially introduced in Ref. [27]—and standard manufacturer data can be utilized to approximate the variables of the double-cage asynchronous device. Depending on the number of physical factors, the variables

of the induction motor were obtained from a poor initial approximation using nonlinear regression approaches based on a least-squares two-step strategy.

It is crucial to have a precise understanding of the electrical properties of induction machines due to their wide applications in industry [28]. The electrical equivalent circuit characteristics of the induction motor are necessary to compute the motor's efficiency [29], load variations' response, control driver parameters [30,31], and estimate breakdown behavior [32]. Consequently, determining electrical and mechanical parameters requires accurate parameter estimations and induction motor modeling [33]. However, their nonlinear models make obtaining these motors' mathematical methods somewhat difficult [34]. This study examined how training affected the parameter determination of squirrel-cage IMs.

This research improves the modeling of IMs utilizing analogous circuits, neural networks, and numerical simulations. ANNs have been used to solve numerous engineering issues [35,36]. Before being expanded to accommodate more polyphase rotating induction motors [37], a neural network was first utilized to produce a single-sided linear IM [38]. In these early research articles, the neural network transferred the input machine's geometrical design variables to the output machine's efficiency. The closely linked and interdependent input–output variables in this model are one of its drawbacks; this makes it challenging and ineffective to create data patterns and train the network. The same circuit approach can considerably reduce the coupling effect. It links machine performance through circuit variables to geometric design elements. The circuit parameters of an induction motor can all be connected to particular machine parts [39]. The neural network will be trained more effectively as a result. It is possible to predict and measure the impacts of the airgap, stator, and rotor independently on the behavior and performance of a machine. Using this method will enable you to identify the necessary efficiency attributes swiftly. Because the circuit variables are estimated using the finite element approach, which also considers the complex geometry and excitations, as well as the non-linear magnetic characteristics, this model retains a high level of accuracy.

Using a novel approach proposed in this publication, variables of the double and single-cage patterns of induction motors are estimated. The proposed method uses six network (ANN) techniques. To do this, the manufacturer data [4] use twenty induction motors with synchronous speeds varying from two to eight poles and a 400 V line voltage.

2. Structure and Equivalent Circuit Models of Squirrel-Cage Induction Motors

Induction motors are preferred in the industry since they require less maintenance than many other motors, in addition to their simple, robust, and inexpensive structures. The structure of squirrel-cage induction motors, one of the most preferred, is also straightforward, robust, and valuable. In addition to these features of squirrel-cage induction motors, their load-dependent speed regulation is also quite good. Squirrel-cage rotors can be manufactured as single or double-cage rotors. The squirrel cage machine's rotor is manufactured by cutting laminated steel sheets into the appropriate geometry and pressing them into blocks. The cage structure is formed by placing aluminum or copper rods in the channels opened following the surface of the rotor, which has been turned into a block, using casting or other techniques [40]. The rotor formed in this way is called a squirrel-cage rotor. Squirrel-cage bars may have different combinations and geometries. The stator of the induction motor of the squirrel-cage approach and the double and single-cage rotor structure are illustrated in Figure 1.

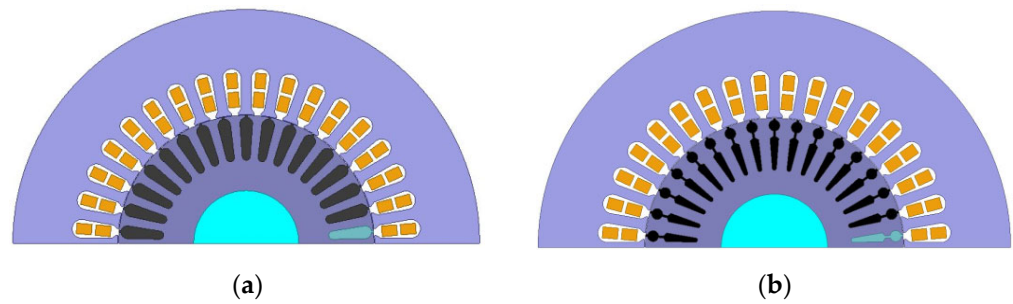


Figure 1. Structure of the induction motor: (a) Single-cage, (b) Double-cage.

Obtaining equivalent circuit parameters with a high degree of accuracy from the manufacturer’s data sheet of a squirrel-cage machine allows electrical and mechanical analysis of the machine without purchasing or using it directly in a system—in other words, without risking it. This gives us significant advantages in terms of cost, time, and practicality in many applications.

The induction motor is one of the most commonly utilized motor types in high-efficiency drive applications, which requires full knowledge of some, if not all of the induction motor variables of the control schemes of drivers [41]. It is necessary to establish the equivalent circuit method of the machine and know the parameters of this model correctly in terms of detecting critical operating points of the machine, performance analysis, control, protection, malfunction, and operation [7]. Depending on the single or double-cage rotor structure, squirrel-cage induction motors can be modeled as a single or double-cage equivalent circuit. The most commonly utilized equivalent circuit model of the squirrel-cage induction motor is the constant parameter equivalent circuit model with a single-cage rotor structure [11]. Contrary to popular belief, obtaining equivalent circuit parameters is complex, and it is even more challenging to find the typical amounts of the double-cage induction motor variables [42,43]. Since the single-cage equivalent circuit method does not adequately present induction motors, the double-cage equivalent circuit method should be utilized. The equivalent circuit models of the induction motor with a single and double-cage rotor structure are shown in Figures 2 and 3. Under steady-state conditions, the equivalent circuit model consists of five electrical variables, including R_r , R_s , X_{sd} , X_m , and X_{rd} , for a single-cage structure, and seven electrical parameters, including R_s , R_2 , R_1 , X_{sd} , X_m , X_{2d} , and X_{1d} , for a double-cage structure. Among these parameters, X_{1d} and R_1 demonstrate the inner cage, and the variables of X_{2d} and R_2 demonstrate the outer cage [4,44].

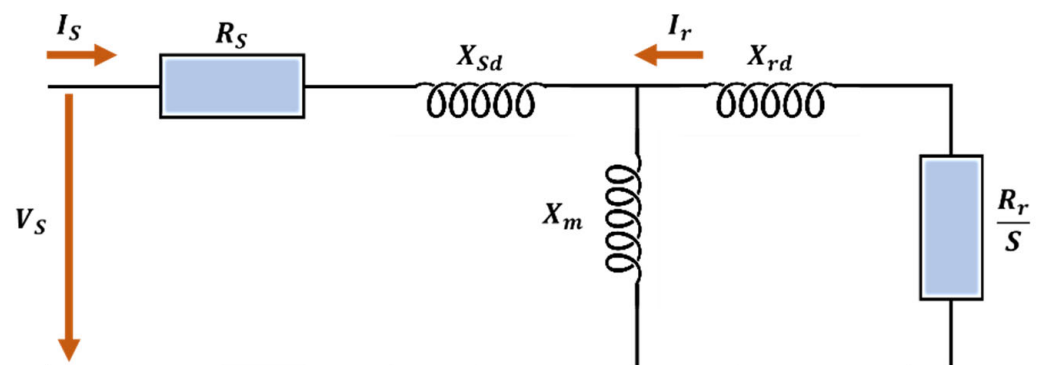


Figure 2. The steady-state pattern of the single-cage induction motor.

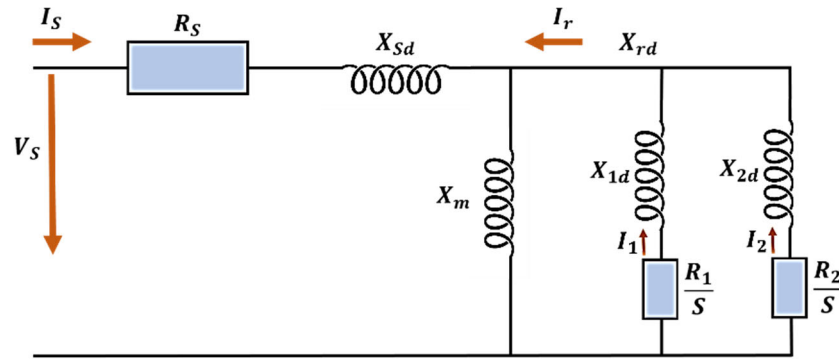


Figure 3. The steady-state pattern of the double-cage induction motor.

Equations (1)–(10) illustrated the mathematical induction motor model [45]. The mathematical modeling of an induction motor involves expressing its electrical and mechanical characteristics through a set of equations. An induction motor is an alternating current (AC) motor widely used in various industrial applications. The basic mathematical model consists of electrical equations representing the stator and rotor windings, and mechanical equations describing the motor’s motion. A proper predictive model can avoid complicated mathematical processes. The proposed predictive network obtained from this study can be implemented as a practical solution by producing design charts.

$$\begin{bmatrix} V_{S0} \\ V_{sd} \\ V_{sq} \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos\theta_s & \cos(\theta_s - \frac{2\pi}{3}) & \cos(\theta_s + \frac{2\pi}{3}) \\ -\sin\theta_s & -\sin(\theta_s - \frac{2\pi}{3}) & -\sin(\theta_s + \frac{2\pi}{3}) \end{bmatrix} \cdot \begin{bmatrix} V_{as} \\ V_{bs} \\ V_{cs} \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \cos\theta_s & -\sin\theta_s \\ \frac{1}{\sqrt{2}} & \cos(\theta_s - \frac{2\pi}{3}) & -\sin(\theta_s - \frac{2\pi}{3}) \\ \frac{1}{\sqrt{2}} & \cos(\theta_s + \frac{2\pi}{3}) & -\sin(\theta_s + \frac{2\pi}{3}) \end{bmatrix} \cdot \begin{bmatrix} i_{s0} \\ i_{sd} \\ i_{sq} \end{bmatrix} \quad (2)$$

$$RE = R_s + R'_r \cdot \frac{L_m}{L_r} \quad (3)$$

$$\sigma = 1 - \frac{L_m^2}{L_s \cdot L'_r} \quad (4)$$

$$V_{sd} = R_s \cdot i_{sd} - \omega_s \cdot \left[\sigma \cdot L_s \cdot i_{sq} + \frac{L_m}{L'_r} \cdot \psi_{rq} \right] + \frac{d}{dt} \left[\sigma \cdot L_s \cdot i_{sd} + \frac{L_m}{L'_r} \cdot \psi_{rd} \right] \quad (5)$$

$$V_{sq} = R_s \cdot i_{sq} + \omega_s \cdot \left[\sigma \cdot L_s \cdot i_{sd} + \frac{L_m}{L'_r} \cdot \psi_{rd} \right] + \frac{d}{dt} \left[\sigma \cdot L_s \cdot i_{sq} + \frac{L_m}{L'_r} \cdot \psi_{rq} \right] \quad (6)$$

$$\frac{d\psi_{rd}}{dt} = \frac{R'_r \cdot L_m}{L'_r} \cdot i_{sd} - \frac{R'_r}{L'_r} \cdot \psi_{rd} + \omega_r \cdot \psi_{rq} \quad (7)$$

$$\frac{d\psi_{rq}}{dt} = \frac{R'_r \cdot L_m}{L'_r} \cdot i_{sq} - \omega_r \cdot \psi_{rd} + \frac{R'_r}{L'_r} \cdot \psi_{rq} \quad (8)$$

$$T_e - T_L = J \cdot \frac{d\omega}{dt} + B \cdot \omega \quad (9)$$

$$T_e = p \cdot \frac{L_m}{L'_r} \cdot (i_{sq} \cdot \psi_{rd} - i_{sd} \cdot \psi_{rq}) \quad (10)$$

For balanced systems, V_{S0} and i_{s0} are zero; $i_a, i_b, i_c - V_{as}, V_{bs}, V_{cs}$ are the stator phase currents and voltages; V_{sd} and V_{sq} are the d-q axis stator voltages; i_{sd}, i_{sq} are the d-q axis stator currents; ψ_{rd}, ψ_{rq} are the d-q axis rotor fluxes; R'_r is the rotor winding resistance with referred to stator; RE is the equivalent resistance; σ is the leakage factor; R_s is the stator phase winding resistance; ω_s and ω_r are the angular frequency of stator and rotor currents;

ω is the angular speed of the rotor; T_e is the electromagnetic moment; T_L is the load moment; B is the damping coefficient; J is the moment of inertia; L_s is the stator inductance; L'_r is the rotor inductance with referred to stator; L_m is the magnetization inductance; and p is the number of double poles.

3. Established Dataset

The data utilized in this study are taken from Ref [4]. The producer’s evaluations, displayed in Table 1, are used to calculate the experimental data using a numerical method presented by Monjo, Kojooyan-Jafari [46], Pedra and Corcoles [47]. The models’ input parameters have rated power $P(KW)$, full load power factor $\cos(\rho FL)$, maximum torque to total load torque T_m/T_{FL} , the proportion of initial torque to total load torque T_{ST}/T_{FL} , the proportion of initial current to total load current I_{ST}/I_{FL} , angular velocity ω_{FL} (rpm), and full load efficiency η_{FL} . The induction motor’s output variable is X_m . The empirical data are split into two categories to train and test the networks. Eighty percent of the empirical findings are employed for network training, and the remaining twenty percent are employed to assess the efficiency of the trained methods. To find the best ANN structures, various ANN structures (networks with various numbers of concealed neurons and layers in each concealed layer) were tested and optimized in this research.

Table 1. Input and output data values.

Input							Output	
$P(KW)$	$\cos(\rho FL)$	T_m/T_{FL}	T_{ST}/T_{FL}	I_{ST}/I_{FL}	ω_{FL} (rpm)	η_{FL}	Cage Number	Reactance (ohms) X_m
8	0.74	2.5	2.1	4.6	960	0.86	1	1.056
11	0.9	3.1	2.2	7	2945	0.91	1	2.5856
15	0.92	2.9	2.2	6.6	2910	0.904	1	3.1806
19	0.84	3.2	2.7	6.9	1460	0.905	1	1.6808
22	0.77	2.9	2.8	5.5	975	0.908	1	1.2526
30	0.88	2.7	2.3	6	2940	0.91	1	2.3497
37	0.86	3.1	2.5	7	1475	0.929	1	1.9975
45	0.81	2.3	2.1	6	740	0.92	1	1.6728
55	0.82	2.4	2.2	6	738	0.931	1	1.7594
75	0.86	2.4	2.1	6.3	1482	0.947	1	2.3228
90	0.86	2.7	2.2	6.8	1480	0.94	1	2.1573
110	0.86	3	2	7.6	2982	0.955	1	2.1294
132	0.86	3	2.7	7.2	1486	0.955	1	2.1209
160	0.86	2.7	2.4	7	1487	0.96	1	2.2378
200	0.87	2.7	2.7	7	1488	0.962	1	2.4082
250	0.8	3	2.2	7.3	991	0.91	1	1.4423
315	0.84	3	2	7.3	991	0.962	1	1.9026
355	0.87	2.7	2.2	6.8	1486	0.967	1	2.4236
400	0.82	2.6	2.1	6.5	742	0.962	1	1.7943
500	0.87	2.7	2.3	6.5	992	0.966	1	2.3801
8	0.74	2.5	2.1	4.6	960	0.86	2	1.0415
11	0.9	3.1	2.2	7	2945	0.91	2	2.5947
15	0.92	2.9	2.2	6.6	2910	0.904	2	3.0787
19	0.84	3.2	2.7	6.9	1460	0.905	2	1.6559
22	0.77	2.9	2.8	5.5	975	0.908	2	1.2665
30	0.88	2.7	2.3	6	2940	0.91	2	2.3576
37	0.86	3.1	2.5	7	1475	0.929	2	2.0088
45	0.81	2.3	2.1	6	740	0.92	2	1.7001
55	0.82	2.4	2.2	6	738	0.931	2	1.7804
75	0.86	2.4	2.1	6.3	1482	0.947	2	2.3514
90	0.86	2.7	2.2	6.8	1480	0.94	2	2.1727
110	0.86	3	2	7.6	2982	0.955	2	2.1472
132	0.86	3	2.7	7.2	1486	0.955	2	2.1405
160	0.86	2.7	2.4	7	1487	0.96	2	2.261
200	0.87	2.7	2.7	7	1488	0.962	2	2.4351
250	0.8	3	2.2	7.3	991	0.91	2	1.4611
315	0.84	3	2	7.3	991	0.962	2	1.9158
355	0.87	2.7	2.2	6.8	1486	0.967	2	2.442
400	0.82	2.6	2.1	6.5	742	0.962	2	1.8144
500	0.87	2.7	2.3	6.5	992	0.966	2	2.4021

The experimental data are randomly chosen to form the data training (the more significant data collection) and data assessment. The various ANN structures are initially developed using the training data (the training procedure). The precision of the developed (trained) network is then evaluated using the data set evaluation, which is unidentified to the network. A computer program was created using MATLAB software R2020 to train the ANN methods. Table 2 displays the optimal architectures of the ANN method.

Table 2. ANN network outcomes concerning RMSE output.

Neurons' Number	Network Results			Scoring			Total Score	Rank
	RMSE Total	RMSE Train	RMSE Test	RMSE Total	RMSE Train	RMSE Test		
1	0.042	0.159	0.094	5	3	4	12	7
2	0.070	0.087	0.076	3	5	5	13	6
3	0.026	0.031	0.027	7	10	9	26	2
4	0.023	0.042	0.030	8	8	7	23	4
5	0.022	0.042	0.029	9	7	8	24	3
6	0.028	0.189	0.106	6	2	3	11	8
7	0.002	0.034	0.019	10	9	10	29	1
8	0.413	0.557	0.461	1	1	1	3	10
9	0.085	0.158	0.112	2	4	2	8	9
10	0.055	0.054	0.055	4	6	6	16	5

4. Methodology

The graphical depiction of the techniques used in the current article is shown in Figure 4. This study's first stage is to collect data, which will be utilized in simulations and analyses. These data include $P(KW)$, $\cos(\rho FL)$, T_m/T_{FL} , T_{ST}/T_{FL} , I_{ST}/I_{FL} , ω_{FL} (rpm), and η_{FL} . To identify the model's highest possible reliability, the model employs six hybrid methods using the MVO-MLP, COA-MLP, HBO-MLP, LCA-MLP, OOA-MLP, and STOA-MLP networks.

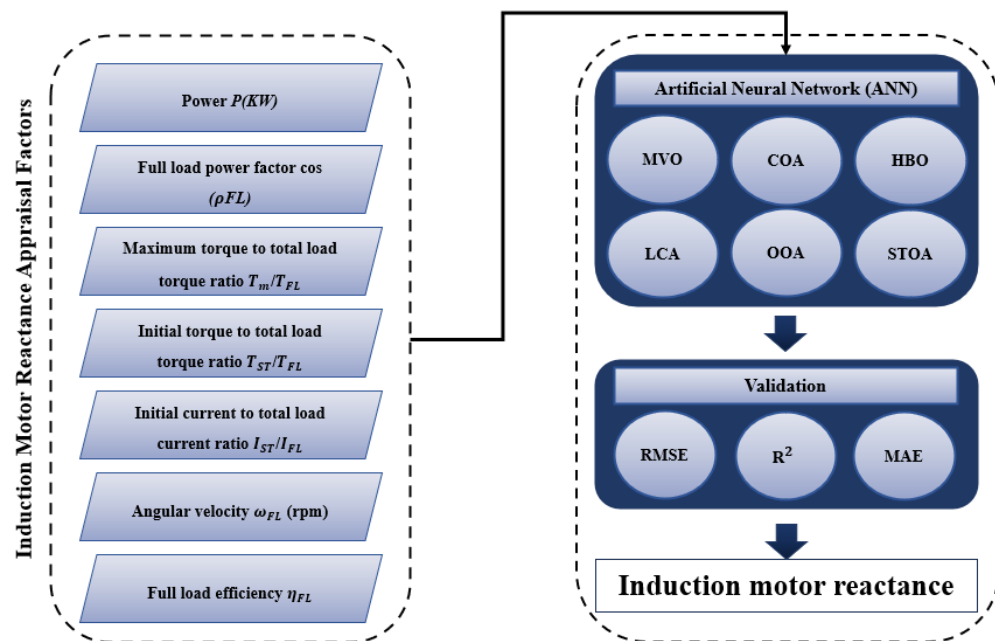


Figure 4. Investigation flowchart encompassing input parameters to anticipate the output of motor reactance.

4.1. Artificial Neural Network (ANN)

A system relying on how biological neural networks function is known as an ANN [48]. The most popular neural networks comprising numerous processing units called neurons are multi-layer perceptron (MLP) networks [4,49]. An MLP network consists of at least three layers, usually called the layer of input, the concealed layer, and the layer of output, as demonstrated in Figure 5.

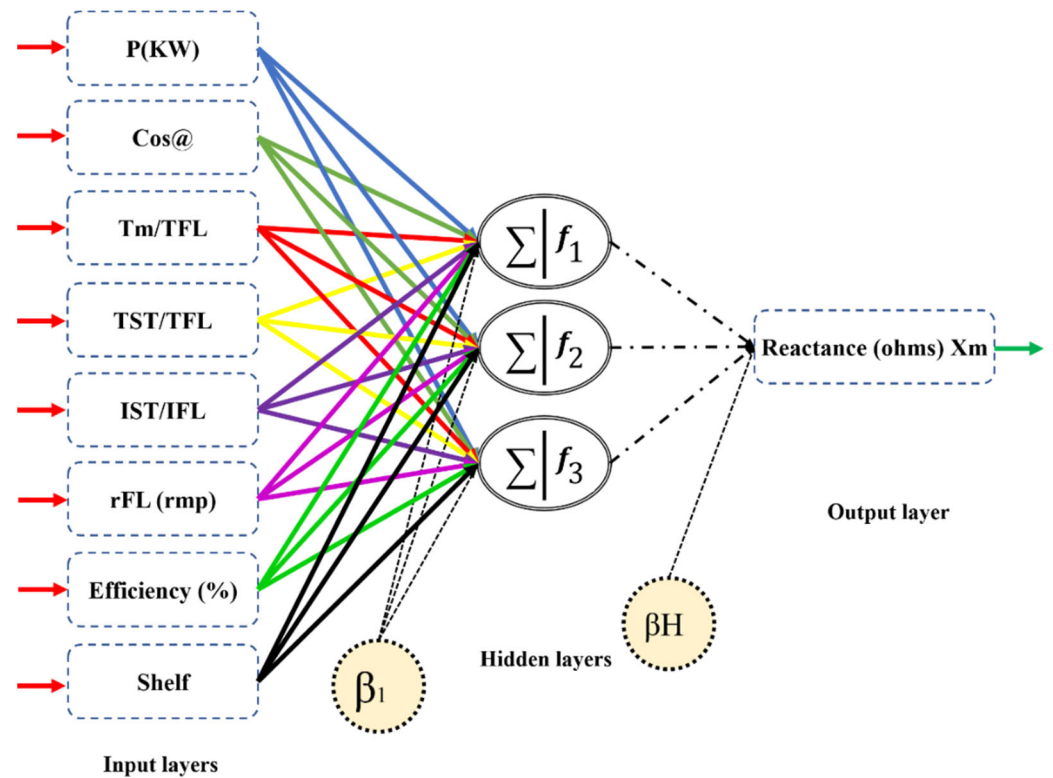


Figure 5. A simple structure of ANN is in the present example.

The neurons’ frequency in the MLP network varies depending on the layer. The node’s input in the concealed layer is presented as follows [4]:

$$\rho_d = a_d + \sum_{k=1}^i (X_k \cdot w_{kd}) \quad d = 1, 2, \dots, n \tag{11}$$

where X represents the inputs, n represents the neurons’ frequency in the concealed layer, i represents the neurons’ frequency in the input layer, a represents the bias, and w represents the weighting factor [4]. The concealed layer’s d th neuron’s output is presented as follows:

$$\theta_d = f(\rho_d), \tag{12}$$

where f represents the concealed layer’s activation function.

The output layer’s c th neuron output is presented as follows [4]:

$$Y_c = b_c + \sum_{k=1}^i (\theta_k \cdot w_{kc}) \quad c = 1, 2, \dots, m \tag{13}$$

Here, b represents the term of bias, w represents the weighting factor, m represents the output layer’s neuron frequency, and i represents the neurons’ frequency in the concealed layer.

4.2. Hybrid Model Development

The gradient descent rule and the Gauss–Newton algorithm are combined in the Levenberg–Marquardt (LM) algorithm, which trains neural networks [50]. The LM algorithm removes the constraints of these two techniques. However, as earlier research has shown, the ANN has many drawbacks, including prematurely converging and becoming stuck in local minima [51,52]. Combining evolutionary algorithms (EAs) with ANNs, which are a part of bio-inspired and evolutionary computing, can partially offset these weaknesses. EAs are mechanisms inspired by nature that solve issues through procedures that mimic living organisms’ behavior. In this section, the six hybrid models (i.e., heap-based optimization (HBO), leagues championship algorithm (LCA), multi-verse optimization (MVO), osprey optimization algorithm (OOA), cuckoo optimization algorithm (COA), and sooty tern optimization algorithm (STOA)) that are combined with conventional ANN are illustrated. The open-source codes for these algorithms alone can be found at mathworks.com. The calculation process was (i) to optimize the ANN (e.g., through the trial-and-error process and changing the hidden layers and neurons’ number) and (ii) utilizing the optimized ANN network to be combined with proposed Metaheuristic solutions. In each section, and to assess the proposed predictive network accuracies, several statistical indices, namely mean average error (MAE), root mean square error (RMSE), and coefficient of determination (R^2), were used. The equations of each item are given as follows:

$$RMSE = \left[N^{-1} \sum_{i=1}^N (y_{measured, i} - y_{predicted, i})^2 \right]^{0.5} \tag{14}$$

$$MAE = \frac{1}{N} \sum_{i=1}^N (y_{measured, i} - y_{predicted, i}) \tag{15}$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_{predicted, i} - y_{measured, i})^2}{\sum_{i=1}^N (y_{measured, i} - \underline{y}_{measured, i})^2} \tag{16}$$

where n represents the observation frequency, $y_{measured, i}$ indicates the determined amount of the i th data number, $y_{predicted, i}$ represents the anticipated amount of the i th data number, and $\underline{y}_{measured, i}$ is the average value of the $y_{measured, i}$. In the most optimal method, RMSE should be equal to 0, and R^2 should equal 1.

4.2.1. Multi-VERSE OPTIMIZATION (MVO)

The three cosmological concepts of black, white, and wormholes are the foundation for this article’s new metaheuristic MVO method. Local search, exploitation, and exploration are implemented based on analytic models of these concepts. The universe’s objects serve as the algorithm’s solution variables, because the algorithm and the universe are analogous [53].

In the literature, the scenario of the roulette wheel is employed to determine the best solution to the black and white hole tunnels’ analytical model and exchange the objects between the worlds [53]. Following is a random representation of the globe found in the solution space:

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^n \\ x_2^1 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^n \end{bmatrix} \tag{17}$$

Here, U represents the world, n is the frequency of search components, d represents the control parameters' measurements, and X_i^j represents the j th parameter of i th the world that adopts the below shape [53]:

$$x_i^j = \begin{cases} x_k^j & r_1 < NI(U_i) \\ x_i^j & r_1 > NI(U_i) \end{cases} \tag{18}$$

where U_i represents the i th world, NI represents the normalized inflation rate, r_1 represents an accidental number in the $[0, 1]$ range, and X_k^j represents the j th variable of the k th world, chosen by employing the scenario of the roulette wheel.

The mechanism of the transmitted wormholes' objects [53] is demonstrated as follows:

$$x_i^j = \begin{cases} \begin{cases} x_j + TDR \cdot [(ub_j - lb_j) \cdot r_4 + lb_j] & r_3 < 0.5 \\ x_j - TDR \cdot [(ub_j - lb_j) \cdot r_4 + lb_j] & r_3 \geq 0.5 \end{cases} & r_2 < WEP \\ x_i^j & r_2 \geq WEP \end{cases} \tag{19}$$

where X_j is the j th the variable of the optimum world, WEP (Wormhole Existence Probability) and TDR (Traveling Distance Rate) are coefficients, ub_j and lb_j indicate the upper and lower limit of j th variable, x_i^j indicates the j th parameter of i th world, and r_4 , r_3 , and r_2 are accidental numbers belonging to $[0, 1]$.

The WEP increases the exploitation and adopts the below shape [53]:

$$WEP = min + t \times \left(\frac{max - min}{T_{max}} \right) \tag{20}$$

where t indicates the current repetition, T_{max} indicates the highest repetition frequency, and min and max indicate the minimum and maximum number of controlled variables. TDR is acquired as follows [53]:

$$TDR = 1 - \frac{t^{(1/p)}}{T_{max}^{(1/p)}} \tag{21}$$

where p indicates the exploitation precision over the repetition; as the p enhances, more precise exploitation and quicker local search occur.

4.2.2. Cuckoo Optimization Algorithm (COA)

In 2009, Yang and Deb [54] developed the meta-heuristic approach of COA—Cuckoo Optimization Algorithm, inspired by nature. The obligate brood parasitism present in several species of cuckoos, which causes them to lay their eggs in the nests of other host birds, served as the basis for the algorithm. The female cuckoos of the parasitic species have developed a specialization in imitating the other cuckoo species' color and egg. This reduces the likelihood of relinquishment and boosts their reproductive capacity. Three idolized rules are established to explain their application in optimization [55].

A metaheuristic algorithm for randomly exploring the optimal strategies in a search environment relies heavily on randomization. Randomized walks are used to achieve this randomization, comprised of trying to take a sequence of accidental stages in a random dispersion. To maximize the effectiveness of resource searches in unpredictable circumstances, the random walk is offered for the Cuckoo optimization algorithm by employing flight of Lévy, whose duration of step proceeded from the Lévy dispersion and possesses a much-extended step duration. As Equation (22) demonstrated, the Lévy flight offers a walk with accidental stages duration derived from Lévy's power law dispersion [54].

$$Lévy \sim u = t^{-\lambda}, \quad (1 < \lambda \leq 3) \tag{22}$$

In Equation (22), u indicates Lévy dispersion. New ones surround the best solution, accelerating the local search [54].

A novel strategy by employing Lévy flights and subsequently its fitness is compared with the remaining eggs in the nest for an objective function $f(x) = (x_1, \dots, x_d)^t$ and preliminary cuckoo population n for a nest x_i . A new egg substitutes an egg if the recently developed solution is more suitable than the randomly selected nest (solution). The best solutions are maintained for additional analysis, while the majority of the worst nests are disregarded.

The recently developed solution, $x_i^{(t+1)}$ from Lévy flights, is conducted by employing the following equation [54]:

$$x_i^{(t+1)} = x_i^t + \alpha \oplus \text{Lévy}(\lambda) \tag{23}$$

where $\alpha > 0$ represents the step size associated with the considered issue's scales, considered one, and the product \oplus indicates entry-wise multiplication, i.e., Exclusive-OR operation.

4.2.3. Heap-Based Optimization (HBO)

Askari [56] developed the heap-based optimization algorithm, which drew inspiration from human social attitudes toward organizational hierarchies. This algorithm emulates the corporate rank hierarchy (CRH). According to the CRH, team members in a given organization organize themselves hierarchically to accomplish a certain goal. The suggested HBO technique bases its hierarchical search candidate arrangement on fitness utilizing the CRH idea. The hierarchy is constructed using the heap-based data structure.

The entire concept involves three phases in addition to the modeling of the CRH: (i) modeling the cooperation among the subordinators and their direct manager; (ii) modeling the employee interaction, and (iii) modeling the subordinator's self-contribution to accomplish the desired function. These steps are explained in brief in the following subsections.

The CRH Concept's Modeling

Corporate rank hierarchy is constructed using a non-linear tree-shaped data structure. The improved CRH is considered the swarm in the suggested method. The heap node is represented by the search agent in the search spaces throughout the simulation, and the master key to the heap node is the optimization issue's fitness function. The population index of the search candidate was determined by taking the heap node's amount.

The Interaction's Modeling with the Direct Manager

Senior leadership enforces regulations and laws uniformly on employees in large organizations with centralized organizational structures, and employees must follow directions from their superiors. By changing the search candidates' locations, the next phase can be mathematically described [56]:

$$x_i^k(t+1) = B^k + \gamma(2r - 1) \left| B^k - x_i^k(t) \right| \tag{24}$$

where B is the parent node, t and k are the current repetition and the vector of the component, respectively, and x is the search agent's location. Equation (25) describes the term $(2r - 1)$, which denotes the k -th component of the vector γ , and is produced accidentally [56].

$$\lambda^k = 2r - 1 \tag{25}$$

where r represents any arbitrary parameter in a uniformly distributed form within the interval $[0, 1]$. γ can be written like this [56]:

$$\gamma = \left| 2 - \frac{\left(t \bmod \frac{T}{C} \right)}{\frac{T}{4c}} \right| \tag{26}$$

where T is the maximum iterations' number, and C is an adjustable parameter that the user defines concerning the case under study, which is dependent upon the number of iterations using the following formula [56]:

$$C = \frac{T}{25} \tag{27}$$

Modeling of the Interactivity between the Subordinators

Colleagues who work together as subordinators in a certain organization accomplish official responsibilities. The nodes at the same position in the heap are regarded to be colleagues in the suggested procedure. Equation (28) presents the updated position (x_i) of each search agent depending on the randomly picked teammate (S_r) [56].

$$x_i^k(t+1) = \begin{cases} S_r^k + \gamma\lambda^k \left| S_r^k + x_i^k(t) \right|, & f(S_r) < f(x_i^k(t)) \\ x_i^k + \gamma\lambda^k \left| S_r^k - x_i^k(t) \right|, & f(S_r) \geq f(x_i^k(t)) \end{cases} \tag{28}$$

The Employee's Self-Contribution Modeling

Equation (29) describes how each subordinator in the organization contributes on their own [56]:

$$x_i^k(t+1) = x_i(t) \tag{29}$$

Position Update

The three-position upgrade formulas covered in the preceding subsections are combined into one formula in this subsection. A roulette wheel was employed to balance the exploitation and exploration stages. The equilibrium between these phases was established by applying the three probabilities P_1 , P_2 , and P_3 . The search agents' positions within the population were updated using the first probability, P_1 , which may be stated as follows [56]:

$$P_1 = 1 - \frac{t}{T} \tag{30}$$

This is how the second proportion, P_2 , was determined [56]:

$$P_2 = P_1 + \frac{1 - P_1}{2} \tag{31}$$

In the end, the probability P_3 was determined using Equation (32) [56]:

$$P_3 = P_2 + \frac{1 - P_1}{2} = 1 \tag{32}$$

Equation (33) presents the position upgrade's generic equation for the suggested HBO [56].

$$x_i^k(t+1) = \begin{cases} x_i^k(t), & P < P_1 \\ B^k + \gamma\lambda^k \left| B^k - x_i^k(t) \right|, & P_2 < P < P_3 \\ S_r^k + \gamma\lambda^k \left| S_r^k + x_i^k(t) \right|, & P_2 < P < P_3 \text{ and } f(S_r) < f(x_i^k(t)) \\ x_i^k + \gamma\lambda^k \left| S_r^k - x_i^k(t) \right|, & P_2 < P < P_3 \text{ and } f(S_r) \geq f(x_i^k(t)) \end{cases} \tag{33}$$

where P represents any number in the interval $[0, 1]$.

4.2.4. Leagues Championship Algorithm (LCA)

Kashan [57] introduced the LCA technique, a novel metaheuristic approach for resolving continuous optimization issues. It functions similarly to other algorithms inspired by nature, with a population of solutions evolving towards the best one. With n players equal to the number of variables, each team (person) in the league (a swarm of L teams)

has a workable solution to the issue being solved. Following the construction of the fake weekly league schedule, team i takes on team j , each with a playing strength corresponding to its fitness rating. The clubs play each other in pairs for $S \times (L - 1)$ weeks according to the league schedule, where S is the seasons' number and t is the week. Playing has an outcome that determines who wins and who loses. Each side forms a new team based on the outcomes of the previous week's matches to prepare for the upcoming game. The present best team configuration is replaced by the anticipated productive team formation, chosen under the direction of a team formation with superior playing strength.

A League Schedule's Generation

The first stage is to create a schedule that includes every game for every season. Every team in the LCA plays every other team once a season under a single round-robin schedule. $L(L - 1)/2$ competitions exist, and L has to be an even integer. After that, the championship goes on for S more seasons. The league scheduling method creates an 8-team ($L = 8$) basic sports league from Figure 6.

1 Week				2 Week				3 Week				4 Week				5 Week				6 Week				7 Week			
1	2	3	4	1	8	2	3	1	7	8	2	1	6	7	8	1	5	6	7	1	4	5	6	1	3	4	5
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
8	7	6	5	7	6	5	4	6	5	4	3	5	4	3	2	4	3	2	8	3	2	8	7	2	8	7	6

Figure 6. A league–schedule simulation for an 8-team league.

Evaluating Winner/Loser

Choosing a loser (and winner) according to the playing strength standard, with formations $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{im}^t)$, X_j^t , and playing strengths $f(X_i^t)$ and $f(X_j^t)$; correspondingly, let us examine teams i and j participating in week t . The possibility that team i will outdo team j in week t is represented by p_i^t , which is determined by Equation (34) [57].

$$p_i^t = \frac{f(X_j^t) - \hat{f}}{f(X_j^t) + f(X_i^t) - 2\hat{f}} \tag{34}$$

where the best team global team formation is represented by \hat{f} . The probability that team j would overcome team i is also determined simultaneously by a uniform accidental number in the range of $[0, 1)$. Team i triumphs, and team j loses if the result is less than or equal to p_i^t . On the other hand, j wins and i loses if the number is bigger than p_i^t .

A New Team Formation

The clubs that played with team i in week $t + 1$, with team i in week t , and with team l in week t , respectively, are indicated by the other three linked indices of i : l , j , and k , based on the league schedule. Presume that the optimal team configurations for teams k , j , and i at week t are B_k^t , B_j^t , and $B_i^t = (b_{i1}^t, b_{i2}^t, \dots, b_{im}^t)$, respectively. It can be argued that for team k to overcome team l , team i must come up with a playing strategy that is comparable to that employed by team k at week t , based on the strengths of team k , which $(B_k^t - B_i^t)$ indicates the gap vector among the playing styles of team k and team i . Similarly, it would make sense to steer clear of adopting a playing style akin to that of team k while concentrating on the deficiencies of that team $(B_i^t - B_k^t)$. As a result, one may understand $(B_j^t - B_i^t)$ and $(B_i^t - B_j^t)$ similarly. The two gap vectors' information is combined with the two constant variables, ψ_1 -retreat and ψ_2 -approach, to generate a new team. When team i wishes to go in the direction of a rival, they employ the approach parameter. On the other hand, if team i wants to distance itself from a competitor, the retreat parameter is employed.

A swarm-based approach was created to achieve global optimization in the basic LCA. Despite its simplicity and effectiveness, The LCA is prone to becoming locked in local optima, leading to an imbalance in global exploration and local exploitation.

4.2.5. Osprey Optimization Algorithm (OOA)

The osprey, often known as the sea hawk, is a widely distributed, nocturnal predatory bird. The strategy of osprey in catching fish and transporting them to a prime spot for consumption is a brilliant natural treatment that can act as the basis for developing a novel optimization algorithm. These clever osprey actions were mathematically analyzed to develop the suggested OOA method.

Following the OOA launch, which is covered in detail in this subsection, the process of upgrading the ospreys' positions in the two stages of exploitation and exploration based on the natural osprey treatment's simulation is provided [58].

The suggested OOA is a swarm-based method that can identify a feasible solution based on the search capacity of its swarm members in the issue-solving space through an iteration-based method. As the OOA's swarm member, each osprey evaluates the amounts of the issue parameters based on where it is in the search space. Consequently, every osprey symbolizes a possible resolution to the issue, expressed mathematically as a vector. Equation (35) describes the OOA swarm, which consists of all ospreys, and may be expressed as a matrix. At the beginning of the OOA implementation, the ospreys' position in the search space is initialized by accidentally utilizing Equation (36) [58].

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix}_{N \times m} \tag{35}$$

$$x_{i,j} = lb_j + r_{i,j} \cdot (ub_j - lb_j), \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, m \tag{36}$$

where m denotes the issue variables' number, $r_{i,j}$ indicates accidental numbers in the $[0, 1]$, ub_j , and lb_j indicates the upper and lower bounds, and X represents the population matrix of the osprey locations. X_i indicates the i th osprey, and $x_{i,j}$ indicates its j th dimension. Equation (37) states that the determined amounts for the objective function of the issue can be presented as a vector [58].

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N \times 1} \tag{37}$$

where F is the vector containing the objective function amounts and F_i is the computed objective function amount for the i th osprey.

The values evaluated for the objective function serve as the main criterion for evaluating the possible solutions' quality. The greatest value found for the objective function is correlated with the best candidate solution or the best member. In contrast, the worst amount found for the objective function is correlated with the worst member or candidate solution. Since the location of the ospreys in the search space is upgraded every repetition, the best candidate solution needs to be modified.

Stage 1: Positions’ Identification and Fish Hunting (Exploration)

Ospreys are strong predators with excellent vision, enabling them to find fish underwater. They find the fish, dive under the surface to attack it, and then pursue it. The true natural behavior of ospreys’ simulation was used to model the first phase of the OOA’s swarm upgrade. Modeling the osprey attack on fish led to a significant change in the osprey’s situation in the search space, increasing the OOA’s ability to explore and find the optimal location while avoiding the local optima.

In the OOA’s design, each osprey’s placements beside other ospreys with a higher objective function amount were considered undersea fish. Equation (38) was used to calculate the fish’s set for each osprey [58].

$$FP_i = \{X_k | k \in \{1, 2, \dots, N\} \wedge F_k < F_i\} \cup \{X_{best}\} \tag{38}$$

where X_{best} is the optimal osprey solution and FP_i is the fish situation set for the i th osprey.

The osprey locates one of these fish at random and strikes it. Equation (39) was used to simulate the osprey’s movement towards the fish to determine a new situation for the corresponding osprey. Equation (40) states that if the osprey’s new location increases the amount of the objective function, it will relocate [58].

$$x_{i,j}^{P1} = x_{i,j} + r_{i,j} \cdot (SF_{i,j} - I_{i,j} \cdot x_{i,j}), \quad x_{i,j}^{P1} = \begin{cases} x_{i,j}^{P1}, lb_j \leq x_{i,j}^{P1} \leq ub_j \\ lb_j, x_{i,j}^{P1} < lb_j \\ ub_j, x_{i,j}^{P1} > ub_j \end{cases} \tag{39}$$

$$X_i = \begin{cases} X_i^{P1}, F_i^{P1} < F_i \\ X_i, else, \end{cases} \tag{40}$$

where F_i^{P1} is its fitness function, $X_{i,j}^{P1}$ is its j th dimension, $r_{i,j}$ are accidental digits in the range of $[0, 1]$, $I_{i,j}$ are accidental digits from the set $\{1,2\}$, and X_i^{P1} is the i th osprey new situation based on the first stage of OOA.

Stage 2: Transporting the Fish to the Appropriate Spot (Exploitation)

The osprey transports fish it has hunted to a suitable feeding site. The second step of upgrading the swarm in the OOA was modeled using this actual behavior simulation. Minute adjustments established the osprey’s location in the search space by simulating the fish’s transportation to the appropriate spot. This improved the OOA’s ability to exploit the local search and led to convergence towards superior solutions near the found solutions. Equation (41) was used in the OOA design to identify a new accidental situation for every swarm member as a “suitable situation for eating fish”. This mimicked the ospreys’ natural treatment.

Then, following Equation (42), if the goal function’s value increased in this new location, it replaced the connected element’s previous location [58].

$$x_{i,j}^{P2} = x_{i,j} + \frac{lb_j + r \cdot (ub_j - lb_j)}{t}, \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, m, \quad t = 1, 2, \dots, T, \tag{41}$$

$$x_{i,j}^{P2} = \begin{cases} x_{i,j}^{P2}, lb_j \leq x_{i,j}^{P2} \leq ub_j \\ lb_j, x_{i,j}^{P2} < lb_j \\ ub_j, x_{i,j}^{P2} > ub_j \end{cases}$$

$$X_i = \begin{cases} X_i^{P2}, F_i^{P2} < F_i \\ X_i, else, \end{cases} \tag{42}$$

where $r_{i,j}$ are accidental integers in the range of $[0, 1]$, X_i^{P2} is the osprey’s new situation based on the second stage of the OOA, $x_{i,j}^{P2}$ is the j th proportion, F_i^{P2} is its fitness function, t is the method’s repetition counter, and T is the total number of repetitions.

4.2.6. Sooty Tern Optimization Algorithm (STOA)

The sooty tern’s natural behavior served as the model for the STOA algorithm. Dhiman originally suggested it for industrial engineering issues [59]. Omnivorous sooty terns consume earthworms, insects, fish, and other food items. The main advantages of STOA over other bionic optimization algorithms are its exploration and exploitation capabilities.

Migration Behavior (Exploration)

The three elements that make up the exploration portion of migrating behavior are as follows:

Avoiding collisions: S_A is used to determine the new location to prevent search agents from colliding [59].

$$\vec{C}_{st} = S_A \times \vec{P}_{st} | (Z) \tag{43}$$

where Z denotes the current iteration, S_A depicts the search agent’s motion within a specific search space, \vec{P}_{st} indicates the search agent’s current location, and \vec{C}_{st} is the situation of the search agent without colliding with other search agents [59].

$$S_A = C_f - \left(Z \times \left(\frac{C_f}{Max_{iterations}} \right) \right) \tag{44}$$

where

$$Z = 0, 1, 2, \dots, Max_{iterations} \tag{45}$$

where C_f is the control variable that is linearly reduced from C_f to 0 to modify S_A . In the meantime, this paper’s C_f value is set to 2.

Converge in the best neighbor’s direction: Following collision avoidance, the search agents proceed in the direction of the best neighbor [59].

$$\vec{M}_{st} = C_B \times \left(\vec{P}_{bst}(Z) - \vec{P}_{st}(Z) \right) \tag{46}$$

where \vec{M}_{st} denotes the movement of search agents \vec{P}_{st} toward the most suitable search agent \vec{P}_{bst} from their respective places. Better exploration, which is defined as follows, is the responsibility of C_B [59].

$$C_B = 0.5 \times R_{and} \tag{47}$$

where R_{and} is a chance value in the interval $[0, 1]$.

Lastly, the search agents adjust their situations to point to the orientation of the top sooty terns [59].

$$\vec{D}_{st} = \vec{C}_{st} + \vec{M}_{st} \tag{48}$$

The distance between the fittest search agent and the search agent is denoted by \vec{D}_{st} .

Attacking Treatment (Exploitation)

Sooty terns can change their attack angle and speed while migrating. With the help of their wings, they can soar higher. Their spiral behaviors when attacking the prey are presented as follows [60]:

$$x' = R_{adius} \times \sin(i) \tag{49}$$

$$y' = R_{adius} \times \cos(i) \tag{50}$$

$$z' = R_{adius} \times i \tag{51}$$

$$R_{adius} = u \times e^{kv} \tag{52}$$

where I is a variable between $[0, 2\pi]$, and R_{adius} shows the radius of each spiral. The constants u and v determine the spiral's shape, and the natural logarithm's base is e . As a result, the search agent's position will be updated as follows [60]:

$$\vec{P}_{st}(Z) = \left(\vec{D}_{st} \times (x' + y' + z') \right) \times \vec{P}_{bst}(Z) \quad (53)$$

5. Results and Discussion

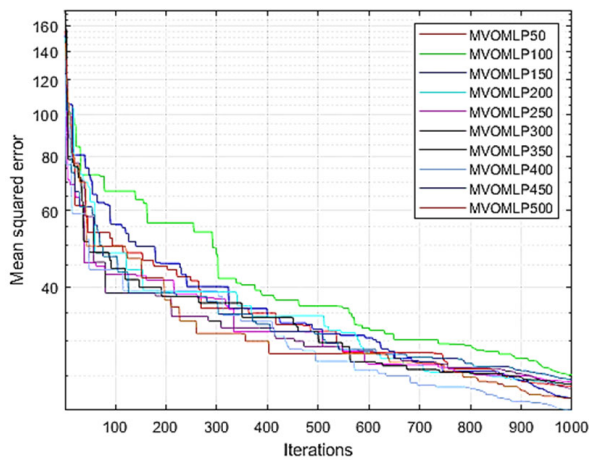
The motor reactance (X_m) was predicted using several hybrid prediction models combined with multilayer perceptrons such as MVO, COA, HBO, LCA, OOA, and STOA. MATLAB was used to develop the algorithms. The correlation between the dependent and independent parameters was then determined using the networks fed by these datasets. A trial and error procedure was employed to determine the optimal intricacy of the predictive models. The step to execute an MLP is determining the testing and training phases. The training and testing datasets, which accounted for 80% and 20% of the dataset, respectively, were selected in line with earlier reports from other researchers. The designed models are dependable and straightforward, and agree with the original experimental results. This demonstrates that the developed technique is a versatile and dependable instrument. To determine the suitable population size, this study used parametric research. Different population sizes, such as 500, 450, 400, 350, 300, 250, 200, 150, 100, and 50, were used for several MLP analyses. The effectiveness of the error mentioned above and the trial procedure was approximated using the MSE reduction process. In light of the error procedures depicted in Figure 7, the optimal efficiency was demonstrated in the smaller amount of the means square error (MSE = 429.2693) reached by the multi-verse optimization with a population size of 400. This structure was introduced as the ideal MVO-MLP architecture to facilitate future assessments of motor reactance. The other five methods, COA-MLP, HBO-MLP, LCA-MLP, OOA-MLP, and STOA-MLP, also showed an appropriate result in terms of MSE with values of 665.0085, 2080.2834, 795.005, 3298.092, and 642.6149, and relative population sizes of 350, 400, 450, 500, and 500, respectively.

To evaluate the proposed ANN approaches' performance, the attained results and the known results are compared with each other. Figures 8–13 compare the outcomes of the suggested ANN approaches with the empirical data for testing and training.

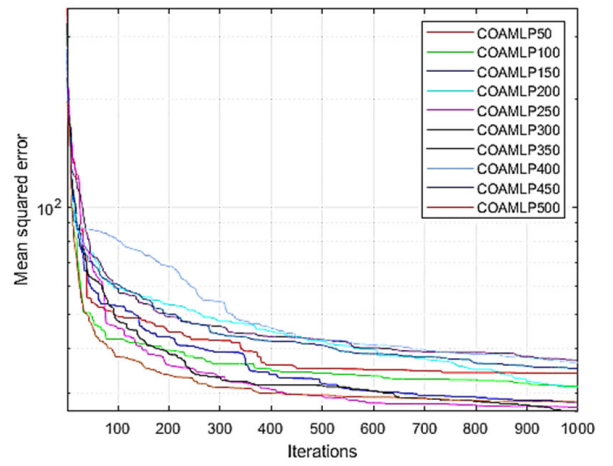
Figures 8–13 show a regression analysis of empirical and forecasted amounts to provide more detailed research into the suggested ANN models. The correlation factor (CF) can be used to validate the appropriateness of the suggested ANN models. The CF is determined as follows:

$$CF = 1 - \left[\frac{\sum_{i=1}^n [X_i(Exp) - X_i(pred)]^2}{\sum_{i=1}^n [X_i(Exp)]^2} \right] \quad (54)$$

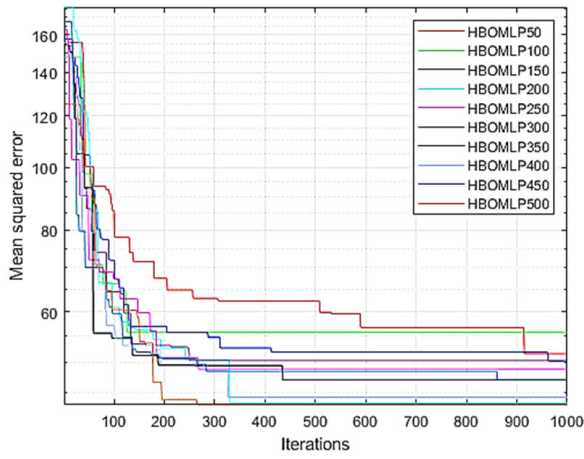
where N represents the data frequency, and ' $X(Exp)$ ' and ' $X(pred)$ ' indicates the empirical and anticipated (ANN) amounts, respectively. Figures 8–13 demonstrate a positive correlation between the recommended MVO-MLP, COA-MLP, HBO-MLP, LCA-MLP, OOA-MLP, and STOA-MLP methods, respectively, and the empirical amounts for the motors' reactance. Figures 8–13 make it understandable that the suggested ANN models' anticipated outputs are reasonably close to the empirical findings, demonstrating the appropriateness of ANN as a precise and dependable method for the simulation of the induction motors for Squirrel-cage kind. It is evident that the coefficient of determination was 0.9962, 0.9937, 0.9858, 0.9929, 0.9691, and 0.9946 in the testing phase, and 0.99598, 0.99358, 0.98561, 0.99252, 0.97008, and 0.99395 in the training phase for the MVO-MLP, COA-MLP, HBO-MLP, LCA-MLP, OOA-MLP, and STOA-MLP techniques, respectively. Additionally, based on the enquired findings, it can be observed that the recommended MVO-MLP approach is more precise than the other approaches regarding the R^2 values.



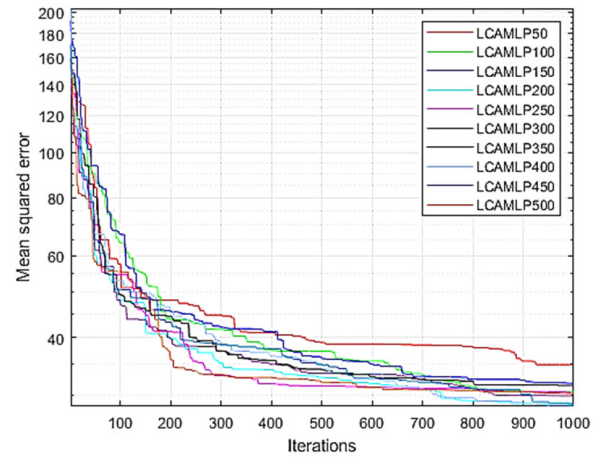
(a)



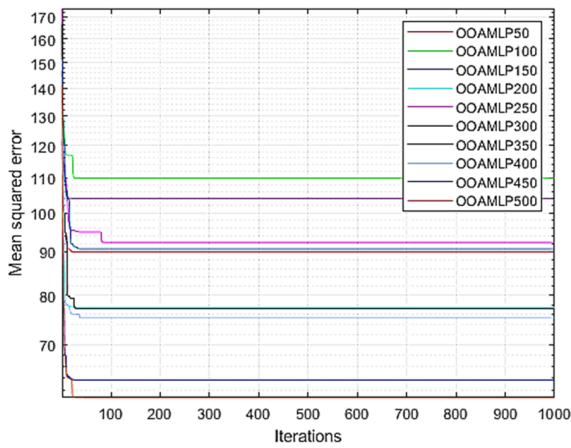
(b)



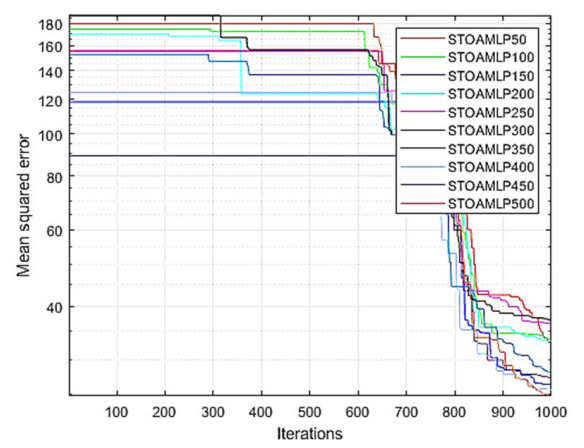
(c)



(d)

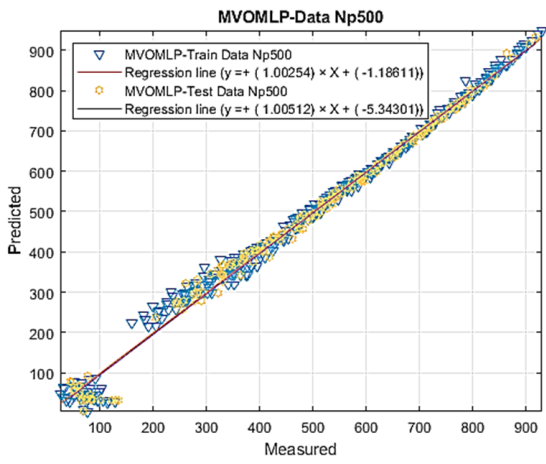


(e)

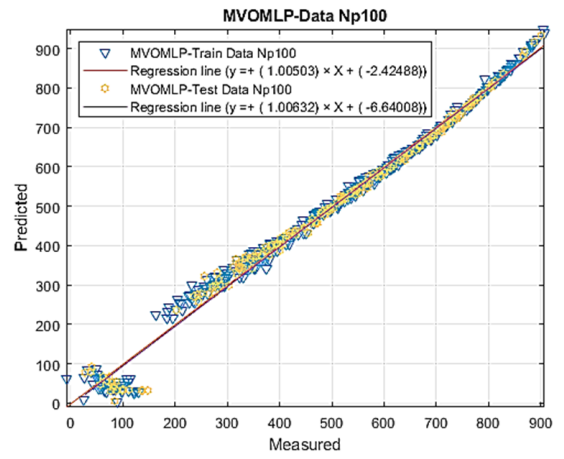


(f)

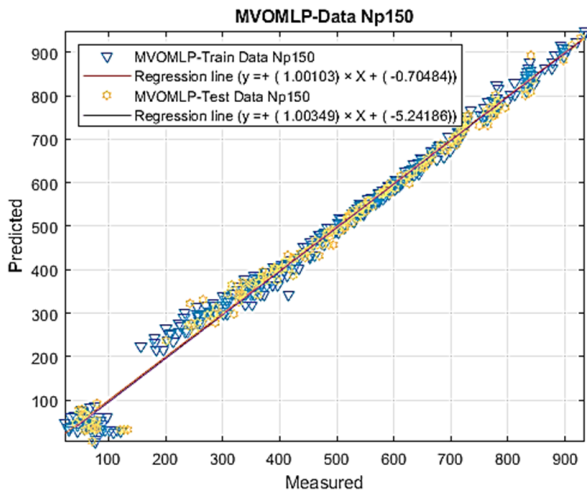
Figure 7. The best-fit structure for the (a) MVO-MLP, (b) COA-MLP, (c) HBO-MLP, (d) LCA-MLP (e) OOA-MLP, and (f) STOA-MLP.



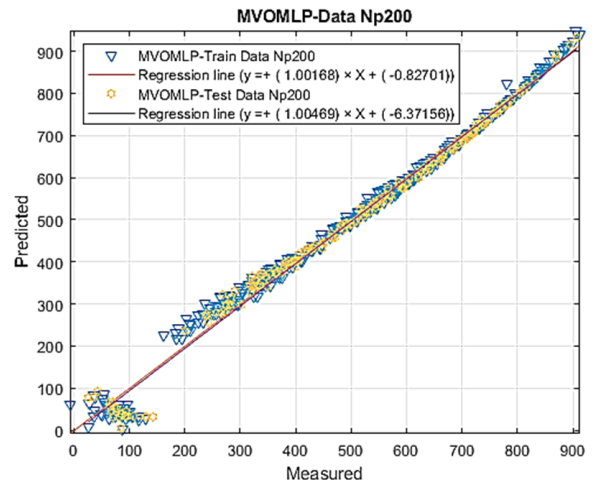
(a) MVO-MLP-N_p50



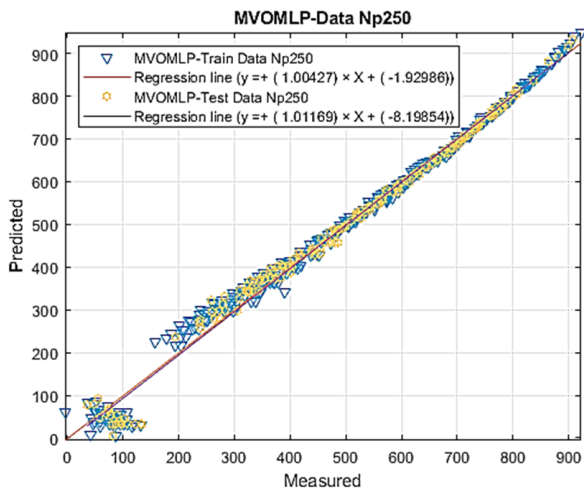
(b) MVO-MLP-N_p100



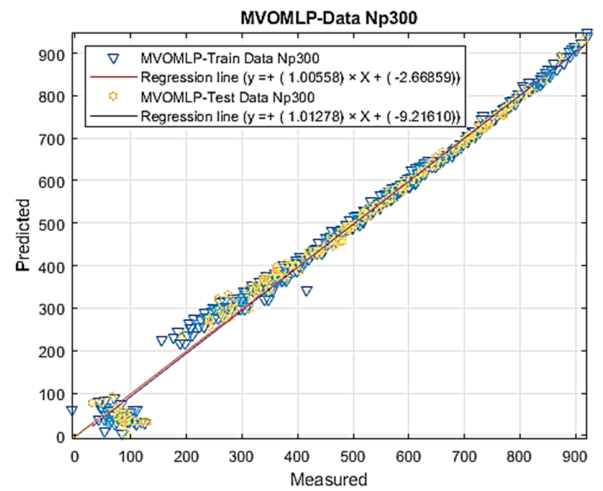
(c) MVO-MLP-N_p150



(d) MVO-MLP-N_p 200

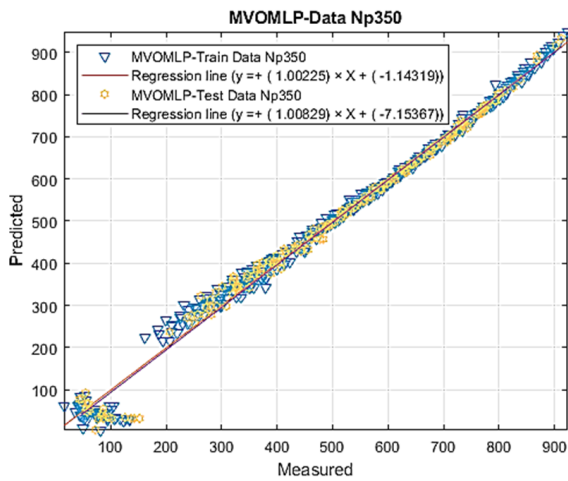


(e) MVO-MLP-N_p250

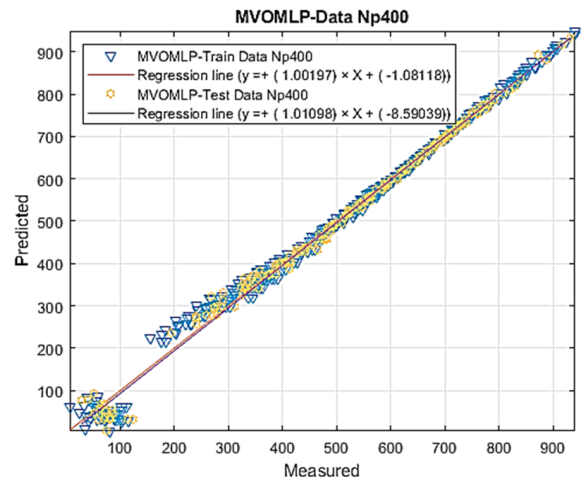


(f) MVO-MLP-N_p300

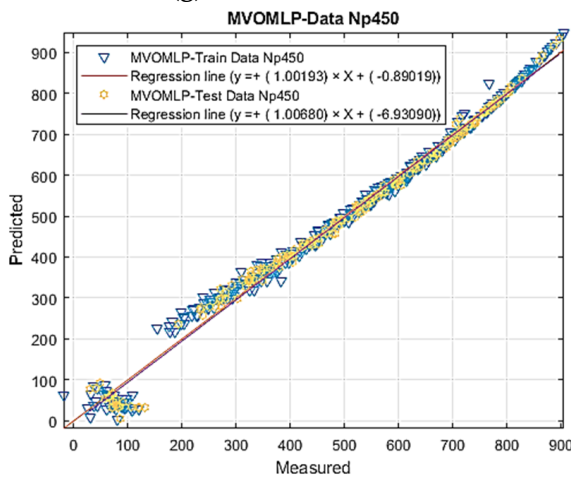
Figure 8. Cont.



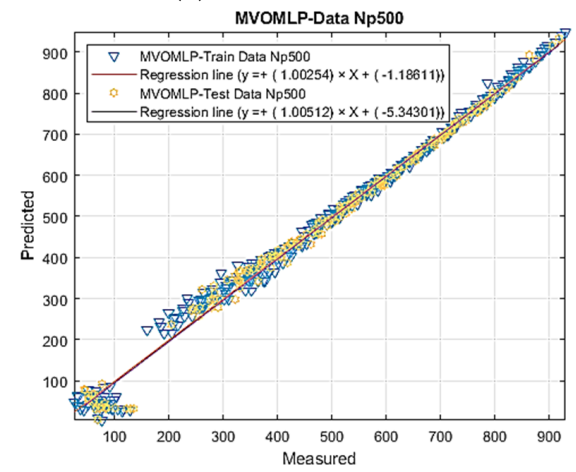
(g) MVO-MLP-N_p350



(h) MVO-MLP-N_p400

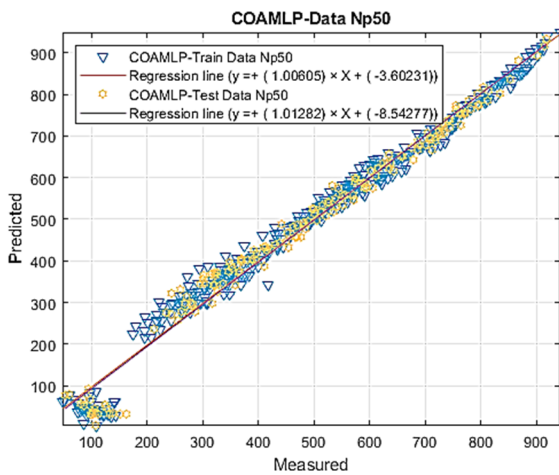


(i) MVO-MLP-N_p450

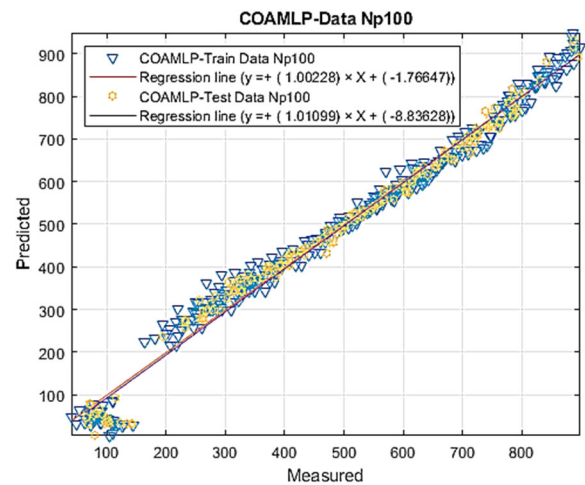


(j) MVO-MLP-N_p500

Figure 8. Precision findings of training and evaluation database for the distinct suggested MVO-MLP structure.

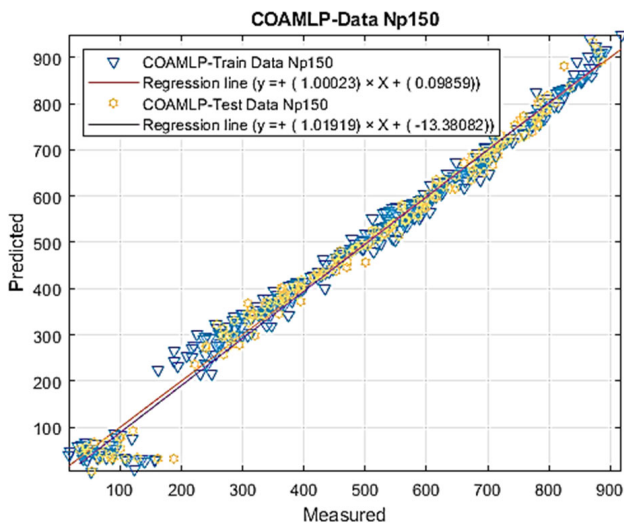


(a) COA-MLP-N_p50

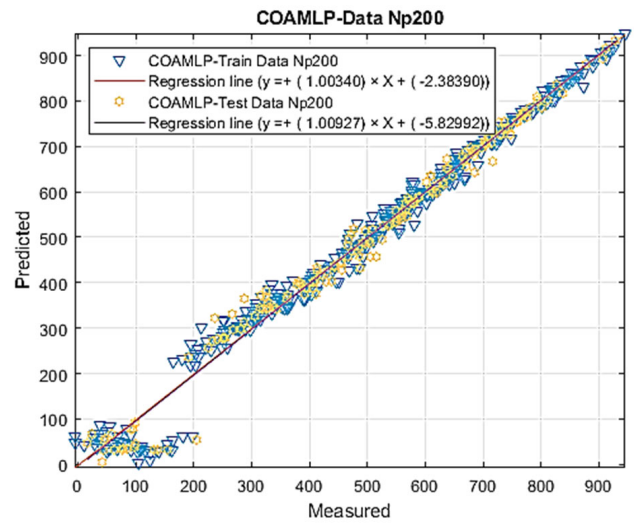


(b) COA-MLP-N_p100

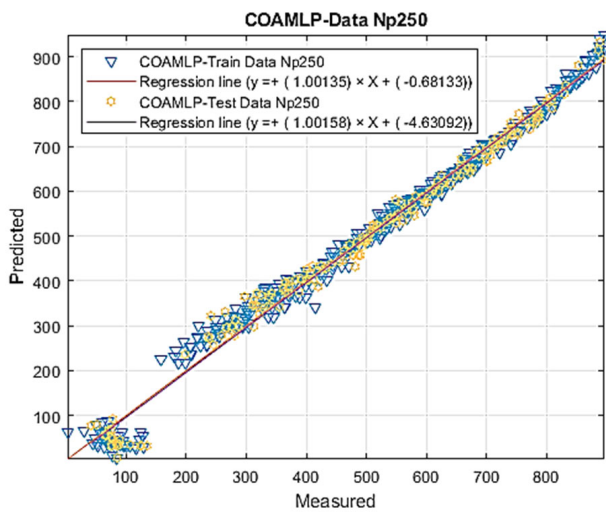
Figure 9. Cont.



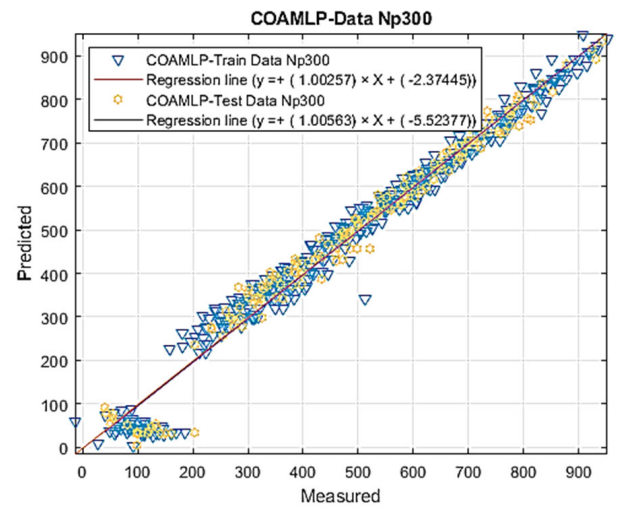
(c) COA-MLP- N_p150



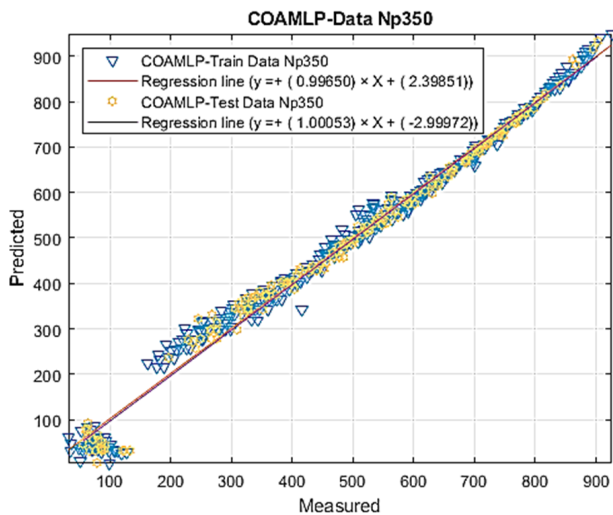
(d) COA-MLP- N_p200



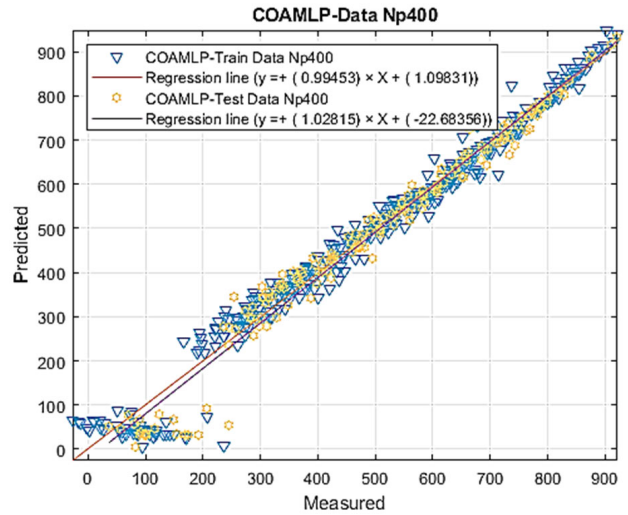
(e) COA-MLP- N_p250



(f) COA-MLP- N_p300



(g) COA-MLP- N_p350



(h) COA-MLP- N_p400

Figure 9. Cont.

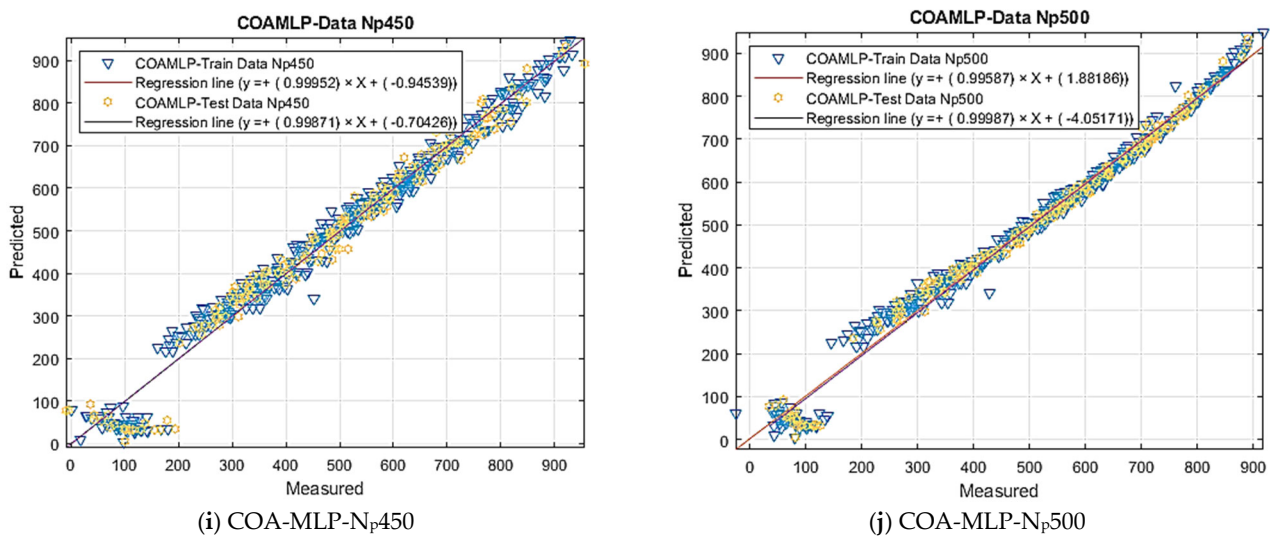


Figure 9. Precision findings of training and evaluation database for distinct suggested COA-MLP structure.

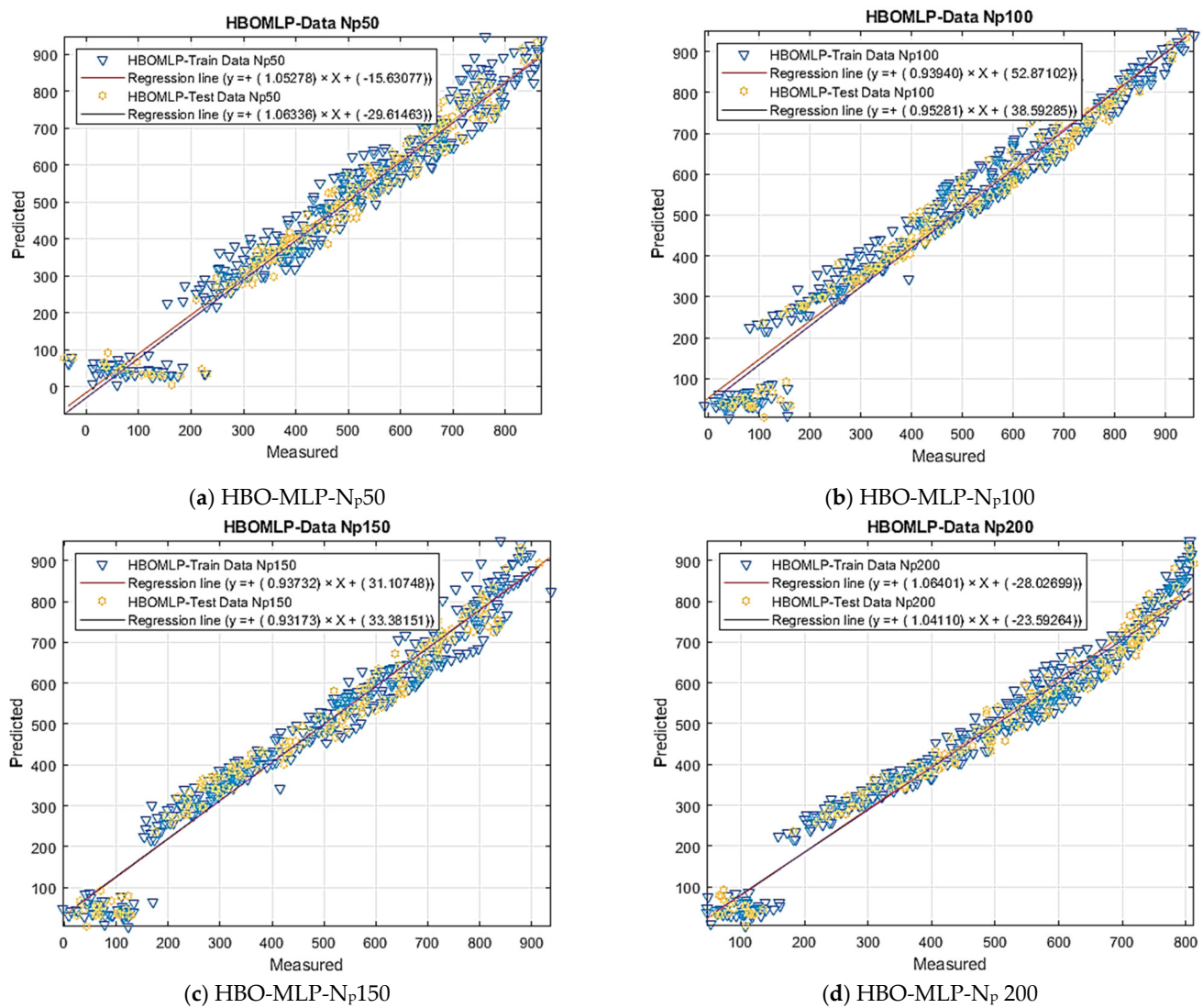


Figure 10. Cont.

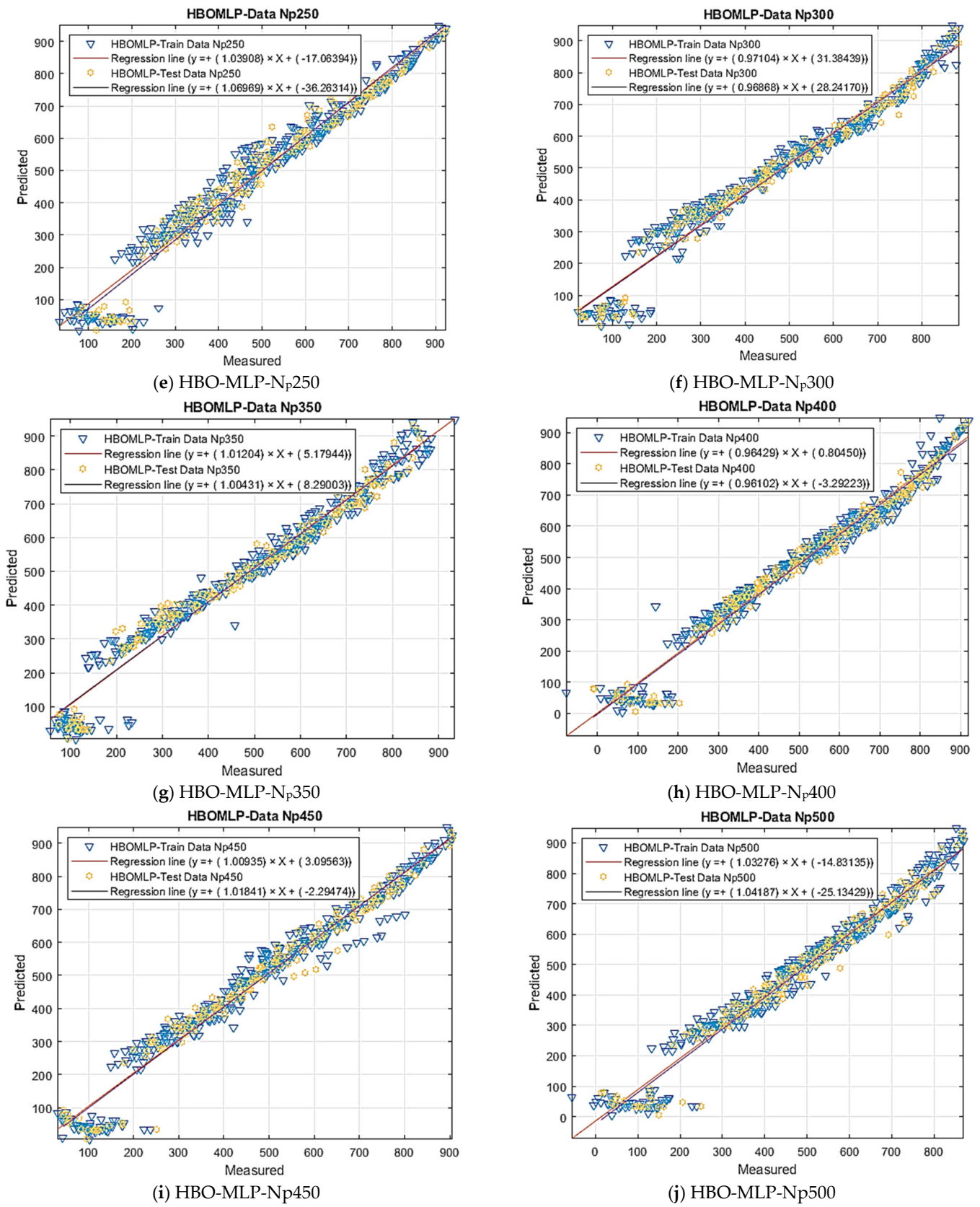
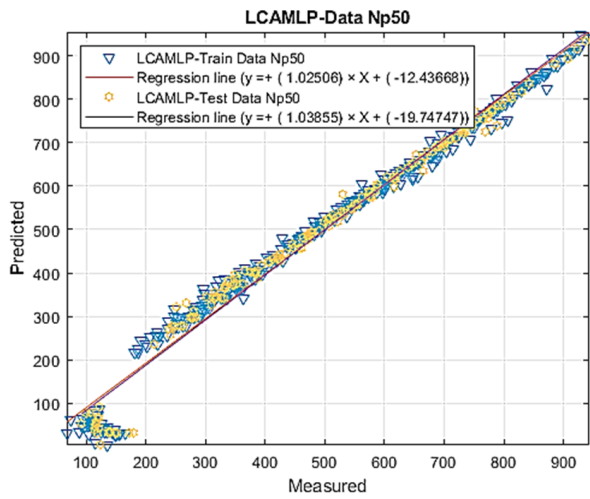
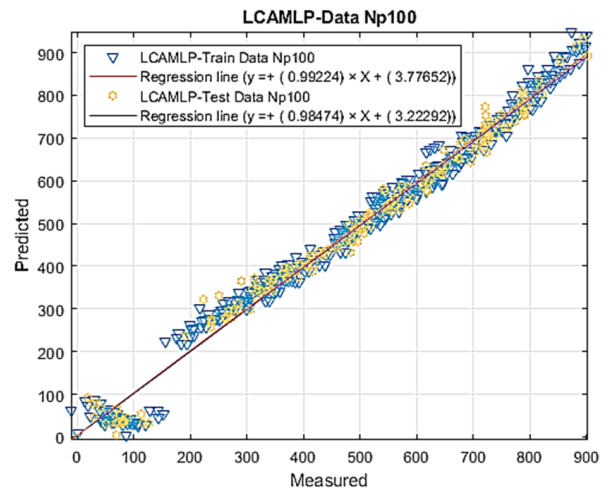


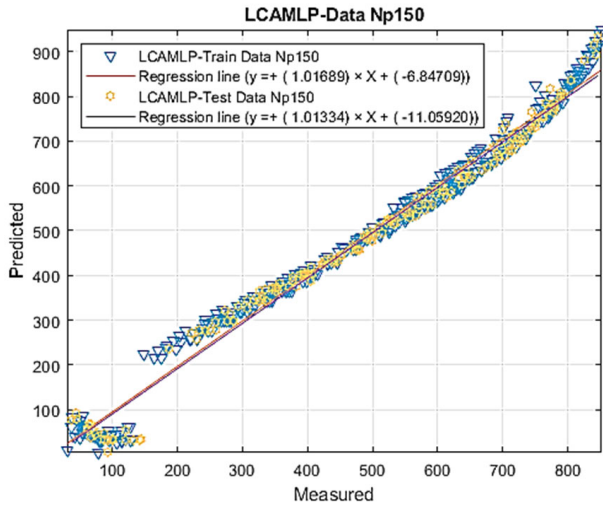
Figure 10. Precision findings of training and evaluation database for the distinct suggested HBO-MLP structure.



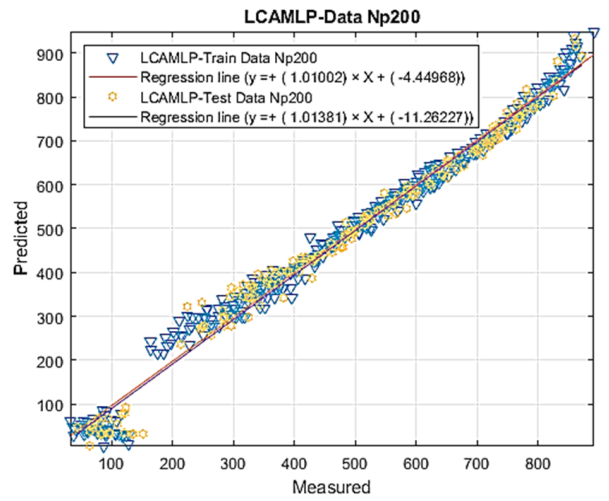
(a) LCA-MLP-N_p50



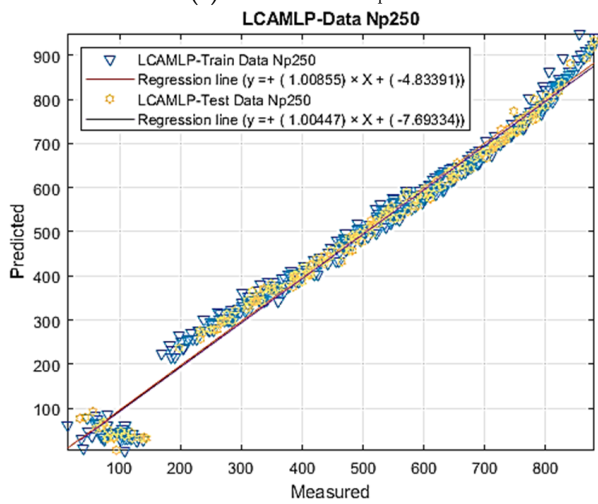
(b) LCA-MLP-N_p100



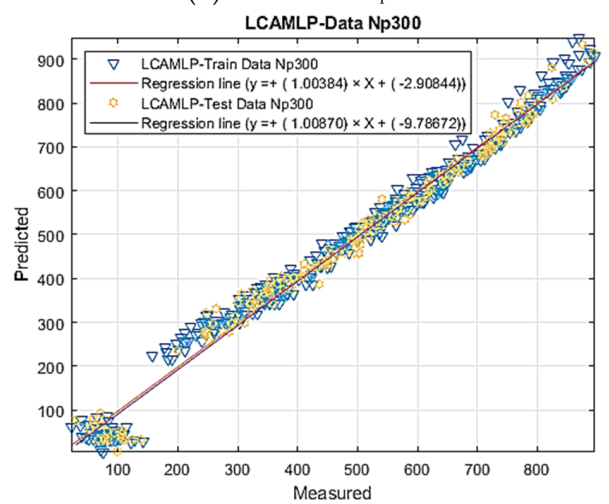
(c) LCA-MLP-N_p150



(d) LCA-MLP-N_p 200

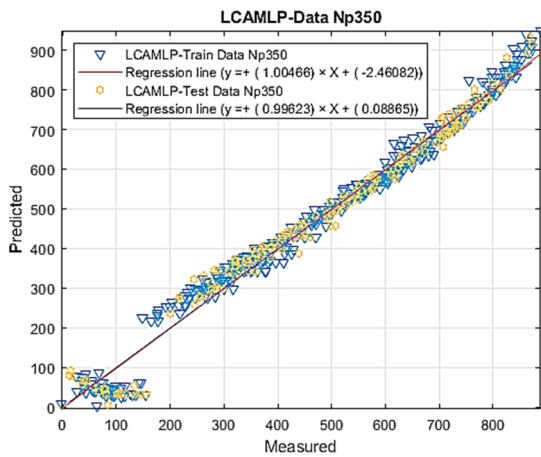


(e) LCA-MLP-N_p250

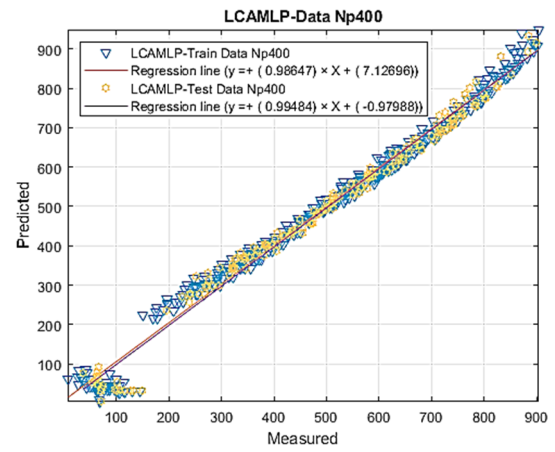


(f) LCA-MLP-N_p300

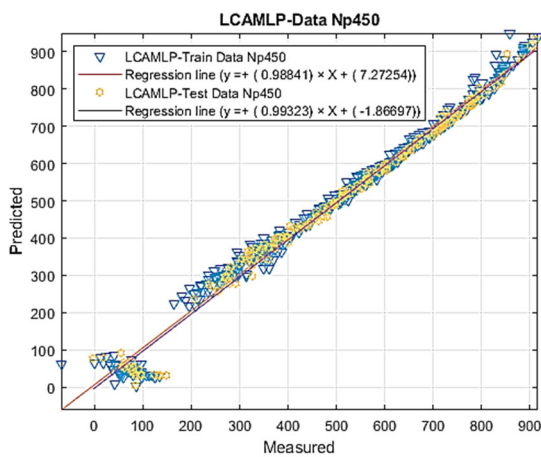
Figure 11. Cont.



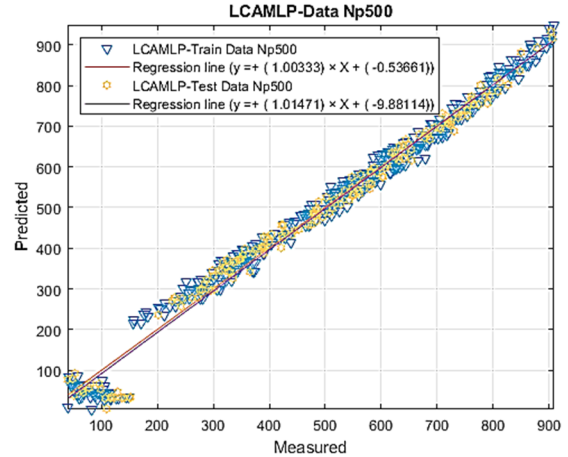
(g) LCA-MLP-N_p350



(h) LCA-MLP-N_p400

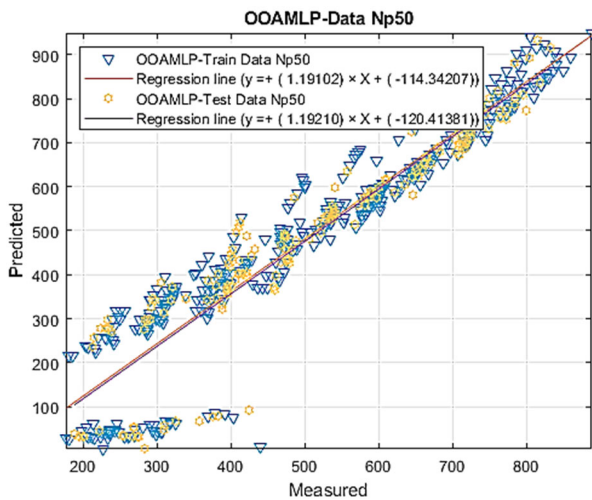


(i) LCA-MLP-N_p450

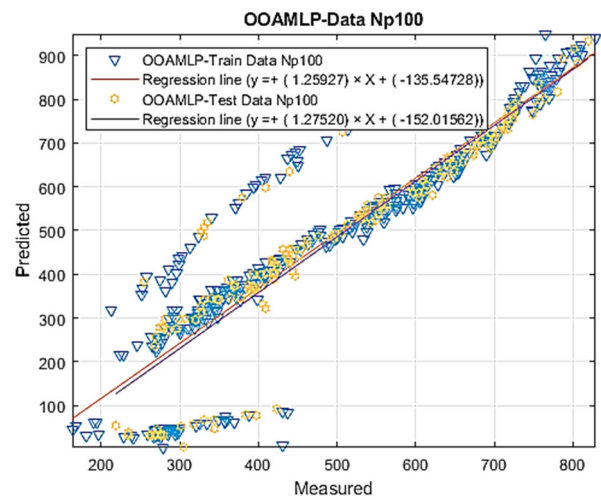


(j) LCA-MLP-N_p500

Figure 11. Precision findings of training and evaluation database for the distinct suggested LCA-MLP structure.

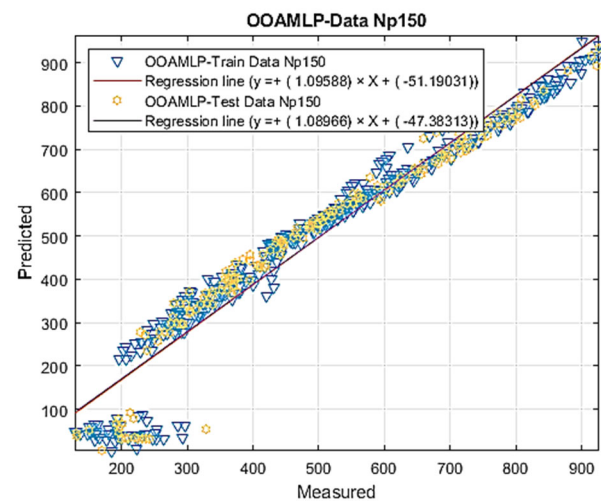


(a) OOA-MLP-N_p50

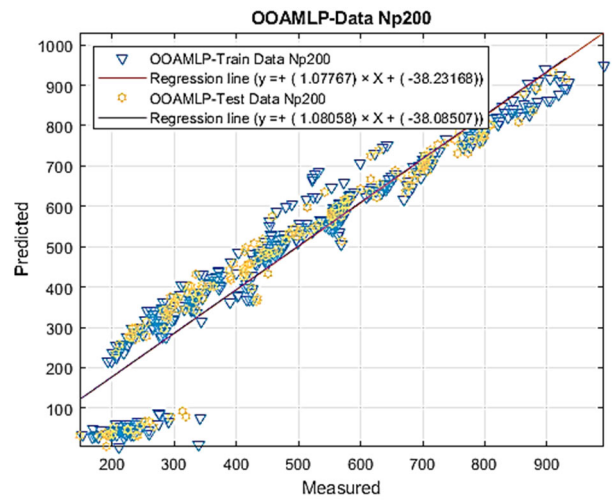


(b) OOA-MLP-N_p100

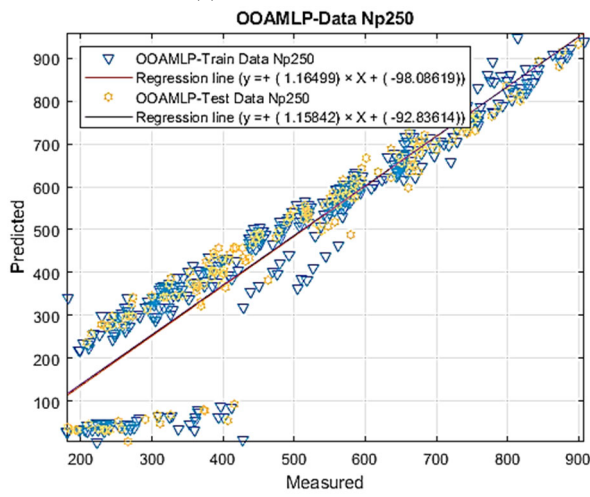
Figure 12. Cont.



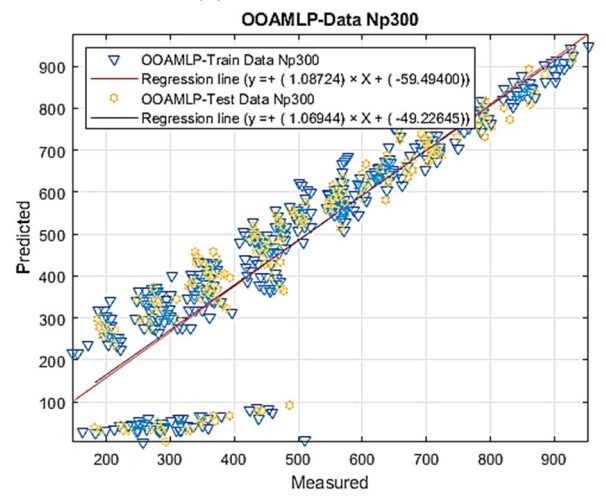
(c) OOA-MLP-N_p150



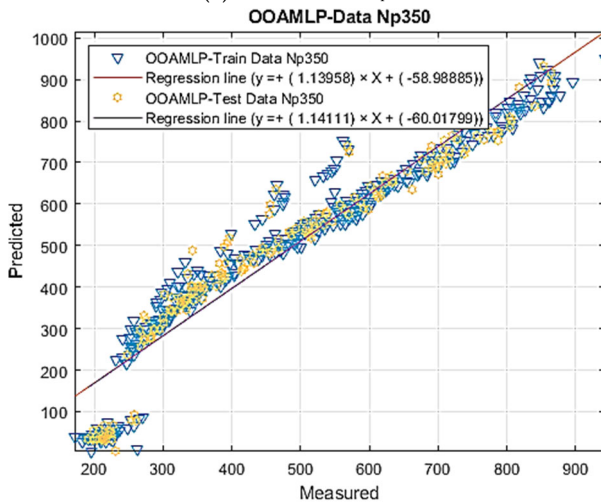
(d) OOA-MLP-N_p 200



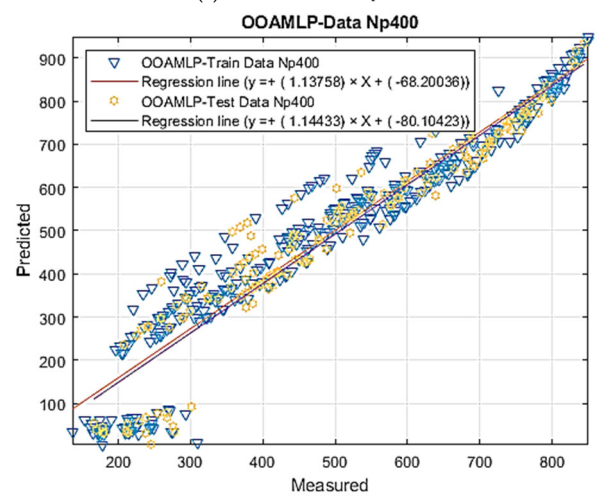
(e) OOA-MLP-N_p250



(f) OOA-MLP-N_p300

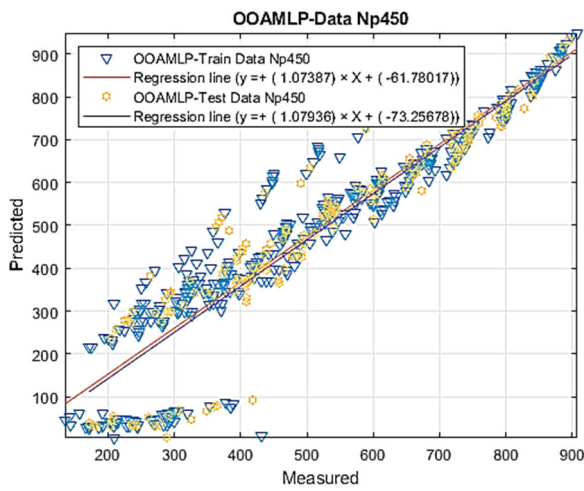


(g) OOA-MLP-N_p350

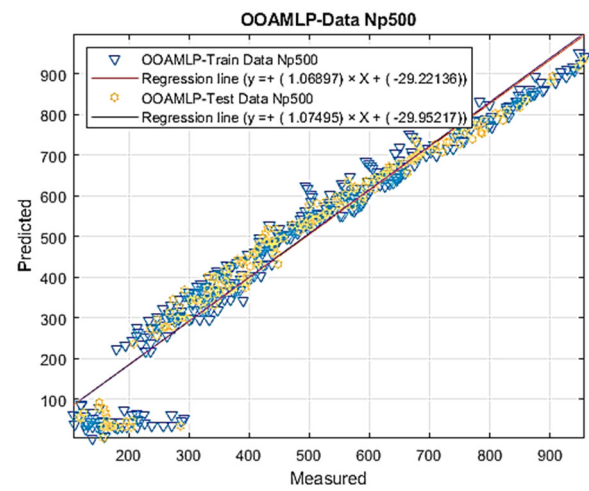


(h) OOA-MLP-N_p400

Figure 12. Cont.

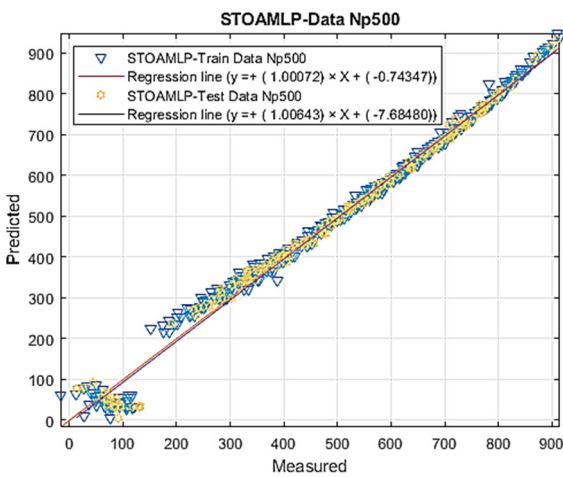


(i) OOA-MLP-N_p450

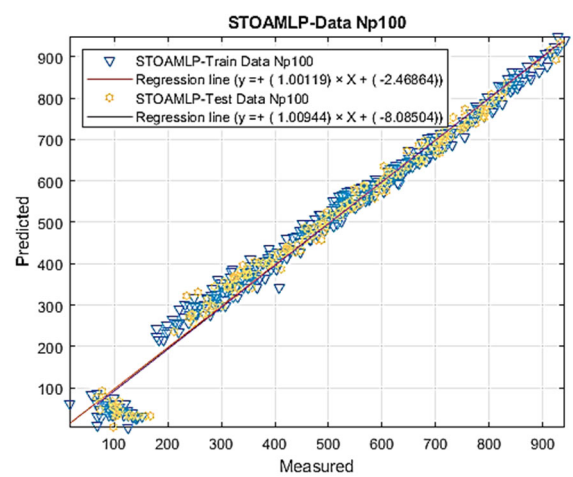


(j) OOA-MLP-N_p500

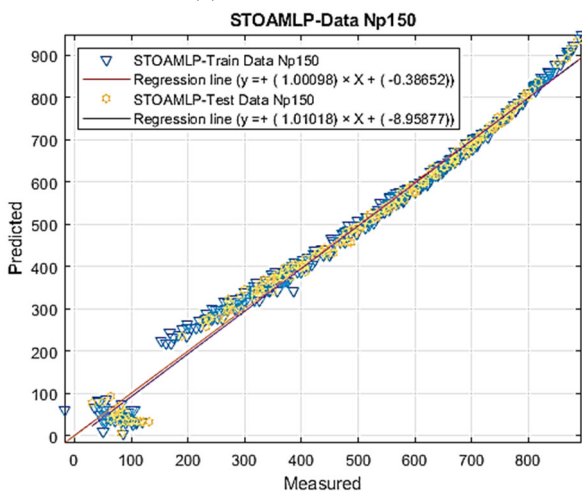
Figure 12. Precision findings of training and evaluation database for the distinct suggested OOA-MLP structure.



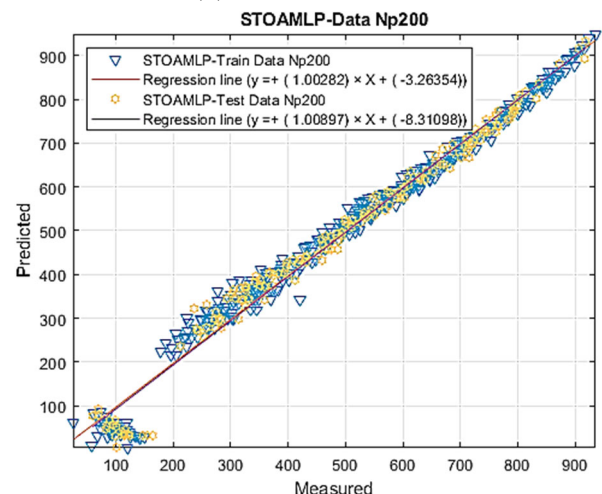
(a) STOA-MLP-N_p50



(b) STOA-MLP-N_p100

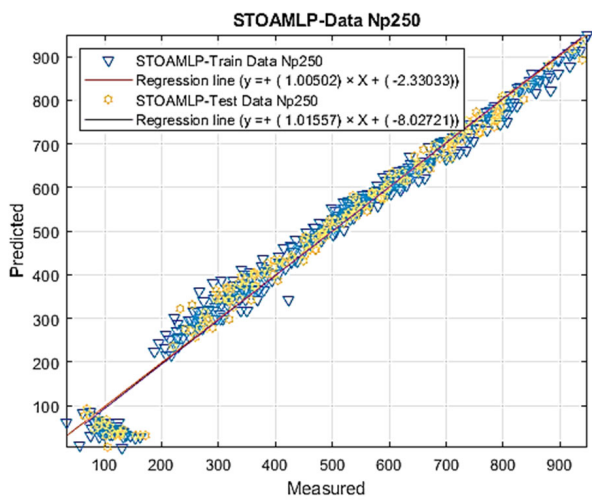


(c) STOA-MLP-N_p150

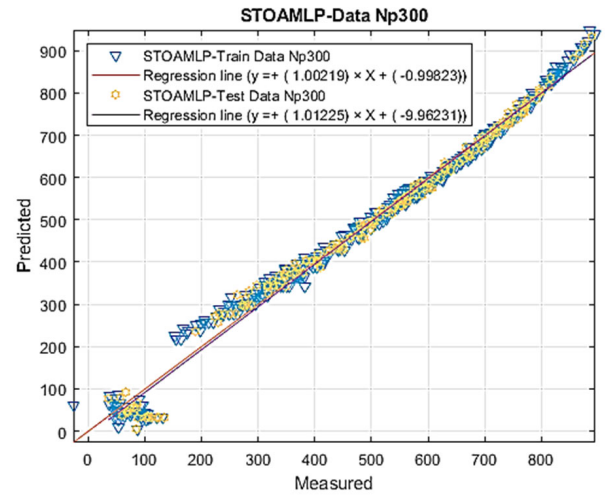


(d) STOA-MLP-N_p200

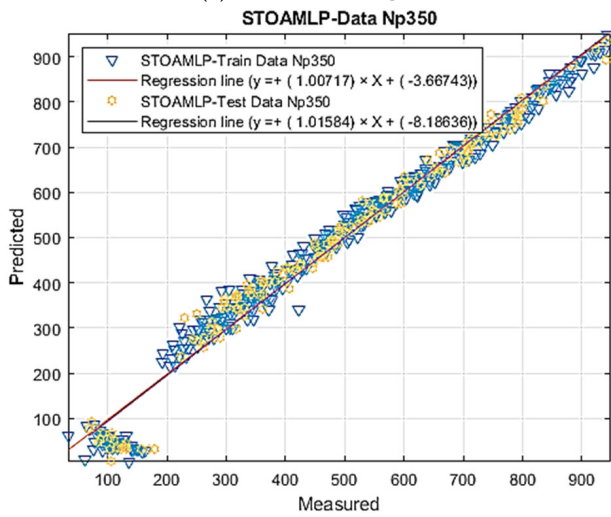
Figure 13. Cont.



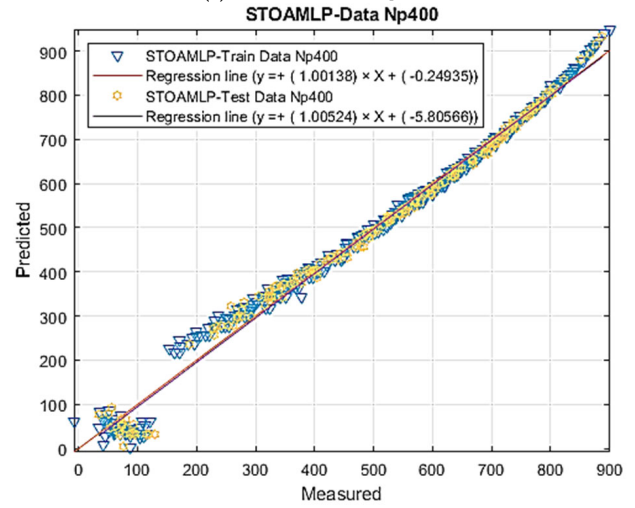
(e) STOA-MLP-N_p250



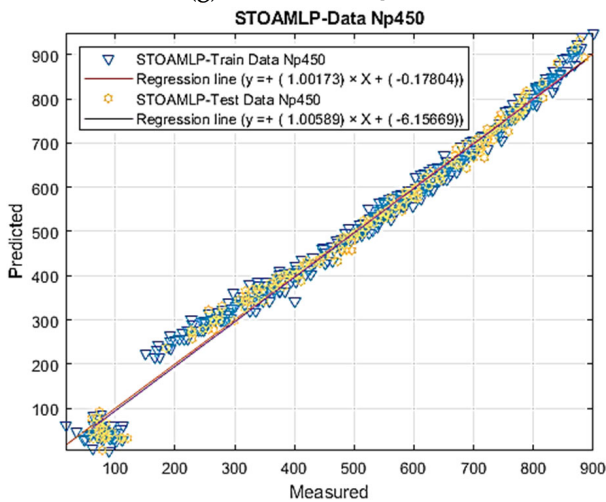
(f) STOA-MLP-N_p300



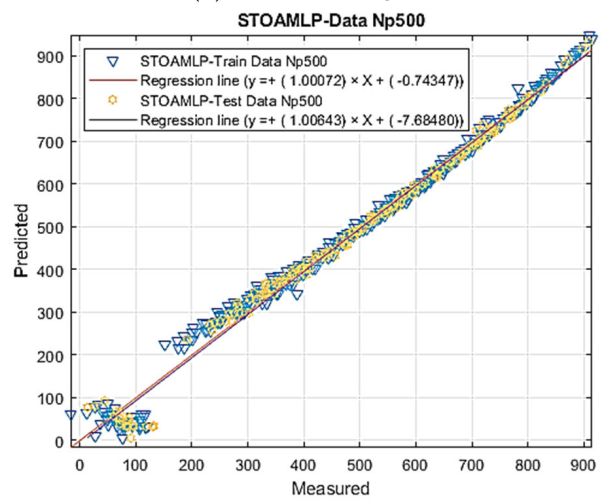
(g) STOA-MLP-N_p350



(h) STOA-MLP-N_p400



(i) STOA-MLP-N_p450



(j) STOA-MLP-N_p500

Figure 13. Precision findings of training and evaluation database for the distinct suggested STOA-MLP structure.

To confirm the consistency of the learning system, the training mechanism was repeated several times for each structure. Six models introduced in Tables 3–8 all underwent the same procedure. It is worth mentioning that a greater frequency of neurons results in more complex networks with elevated accuracy. The researchers opted for seven concealed layers as the fittest structure based on the precision of testing results and a modest increase in testing R^2 value (and a negligible reduction in the RMSE value). Consequently, an MLP architecture with a network structure of $8 \times 7 \times 1$ was selected as the best possible structure for an overall better hybridization process (for example, MVO-MLP, COA-MLP, HBO-MLP, LCA-MLP, OOA-MLP, and STOA-MLP). For the testing and training datasets, the proposed MVO-MLP model yielded R^2 of 0.9962 and 0.99598 and RMSE of 20.80626 and 20.31492, respectively. The COA-MLP, HBO-MLP, LCA-MLP, and OOA-MLP model, nevertheless, had R^2 of 0.9937, 0.9858, 0.9929, and 0.9691 in training, and 0.99358, 0.98561, 0.99252, and 0.97008 in testing; and RMSE amounts of 26.6653, 39.97074, 28.3406, and 58.72181 in training, and 25.64152, 38.31342, 27.67516, and 55.03676, in testing. Additionally, for the training and testing stages of the STOA-MLP, the R^2 are 0.9946 and 0.99395, and the RMSE are 24.80362 and 24.89665, respectively. According to Table 9, which summarizes the findings of all six tables, the formed hybrid MVO-MLP model precisely anticipates the motor’s reactance. In light of this, the proposed hybrid MVO-MLP model can be used or recommended as a cutting-edge, accurate model for predicting the motor’s reactance.

Table 3. Network outcomes for ten proposed MVO-MLP swarm sizes.

Swarm Size	Training Dataset		Testing Dataset		Scoring				Total Score	Rank
	RMSE	R^2	RMSE	R^2	Training	Testing	Training	Testing		
50	23.63823	0.9951	24.11876	0.99432	6	6	5	5	22	5
100	25.11094	0.9944	26.27564	0.99326	1	1	1	1	4	10
150	22.22501	0.9956	23.80023	0.99447	8	8	7	7	30	3
200	23.84449	0.9950	24.57425	0.99411	5	5	4	4	18	6
250	24.1811	0.9948	24.76236	0.99402	3	3	3	3	12	8
300	23.33629	0.9952	23.65243	0.99454	7	7	8	8	30	3
350	23.90106	0.9949	25.61831	0.99359	4	4	2	2	12	8
400	20.80626	0.9962	20.31492	0.99598	10	10	10	10	40	1
450	24.51922	0.9947	24.03486	0.99436	2	2	6	6	16	7
500	22.14183	0.9957	22.65417	0.99499	9	9	9	9	36	2

Table 4. Network outcomes for ten proposed COA-MLP swarm sizes.

Swarm Size	Training Dataset		Testing Dataset		Scoring				Total Score	Rank
	RMSE	R^2	RMSE	R^2	Training	Testing	Training	Testing		
50	34.08308	0.9897	34.06019	0.98865	4	4	4	4	16	7
100	31.44115	0.9912	29.71232	0.99137	5	5	7	7	24	5
150	28.26581	0.9929	30.61575	0.99084	8	8	6	6	28	4
200	31.10083	0.9914	30.84898	0.9907	6	6	5	5	22	6
250	27.41033	0.9933	27.70597	0.9925	9	9	8	8	34	2
300	37.17817	0.9877	38.33216	0.9856	1	1	3	3	8	8
350	26.6653	0.9937	25.64152	0.99358	10	10	10	10	40	1
400	36.48612	0.9882	38.98085	0.9851	2	2	2	2	8	8
450	35.18528	0.9890	40.22532	0.98413	3	3	1	1	8	8
500	28.3602	0.9929	27.34985	0.99269	7	7	9	9	32	3

Table 5. Network outcomes for ten proposed HBO-MLP swarm sizes.

Swarm Size	Training Dataset		Testing Dataset		Scoring		Total Score	Rank		
	RMSE	R ²	RMSE	R ²	Training	Testing				
50	49.45456	0.9782	46.48186	0.97875	1	1	2	2	6	9
100	47.41478	0.9799	46.34932	0.97887	4	4	3	3	14	7
150	47.87484	0.9796	44.38386	0.98064	3	3	4	4	14	7
200	40.9752	0.9851	38.16262	0.98573	9	9	10	10	38	1
250	48.13358	0.9793	48.15308	0.97718	2	2	1	1	6	9
300	46.59209	0.9806	39.55079	0.98466	5	5	8	8	26	4
350	45.74193	0.9813	43.03009	0.98182	7	7	5	5	24	5
400	39.97074	0.9858	38.31342	0.98561	10	10	9	9	38	1
450	46.52896	0.9807	42.55685	0.98222	6	6	6	6	24	5
500	42.47182	0.9839	41.92675	0.98275	8	8	7	7	30	3

Table 6. Network outcomes for ten proposed LCA-MLP swarm sizes.

Swarm Size	Training Dataset		Testing Dataset		Scoring		Total Score	Rank		
	RMSE	R ²	RMSE	R ²	Training	Testing				
50	34.47294	0.9895	35.40373	0.98773	1	1	1	1	4	10
100	30.54225	0.9917	30.22369	0.99107	4	4	6	6	20	6
150	31.50901	0.9912	31.1856	0.99049	3	3	5	5	16	8
200	28.6116	0.9927	31.57711	0.99025	8	8	4	4	24	5
250	30.26023	0.9919	28.95705	0.99181	5	5	9	9	28	3
300	29.91963	0.9921	29.47933	0.99151	7	7	7	7	28	3
350	31.53647	0.9912	34.09141	0.98863	2	2	2	2	8	9
400	28.26144	0.9929	29.3445	0.99159	10	10	8	8	36	2
450	28.3406	0.9929	27.67516	0.99252	9	9	10	10	38	1
500	30.24985	0.9919	31.94492	0.99002	6	5	3	3	17	7

Table 7. Network outcomes for ten proposed OOA-MLP swarm sizes.

Swarm Size	Training Dataset		Testing Dataset		Scoring		Total Score	Rank		
	RMSE	R ²	RMSE	R ²	Training	Testing				
50	80.24177	0.9414	82.22691	0.93189	5	5	4	4	18	6
100	99.9979	0.9074	93.41085	0.91114	2	2	2	2	8	9
150	60.22237	0.9674	61.38316	0.96264	9	9	9	9	36	2
200	75.74714	0.9480	74.62977	0.94425	6	6	6	6	24	5
250	85.32182	0.9335	81.59779	0.93296	4	4	5	5	18	6
300	101.28839	0.9049	105.2951	0.88556	1	1	1	1	4	10
350	71.63692	0.9536	71.28501	0.94926	7	7	7	7	28	4
400	70.02715	0.9557	70.14969	0.95091	8	8	8	8	32	3
450	85.97935	0.9324	86.26555	0.92475	3	3	3	3	12	8
500	58.72181	0.9691	55.03676	0.97008	10	10	10	10	40	1

Table 9 demonstrates that the optimum population amounts for the MVO-MLP, COA-MLP, HBO-MLP, LCA-MLP, OOA-MLP, and STOA-MLP were estimated to be 400, 350, 400, 450, 500, and 500, respectively. The mistakes occurrence and the minimum mistake number in the MVO-MLP, COA-MLP, HBO-MLP, LCA-MLP, OOA-MLP, and STOA-MLP best-fitted structures are demonstrated in Figures 14–19, respectively. The outcomes obtained from the testing and training database show a great agreement among the observed and estimated amounts of motor reactance. During the training stages, the MAE amounts of 13.6075, 19.9943, 33.1182, 21.0433, 43.0551, and 17.8025 were reached for the MVO-MLP, COA-MLP, HBO-MLP, LCA-MLP, OOA-MLP, and STOA-MLP, respectively. Also, the MSE amounts related to the MVO-MLP, COA-MLP, HBO-MLP, LCA-MLP, OOA-MLP, and STOA-MLP models are correspondingly equal to 433.1356, 712.2449, 1961.1447, 813.8206, 3683.0843,

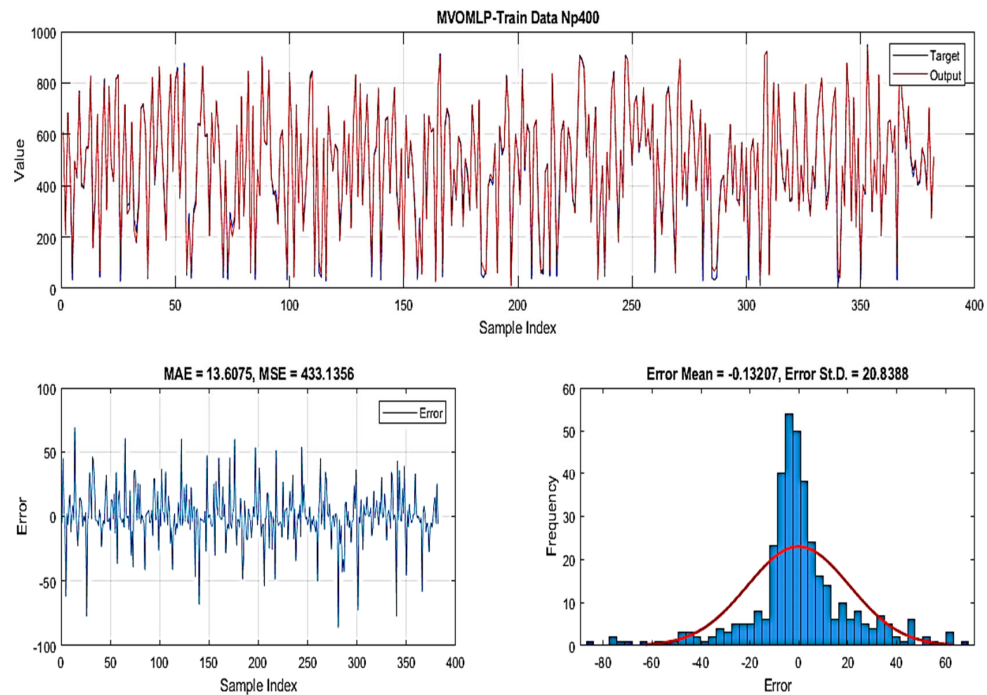
and 615.406. According to the error values, it is evident that MVO-MLP is a more reliable predictive network than the other proposed algorithms for approximating real-world induction motor reactance.

Table 8. Network outcomes for ten proposed STOA-MLP swarm sizes.

Swarm Size	Training Dataset		Testing Dataset		Scoring				Total Score	Rank
	RMSE	R ²	RMSE	R ²	Training	Testing	Training	Testing		
50	32.77473	0.9905	30.24719	0.99106	5	5	5	5	20	6
100	33.41437	0.9901	34.52783	0.98833	3	3	3	3	12	8
150	26.3452	0.9939	25.40095	0.9937	8	8	8	8	32	3
200	33.0703	0.9903	33.86622	0.98878	4	4	4	4	16	7
250	36.46323	0.9882	36.54035	0.98692	2	2	2	2	8	9
300	27.32019	0.9934	26.66101	0.99306	7	7	6	6	26	4
350	37.16543	0.9877	37.65716	0.9861	1	1	1	1	4	10
400	25.61274	0.9942	24.79169	0.994	9	9	10	10	38	1
450	28.03065	0.9930	26.49107	0.99315	6	6	7	7	26	4
500	24.80362	0.9946	24.89665	0.99395	10	10	9	9	38	1

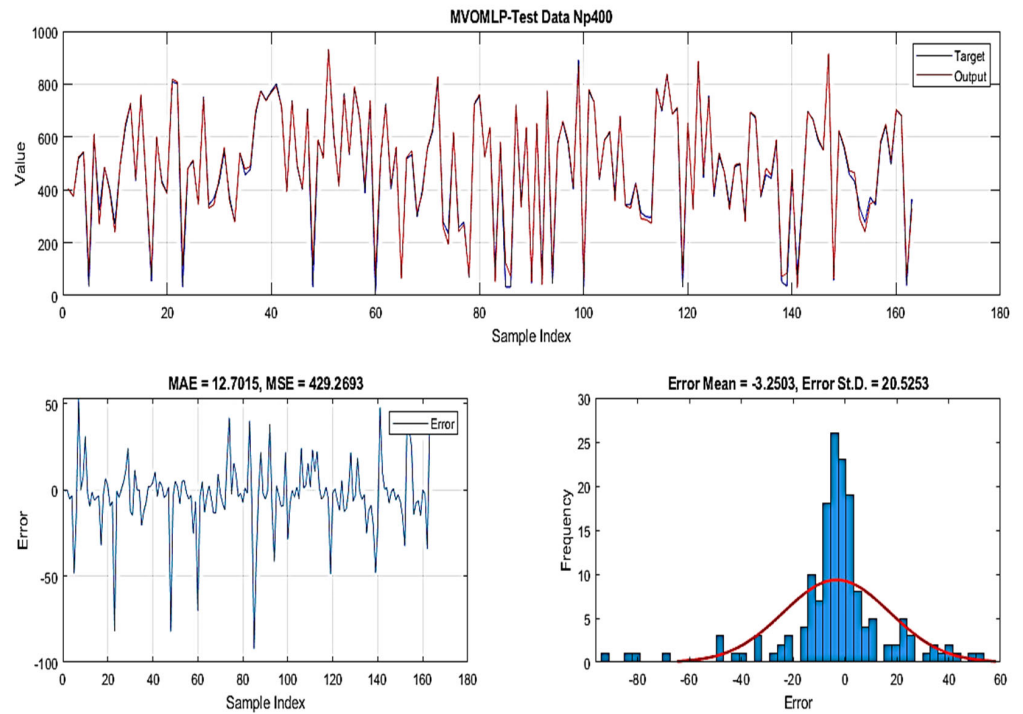
Table 9. R² and RMSE results for different suggested hybrid approaches.

Method	Swarm Size	Training Dataset		Testing Dataset		Scoring				Total Score	Rank
		RMSE	R ²	RMSE	R ²	Training	Testing	Training	Testing		
MVO-MLP	400	20.80626	0.9962	20.31492	0.99598	6	6	6	6	24	1
COA-MLP	350	26.6653	0.9937	25.64152	0.99358	4	4	4	4	16	3
HBO-MLP	400	39.97074	0.9858	38.31342	0.98561	2	2	2	2	8	5
LCA-MLP	450	28.3406	0.9929	27.67516	0.99252	3	3	3	3	12	4
OOA-MLP	500	58.72181	0.9691	55.03676	0.97008	1	1	1	1	4	6
STOA-MLP	500	24.80362	0.9946	24.89665	0.99395	5	5	5	5	20	2



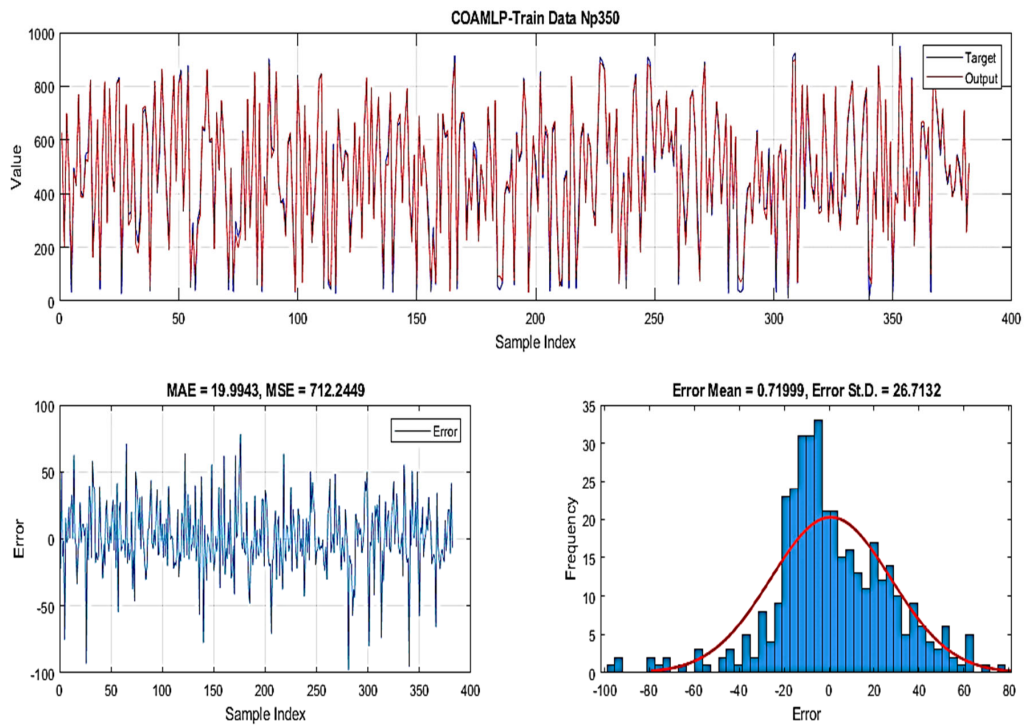
(a) MVO-MLP 400-Training

Figure 14. Cont.



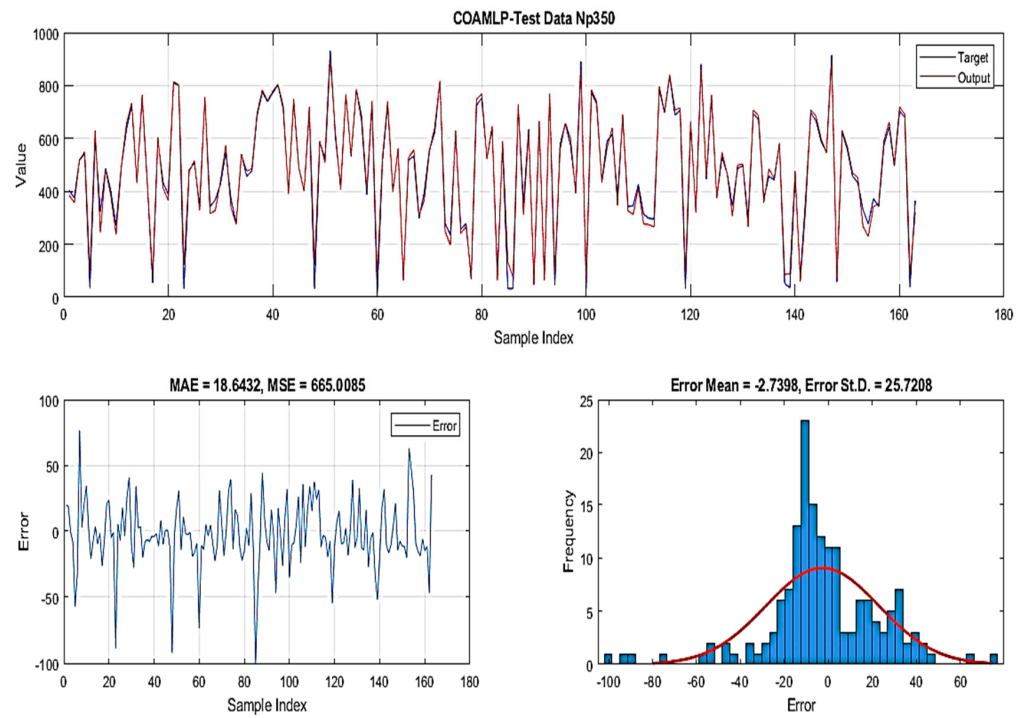
(b) MVO-MLP 400-Testing

Figure 14. The MAE frequency and error for the best fit MVO-MLP suggested method.



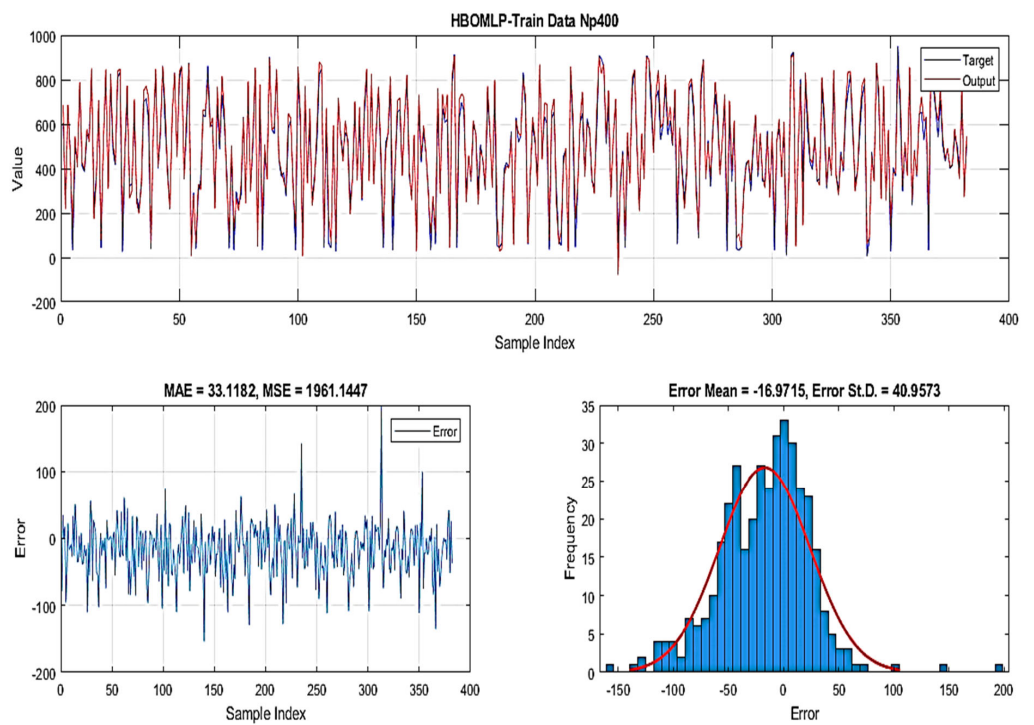
(a) COA-MLP 350-Training

Figure 15. Cont.



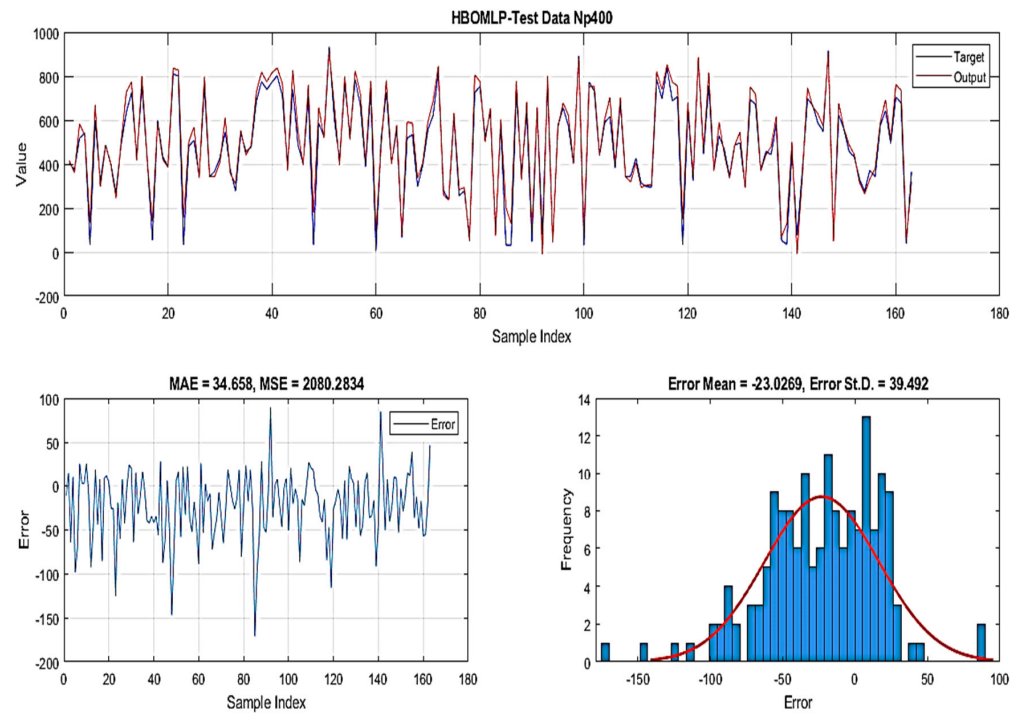
(b) COA-MLP 350-Testing

Figure 15. The MAE frequency and error for the best fit COA-MLP suggested method.



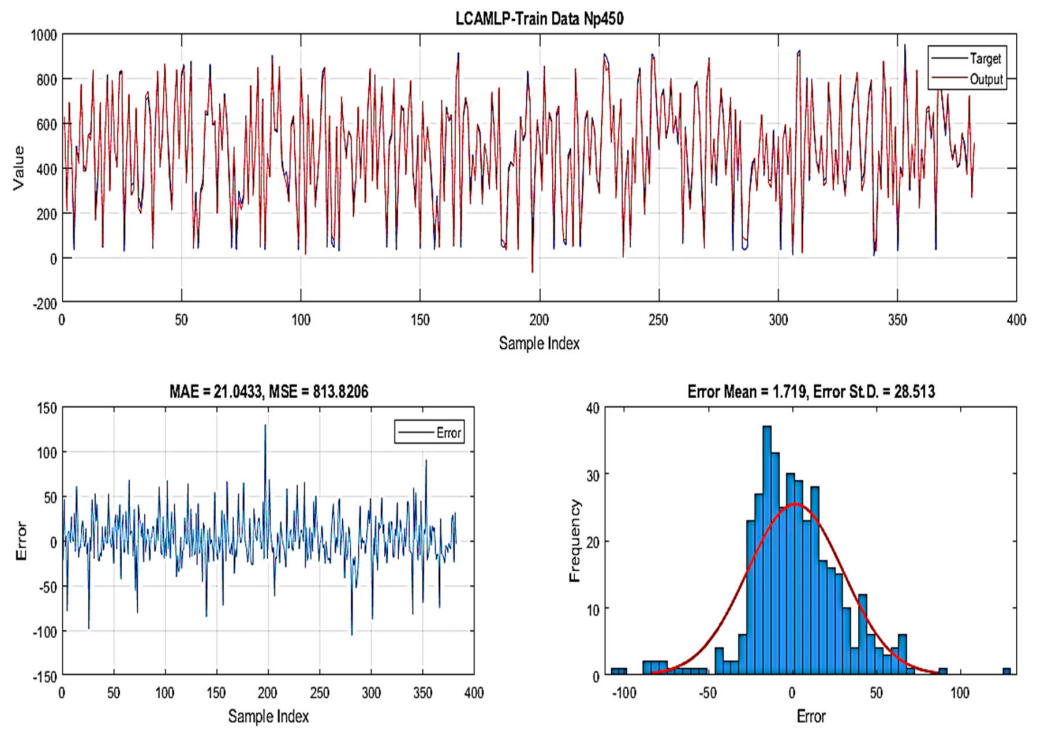
(a) HBO-MLP 400-Training

Figure 16. Cont.



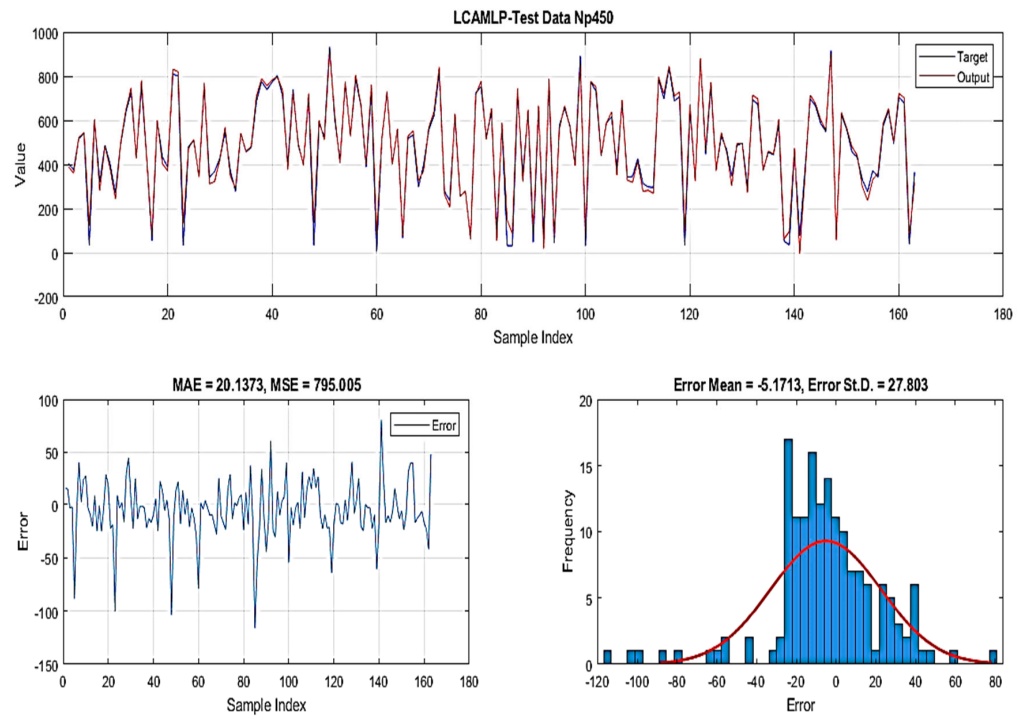
(b) HBO-MLP 400-Testing

Figure 16. The MAE frequency and error for the best fit HBO-MLP suggested method.



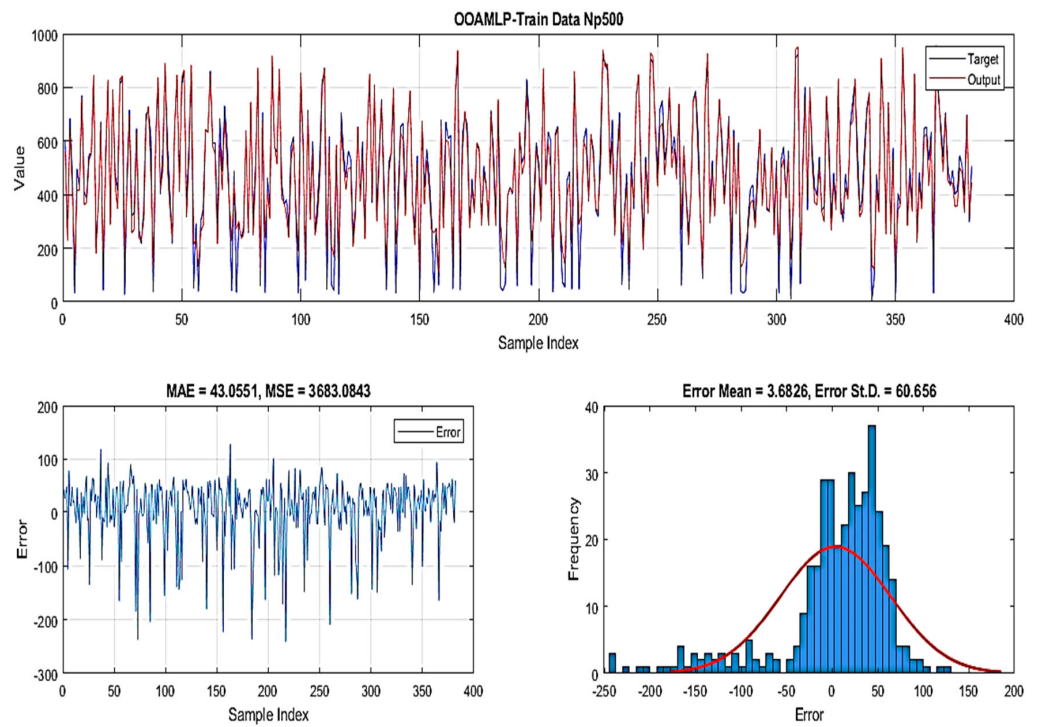
(a) LCA-MLP 450-Training

Figure 17. Cont.



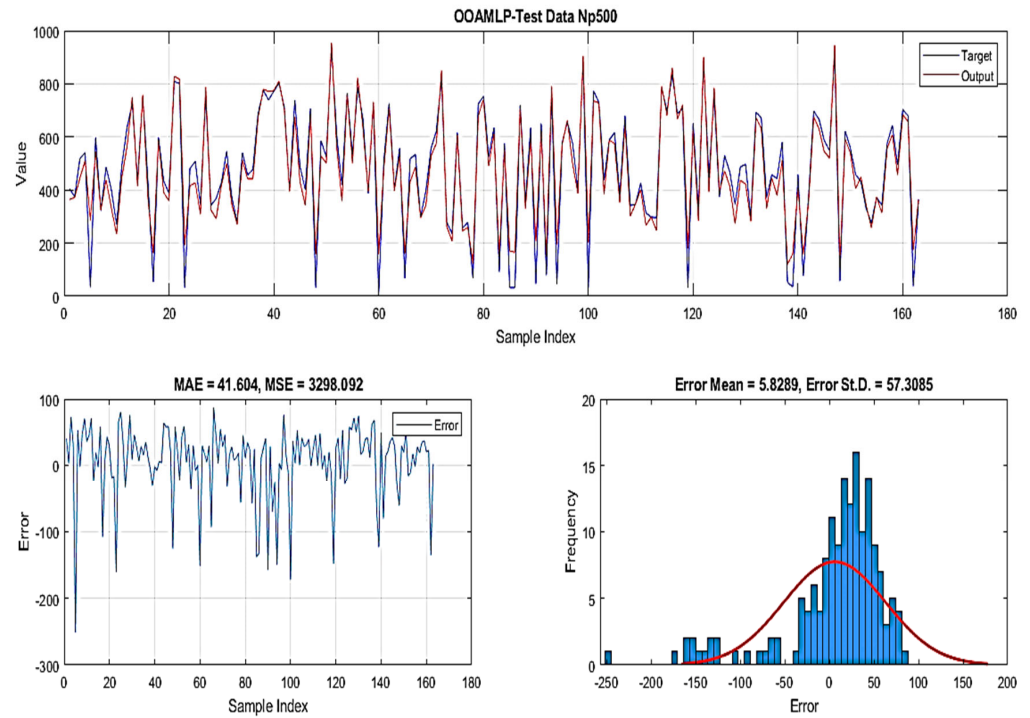
(b) LCA-MLP 450-Testing

Figure 17. The MAE frequency and error for the best fit LCA-MLP suggested method.



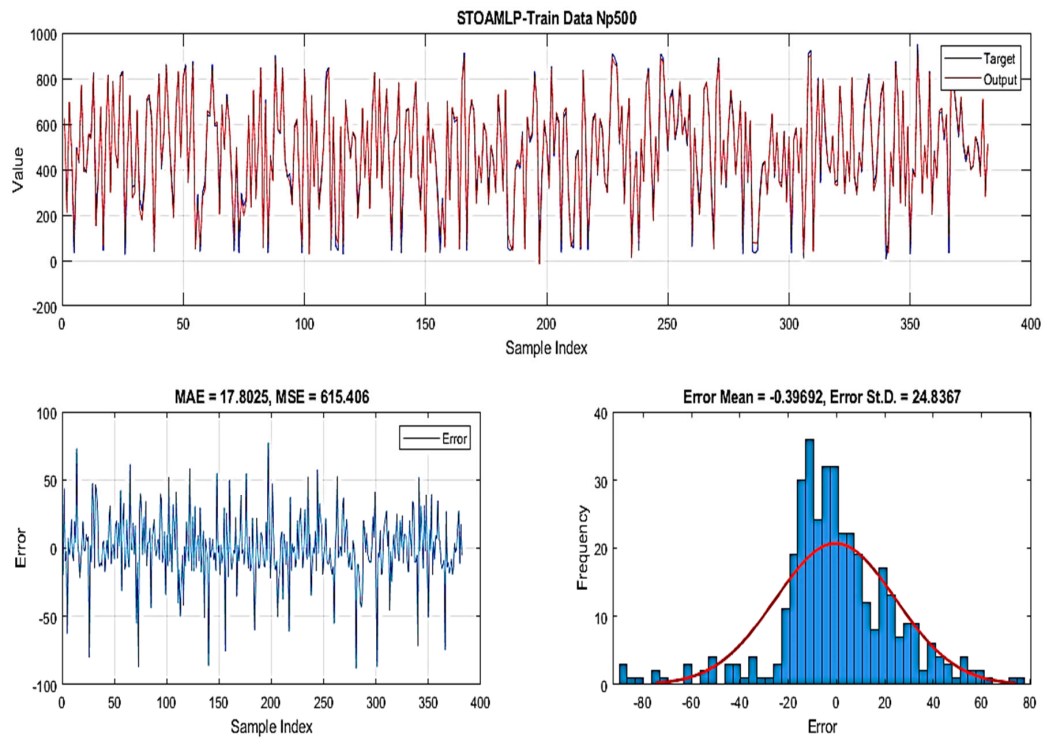
(a) OOA-MLP 500-Training

Figure 18. Cont.



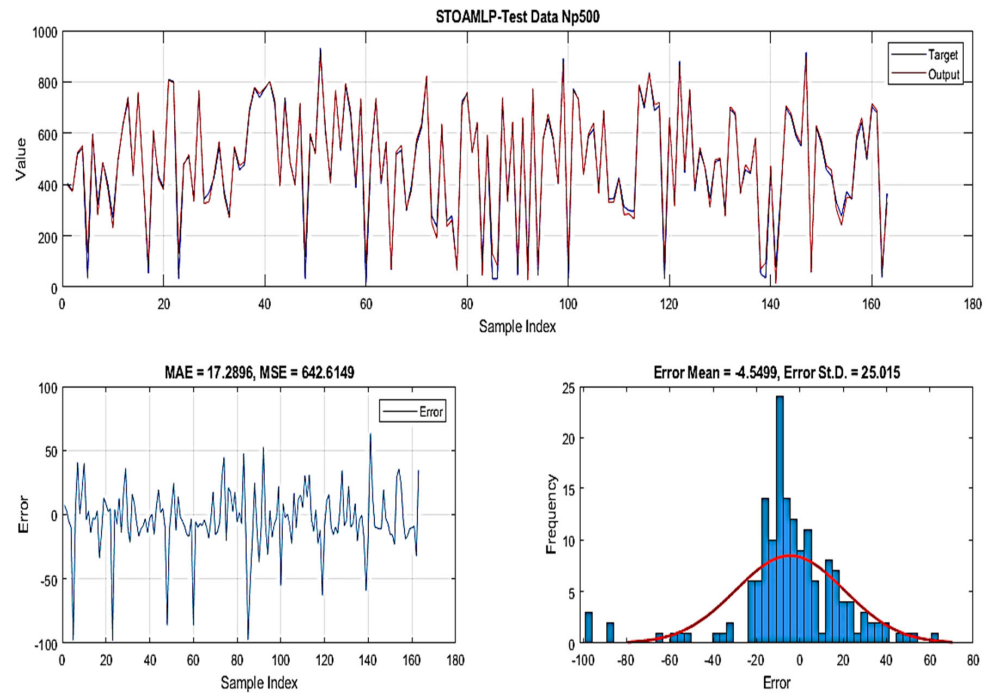
(b) OOA-MLP 500-Testing

Figure 18. The MAE frequency and error for the best fit OOA-MLP suggested method.



(a) STOA-MLP 500-Training

Figure 19. Cont.



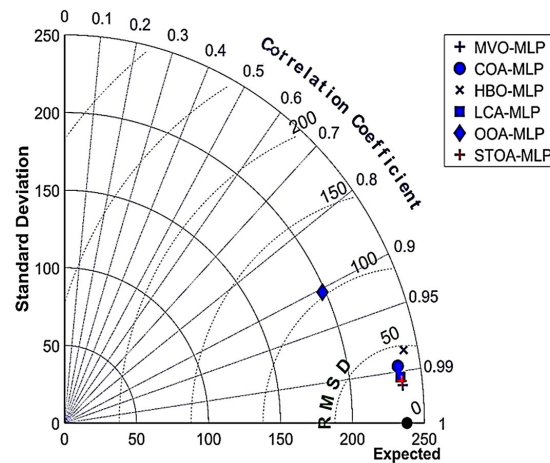
(b) STOAMLP 500-Testing

Figure 19. The MAE frequency and error for the best fit STOAMLP suggested method.

Taylor Diagram

In addition to the statistical parameters (RMSE, MAE, and R^2), the Taylor diagram [61] was utilized to evaluate the mentioned methods' accuracy. This graph accurately maps the predicted and observed data [62]. Taylor represented a solo projection for representing various parameters of evaluation. Considerably, these are qualified for focus on the methods' accuracy utilizing several plot points. The Taylor graph shows the standard deviation, RMSE amount, and correlation coefficient between observed and predicted amounts for better recognition of variations [63].

Figure 20 demonstrates Taylor graphs for different best-fit methods. The radial length from the observed value is the RMSE quantity [61]. As a result, the more accurate method is recognized by the point with the highest R^2 amount ($R^2 = 1$) and the RMSE with the minimum value. It was evident from Figure 20 that all six methods show high accuracy in predicting the induction motor reactance, but MVO-MLP produced the best prediction.



(a)

Figure 20. Cont.

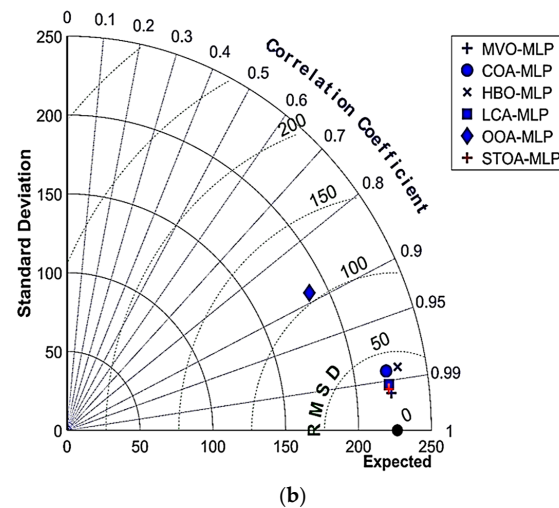


Figure 20. Taylor diagram for the best-fit structures of MVO-MLP, COA-MLP, HBO-MLP, LCA-MLP, OOA-MLP, and STOA-MLP; (a) training dataset, and (b) testing dataset.

6. Practical Implementation

As stated earlier, ANNs (e.g., including all hybridized techniques) can be trained on the historical data of induction motors, including their design specifications, geometrical parameters, operational conditions, and the corresponding reactance values. By learning from this data, the ANN can establish patterns and relationships between the motor characteristics and its reactance. Once trained, the ANN can predict a new motor's reactance based on its parameters. Their ability to learn from large datasets and generalize from learned patterns makes them valuable tools in motor analysis, optimization, and fault detection. It is noteworthy that ANNs have advantages such as flexibility, adaptability, accuracy, generalization, real-time and fast predictions, data-based optimization, and integration with automation systems. These advantages make ANNs useful and valuable for industrial applications' motor monitoring, control, and optimization. Estimating equivalent circuit variables of an induction motor using ANN can be applied in various ways in industry. In this way, the engine can optimize operations, increase reliability, reduce maintenance costs, and improve overall system efficiency. In addition, ANNs can help detect and diagnose engine faults early, prevent unexpected failures, and minimize downtime by monitoring equivalent circuit parameters. By analyzing the trends and patterns in the predicted parameters, it can be decided how to intervene according to the signs of deterioration or inefficiency in the engine's operation.

As expected from the general ANNs, the hybrid methods used in this article are expected to contribute significantly to these applications. The preceding sections have established that the MVO algorithm exhibits a marginally lower convergence curve than other algorithms, as indicated in Table 6. This proposes that, compared to alternative optimization strategies, the algorithm has achieved lower error rates when altering ANN parameters. Thus, the algorithm's outcomes are provided here to create a predictive approach. Regarding ANN optimization computations, the last neuron involves eight variables and outputs. Seven prior neural layers, each with nine variables, supply this neuron. Metaheuristic techniques are used to optimize these sixty-three variables. Equation (45) uses the MVO method to determine reactance based on seven parameters that describe neural responses in the buried layer. These parameters are O_1 , O_2 , O_3 , O_4 , O_5 , O_6 , and O_7 .

Reactance MVO-MLP = $0.2437 \times O_1 + 0.0850 \times O_2 + 0.5791 \times O_3 + 0.0937 \times O_4 - 0.3849 \times O_5 - 0.4100 \times O_6 + 0.4853 \times O_7 + 0.1542$, where considering the labels presented in Table 10,

$$O_i = \text{Tansig} ([W_{i1} \times P] + [W_{i2} [\text{Cos@}(\rho FL)]] + [W_{i3} \times \left[\frac{T_m}{T_{FL}} \right]] + [W_{i4} \times \frac{T_{ST}}{T_{FL}}] + [W_{i5} \times \frac{I_{ST}}{I_{FL}}] + [W_{i6} \times \omega_{FL}] + [W_{i7} \times \eta_{FL}] + [W_{i8} \times \text{cage}] + b_i), \tag{55}$$

Table 10. Labels description.

Input							Output	
$P(KW)$	$\text{Cos@}(\rho FL)$	T_m/T_{FL}	T_{ST}/T_{FL}	I_{ST}/I_{FL}	ω_{FL} (rpm)	η_{FL}	Cage Number	Reactance (ohms) X_m
W_{i1}	W_{i2}	W_{i3}	W_{i4}	W_{i5}	W_{i6}	W_{i7}	W_{i8}	

In which, $W_{i1}, W_{i2}, \dots, W_{i11}$, and b_i are in Table 11.

Table 11. Values of W_i .

i	W_{i1}	W_{i2}	W_{i3}	W_{i4}	W_{i5}	W_{i6}	W_{i7}	W_{i8}	b_i
1	-0.2238	0.7354	-0.9892	0.5626	-0.4626	-0.3124	1.2256	0.2762	1.4507
2	-0.0923	1.0071	0.4144	-0.2165	0.6689	-0.9779	0.1880	-0.6596	-1.0351
3	0.3254	1.5829	-0.5839	0.3391	-0.7296	0.1982	0.5122	0.1488	-0.7934
4	0.1890	-0.7108	-0.8677	0.1120	-0.8135	-1.0654	0.5661	-1.0478	0.0897
5	0.7587	-0.1117	-0.2094	0.3578	-0.8805	-0.8035	-0.4683	-0.2130	0.8239
6	-0.2406	-0.1814	-0.4286	1.2509	-1.0440	0.6871	-0.3261	0.3638	-1.3322
7	0.2390	-0.3724	0.1786	0.9425	-0.5653	0.9022	-0.1640	-0.1949	-2.1776

Indeed, the output of this study can be employed to optimize the design of induction motors by adjusting various parameters to achieve desired reactance values. By training the ANN on a dataset that includes different motor designs and their corresponding reactance values, the network can learn to identify the optimal combination of parameters that yield the desired reactance. This can aid in improving motor efficiency, reducing losses, and enhancing performance.

7. Conclusions

This research improves the modeling of induction motors using neural networks, analogous circuits, and numerical simulations. The closely linked and interdependent input-output variables in this model are one of its drawbacks; this makes it challenging and ineffective to create data patterns and train the network. In this study, the COA, LCA, HBO, MVO, OOA, and STOA were applied to estimate the variables of the reactance of the motor. Comparing the earlier investigation and the ANN estimates demonstrates that the suggested MVO-MLP method with R^2 of (0.9962 and 0.99598) and RMSE of (20.80626 and 20.31492) performed better than COA-MLP ($R^2 = 0.9937$ and 0.99358 and $RMSE = 26.6653$ and 25.64152), HBO-MLP ($R^2 = 0.9858$ and 0.98561 and $RMSE = 39.97074$ and 38.31342), LCA-MLP ($R^2 = 0.9929$ and 0.99252 and $RMSE = 28.3406$ and 27.67516), OOA-MLP ($R^2 = 0.9691$ and 0.97008 and $RMSE = 58.72181$ and 55.03676), and STOA-MLP ($R^2 = 0.9946$ and 0.99395 and $RMSE = 24.80362$ and 24.89665). Not only has the suggested MVO-MLP generated a better outcome than others, but it is also very close to actual results. The suggested ANNs have yielded excellent results for the projected model.

According to the findings, metaheuristic solutions can offer valuable perspectives for optimizing induction motor reactance. Metaheuristic algorithms used in this study are well-suited for handling complex problems. These algorithms can address multimodal optimization problems by searching for multiple solutions simultaneously or adapting their search strategies to explore different regions of the search space. This flexibility allows

for comprehensive design space exploration, leading to a better understanding of the trade-offs involved. Optimizing an induction motor's reactance can enhance its overall efficiency. Metaheuristic algorithms can assist in identifying reactance values that minimize losses, reduce energy consumption, and improve the motor's overall performance. This perspective aligns with the growing emphasis on energy efficiency and sustainability in various industries. Also, metaheuristic algorithms offer a highly efficient alternative by leveraging stochastic search strategies, reducing the computational burden, and providing reasonably good solutions within a reasonable timeframe. By incorporating domain knowledge and problem-specific constraints, these algorithms can be tailored to address the unique challenges and objectives in motor design and control. Further research and experimentation in this area can help to refine and improve the application of metaheuristic solutions in induction motor reactance optimization, leading to more efficient and optimized motor designs.

Funding: This research received no external funding.

Data Availability Statement: Data are available upon request.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

ANN	Artificial neural networks
HBO	Heap-based optimization
LCA	Leagues championship algorithm
MVO	Multi-verse optimization
OOA	Osprey optimization algorithm
COA	Cuckoo optimization algorithm
STOA	Sooty tern optimization algorithm
MLP	Multi-layer perceptron
GA	Genetic algorithm
ABC	Artificial bee colony
AC	Alternating current
DC	Direct current
MAE	Mean average error
RMSE	Root mean square error
R^2	Coefficient of determination
WEP	Wormhole existence probability
TDR	Traveling distance rate
CRH	Corporate rank hierarchy
CF	Correlation factor
IM	Induction motor
X_m	Motor reactance
$P(KW)$	Rated power
$\cos(\rho_{FL})$	Full load power factor
T_m/T_{FL}	Maximum torque to total load torque
T_{ST}/T_{FL}	Initial torque to total load torque
I_{ST}/I_{FL}	Initial current to total load current
ω_{FL}	Angular velocity
η_{FL}	Full load efficiency

References

1. Nardo, M.D.; Marfoli, A.; Degano, A.; Gerada, C. Rotor slot design of squirrel cage induction motors with improved rated efficiency and starting capability. *IEEE Trans. Ind. Appl.* **2022**, *58*, 3383–3393. [[CrossRef](#)]
2. Lee, K.S.; Lee, S.H.; Park, J.H.; Kim, J.M.; Choi, J.Y. Experimental and analytical study of single-phase squirrel-cage induction motor considering end-ring porosity rate. *IEEE Trans. Magn.* **2017**, *53*, 1–4. [[CrossRef](#)]
3. Yang, M.; Wang, Y.; Xiao, X.; Li, Y. A Robust Damping Control for Virtual Synchronous Generators Based on Energy Reshaping. *IEEE Trans. Energy Convers.* **2023**, *38*, 2146–2159. [[CrossRef](#)]

4. Jirdehi, M.A.; Rezaei, A. Parameters estimation of squirrel-cage induction motors using ANN and ANFIS. *Alex. Eng. J.* **2016**, *55*, 357–368. [[CrossRef](#)]
5. Song, X.; Wang, H.; Ma, X.; Yuan, X.; Wu, X. Robust model predictive current control for a nine-phase open-end winding PMSM with high computational efficiency. *IEEE Trans. Power Electron.* **2023**, *38*, 13933–13943. [[CrossRef](#)]
6. Çetin, O.; Dalcalı, A.; Temurtaş, F. A comparative study on parameters estimation of squirrel cage induction motors using neural networks with unmemorized training. *Eng. Sci. Technol. Int. J.* **2020**, *23*, 1126–1133. [[CrossRef](#)]
7. Silva, A.M.; Alberto, J.; Antunes, C.H.; Ferreira, F.J.T.E. A Stochastic Optimization Approach to the Estimation of Squirrel-Cage Induction Motor Equivalent Circuit Parameters. In Proceedings of the 2020 International Conference on Electrical Machines (ICEM), Gothenburg, Sweden, 23–26 August 2020. [[CrossRef](#)]
8. Shen, Y.; Liu, D.; Liang, W.; Zhang, X. Current reconstruction of three-phase voltage source inverters considering current ripple. *IEEE Trans. Transp. Electrification.* **2022**, *9*, 1416–1427. [[CrossRef](#)]
9. Ocaik, C. A FEM-Based Comparative Study of the Effect of Rotor Bar Designs on the Performance of Squirrel Cage Induction Motors. *Energies* **2023**, *16*, 6047. [[CrossRef](#)]
10. Abunike, C.E.; Akuru, U.B.; Okoro, O.I.; Awah, C.C. Sizing, Modeling, and Performance Comparison of Squirrel-Cage Induction and Wound-Field Flux Switching Motors. *Mathematics* **2023**, *11*, 3596. [[CrossRef](#)]
11. Agah, G.R.; Rahideh, A.; Faradonbeh, V.Z.; Hedayati, K.S. Stator Winding Inter-Turn Short-Circuit Fault Modeling and Detection of Squirrel-Cage Induction Motors. *IEEE Trans. Transp. Electrification.* **2023**. [[CrossRef](#)]
12. Du, J.; Li, Y. Analysis on the Variation Laws of Electromagnetic Force Wave and Vibration Response of Squirrel-Cage Induction Motor under Rotor Eccentricity. *Electronics* **2023**, *12*, 1295. [[CrossRef](#)]
13. Pedra, J.; Sainz, L.; Córcoles, F. Study of aggregate models for squirrel-cage induction motors. *IEEE Trans. Power Syst.* **2005**, *20*, 1519–1527. [[CrossRef](#)]
14. Zhang, H.; Wu, H.; Jin, H.; Li, H. High-Dynamic and Low-Cost Sensorless Control Method of High-Speed Brushless DC Motor. *IEEE Trans. Ind. Inform.* **2023**, *19*, 5576–5584. [[CrossRef](#)]
15. Özsoy, M.; Kaplan, O.; Akar, M. FEM-based analysis of rotor cage material and slot geometry on double air gap axial flux induction motors. *Ain Shams Eng. J.* **2024**, *15*, 102393. [[CrossRef](#)]
16. Yang, X.; Wang, X.; Wang, S.; Wang, K.; Sial, M.B. Finite-time adaptive dynamic surface synchronization control for dual-motor servo systems with backlash and time-varying uncertainties. *ISA Trans.* **2023**, *137*, 248–262. [[CrossRef](#)] [[PubMed](#)]
17. Li, M.; Wang, T.; Chu, F.; Han, Q.; Qin, Z.; Zuo, M.J. Scaling-basis chirplet transform. *IEEE Trans. Ind. Electron.* **2021**, *68*, 8777–8788. [[CrossRef](#)]
18. Zheng, W.; Gong, G.; Tian, J.; Lu, S.; Wang, R.; Yin, Z.; Li, X.; Yin, L. Design of a Modified Transformer Architecture Based on Relative Position Coding. *Int. J. Comput. Intell. Syst.* **2023**, *16*, 168. [[CrossRef](#)]
19. Sun, Y.; Peng, Z.; Hu, J.; Ghosh, B.K. Event-triggered critic learning impedance control of lower limb exoskeleton robots in interactive environments. *Neurocomputing* **2024**, *564*, 126963. [[CrossRef](#)]
20. Miaofen, L.; Youmin, L.; Tianyang, W.; Fulei, C.; Zhike, P. Adaptive synchronous demodulation transform with application to analyzing multicomponent signals for machinery fault diagnostics. *Mech. Syst. Signal Process.* **2023**, *191*, 110208. [[CrossRef](#)]
21. Araoye, T.O.; Ashigwuike, E.C.; Adeyemi, A.C.; Egoigwe, S.V.; Ajah, N.G.; Eronu, E. Reduction and control of harmonic on three-phase squirrel cage induction motors with voltage source inverter (VSI) using ANN-grasshopper optimization shunt active filters (ANN-GOSAF). *Sci. Afr.* **2023**, *21*, e01785. [[CrossRef](#)]
22. Milykh, V.I. Numerical-field analysis of active and reactive winding parameters and mechanical characteristics of a squirrel-cage induction motor. *Electr. Eng. Electromech.* **2023**, *4*, 3–13. [[CrossRef](#)]
23. Kojooyan, J.H.; Monjo, L.; Córcoles, F.; Pedra, J. Using the instantaneous power of a free acceleration test for squirrel-cage motor parameters estimation. *IEEE Trans. Energy Convers.* **2015**, *30*, 974–982. [[CrossRef](#)]
24. Tseligorov, N.; Ozersky, A.I.; Chubikin, A.V.; Tseligorova, E.N. Development of a robust scalar control system for an induction squirrel-cage motor based on a linearized vector model. *WSEAS Trans. Comput.* **2022**, *21*, 1–9. [[CrossRef](#)]
25. Fortes, M.Z.; Ferreira, V.H.; Coelho, A.P.F. The induction motor parameter estimation using genetic algorithm. *IEEE Lat. Am. Trans.* **2013**, *11*, 1273–1278. [[CrossRef](#)]
26. Abro, A.G.; Saleh, J.M. Multiple-global-best guided artificial bee colony algorithm for induction motor parameter estimation. *Turk. J. Electr. Eng. Comput. Sci.* **2014**, *22*, 620–636. [[CrossRef](#)]
27. Gomez, G.M.; Jurado, F.; Pérez, I. Shuffled frog-leaping algorithm for parameter estimation of a double-cage asynchronous machine. *IET Electr. Power Appl.* **2012**, *6*, 484–490. [[CrossRef](#)]
28. Yan, Z.; Wen, H. Electricity theft detection base on extreme gradient boosting in AMI. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–9. [[CrossRef](#)]
29. Feng, H.; Cui, X.; Si, J.; Gao, C.; Hu, Y. Equivalent Circuit Model of Novel Solid Rotor Induction Motor with Toroidal Winding Applying Composite Multilayer Theory. *Appl. Sci.* **2019**, *9*, 3288. [[CrossRef](#)]
30. Ganesh, K.P.; Mary, A.D. Speed Estimation and Equivalent Circuit Parameter Determination of Induction Motor Using Virtual Instrumentation. In Proceedings of the 2016 International Conference on Next Generation Intelligent Systems (ICNGIS), Kottayam, India, 1–3 September 2016. [[CrossRef](#)]
31. Aryza, S.; Irwanto, M.; Lubis, Z.; Siahaan, A.P.U.; Rahim, R.; Furqan, M. A Novelty Design of Minimization of Electrical Losses in A Vector Controlled Induction Machine Drive. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *300*, 012067. [[CrossRef](#)]

32. Al-Jufout, S.A.; Al-Rousan, W.H.; Wang, C. Optimization of Induction Motor Equivalent Circuit Parameter Estimation Based on Manufacturer's Data. *Energies* **2018**, *11*, 1792. [CrossRef]
33. Mishra, R.N.; Mohanty, K.B. Real time implementation of an ANFIS-based induction motor drive via feedback linearization for performance enhancement. *Eng. Sci. Technol. Int. J.* **2016**, *19*, 1714–1730. [CrossRef]
34. Ding, Z.; Wu, X.; Chen, C.; Yuan, X. Magnetic Field Analysis of Surface-Mounted Permanent Magnet Motors Based on an Improved Conformal Mapping Method. *IEEE Trans. Ind. Appl.* **2023**, *59*, 1689–1698. [CrossRef]
35. Liu, S.; Liu, C. Direct harmonic current control scheme for dual three-phase PMSM drive system. *IEEE Trans. Power Electron.* **2021**, *36*, 11647–11657. [CrossRef]
36. Wang, H.; Sun, W.; Jiang, D.; Qu, R. A MTPA and flux-weakening curve identification method based on physics-informed network without calibration. *IEEE Trans. Power Electron.* **2023**, *38*, 12370–12375. [CrossRef]
37. Idir, K.; Chang, L.; Dai, H. A Neural Network-Based Optimization Approach for Induction Motor Design. In Proceedings of the 1996 Canadian Conference on Electrical and Computer Engineering, Calgary, AB, Canada, 26–29 May 1996. [CrossRef]
38. Im, D.H.; Park, S.C.; Park, D.J. Optimum Design of Single-Sided Linear Induction Motor Using the Neural Networks and Finite Element Method. In Proceedings of the 1993 International Conference on Neural Networks (IJCNN), Nagoya, Japan, 25–29 October 1993. [CrossRef]
39. Drabek, T. Derating of Squirrel-Cage Induction Motor Due to Rotating Harmonics in Power Voltage Supply. *Energies* **2023**, *16*, 735. [CrossRef]
40. Marfoli, A.; DiNardo, M.; Degano, M.; Gerada, C.; Jara, W. Squirrel cage induction motor: A design-based comparison between aluminium and copper cages. *IEEE Open J. Ind. Appl.* **2021**, *2*, 110–120. [CrossRef]
41. Chen, H.; Zhao, J.; Wang, H.; Zhang, Q.; Luo, X.; Xu, H.; Xiong, Y. Multi-objective optimum design of five-phase squirrel cage induction motor by differential evolution algorithm. *Energy Rep.* **2022**, *8*, 51–62. [CrossRef]
42. Kumar, P.; Hati, A.S. Dilated convolutional neural network based model for bearing faults and broken rotor bar detection in squirrel cage induction motors. *Expert Syst. Appl.* **2022**, *191*, 116290. [CrossRef]
43. Perin, M.; da Silveira, G.B.; Pereira, L.A.; Haffner, S.; Almansa, D.M.S. Estimation of Electrical Parameters of the Double-Cage Model of Induction Motors Using Manufacturer Data and Genetic Algorithm. In Proceedings of the IECON 2022—48th Annual Conference of the IEEE Industrial Electronics Society, Brussels, Belgium, 17–20 October 2022. [CrossRef]
44. Tulicki, J.; Sobczyk, T.J.; Sułowicz, M. Diagnostics of A Double-Cage Induction Motor Under Steady State with the Rotor Asymmetry. In Proceedings of the 2023 IEEE 14th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED), Chania, Greece, 28–31 August 2023. [CrossRef]
45. Karakaya, A. Modeling of Induction Motor and Speed Analysis of Modern Control Methods. *Karaelmas Sci. Eng. J.* **2017**, *7*, 497–502. Available online: <https://dergipark.org.tr/en/download/article-file/1329457> (accessed on 30 January 2024).
46. Monjo, L.; Kojooyan-Jafari, H.; Corcoles, F.; Pedra, J. Squirrel-cage induction motor parameter estimation using a variable frequency test. *IEEE Trans. Energy Convers.* **2014**, *30*, 550–557. [CrossRef]
47. Pedra, J.; Corcoles, F. Estimation of induction motor double-cage model parameters from manufacturer data. *IEEE Trans. Energy Convers.* **2004**, *19*, 310–317. [CrossRef]
48. Mo, J.; Yang, H. Sampled Value Attack Detection for Busbar Differential Protection Based on a Negative Selection Immune System. *J. Mod. Power Syst. Clean Energy* **2023**, *11*, 421–433. [CrossRef]
49. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 1998.
50. Zhou, X.; Liu, X.; Zhang, G.; Jia, L.; Wang, X.; Zhao, Z. An iterative threshold algorithm of log-sum regularization for sparse problem. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 4728–4740. [CrossRef]
51. Moayedi, H.; Ghareh, S.; Foong, L.K. Quick integrative optimizers for minimizing the error of neural computing in pan evaporation modeling. *Eng. Comput.* **2022**, *38*, 1331–1347. [CrossRef]
52. Zhou, G.; Moayedi, H.; Bahiraei, M.; Lyu, Z. Employing artificial bee colony and particle swarm techniques for optimizing a neural network in prediction of heating and cooling loads of residential buildings. *J. Clean. Prod.* **2020**, *254*, 120082. [CrossRef]
53. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Applic.* **2016**, *27*, 495–513. [CrossRef]
54. Yang, X.S.; Deb, S. Cuckoo Search via Lévy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009. [CrossRef]
55. Yang, X.S. *Nature-Inspired Metaheuristic Algorithms*, 2nd ed.; Luniver Press: Frome, UK, 2010.
56. Askari, Q.; Saeed, M.; Younas, I. Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Syst. Appl.* **2020**, *161*, 113702. [CrossRef]
57. Kashan, A.H. League Championship Algorithm: A New Algorithm for Numerical Function Optimization. In Proceedings of the 2009 International Conference of Soft Computing and Pattern Recognition, Malacca, Malaysia, 4–7 December 2009. [CrossRef]
58. Dehghani, M.; Trojovský, P. Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems. *Front. Mech. Eng* **2023**, *8*, 1126450. [CrossRef]
59. Dhiman, G.; Kaur, A. STOA: A bio-inspired based optimization algorithm for industrial engineering problems. *Eng. Appl. Artif. Intell.* **2019**, *82*, 148–174. [CrossRef]
60. Tamura, K.; Yasuda, K. The spiral optimization algorithm: Convergence conditions and settings. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 360–375. [CrossRef]

61. Taylor, K.E. Summarizing multiple aspects of model performance in a single diagram. *J. Geophys. Res. Atmos.* **2001**, *106*, 7183–7192. [[CrossRef](#)]
62. Solomon, S. *Climate Change 2007: The Physical Science Basis*; Cambridge University Press: New York, NY, USA, 2007.
63. Gleckler, P.J.; Taylor, K.E.; Doutriaux, C. Performance metrics for climate models. *J. Geophys. Res. Atmos.* **2008**, *113*, D06104. [[CrossRef](#)]

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