electronics

MDPI

Article

An Improved GWO Algorithm Optimized RVFL Model for Oil Layer Prediction

Pu Lan, Kewen Xia *, Yongke Pan and Shurui Fan *

School of Electronics and Information Engineering, Hebei University of Technology, Tianjin 300401, China; lanpu022@163.com (P.L.); panyongke@hotmail.com (Y.P.)

* Correspondence: kwxia@hebut.edu.cn (K.X.); fansr@hebut.edu.cn (S.F.)

Abstract: In this study, a model based on the improved grey wolf optimizer (GWO) for optimizing RVFL is proposed to enable the problem of poor accuracy of Oil layer prediction due to the randomness of the parameters present in the random vector function link (RVFL) model to be addressed. Firstly, GWO is improved based on the advantages of chaos theory and the marine predator algorithm (MPA) to overcome the problem of low convergence accuracy in the optimization process of the GWO optimization algorithm. The improved GWO algorithm was then used to optimize the input weights and implicit layer biases of the RVFL network model so that the problem of inaccurate and unstable classification of RVFL due to the randomness of the parameters was avoided. MPA-GWO was used for comparison with algorithms of the same type under a function of 15 standard tests. From the results, it was concluded that it outperformed the algorithms of its type in terms of search accuracy and search speed. At the same time, the MPA-GWO-RVFL model was applied to the field of Oil layer prediction. From the comparison tests, it is concluded that the prediction accuracy of the MPA-GWO-RVFL model is on average 2.9%, 3.04%, 2.27%, 8.74%, 1.47% and 10.41% better than that of the MPA-RVFL, GWO-RVFL, PSO-RVFL, WOA-RVFL, GWFOA-RVFL and RVFL algorithms, respectively, and its practical applications are significant.

Keywords: grey wolf optimizer; marine predators algorithm; random vector functional link network; oil layer prediction

Citation: Lan, P.; Xia, K.; Pan, Y.; Fan, S. An Improved GWO Algorithm Optimized RVFL Model for Oil Layer Prediction. *Electronics* **2021**, *10*, 3178. https://doi.org/10.3390/electronics10243178

Academic Editor: Olivier Sename

Received: 9 November 2021
Accepted: 17 December 2021
Published: 20 December 2021

1. Introduction

Petroleum exploration is a complex system that involves a wide range of activities and high-risk values. In logging technology, the approach to petroleum reservoir delineation usually involves the following. Firstly, the raw logging data is used for the correction of non-stratigraphic factors such as environmental influences and for the standardization of the logging data. The models and methods provided by the logs are then used to perform calculations of porosity, permeability and oil saturation (or water saturation). Finally, reservoir categories are determined based on upper and lower criteria for the electrical properties of these reservoir parameters and other logging indicators. Although this method has some advantages, an accurate mathematical model needs to be developed. At the same time, it is subject to strict conditions. Such a structured approach to semi-structured and unstructured problems makes it difficult to achieve satisfactory results. From a mathematical point of view, it follows that the logging interpretation problem actually refers to the solution of a mapping problem. When the hypothetical model matches the actual distribution of the samples, high prediction accuracy is obtained. Nassan et al. [1] simulated the two-phase immiscible flow of water and heavy oil on the famous inverted five-point model. The comparison between the simulation and the actual results showed that the model was in good agreement. Roman et al. [2] analyzed and assessed the mining-geological and mining-technical conditions of the open pit "Severny" using mathematical models. The technical and economic indexes of open-pit mining were effectively improved.

Sun et al. [3] established the mathematical model of the variation of dynamic liquid level height with time in intermittent shutdown periods and intermittent pumping periods and proved that the model has better oil production efficiency and higher economic benefits.

Neural networks have the ability to approximate arbitrary non-linear mappings by learning. When they are applied to pattern recognition and prediction, they are not limited by the constraints of the non-linear mapping model. Therefore, neural networks are necessary to solve the Oil layer prediction problem.

The adoption of an appropriate neural network model becomes critical to the accuracy of the prediction. Chen et al. [4] developed a method to invert the logging tool signals from formation parameters by artificial neural networks, which provided a reliable basis for Oil layer prediction. Pan et al. [5] used an improved BP (backpropagation) neural network for the dynamic prediction of oil reservoir parameters and the better results that they achieve. Osman et al. [6] developed an ANN (artificial neural network) model based on the fluid properties of petroleum reservoirs, which accurately predicted the formation volume factor. However, it has been found over the past decades that the inappropriate learning step size of BP neural networks leads to a very slow convergence rate of the algorithm, which tends to result in local minima. As a result, a large number of iterations are often required to obtain a more satisfactory accuracy. This problem has been the main bottleneck limiting its development in application areas for some time. Later, Huang et al. [7] proposed a simple and efficient single hidden layer feedforward neural network (SLFN) algorithm, which is also known as an extreme learning machine (ELM). ELM randomly selects the input weights and hidden layer bias of the network and computes the output weights by parsing. Its learning speed is extremely fast, and it effectively overcomes the shortcomings of traditional BP neural networks.

The random vector function link (RVFL) network [8] is well known as a very effective and fast prediction model. Tang et al. [9] proposed an ensemble empirical mode decomposition (EEMD) technique for the RVFL model to improve prediction accuracy. Bisoi et al. [10] combined variational mode decomposition (VMD) with an RVFL neural network model to improve both the running time and prediction accuracy of the code. Yu et al. [11] synthesized the impact of five different strategies in the predictive performance of RVFL neural network models from the perspective of the diversity of integration strategies. The results show that the prediction accuracy of the integrated RVFL model with the combination of multiple strategies is significantly improved compared to that of the single RVFL model. Chai et al. [12] imported measurements obtained from the Hong Kong Observatory into the RVFL model to form prediction intervals for solar irradiance time series, and their results proved to be very effective in terms of reliability and clarity, and to be of significant help in the generation of solar energy. Hailiang et al. [13] performed feature extraction on face data and fed the feature set into the RVFL model for recognition, which led to a large improvement in accuracy, sparsity and stability. Zhou [14] proposed a regularized random vector function linkage (RRVFL) chromophore light estimation algorithm, which improved the prediction accuracy of RVFL, but still did not fundamentally solve the parameter randomness of RVFL caused by the algorithm instability problem.

In recent years, computer algorithm researchers have been inspired by the population intelligence of natural organisms and proposed population intelligence algorithms [15], which are widely used in many fields such as signal processing [16], image processing [17], production scheduling [18], pattern recognition [19], automatic control [20] and mechanical design [21]. Successive population intelligence algorithms include genetic algorithms (GA) that mimic the evolutionary mechanisms of organisms in nature [22], differential evolution algorithms (DE) that optimize search through cooperation and competition among individuals within a population [23], immune algorithms (IA) that simulate the learning and cognitive functions of the biological immune system [24], ant colony algorithms (AS) that simulate the collective path-finding behavior of ants [25], particle swarm algorithms (PSO) that simulate the behavior of flocks of birds (PSO) [26], the simulated annealing algorithm (SAA) [27], taboo search algorithms (TS) that simulate the human intellectual

memory process [28], etc. They are mainly used to solve the problem of getting into local optimal solutions, and their principles are simple and have good results in solving non-linear problems.

The MPA [29] and GWO algorithms [30] are used in this study to simulate predator hunting behavior, and both algorithms have the advantage of high search capability, as demonstrated by multiple sets of experiments using multiple functions in the literature. Because they show significant advantages in solving optimization problems, they are widely used to solve continuous and engineering optimization problems. Sharma et al. [31] applied the GWO algorithm to the problem of minimizing the operating cost under solar cell constraints and demonstrated better performance. Barman et al. [32] combined GWO with SVM to provide a solution for forecasting the electric load demand. Zhou et al. [33] proposed A hybrid grey wolf optimizer optimized ELM model that effectively reduces jitter in the sliding mode control of robotic manipulators. Abdel et al. [34] proposed a hybrid COVID-19 detection model based on the improved marine predator algorithm (IMPA) for X-ray image segmentation for effective detection and diagnosis of viral infections. Bayoumi et al. [35] applied the marine predator algorithm (MPA) to extract the parameters of the solar cell tri-photovoltaic model to improve the accuracy of the estimated values. Chen et al. [36] proposed a support vector machine (MPA-SVM) technique based on the MPA for rolling bearing fault diagnosis, and the results demonstrated the effectiveness of the fault diagnosis method, which was able to diagnose bearing faults with 100% accuracy. Aly et al. [37] used MPA to optimize the maximum power point tracking (MPPT) model for fuel cells (FC) to achieve the lowest output power fluctuation with fast tracking speed. Fan et al. [38] proposed a logical opposition-based learning (LOBL) mechanism to improve the MPA model by adding new position update rules, inertia weight coefficients and non-linear step control parameters to achieve the strategy of improving the MPA performance in terms of accuracy, convergence speed and stability. Hoang et al. [39] used MPA to identify a set of suitable SVM hyperparameters (including penalty coefficients and kernel function parameters) to optimize the SVM training phase and applied the improved SVM model to satellite remote sensing data for the purpose of identifying the current state of urban green spaces. Liu et al. [40] used a sine and cosine algorithm with the marine predator algorithm for random initialization population screening, and the optimized model was used for color constancy assessment of dyed fabrics to achieve the best assessment results. Houssein et al. [41] proposed a hybrid model based on the marine predator algorithm (MPA) and convolutional neural network (CNN): MPA-CNN for the ECG-type identification prediction problem, which showed better computational time and accuracy in performance.

Inspired by the above literature, we develop an improved GWO algorithm optimized RVFL model for Oil layer prediction. The main contributions are as follows.

(1) In the population intelligence optimization algorithm, because the quality of the initial population affects the global convergence speed and the final solution quality of the swarm optimization algorithm, a chaotic initialization strategy is introduced to make the beginning distribution of the wolf population more random, the MPA algorithm is used to optimize the GWO and the MPA-GWO algorithm is obtained, which is experimentally compared with the current popular population intelligence algorithm to verify its convergence speed and convergence. The superiority in convergence speed and accuracy is verified.
(2) Since the input weights and deviations in RVFL are set randomly, the prediction accuracy of the model is not high enough, and the stability is poor. In order to overcome this problem, the MPA-GWO algorithm is used to optimize the RVFL to get the optimal input weights and hidden layer deviations.
(3) For the first time, the optimized RVFL model is applied to Oil layer prediction, and the convergence and stability of the algorithm are examined by convergence curves and box plots, and the superiority of the MPA-GWO-RVFL model is demonstrated in comparison with the same type of model.

Part 2 of this paper introduces the MPA, GWO and RVFL algorithms. Part 3 uses the MPA algorithm to improve the GWO algorithm and conducts experimental comparison analysis. Part 4 describes the operation flow and specific process of the MPA-GWO-RVFL model, illustrates the method of Oil layer prediction and verifies the improved prediction model in terms of prediction accuracy, stability and convergence performance by analyzing data processing, algorithm parameter setting and experimental results. Section 5 is the conclusion of the paper.

2. Preparatory Knowledge

2.1. Random Vector Functional Link Network RVFL Algorithm

The backpropagation algorithm in ANN has the disadvantages of slow convergence and long learning time. In contrast, the RVFL neural network randomly assigns input weights and biases, uses least squares to train the output weights, does not perform the connection of processing units in the same layer and the feedback connection between different layers, which can make up for the ANN defects [42], has a good non-linear fitting ability. Ren et al. [43] compared the application of RVFL with an ordinary artificial neural network (ANN) in the field of wind power and found that RVFL has better performance. Zhang et al. [44] proved that RVFL is superior to ELM through experiments and comparison of 16 different benchmarks from different fields. Peng et al. [45] applied RVFL and ELM to the emotion recognition task of EEG and found that RVFL was superior to ELM in performance, while both showed excellent performance. The RVFL model structure is shown in Figure 1.

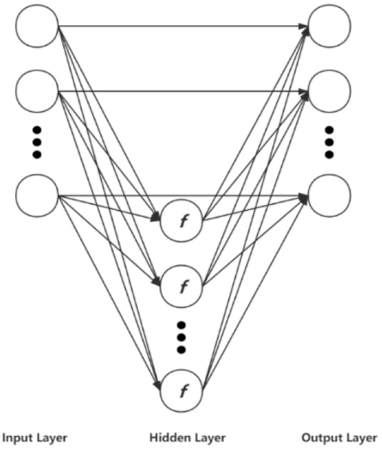


Figure 1. Structure of RVFL neural network.

In the following, each layer of the RVFL model is interpreted.

(1) Input layer

The main role of the input layer is to input a training set $\{(x^n, y^n)\}$ with U training samples $u = 1, 2, \ldots, U$; x is an n dimensional input variable, $x \in R^n$; y is the desired output variable, $y \in R^U$. The analysis in this paper yields a training sample space of $\{(x^\tau, y^\tau)\}_{\tau=1}^{U}$, x^τ is the 5-dimensional input variable at τ time, $x^\tau = (T_{a_i}^\tau, T_{e_i}^\tau, T_{w_i}^\tau, T_{a_{i,in}}^\tau, T_{a_{i,out}}^\tau)$; y^τ is the output variable at τ time, $y^\tau = T_{e_i}^{\tau+1} - T_{e_i}^\tau$.

(2) Hidden layer

The implicit layer can establish the activation function h value of the output of each implicit layer node, which is obtained in this paper by the *sigmoid* function h, which serves to transform the input variables linearly and can be expressed as bellow.

$$h(x, w, b) = \frac{1}{1 + \exp\{-w^T x + b\}^*} \quad (1)$$

where w and b are the weights and biases from the input layer to the hidden layer, respectively, independent of the training data and are determined before the learning process begins. Ultimately, the implicit layer kernel mapping matrix H is calculated as follows for the output layer component.

$$H = \begin{bmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \cdots & \vdots \\ h_1(x_U) & \cdots & h_L(x_U) \end{bmatrix} \tag{2}$$

where L denotes the number of nodes in the hidden layer.

(3) Output layer

Calculating the weights from the hidden layer to the output layer β is a central part of the learning process of the RVFL neural network and according to the standard regularized least-squares principle to find β.

$$\beta^* = \underset{\beta \in R}{\operatorname{argmin}} \frac{1}{2} \|H\beta - Y\|_2^2 + \frac{\lambda}{2} \|\beta\|_2^2 \; \lambda > 0 \tag{3}$$

where Y is a column vector y^u consisting of the training sample space corresponding to x^u; λ denotes a constant. The final weights can be obtained β denoted as bellow.

$$\beta = \left(H^T H + \lambda I\right)^{-1} H^T Y \tag{4}$$

where I denotes the unit matrix. At this point, the learning process is complete, and the test output of the RVFL model is obtained, denoted as below.

$$\hat{y} = \sum_{l=1}^{L} \beta_l h(x, \omega_l, b_l) \tag{5}$$

Although the RVFL model has a fast convergence speed and short learning time, its input weights and hidden layer bias are randomly determined, which largely affects its performance. Therefore, this paper uses an intelligent optimization algorithm to filter out the best parameter values after iteration so as to ensure the accuracy and stability of RVFL applications.

2.2. Grey Wolf Optimizer GWO

The basic idea of the GWO is to simulate the predatory behavior of a grey wolf pack by finding a better location from the prey and moving in the direction closest to the prey to achieve goal optimization. Grey wolves live mainly in packs, and usually, a grey wolf pack contains 5 to 12 grey wolves. In a small pack, there is always only one top wolf, and they are responsible for all decision-making matters of the entire pack, including hunting, migration and foraging. The other, lower-level grey wolves are divided into three classes, beyond which are the pups. In order to describe the hierarchy of grey wolves more scientifically, the pack is divided into four levels from the highest to the lowest: α, β, δ and θ, the first level is the wolf α, which is also known as the "head wolf"; β is the next level of α wolf, which acts as the α wolf's "assistant"; the common wolf δ is in the third level; and the lowest level is called θ, representing the pack, which is responsible for following the above three levels of wolves. Grey wolf optimizer is divided into three main steps: hunting, rounding up and attacking.

2.2.1. Hunting Process

An important criterion for the grey wolf searching for prey is the distance between them and the prey. The position of the grey wolf during the first t iteration of the search is

set to $X(t)$, and the position of the prey is $X_P(t)$, then the distance between the grey wolf and the prey D can be expressed as:

$$\begin{cases} D = \left| C \cdot X_p(t) - X(t) \right| \\ C = 2r_2 \\ r_2 = rand(0,1) \end{cases} \tag{6}$$

where C denotes the random weight, which is a random number in $[0,2]$, and its randomness helps the algorithm jump out when it falls into a local optimum, providing an important role in avoiding local optima.

2.2.2. Roundup Process

In the process of grey wolf encircling prey, the relationship between grey wolf and prey can be modeled by different step lengths and distances to achieve the purpose of encircling prey. The formula is as follows.

$$\begin{cases} X_i^d(t+1) = X_P^d(t) - A_i^d \cdot D_i^d \\ D_i^d = \left| C_i^d \cdot X_p^d(t) - X_i^d(t) \right| \\ A_i^d = 2ar_1 - a \\ C_i^d = 2r_2 \\ a = 2 - t/t_{\max} \\ r_1 \cdot r_2 = rand(0,1) \end{cases} \tag{7}$$

where $A_i^d \cdot D_i^d$ denotes the enclosing step; $t_{\max}$ denotes the maximum number of iterations; t denotes the current number of iterations; parameter a denotes the convergence factor, its value decreases linearly from 2 to 0 during the exploration process of the grey wolf. Random initialization of A_i^d and C_i^d ensures that the grey wolf can easily reach the global optimal position during the exploration process.

2.2.3. Attack Process

By updating the location information through the α, β, δ wolves, it is able to accurately determine the location of the target prey and achieve an attack on it. The specific mathematical relationship equation is as follows.

$$\begin{cases} X_1 = X_\alpha - A_1 \cdot D_\alpha \\ X_2 = X_\beta - A_2 \cdot D_\beta \\ X_3 = X_\delta - A_3 \cdot D_\delta \\ X = (X_1 + X_2 + X_3)/3 \end{cases} \tag{8}$$

where X_1, X_2, X_3 denote the positions of the α, β, δ wolves; A_1, A_2, A_3 denote the three random numbers; $A_1 \cdot D_\alpha$, $A_2 \cdot D_\beta$, $A_3 \cdot D_\delta$ denote the prey encirclement steps of α, β, δ; and X denote the final position of the θ wolf pack in the prey attack. The above equations all use vectors and therefore apply to arbitrary dimensions.

GWO has the following advantages when compared with other optimization algorithms: (1) It has faster convergence speed and stronger local search capability. (2) It has lower space complexity. (3) The principle of GWO is simple, with few parameters, easy to operate and implement. However, GWO has the shortcoming of insufficient global searchability. In a GWO calculation, the initialized three groups of head wolves are replaced by individuals with better adaptation values in the iteration; that is, if all three fall into a local optimum, the whole population cannot seek a better solution at this time. This can be understood as follows: when the decision maker of the pack misjudges the location of the prey emergence, then all the encirclement actions of the grey wolves will be ineffective. Furthermore, experience shows that it may still face the difficulty of falling into a local optimum in the face of highly complex functions. For this reason, the MPA algorithm is used below to improve it and enhance the optimality finding ability.

2.3. MPA Algorithm

2.3.1. Population Location Initialization Phase

Similar to most metaheuristic algorithms, MPA randomly initializes prey locations within the search space to initiate the optimization process. The mathematical description is as follows.

$$X_{ij} = lb + rand(ub - lb)\ i = 0 \ldots n, j = 0 \ldots d \tag{9}$$

where X_{ij} denotes the j dimensional coordinates of the first i population, n is the size of the population, d is the dimension, i.e., the dimension of the solution. ub and lb are the upper and lower boundaries of the search space and $rand$ is a random number between [0, 1].

2.3.2. Optimization Phase

At the beginning of the iteration, i.e., $Iter < \frac{1}{3}Max\_Iter$, this phase is mainly used for the global search of the solution space, which is mathematically described as follows.

$$\begin{cases} \vec{stepsize_i} = \vec{R}_B \otimes (\vec{Elite_i} - \vec{R}_B \otimes \vec{Prey_i})\ i = 1, \ldots n \\ \vec{Prey_i} = \vec{Prey_i} + P \cdot \vec{R} \otimes \vec{stepsize_i} \end{cases} \tag{10}$$

where $\vec{stepsize_i}$ is the step size of this stage; R_B is a vector of random numbers generated by the normal distribution of Brownian motion; $\otimes$ represents the term-by-term multiplicative operator; P is the step control factor, a constant 0.5; R is a random uniformly distributed value within $[0, 1]$; $Iter$ is the current iteration number; Max_Iter is the maximum iteration number.

In the middle of the iteration, i.e., $\frac{1}{3}Max\_Iter < Iter < \frac{2}{3}Max\_Iter$, this phase transitions from a global search of the solution space to a local search of the current optimal solution position in the solution space. The position is updated by the following equation.

$$\begin{cases} \vec{stepsize_i} = \vec{R}_L \otimes (\vec{Elite_i} - \vec{R}_L \otimes \vec{Prey_i})\ i = 1, \ldots n/2 \\ \vec{Prey_i} = \vec{Prey_i} + P \cdot \vec{R} \otimes \vec{stepsize_i} \end{cases} \tag{11}$$

$$\begin{cases} \vec{stepsize_i} = \vec{R}_B \otimes (\vec{Elite_i} - \vec{R}_B \otimes \vec{Prey_i})\ i = n/2, \ldots n \\ \vec{Prey_i} = \vec{Prey_i} + P \cdot CF \otimes \vec{stepsize_i} \end{cases} \tag{12}$$

$$where\ CF = (1 - \frac{Iter}{Max\_Iter})^{(2\frac{Iter}{Max\_Iter})} \tag{13}$$

where R_L is a vector of random numbers generated during the Lévy flight phase; CF is an adaptive parameter used to control the predator's movement stride.

At the end of the iteration, i.e., $Iter > \frac{2}{3}Max\_Iter$, this phase focuses on a local search for the location of the current optimal solution in the solution space. It is mathematically described as follows.

$$\begin{cases} \vec{stepsize_i} = \vec{R}_L \otimes (\vec{R}_L \otimes \vec{Elite_i} - \vec{Prey_i})\ i = 1, \ldots n \\ \vec{Prey_i} = \vec{Elite_i} + P \cdot CF \otimes stepsize_i \end{cases} \tag{14}$$

2.3.3. Vortex Formation and Fish Aggregation Device Effects (FADS)

Fish aggregation devices (FADs) or eddy effects typically alter the foraging behavior of marine predators, a strategy that enables MPAs to overcome the early convergence problem and escape local extremes during the search for an optimal value. It is mathematically described as follows.

$$\vec{Prey_i} = \begin{cases} \vec{Prey_i} + CF[\vec{X}_{\min} + \vec{R} \otimes (\vec{X}_{\max} - \vec{X}_{\min})] \otimes \vec{U}\ ifr \le FADS \\ \vec{Prey_i} + [FADs(1-r) + r](\vec{Prey_{r1}} - \vec{Prey_{r2}})\ ifr > FADS \end{cases} \tag{15}$$

where FADS denotes the probability of influencing the search process, which is usually set to 0.2, $\vec{X}_{\max}$ is the vector consisting of the maximum value of the searched boundary, $\vec{X}_{\min}$ is the vector consisting of the minimum value of the searched boundary and the subscripts $r1$ and $r2$ denote the random indices consisting of the predator matrices, which are binary vectors consisting of 0 and 1.

3. GWO Improved by MPA

3.1. GWO Improved by MPA

The problems with the GWO optimization algorithm are as below.

(1) The initial population individuals of the basic GWO algorithm are generated randomly before the population evolves iteratively, which may lead to a poor diversity of the population.
(2) From Equations (2)–(8), we can see that ω wolves update their positions under the leadership of α wolves, β wolves and δ wolves, and when they are all in the local optimum, each wolf in the pack may tend to be in the local optimum due to the influence of these three wolves, and the global searchability is insufficient.

In this paper, to solve the above problems, the following improvements are made to the GWO algorithm.

(1) Introduce a chaos strategy when initializing the population so that individuals are distributed as evenly as possible in the search space.
(2) To further increase the merit-seeking capability, MPA is used to find three optimal wolves α, β, δ to enhance the global exploration capability of the GWO algorithm.

3.2. Population Initialization Based on Chaos Theory

Chaos is a universal phenomenon, which is a highly unstable and unpredictable motion of a deterministic system in a finite phase space with characteristics such as periodicity, randomness and regularity. In this paper, we will fully extract and capture the information in the solution space through chaotic mapping. One of the widely used mapping mechanisms in the study of chaos theory is *logistic* mapping; its mathematical iterative equation is:

$$\lambda_{t+1} = \mu \times \lambda_t(1 - \lambda_t), t = 0, 1, 2, \cdots, T \tag{16}$$

where λ_t is a uniformly distributed random number on the interval [0, 1] and $\lambda_0 \notin \{0, 0.25, 0.5, 0.75, 1\}T$ is required to be the predetermined maximum number of chaotic iterations. μ is required to $T \geq D$ be the chaotic control parameter in this algorithm. When $\mu = 4$, the system will be in a fully chaotic state.

We use the chaotic variables generated λ by Equation (16), randomly selected data among the elite predators for chaos processing, mapped to the chaotic interval $[f_{\min}, f_{\max}]$ according to Equation (9). The chaos treatment allows for finding more random solutions while maintaining the optimal information of the optimal solution. The expression is:

$$X_i^j = f_{\min} + \lambda_j \times (f_{\max} - f_{\min})\ i = 0 \ldots n, j = 0 \ldots d \tag{17}$$

where X_i^j is the coordinate of the j-th dimension of the i-th search agent and λ_j is the coordinate of the j-th dimension λ after internal random ordering.

Chaotic sequences are used to initialize each subpopulation so that the initialized individuals can be uniformly distributed in the search space to improve the diversity of the population.

3.3. MPA-GWO Algorithm Flow

The main process of the MPA-GWO algorithm is as follows.

(1) Parameter initialization. Initialization settings were made for the population N_M in the MPA algorithm, maximum number of *Iter_Max* iterations, fish aggregation

device effect coefficients FADs, initialization P iterations, number of initialization iterations *iter*, and population size N_G in the GWO algorithm, total number of iterations $Iter\_Max$, constants ε, initialization a, C, A.

(2) Random population generation using chaotic strategies in the search space of the problem to be solved.
(3) Calculate the prey location and construct the prey matrix.
(4) Calculate the fitness values, search for the best in the population, and construct the elite matrix.
(5) Computational updates of prey locations from the beginning, middle and end of the iteration; updates of prey locations based on FADS; and completion of memory storage and updates of elite locations based on prey locations.
(6) Whether the current number of iterations *iter* is equal to the maximum number of iterations *Iter_Max*, if satisfied, the three elite matrices are output, and the optimal solution is assigned to $wolf - a$, the 2nd ranked optimal solution is given to $wolf - b$ and the third optimal path is given to $wolf - c$. Position X_1, X_2, X_3 of the grey wolf α, β, δ being composed.
(7) Calculate the fitness value of each individual $\{f(X_I), i = 1 \ldots N\}$ and rank them, and record the top three individuals in terms of fitness value as α, β and δ respectively, and record their positions as X_α, X_β and X_δ.
(8) Update the location of individual grey wolves to find the optimal solution.
(9) The maximum number of GWO iterations is reached, and the optimal result is saved and output.

The MPA-GWO algorithm flow is shown in Figure 2.

3.4. Experimental Comparison Numerical Optimization Experiments

In this study, five other optimization algorithms are set up for comparison experiments with the proposed MPA-GWO-RVFL algorithm, including MPA-GWO, WOA, MPA, GWO, PSO and GWFOA [46]. The experimental environment is a PC with the following configuration: Windows 10 64-bit, Intel Core I5-3210M2.50 GHz, 8 G RAM, MATLAB R2012a.

3.4.1. Experimental Settings and Algorithm Parameters

Fifteen standard test functions were selected to test the algorithms to verify their performance. They were shown in Table 1. Since the results of each algorithm are randomized during the run, in order to obtain a fair comparison, each comparison algorithm will be run 30 times independently across all test functions with a population size of 50 and a maximum number of iterations set to 1000, and then the resulting data will be averaged and standard deviated.

3.4.2. Benchmark Test Functions

The parameter settings of these 15 standard test functions have been widely used in verifying the validity of metaheuristics. It is known that it is difficult for an algorithm to fit all the test functions. Therefore, the 15 test functions are selected with diversity so that the experimental results obtained can reflect the algorithm's merit-seeking ability more objectively and comprehensively.

Five high-dimensional single-peaked test functions ($F_1(x)$~$F_5(x)$), which have only one global optimum but no local optimum, can be used to test the local searchability and convergence speed of the algorithm. Three high-dimensional multi-peaked test functions ($F_6(x)$~$F_8(x)$) have multiple local optima, in contrast to high-dimensional single-peaked functions, which makes it more difficult for the algorithm to solve high-dimensional multi-peaked test functions than to solve single-peaked functions. Therefore, such a function can be used to test the detection ability of the algorithm, i.e., global searchability. The fixed-dimensional multimodal function ($F_9(x)$~$F_{15}(x)$) has multiple local optima like the high-dimensional multimodal test function but differs in that it has a lower number of dimensions than the high-dimensional multimodal function, and therefore, a relatively

smaller number of local optima. Then, like the high-dimensional multimodal test function, the fixed-dimensional multimodal test function can also be used to test the global search performance of the algorithm. The table *Dim* refers to the dimensionality of the standard test function, $f_{\min}$ refers to the theoretical optimum of the standard test function and Range is the range of the search space.

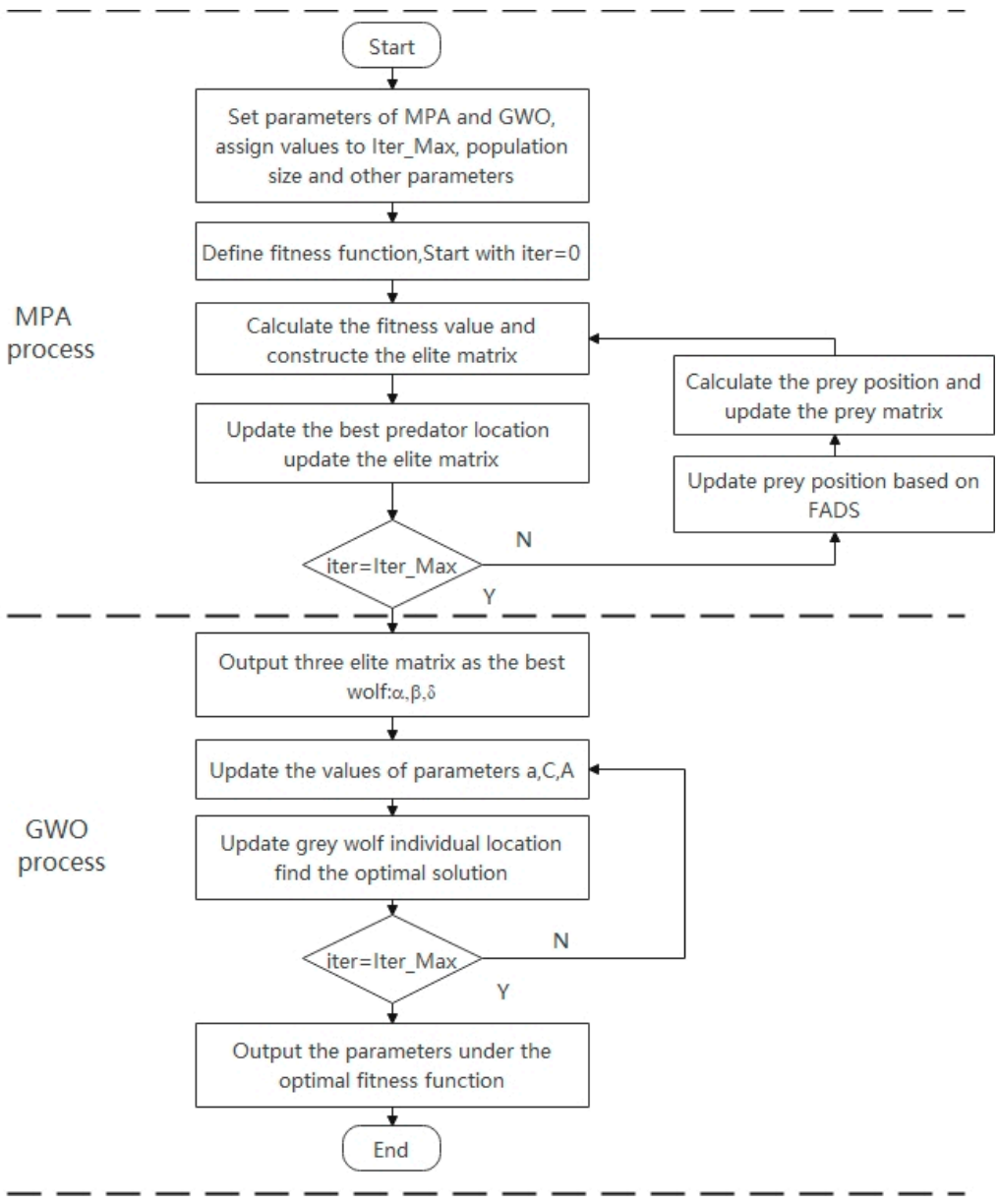


Figure 2. MPA improvement GWO flow chart.

3.4.3. Experimental Results and Analyses

In this subsection, comparative numerical optimization experiments are performed for MPA-GWO, WOA, MPA, GWO, PSO and GWFOA. The convergence curves are shown in Figure A1.

Table 1. Description of unimodal benchmark functions.

Function	Dim	Range	$f_{\min}$
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	30	$[-100, 100]$	0
$F_4(x) = \max_i \{ \lvert x_i \rvert, \quad 1 \le i \le n \}$	30	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]$	0
$F_6(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	30	$[-5.12, 5.12]$	0
$F_7(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	30	$[-32, 32]$	0
$F_8(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	30	$[-600, 600]$	-418.9829
$F_9(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right)^{-1}$	2	$[-65, 65]$	1
$F_{10}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	$[-5, 5]$	0.00030
$F_{11}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
$F_{12}(x) = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2) \right] \times \left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2) \right]$	2	$[-2, 2]$	3
$F_{13}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij} \left(x_j - p_{ij} \right)^2 \right)$	3	[1, 3]	-3.86
$F_{14}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij} \left(x_j - p_{ij} \right)^2 \right)$	6	[0, 1]	-3.32
$F_{15}(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	-10.5363

From Figure A1, it can be seen that on $F_1(x)$~$F_5(x)$, MPA-GWO converges significantly faster than other algorithms as well as has better convergence accuracy. On the $F_5(x)$ upper, all functions converge more slowly, and MPA-GWO converges to the best value, which reflects that MPA-GWO is not easily trapped in a local optimum overall. In the test of high-dimensional single-peaked functions, the MPA-GWO algorithm shows its superior and stable performance, which indicates that it is effective and feasible in solving high-dimensional space problems. The MPA-GWO convergence speed on $F_6(x)$~$F_8(x)$ is the fastest, and there is also better convergence accuracy, which shows that MPA-GWO is better than other algorithms in jumping out of the local optimum and finding better solutions. The overall reflects that the MPA-GWO algorithm is significantly superior and stable in performance than other algorithms in solving Kor-dimensional multi-peaked test functions, which verifies its strong global search capability. From $F_9(x)$~$F_{15}(x)$, on the first five functions, MPA-GWO reflects the fastest convergence speed and higher convergence accuracy. Although MPA-GWO on $F_{15}(x)$ converges slowly, it jumps out of the local optimum at a later stage and finds the optimal value closer to theory. In general, on the fixed-dimensional multimodal functions, MPA-GWO's ability to jump out of the local optimum is better than other algorithms, and the stability and effectiveness of the algorithm are verified.

The test results for the single-peak function, the high-dimensional multi-peak function and the fixed-dimensional multi-peak function are listed in Table A1. *Ave* and *Std* denote the average solution over 30 independent experiments and the standard deviation of the results over 30 runs, respectively. Due to the stochastic nature of the swarm intelligence algorithm, such statistical experiments are necessary to ensure the validity of the data.

For these five high-dimensional single-peak test functions $F_1(x)$~$F_5(x)$, it can be seen from Table A1 that the MPA-GWO algorithm exhibits superior performance over other algorithms on these five test functions. In these five test functions, it is clearly shown that MPA-GWO obtains better results than other algorithms in both mean and standard deviation, and the convergence accuracy is substantially improved compared with other algorithms. More noteworthy, in four of the functions tested, IEO takes the ideal optimal value of 0 every time. The variance of the MPA-GWO algorithm is much smaller than other algorithms, which fully illustrates the stability of the MPA-GWO algorithm's stability.

For the high-dimensional multi-peak test function $F_6(x)$~$F_8(x)$, it can be seen from Table A1 that MPA-GWO performs significantly better than the other algorithms on these three standard test functions. Compared with other algorithms, it can be seen that MPA-GWO performs optimally and most consistently, as evidenced by being the smallest in both mean and standard deviation. Notably, the theoretical optimum of zero is achieved on both $F_6(x)$, $F_8(x)$.

For the fixed-dimensional multimodal function ($F_9(x)$~$F_{15}(x)$), Table A1 shows the data comparison of all algorithms to optimize the fixed-dimensional multimodal function. According to the comparison of mean and standard deviation in the table, MPA-GWO achieves better mean and standard deviation for the $F_9(x)$, $F_{12}(x)$ and $F_{13}(x)$ test functions; in medium $F_{10}(x)$, MPA achieves the optimal mean and standard deviation, MPA-GWO is second; in medium $F_{11}(x)$, MPA-GWO has the same mean and less standard deviation than the four functions GWO, WOA and GWFOA; in medium $F_{14}(x)$, GWFOA achieves the optimal mean and MPA-GWO is the next best; in $F_{15}(x)$, MPA-GWO achieved the optimal mean and standard deviation together with MPA. The overall reflects that MPA-GWO has stronger global searchability among the fixed-dimensional multimodal functions.

4. RVFL Based on MPA-GWO for Oil Reservoir Prediction

4.1. Design of Oil Layer Recognition System

As mentioned above, the prediction performance of RVFL is mainly affected by the input weights and the hidden layer bias, which directly affect the prediction effect of the model. To this end, we propose an improved MPA-GWO-RVFL model, whose main idea is to optimize the two-parameter pairings of RVFL by using the good optimizing ability of the above optimization algorithm, and after a certain number of iterations, the best parameter values are filtered out, so as to improve the RVFL prediction capability. Then, we apply MPA-GWO-RVFL to oil logging and verify the effectiveness of this algorithm by using oil data provided by an oil field.

The block diagram of the MPA-GWO-RVFL-based Oil layer prediction system is shown in Figure 3.

4.2. RVFL Model Optimization

The steps of the MPA-GWO-RVFL model are as follows.

1. Data acquisition and pre-processing

The oil logging data in this paper is obtained from the actual data measured by logging tools in an oil field in China (Xinjiang). The data pre-processing mainly focuses on denoising. In addition, because the attributes have different magnitudes and value ranges, these data need to be normalized first so that the sample data range is between [0, 1], and then the normalized influence factor data are substituted into the network for

training and testing to produce the results. One of the formulas for sample normalization is shown below.

$$x = \frac{(x - x_{\min})}{x_{\max} - x_{\min}} \tag{18}$$

where $x \in [x_{\min}, x_{\max}]$, $x_{\min}$ is the minimum value of the data sample attribute and $x_{\max}$ is the maximum value of the data sample attribute.

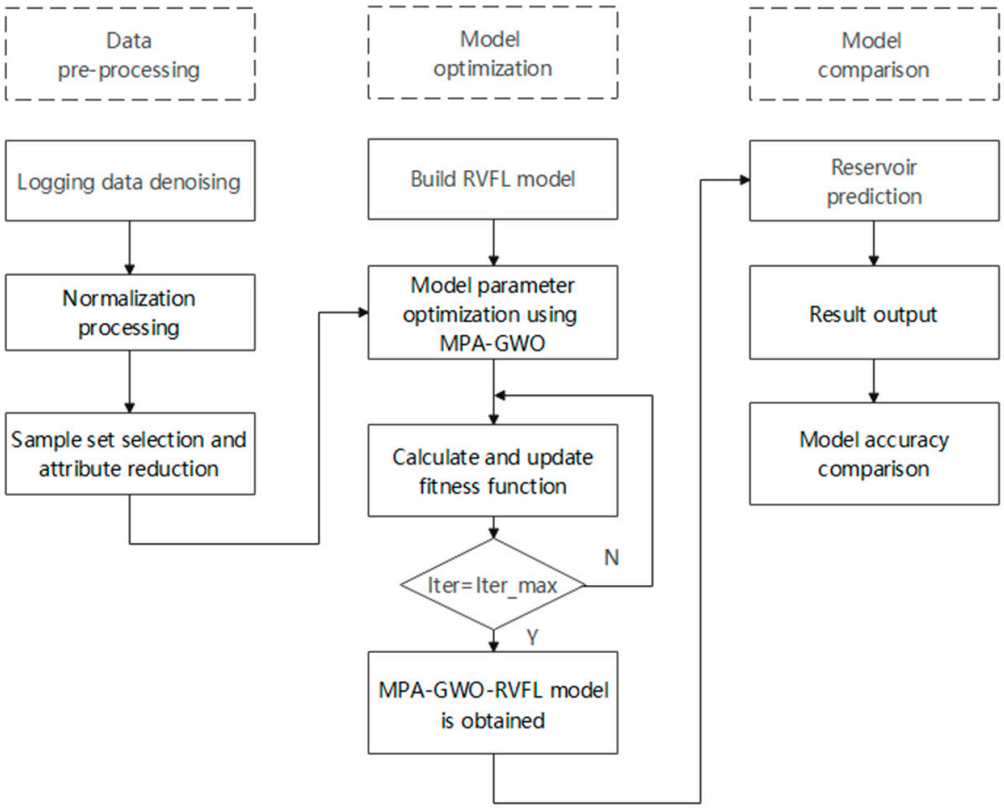


Figure 3. Block diagram of the MPA-GWO-RVFL-based Oil layer prediction system.

2. Selection of sample set and attribute approximation

The selection of the sample set used for training should be complete and comprehensive and should be closely related to the oil layer assessment. In addition, the degree of determination of Oil layer prediction varies for each condition attribute of the oil layer. Usually, there are dozens of logging condition attributes in logging data, but not all of them play a decisive role, so attribute approximation must be performed. In this paper, we use an inflection point-based discretization algorithm followed by an attribute dependency-based reduction method to reduce the logging attributes.

3. MPA-GWO-RVFL modeling

Firstly, the MPA-GWO-RVFL model is established, the function activated, the number of hidden layer nodes and the population size are determined, the population dimension dim $= (n+1) \times L$, n and L are set to represent the number of input layer nodes and hidden layer nodes, respectively, and the maximum number of iterations of the algorithm is T. The position of the population is updated according to the MPA-improved GWO, and each search agent is rearranged into matrix form, and the error rate of the test set prediction results in the training sample is used as the fitness function during the iterative solution process as follows.

$$f = \frac{\sum_{i=1}^{M} T_i}{N} 100\% \tag{19}$$

where f is the prediction error rate, M is the number of sample categories, T_i is the number of samples with prediction errors in each category and N is the number of samples in the test set in the training sample.

4. Derive the output weights

When the algorithm reaches the termination condition, the optimal search agent position is saved at this point and rearranged into matrix form as the optimal solution, i.e., the optimal input weights W and biases are obtained B, and the output weights are computed β.

5. Bring the output weights β into the RVFL model

The MPA-GWO optimized RVFL process is shown in Figure 4.

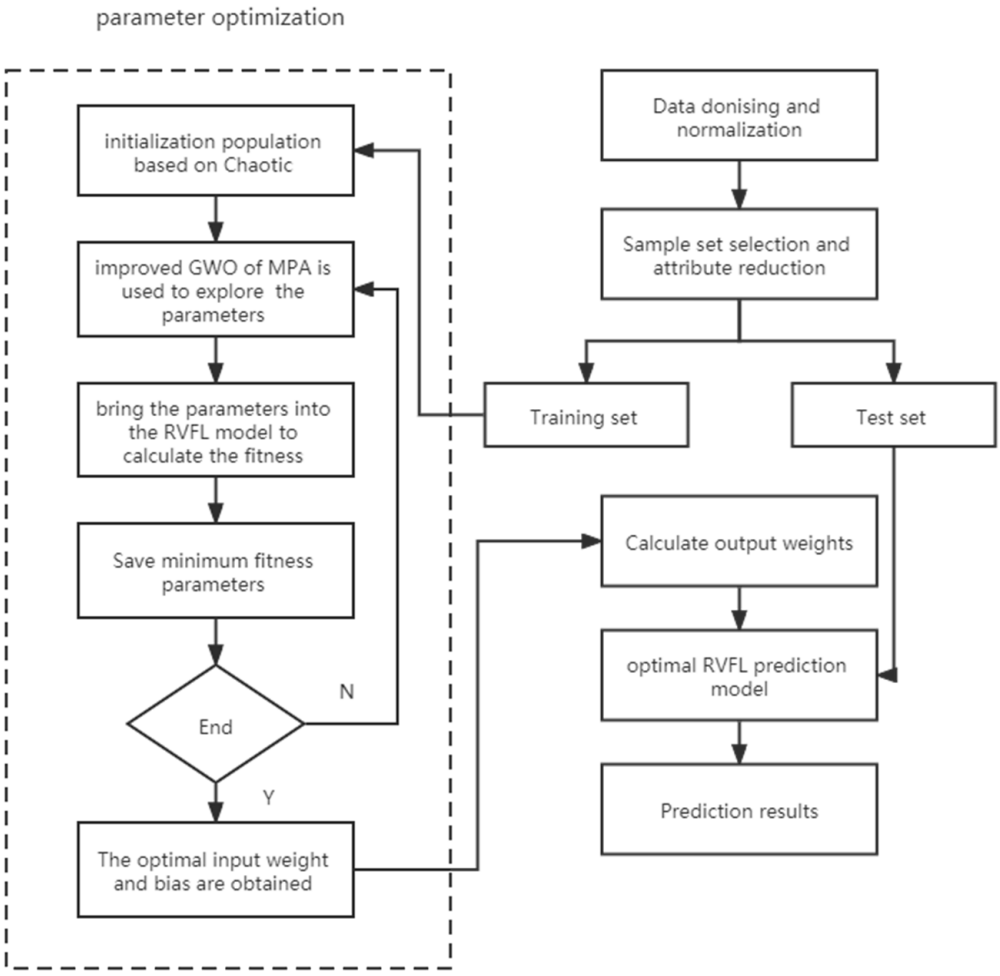


Figure 4. MPA-GWO-RVFL Oil layer prediction model.

6. Predict

The trained MPA-GWO-RVFL model is used for reservoir prediction, and the results are output and compared with the actual data.

4.3. Data Processing

In order to verify the application effect of the improved algorithm optimization, logging data were selected from the database for training and testing.

Table 2 gives the conditional attributes, including redundant attributes and important attributes, as well as the value range of important attributes, *GR* represents natural gamma, *DT* represents acoustic time difference, *SP* represents natural potential, *LLD* represents

deep lateral resistivity, *LLS* represents shallow lateral resistivity, *DEN* represents compensation density and *K* represents potassium. We classify the datasets containing important attributes into training sets and testing machines, in which the value depth of training sets is 3150 to 3330 and the value depth of testing sets is 3330 to 3460.

Table 2. Attribute reduction and value range.

Category	Condition Attributes						
Redundant Attributes	NPHI	PE	U	TH	CALI		
Important Attributes	GR	DT	SP	LLD	LLS	DEN	K
Boundary	[6, 200]	[152, 462]	[−167, −68]	[0, 25000]	[0, 3307]	[1, 4]	[0, 5]

4.4. MPA-GWO-RVFL Algorithm Parameter Analysis

4.4.1. Selection of RVFL Activation Function

In order to make the model achieve better prediction results, it is first necessary to find the best activation function for the WPA-GWO-RVFL model. In this experiment, the number of hidden layer nodes is set to 100, and the prediction accuracy under each activation function is obtained by using a 5-fold cross-validation method. Table A2 shows the results of 10 runs under various activation functions. According to the results, it is seen that when the activation function is set to sigmoid, the average prediction accuracy is the highest, and the standard deviation is also smaller. Therefore, sigmoid is determined as the activation function in the next experiments.

4.4.2. Selection of the Number of RVFL Hidden Layer Nodes

Another metric that affects the prediction accuracy of RVFL is the number of hidden layer neurons, too many or too few neurons will affect the accuracy or processing speed; this experiment analyzes the prediction accuracy of multiple sets of algorithms with different hidden layer nodes. The number of nodes is increased from 10 to 150, the activation function of each improved model algorithm is set to sigmoid, the dataset is divided into five folds for cross-validation, and it is run 10 times to take the average as the final result. The experimental results are shown in Table A3 and Figure 5, which visually show the trend of prediction accuracy of various improved RVFL algorithms with the number of nodes.

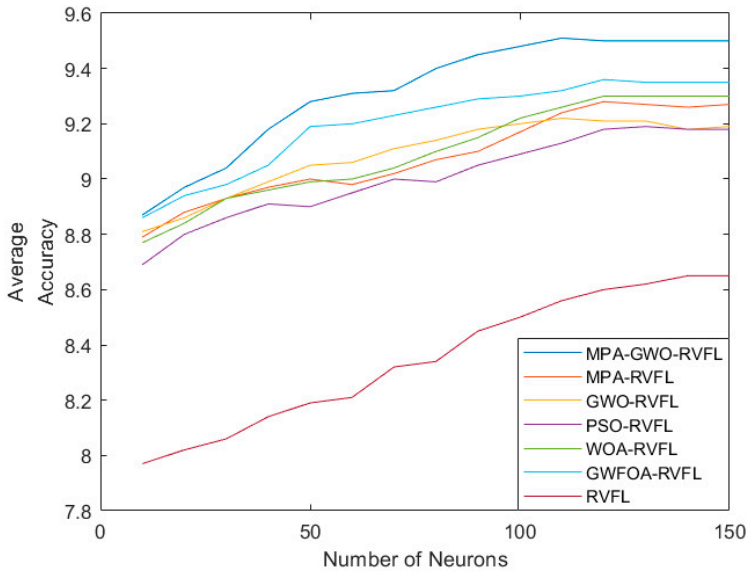


Figure 5. Influence of node number on prediction accuracy.

As seen in Table A3, when the number of nodes is in the interval [10, 110], the average prediction accuracy of each algorithm improves significantly as the number of nodes in the hidden layer increases. As seen in Figure 6, the average prediction accuracy of each algorithm grows slowly as the number of nodes in the hidden layer increases in the interval [110, 150] and successively reaches a steady-state and stabilizes within a certain range. From Table A3, a cross-sectional comparison shows that the average prediction accuracy of the MPA-GWO-RVFL algorithm consistently outperforms the other algorithms, and the average prediction accuracy of the MPA-GWO-RVF algorithm enters a steady-state relatively quickly compared to the other algorithms. This indicates that this algorithm can use a smaller network to get the optimal prediction accuracy. At the number of nodes of 110, MPA-GWO-RVFL enters a smooth state first, so 110 is chosen as the final set number of nodes.

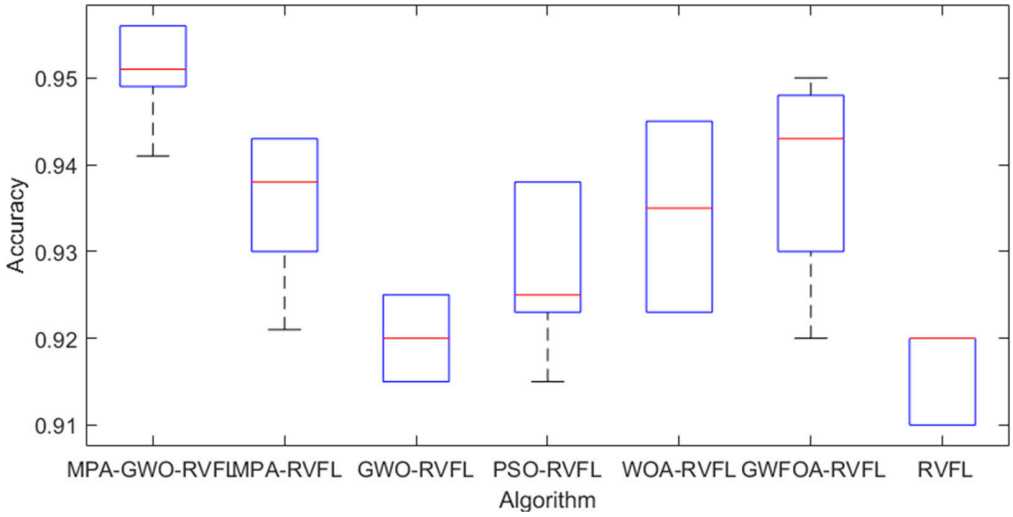


Figure 6. Stability analysis box diagram.

4.4.3. Selection of Model Population

Another important parameter that affects the prediction accuracy is the number of initialized populations. In this experiment, in order to get the effect of the number of populations on the prediction accuracy, the number of populations was set to five categories of 10, 20, 30, 40 and 50 in MPA-GWO-RVFL for 5-fold cross-validation, run 10 times to take the average as the final result, the number of iterations was set to 60 and the prediction accuracy is shown in Table 3.

Table 3. Effect of population size on prediction accuracy.

Number of Enhancement Nodes	10	20	30	40	50
Average Accuracy	9.278×10^{-1}	9.359×10^{-1}	9.418×10^{-1}	9.500×10^{-1}	9.500×10^{-1}

As shown in the table, the prediction accuracy gradually increases as the population size increases. At population size 40 and 50, the prediction accuracy converges, so the algorithm population size is set at 40.

For better comparison tests, the RVFL model, MPA-RVFL model, GWO-RVFL model, PSO-RVFL, WOA-RVFL and GWFOA-RVFL were built and compared with the MPA-GWO-RVFL model, and then these optimality-seeking prediction models were used for the test set of Oil layer prediction.

4.5. Model Comparison

4.5.1. Accuracy Analysis of Algorithm

After the above analysis of the algorithm parameters, the activation function used to conduct the final prediction result comparison experiment is Sigmoid, the number of nodes is 110, the population size is 40, in addition, the learning rate is set to 0.1 and the rest of the parameters are taken as the default values of the Matlab toolbox. The dataset was divided into five folds in the experiment, and each algorithm was used to obtain the prediction accuracy of the test set using 5-fold cross-validation and run 10 times to ensure convincing results. Table 4 shows the maximum, minimum, mean and standard deviation obtained for each algorithm run 10 times.

Table 4. Model comparison experimental results.

Measure	MPA-GWO-RVFL	MPA-RVFL	GWO-RVFL	PSO-RVFL	WOA-RVFL	GWFOA-RVFL	RVFL
Max	0.9561×10^{-1}	0.938×10^{-1}	0.9397×10^{-1}	0.9395×10^{-1}	0.8912×10^{-1}	0.9494×10^{-1}	0.8593×10^{-1}
Min	0.9292×10^{-1}	0.8946×10^{-1}	0.9033×10^{-1}	0.9079×10^{-1}	0.8326×10^{-1}	0.9119×10^{-1}	0.8196×10^{-1}
Avg	0.9464×10^{-1}	0.9174×10^{-1}	0.916×10^{-1}	0.9237×10^{-1}	0.859×10^{-1}	0.9317×10^{-1}	0.8423×10^{-1}
Stdv	0.0104	0.0130	0.0150	0.0201	0.0110	0.0109	0.0210

4.5.2. Stability and Analysis of the Algorithm

Figure 6 shows the prediction accuracy box plots for each algorithm separately, each algorithm is validated using a 5-fold crossover, and both are run 10 times. The top and bottom of the figure are the maximum and minimum values, respectively, the top and bottom edges of the box represent the upper and lower quartiles of prediction accuracy, respectively, and the red line in the middle of the box represents the median.

The compactness of the boxplot in Figure 6 shows that MPA-GWO-RVFL is more stable than MPA-RVFL, GWO-RVFL, PSO-RVFL, WOA-RVFL, GWFOA-RVFL and RVFL.

4.5.3. Convergence Analysis of the Algorithm

Figure 7 shows the relationship between fitness and the number of iterations for MPA-GWO-RVFL and the comparison test, and fitness represents the prediction error rate. From Figure 7, it can be seen that as the iterations proceed, the minimum fitness achieved by the MPA-GWO-RVFL algorithm is smaller than the minimum fitness of the other algorithms, and it is not difficult to find that the MPA-GWO-RVFL algorithm converges the fastest among all the algorithms, and the MPA-GWO-RVFL algorithm obtains the minimum fitness with fewer iterations. This shows the excellent prediction performance of the MPA-GWO-RVFL algorithm.

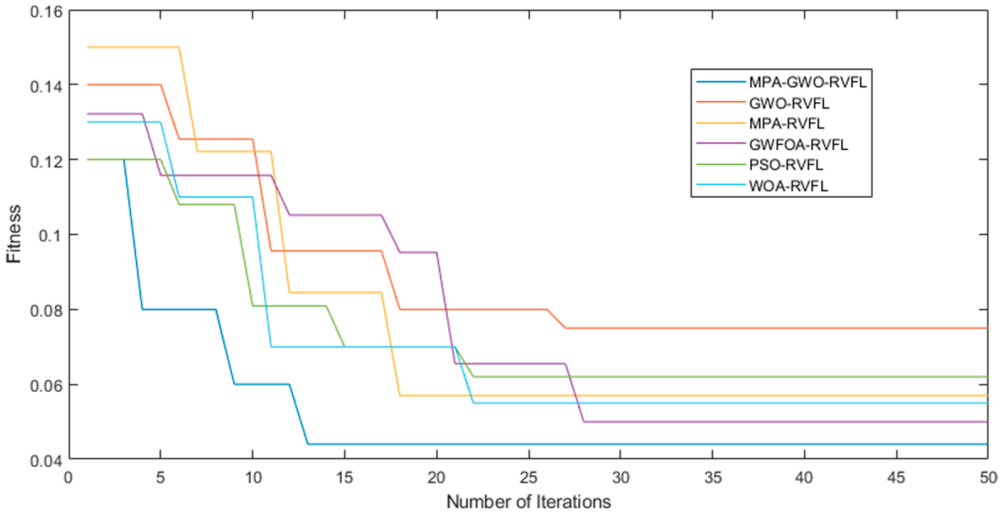


Figure 7. Iterative convergence curve.

4.5.4. Discussion

By comprehensively comparing the experimental results, algorithm stability and convergence of the proposed algorithm, it can be seen that the proposed MPA-GWO algorithm has the overall characteristics of high prediction accuracy, good stability and convergence compared with the same type of algorithms. It has broad application prospects for general optimization problems encountered in other fields for practical application, especially in the hyperparametric optimization of neural networks.

5. Conclusions

In this study, an RVFL to Oil layer prediction model optimized by the GWO algorithm based on the MPA improvement is proposed. The experimental results show the following conclusions.

(1) In this paper, an improved grey wolf optimizer is presented. The algorithm is applied to chaos theory to initialize the population and MPA is used to enhance its global exploration capabilities. Six popular population intelligence algorithms (MPA-GWO, WOA, MPA, GWO, PSO and GWFOA) are used to conduct 30 independent experiments on 15 benchmarks and are compared. From the results of the experiments, it was concluded that the MPA-GWO algorithm showed a significant improvement in convergence speed and convergence accuracy compared to the other intelligent optimization algorithms.
(2) MPA-GWO is used for RVFL input weighting and hidden layer bias finding, and the MPA-GWO-RVFL model is developed. The validity of MPA-GWO-RVFL was verified. The improved model has a higher prediction accuracy compared to the MPA-RVFL, GWO-RVFL, PSO-RVFL, WOA-RVFL, GWFOA-RVFL and RVFL models. The highest accuracy reached 95.61%, and the average accuracy was 94.64%.
(3) The convergence curves and box plots reflect that the convergence speed of the algorithm proposed in this paper is somewhat faster relative to the comparison algorithms. Moreover, its stability has some advantages over most of the comparison algorithms.

MPA-GWO-RVFL model and its application in reservoir prediction has been investigated in this paper. In the future, it is interesting to develop a hybrid neural network model for reservoir prediction. Additionally, it is another significant subject of further investigation to design a more efficient meta-heuristic algorithm and deep-learning-based non-linear combined mechanism to further improve forecasting performance.

Author Contributions: Conceptualization, P.L. and K.X.; methodology, P.L. and K.X.; software, P.L.; validation, P.L., Y.P. and S.F.; formal analysis, P.L. and K.X.; investigation, P.L. and K.X.; resources, P.L., Y.P. and S.F.; data curation, P.L., Y.P. and S.F.; writing—original draft preparation, P.L.; visualization, Y.P. and S.F. supervision, K.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. U1813222, No. 42075129), Hebei Province Natural Science Foundation (No. E2021202179), Key Research and Development Project from Hebei Province (No. 19210404D, No. 20351802D, No. 21351803D).

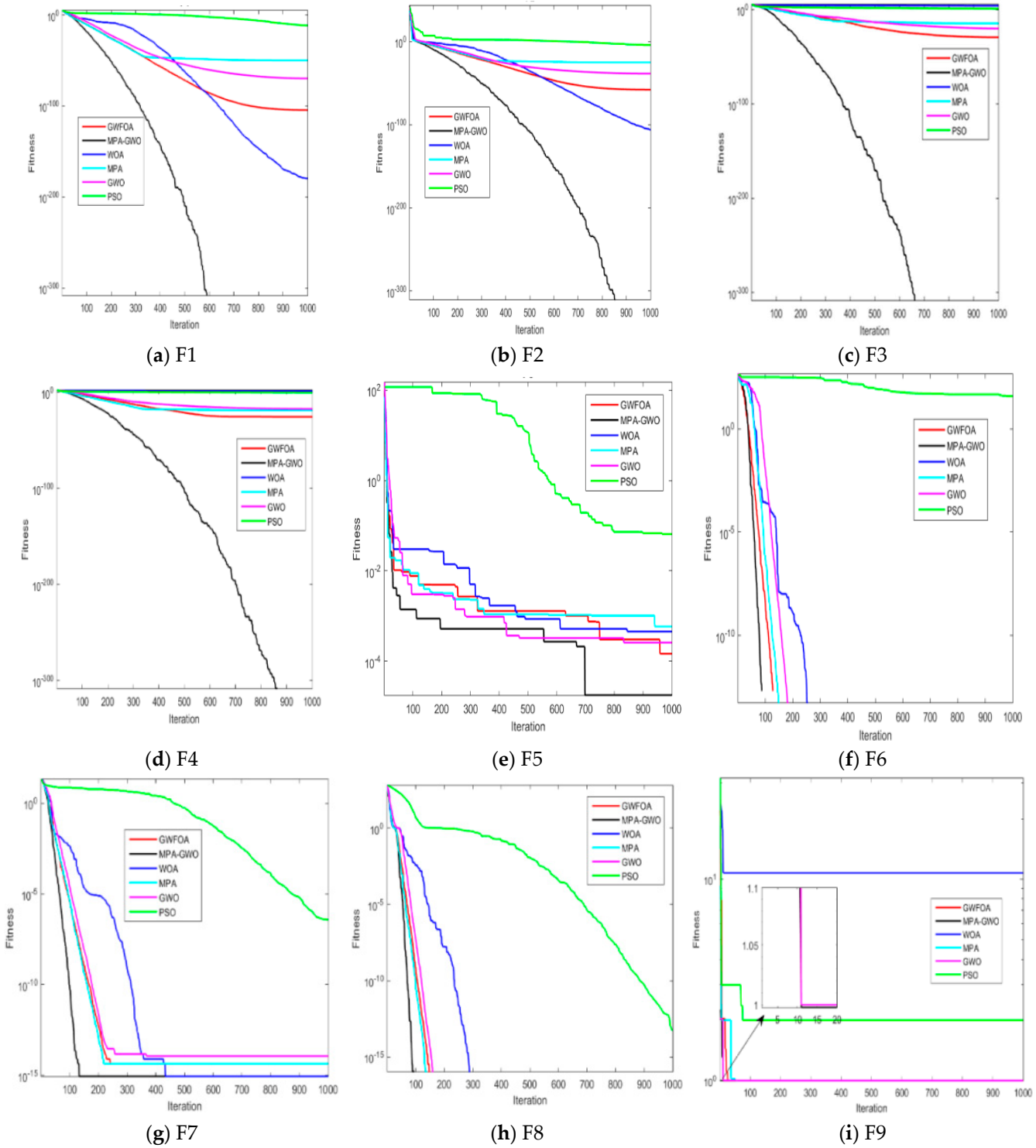
Informed Consent Statement: Written informed consent has been obtained from the patients to publish this paper.

Data Availability Statement: All data are available upon request from the corresponding author.

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

Appendix A

The experimental results in Section 2 are presented below, including statistics for each algorithm, convergence curves and box line plots.

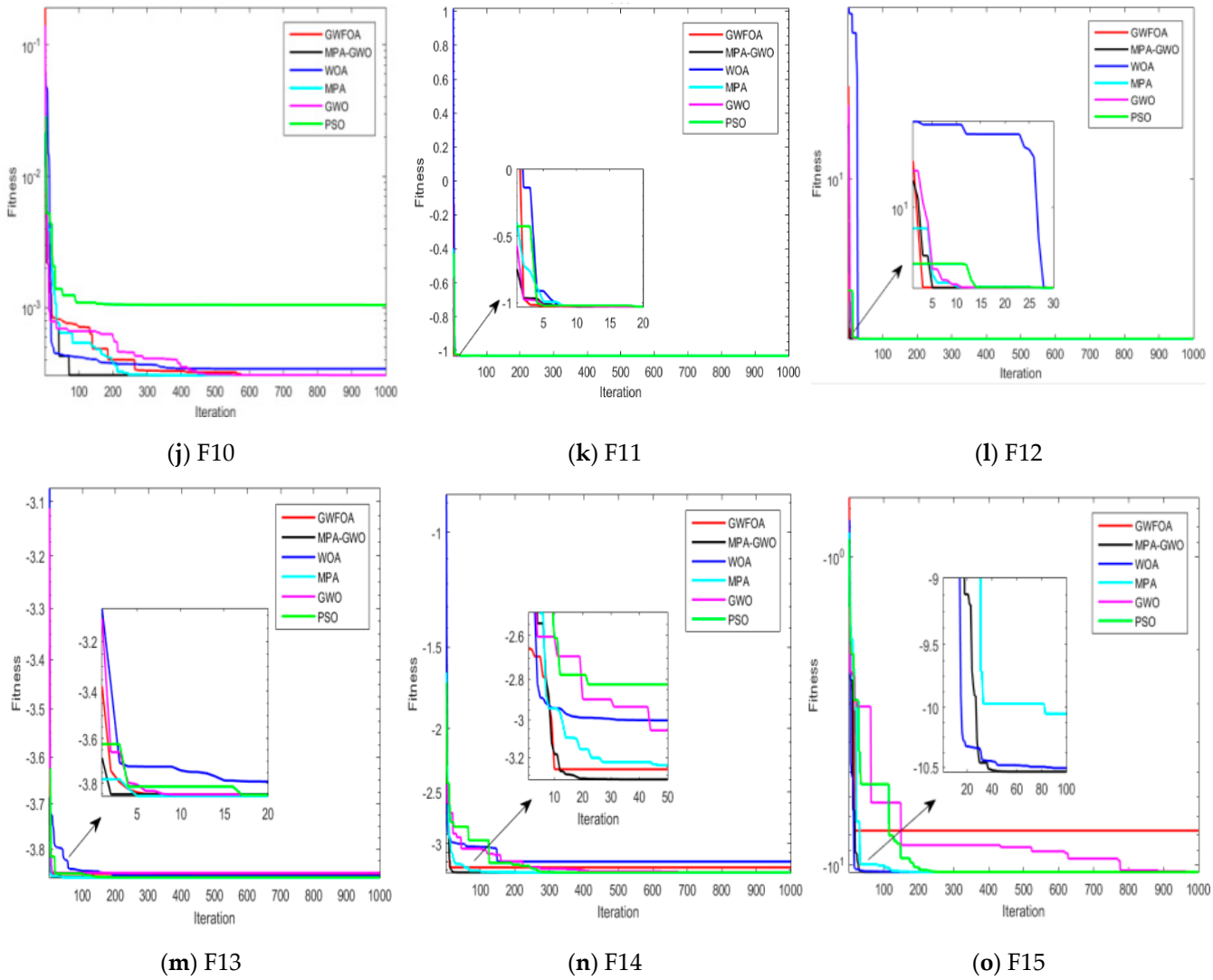


(**a**) F1 (**b**) F2 (**c**) F3

(**d**) F4 (**e**) F5 (**f**) F6

(**g**) F7 (**h**) F8 (**i**) F9

Figure A1. *Cont*.



(**j**) F10 (**k**) F11 (**l**) F12

(**m**) F13 (**n**) F14 (**o**) F15

Figure A1. Convergence curves for 15 benchmark functions. (**a**) Convergence curves for F1. (**b**) Convergence curves for F2. (**c**) Convergence curves for F3. (**d**) Convergence curves for F4. (**e**) Convergence curves for F5. (**f**) Convergence curves for F6. (**g**) Convergence curves for F7. (**h**) Convergence curves for F8. (**i**) Convergence curves for F9. (**j**) Convergence curves for F10. (**k**) Convergence curves for F11. (**l**) Convergence curves for F12. (**m**) Convergence curves for F13. (**n**) Convergence curves for F14. (**o**) Convergence curves for F15.

Table A1. Optimization results and comparison for functions.

	Function		MPA-GWO	GWO	MPA	WOA	PSO	GWFOA
Unimodal Benchmark Functions	F1	Ave	$0.00 \times 10^{+00}$	2.45×10^{-85}	3.27×10^{-21}	1.41×10^{-30}	$0.32 \times 10^{+00}$	$0.55 \times 10^{+00}$
		Std	$0.00 \times 10^{+00}$	9.08×10^{-85}	4.61×10^{-21}	4.91×10^{-30}	$0.21 \times 10^{+00}$	$1.23 \times 10^{+00}$
	F2	Ave	$0.00 \times 10^{+00}$	2.91×10^{-48}	$0.07 \times 10^{+00}$	1.06×10^{-21}	$1.04 \times 10^{+00}$	$0.01 \times 10^{+00}$
		Std	$0.00 \times 10^{+00}$	2.92×10^{-48}	$0.11 \times 10^{+00}$	2.39×10^{-21}	$0.46 \times 10^{+00}$	$0.01 \times 10^{+00}$
	F3	Ave	$0.00 \times 10^{+00}$	1.77×10^{-21}	$2.78 \times 10^{+02}$	5.39×10^{-07}	$8.14 \times 10^{+01}$	$8.46 \times 10^{+02}$
		Std	$0.00 \times 10^{+00}$	8.45×10^{-21}	$4.00 \times 10^{+02}$	2.93×10^{-06}	$2.13 \times 10^{+01}$	$1.62 \times 10^{+02}$
	F4	Ave	$0.00 \times 10^{+00}$	2.84×10^{-21}	$6.78 \times 10^{+00}$	0.73×10^{-01}	$1.51 \times 10^{+00}$	$4.56 \times 10^{+00}$
		Std	$0.00 \times 10^{+00}$	9.26×10^{-21}	$2.94 \times 10^{+00}$	$0.40 \times 10^{+00}$	$0.22 \times 10^{+00}$	$0.59 \times 10^{+00}$
	F5	Ave	6.70×10^{-05}	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.07 \times 10^{+00}$	$0.11 \times 10^{+00}$
		Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.01 \times 10^{+00}$	$0.04 \times 10^{+00}$

Table A1. *Cont.*

	Function		MPA-GWO	GWO	MPA	WOA	PSO	GWFOA
Multimodal Benchmark Functions	F6	Ave	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$4.84 \times 10^{+01}$	$0.00 \times 10^{+00}$
		Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$3.44 \times 10^{+00}$	$3.97 \times 10^{+00}$
	F7	Ave	8.84×10^{-16}	5.15×10^{-15}	9.69×10^{-12}	$7.40 \times 10^{+00}$	$1.20 \times 10^{+00}$	$0.18 \times 10^{+00}$
		Std	$0.00 \times 10^{+00}$	1.45×10^{-15}	6.13×10^{-12}	$9.90 \times 10^{+00}$	$0.73 \times 10^{+00}$	$0.15 \times 10^{+00}$
	F8	Ave	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.01 \times 10^{+00}$	$0.66 \times 10^{+00}$
		Std	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.01 \times 10^{+00}$	$0.19 \times 10^{+00}$
Fixed-Dimension Multimodal Benchmark Functions	F9	Ave	$1.00 \times 10^{+00}$	$1.00 \times 10^{+00}$	$1.00 \times 10^{+00}$	$2.11 \times 10^{+00}$	$2.18 \times 10^{+00}$	$1.00 \times 10^{+00}$
		Std	2.12×10^{-17}	1.11×10^{-16}	2.47×10^{-16}	$2.50 \times 10^{+00}$	$2.01 \times 10^{+00}$	8.84×10^{-12}
	F10	Ave	3.58×10^{-04}	3.69×10^{-49}	3.07×10^{-04}	$0.00 \times 10^{+00}$	5.61×10^{-04}	2.69×10^{-03}
		Std	1.54×10^{-04}	2.36×10^{-04}	4.09×10^{-15}	$0.00 \times 10^{+00}$	4.38×10^{-04}	4.84×10^{-03}
	F11	Ave	$-1.03 \times 10^{+00}$	$-1.03 \times 10^{+00}$	$-1.03 \times 10^{+00}$	$-1.03 \times 10^{+00}$	$-1.03 \times 10^{+00}$	$-1.03 \times 10^{+00}$
		Std	5.63×10^{-06}	2.04×10^{-06}	4.46×10^{-01}	4.2×10^{-07}	6.64×10^{-06}	2.39×10^{-08}
	F12	Ave	$3.00 \times 10^{+00}$	$3.00 \times 10^{+00}$	$3.00 \times 10^{+00}$	$3.00 \times 10^{+00}$	$3.00 \times 10^{+00}$	$3.00 \times 10^{+00}$
		Std	5.95×10^{-17}	4.96×10^{-16}	1.95×10^{-15}	4.22×10^{-15}	1.38×10^{-15}	2.45×10^{-07}
	F13	Ave	$-3.86 \times 10^{+00}$	$-3.86 \times 10^{+00}$	$-3.86 \times 10^{+00}$	$-3.85 \times 10^{+00}$	$-3.86 \times 10^{+00}$	$-3.86 \times 10^{+00}$
		Std	3.17×10^{-16}	2.15×10^{-15}	2.42×10^{-15}	$0.00 \times 10^{+00}$	2.68×10^{-15}	2.85×10^{-8}
	F14	Ave	$-3.32 \times 10^{+00}$	$-3.28 \times 10^{+00}$	$-3.32 \times 10^{+00}$	$-2.98 \times 10^{+00}$	$-3.26 \times 10^{+00}$	$-3.34 \times 10^{+00}$
		Std	1.59×10^{-15}	$0.06 \times 10^{+00}$	1.14×10^{-11}	$0.38 \times 10^{+00}$	6.05×10^{-2}	5.99×10^{-2}
	F15	Ave	$-1.05 \times 10^{+1}$	$-1.03 \times 10^{+01}$	$-1.05 \times 10^{+1}$	$-9.34 \times 10^{+00}$	$-7.25 \times 10^{+00}$	$-7.77 \times 10^{+00}$
		Std	1.92×10^{-07}	$1.48 \times 10^{+00}$	3.89×10^{-11}	$2.41 \times 10^{+00}$	$3.66 \times 10^{+00}$	$3.73 \times 10^{+00}$

Table A2. Influence of activation function on prediction accuracy.

Times	Activation Function					
	Sig	Sin	Hardlim	Tribas	Radbas	ReLU
1	8.68×10^{-1}	7.90×10^{-1}	8.18×10^{-1}	8.04×10^{-1}	8.31×10^{-1}	8.77×10^{-1}
2	8.52×10^{-1}	8.52×10^{-1}	8.32×10^{-1}	7.46×10^{-1}	8.22×10^{-1}	8.19×10^{-1}
3	8.69×10^{-1}	8.54×10^{-1}	8.26×10^{-1}	7.65×10^{-1}	8.66×10^{-1}	8.37×10^{-1}
4	8.61×10^{-1}	8.11×10^{-1}	7.99×10^{-1}	7.92×10^{-1}	8.28×10^{-1}	8.64×10^{-1}
5	8.53×10^{-1}	8.23×10^{-1}	8.11×10^{-1}	7.88×10^{-1}	8.21×10^{-1}	8.61×10^{-1}
6	8.94×10^{-1}	8.15×10^{-1}	8.04×10^{-1}	7.59×10^{-1}	8.51×10^{-1}	8.31×10^{-1}
7	8.45×10^{-1}	8.18×10^{-1}	8.25×10^{-1}	7.62×10^{-1}	8.44×10^{-1}	8.34×10^{-1}
8	8.69×10^{-1}	8.50×10^{-1}	8.33×10^{-1}	7.85×10^{-1}	8.08×10^{-1}	8.57×10^{-1}
9	8.87×10^{-1}	7.93×10^{-1}	8.67×10^{-1}	8.21×10^{-1}	8.64×10^{-1}	8.94×10^{-1}
10	8.78×10^{-1}	7.70×10^{-1}	7.92×10^{-1}	7.78×10^{-1}	8.31×10^{-1}	8.51×10^{-1}
Average	8.68×10^{-1}	8.18×10^{-1}	8.21×10^{-1}	8.31×10^{-1}	8.37×10^{-1}	8.53×10^{-1}
Std	1.3×10^{-1}	2.5×10^{-1}	1.9×10^{-1}	2.4×10^{-1}	1.7×10^{-1}	2.0×10^{-1}

Table A3. Influence of node number on prediction accuracy.

Number of Enhancement Nodes	MPA-GWO-RVFL	MPA-RVFL	GWO-RVFL	PSO-RVFL	WOA-RVFL	GWFOA-RVFL	RVFL
10	8.87×10^{-1}	8.79×10^{-1}	8.81×10^{-1}	8.69×10^{-1}	8.77×10^{-1}	8.86×10^{-1}	7.97×10^{-1}
20	8.97×10^{-1}	8.88×10^{-1}	8.86×10^{-1}	8.80×10^{-1}	8.84×10^{-1}	8.94×10^{-1}	8.02×10^{-1}
30	9.04×10^{-1}	8.93×10^{-1}	8.93×10^{-1}	8.86×10^{-1}	8.93×10^{-1}	8.98×10^{-1}	8.06×10^{-1}
40	9.18×10^{-1}	8.97×10^{-1}	8.99×10^{-1}	8.91×10^{-1}	8.96×10^{-1}	9.05×10^{-1}	8.14×10^{-1}
50	9.28×10^{-1}	9.00×10^{-1}	9.05×10^{-1}	8.90×10^{-1}	8.99×10^{-1}	9.19×10^{-1}	8.19×10^{-1}
60	9.31×10^{-1}	8.98×10^{-1}	9.06×10^{-1}	8.95×10^{-1}	9.00×10^{-1}	9.20×10^{-1}	8.21×10^{-1}
70	9.32×10^{-1}	9.02×10^{-1}	9.11×10^{-1}	9.00×10^{-1}	9.04×10^{-1}	9.23×10^{-1}	8.32×10^{-1}
80	9.40×10^{-1}	9.07×10^{-1}	9.14×10^{-1}	8.99×10^{-1}	9.10×10^{-1}	9.26×10^{-1}	8.34×10^{-1}
90	9.45×10^{-1}	9.10×10^{-1}	9.18×10^{-1}	9.05×10^{-1}	9.15×10^{-1}	9.29×10^{-1}	8.45×10^{-1}
100	9.48×10^{-1}	9.17×10^{-1}	9.20×10^{-1}	9.09×10^{-1}	9.22×10^{-1}	9.30×10^{-1}	8.50×10^{-1}
110	9.51×10^{-1}	9.24×10^{-1}	9.22×10^{-1}	9.13×10^{-1}	9.26×10^{-1}	9.32×10^{-1}	8.56×10^{-1}
120	9.50×10^{-1}	9.28×10^{-1}	9.21×10^{-1}	9.18×10^{-1}	9.30×10^{-1}	9.36×10^{-1}	8.60×10^{-1}
130	9.50×10^{-1}	9.27×10^{-1}	9.21×10^{-1}	9.19×10^{-1}	9.30×10^{-1}	9.35×10^{-1}	8.62×10^{-1}
140	9.50×10^{-1}	9.26×10^{-1}	9.18×10^{-1}	9.18×10^{-1}	9.30×10^{-1}	9.35×10^{-1}	8.65×10^{-1}
150	9.50×10^{-1}	9.27×10^{-1}	9.19×10^{-1}	9.18×10^{-1}	9.30×10^{-1}	9.35×10^{-1}	8.65×10^{-1}

References

1. Nassan, T.H.; Amro, M. Finite Element Simulation of Multiphase Flow in Oil Reservoirs-Comsol Multiphysics as Fast Prototyping Tool in Reservoir Simulation. *Gorn. Nauk. Tekhnologii Min. Sci. Technol.* **2020**, *4*, 220–226. [CrossRef]
2. Klyuev, R.V.; Bosikov, I.I.; Egorova, E.V.; Gavrina, O.A. Assessment of mining-geological and mining technical conditions of the Severny pit with the use of mathematical models. *Sustain. Dev. Mt. Territ.* **2020**, *3*, 418–427. [CrossRef]
3. Sun, W.; Ren, T.; Zhang, X.; Song, H. Optimization of intermittent oil production pattern based on data mining technology. In Proceedings of the 3rd International Conference on Intelligent Control, Measurement and Signal Processing and Intelligent Oil Field (ICMSP), Xi'an, China, 23–25 July 2021; pp. 361–364.
4. Chen, D.; San Martin, L.E.; Merchant, G.A.; Strickland, R. Processing well logging data with neural network. U.S. Patent 7,814,036, 12 October 2010. pp. 1–25.
5. Pan, S.; Liang, H.; Liang, L.I.; Wang, J. Dynamic prediction on reservoir parameter by improved PSO-BP neural network. *Comput. Eng. Appl.* **2014**, *50*, 52–56.
6. Osman, E.A.; Abdel-Wahhab, O.A.; Al-Marhoun, M.A. Prediction of oil PVT properties using neural networks. In *SPE Middle East Oil Show*; OnePetro: Manama, Bahrain, 2001; pp. 17–20.
7. Huang, G.B.; Zhou, Q.Y.; Siew, C.K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 25–29 July 2004; pp. 985–990.
8. Pao, Y.-H.; Park, G.-H.; Sobajic, D.J. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing* **1994**, *6*, 163–180. [CrossRef]
9. Tang, L.; Wu, Y.; Yu, L. A non-iterative decomposition-ensemble learning paradigm using RVFL network for crude oil price forecasting. *Appl. Soft Comput.* **2017**, *70*, 1097–1108. [CrossRef]
10. Bisoi, R.; Dash, P.K.; Mishra, S.P. Modes decomposition method in fusion with robust random vector functional link network for crude oil price forecasting. *Appl. Soft Comput.* **2019**, *80*, 475–493. [CrossRef]
11. Yu, L.; Wu, Y.; Tang, L.; Lai, K.K. Investigation of diversity strategies in RVFL network ensemble learning for crude oil price forecasting. *Soft Comput.* **2021**, *25*, 3609–3622. [CrossRef]
12. Chai, S.; Zhao, X.; Wong, W.K. Optimal Granule-Based PIs Construction for Solar Irradiance Forecast. *IEEE Trans. Power Syst.* **2016**, *31*, 3332–3333. [CrossRef]
13. Ye, H.; Cao, F.; Wang, D. A hybrid regularization approach for random vector functional-link networks—ScienceDirect. *Expert Syst. Appl.* **2020**, *140*, 112912. [CrossRef]
14. Zhou, Z.; Liu, D.; Guo, J.; Zhang, J.; Zhu, Z.; Wang, C. Dyed fabric illumination estimation with regularized random vector function link network. *Color Res. Appl.* **2020**, *46*, 376–387. [CrossRef]
15. Chakraborty, A.; Kar, A.K. Swarm intelligence: A review of algorithms. *Nat.-Inspired Comput. Optim.* **2017**, *10*, 475–494.
16. Véhel, J.L.; Lutton, E. Evolutionary signal enhancement based on Hölder regularity analysis. In *Workshops on Applications of Evolutionary Computation*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 325–334.
17. Xie, M. Image thresholding segmentation based on multi-objective artificial bee colony optimization. *Digit. Video* **2018**, *42*, 6–14.
18. Liu, H.; Abraham, A.; Choi, O.; Moon, S.H. Variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems. In Proceedings of the Asia-Pacific Conference on Simulated Evolution and Learning, Hefei, China, 15–18 October 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 197–204.
19. De la Fraga, L.G.; Coello, C.A.C. A review of applications of evolutionary algorithms in pattern recognition. *Pattern Recognit. Mach. Intell. Biom.* **2011**, *10*, 3–28.
20. Iqbal, M.F.; Zahid, M.; Habib, D.; John, L.K. Efficient prediction of network traffic for real-time applications. *J. Comput. Netw. Commun.* **2019**, *2019*, 4067135. [CrossRef]
21. Deb, K.; Bandaru, S.; Greiner, D.; Gaspar-Cunha, A.; Tutum, C.C. An integrated approach to automated innovization for discovering useful design principles: Case studies from engineering. *Appl. Soft Comput.* **2014**, *15*, 42–56. [CrossRef]
22. Sastry, K.; Goldberg, D.; Kendall, G. Genetic algorithms. In *Search Methodologies*; Springer: Boston, MA, USA, 2005; pp. 97–125.
23. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
24. Liu, Z.; Li, Z.; Chen, W.; Zhao, Y.; Yue, H.; Wu, Z. Path Optimization of Medical Waste Transport Routes in the Emergent Public Health Event of COVID-19: A Hybrid Optimization Algorithm Based on the Immune–Ant Colony Algorithm. *Int. J. Environ. Res. Public Health* **2020**, *17*, 5831. [CrossRef]
25. Dorigo, M.; Maniezzo, V. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans.* **1996**, *26*, 29. [CrossRef] [PubMed]
26. Pham, B.T.; Qi, C.; Ho, L.S.; Nguyen-Thoi, T.; Al-Ansari, N.; Nguyen, M.D.; Nguyen, H.D.; Ly, H.B.; Le, H.V.; Prakash, I. A novel hybrid soft computing model using random forest and particle swarm optimization for estimation of undrained shear strength of soil. *Sustainability* **2020**, *12*, 2218. [CrossRef]
27. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing SE—New Series. *Science* **1983**, *220*, 671–680. [CrossRef]
28. Glover, F. Tabu Search—Part I. *Orsa J. Comput.* **1989**, *1*, 89–98. [CrossRef]
29. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [CrossRef]

30. Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
31. Sharma, S.; Bhattacharjee, S.; Bhattacharya, A. Grey wolf optimisation for optimal sizing of battery energy storage device to minimise operation cost of microgrid. *IET Gener. Transm. Distrib.* **2016**, *10*, 625–637. [CrossRef]
32. Barman, M.; Choudhury, N. A similarity based hybrid GWO-SVM method of power system load forecasting for regional special event days in anomalous load situations in Assam, India. *Sustain. Cities Soc.* **2020**, *61*, 102311. [CrossRef]
33. Zhou, Z.; Wang, C.; Zhu, Z.; Wang, Y.; Yang, D. Sliding mode control based on a hybrid grey-wolf-optimized extreme learning machine for robot manipulators. *Optik* **2019**, *185*, 364–380. [CrossRef]
34. Ho, L.V.; Nguyen, D.H.; Mousavi, M.; De Roeck, G.; Bui-Tien, T.; Gandomi, A.H.; Wahab, M.A. A hybrid computational intelligence approach for structural damage detection using marine predator algorithm and feedforward neural networks. *Comput. Struct.* **2021**, *252*, 106568. [CrossRef]
35. Bayoumi, A.S.A.; El-Sehiemy, R.A.; Abaza, A. Effective PV Parameter Estimation Algorithm Based on Marine Predators Optimizer Considering Normal and Low Radiation Operating Conditions. *Arab. J. Sci. Eng.* **2021**, *21*, 1–16. [CrossRef]
36. Chen, X.; Qi, X.; Wang, Z.; Cui, C.; Wu, B.; Yang, Y. Fault diagnosis of rolling bearing using marine predators algorithm-based support vector machine and topology learning and out-of-sample embedding. *Measurement* **2021**, *176*, 109116. [CrossRef]
37. Yousri, D.; Fathy, A.; Rezk, H. A new comprehensive learning marine predator algorithm for extracting the optimal parameters of supercapacitor model. *J. Energy Storage* **2021**, *42*, 103035. [CrossRef]
38. Fan, Q.; Huang, H.; Chen, Q.; Yao, L.; Yang, K.; Huang, D. A modified self-adaptive marine predators algorithm: Framework and engineering applications. *Eng. Comput.* **2021**, *5*, 1–26. [CrossRef]
39. Hoang, N.D.; Tran, X.L. Remote Sensing–Based Urban Green Space Detection Using Marine Predators Algorithm Optimized Machine Learning Approach. *Math. Probl. Eng.* **2021**, *2021*, 5586913. [CrossRef]
40. Liu, X.; Yang, D. Color constancy computation for dyed fabrics via improved marine predators algorithm optimized random vector functional-link network. *Color Res. Appl.* **2021**, *22653*, 1066–1078. [CrossRef]
41. Houssein, E.H.; Hassaballah, M.; Ibrahim, I.E.; AbdElminaam, D.S.; Wazery, Y.M. An automatic arrhythmia classification model based on improved Marine Predators Algorithm and Convolutions Neural Networks. *Expert Syst. Appl.* **2021**, *187*, 115936. [CrossRef]
42. Wang, D.; Alhandoosh, M. Evolutionary extreme learning machine ensembles with size control. *Neurocomputing* **2013**, *102*, 98–110. [CrossRef]
43. Ren, Y.; Qiu, X.; Suganthan, P.; Amaratunga, G. Detecting wind power ramp with random vector functional link (rvfl) network. In Proceedings of the 2015 IEEE Symposium Series on Computational Intelligence, Cape Town, South Africa, 7–10 December 2015; pp. 687–694.
44. Zhang, Y.; Wu, J.; Cai, Z.; Du, B.; Yu, P.S. An unsupervised parameter learning model for RVFL neural network. *Neural Netw.* **2019**, *112*, 85–97. [CrossRef]
45. Peng, Y.; Li, Q.; Kong, W.; Qin, F.; Zhang, J.; Cichocki, A. A joint optimization framework to semi-supervised RVFL and ELM networks for efficient data classification. *Appl. Soft Comput.* **2020**, *97*, 106756. [CrossRef]
46. Ge, F.; Li, K.; Xu, W.; Wang, Y. Path Planning of UAV for Oilfield Inspection Based on Improved Grey Wolf Optimization Algorithm. In Proceedings of the 31st Chinese Control and Decision Conference, Nanchang, China, 3–5 June 2019; pp. 3666–3671.