

Article

Intelligent Dendritic Neural Model for Classification Problems

Weixiang Xu ¹, Dongbao Jia ^{1,2,*}, Zhaoman Zhong ¹, Cunhua Li ¹ and Zhongxun Xu ¹
¹ School of Computer Engineering, Jiangsu Ocean University, Lianyungang 222005, China; xuwx@jou.edu.cn (W.X.); zhongzhaoman@163.com (Z.Z.); cli@jou.edu.cn (C.L.); xuzx@jou.edu.cn (Z.X.)

² The Ministry of Education Key Laboratory of TianQin Project, Sun Yat-sen University, Zhuhai 519082, China

* Correspondence: dbjia@jou.edu.cn

Abstract: In recent years, the dendritic neural model has been widely employed in various fields because of its simple structure and inexpensive cost. Traditional numerical optimization is ineffective for the parameter optimization problem of the dendritic neural model; it is easy to fall into local in the optimization process, resulting in poor performance of the model. This paper proposes an intelligent dendritic neural model firstly, which uses the intelligent optimization algorithm to optimize the model instead of the traditional dendritic neural model with a backpropagation algorithm. The experiment compares the performance of ten representative intelligent optimization algorithms in six classification datasets. The optimal combination of user-defined parameters for the model evaluates by using Taguchi's method, systemically. The results show that the performance of an intelligent dendritic neural model is significantly better than a traditional dendritic neural model. The intelligent dendritic neural model has small classification errors and high accuracy, which provides an effective approach for the application of dendritic neural model in engineering classification problems. In addition, among ten intelligent optimization algorithms, an evolutionary algorithm called biogeographic optimization algorithm has excellent performance, and can quickly obtain high-quality solutions and excellent convergence speed.

Keywords: dendritic neural model (DNM); classification problems; intelligent optimization algorithm



Citation: Xu, W.; Jia, D.; Zhong, Z.;

Li, C.; Xu, Z. Intelligent Dendritic

Neural Model for Classification

Problems. *Symmetry* **2022**, *14*, 11.

<https://doi.org/10.3390/sym14010011>

sym14010011

Academic Editors: Lianbo Ma,
Shi Cheng, Shangce Gao and Yu Guo

Received: 23 October 2021

Accepted: 7 December 2021

Published: 22 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Since the dawn of the big data era, every corner of human society has accumulated a large amount of data. There is an urgent need for computer algorithms that can properly evaluate and utilize data, while machine learning just meets the urgent need of the era. Classification problems, as a hot topic of machine learning, have a widespread application in reality [1], such as common spam recognition speech recognition, tumor recognition, bank credit loan business and so on [2,3]. For solving various classification problems, lots of machine learning techniques have been proposed, such as decision tree [4], naïve Bayesian classifiers [5], support vector machine [6], artificial neural networks [7], k-nearest neighbor [8], ensemble learning [9] and so on.

Due to the high-dimensional characteristics of complex nonlinear problems, traditional methods cannot effectively solve these problems. Among them, artificial neural network simulates the information processing mechanism of biological neural network, which has a good fitting effect on nonlinear problems and has been successfully used in text classification, pattern and speech recognition. In 1943, an artificial neural network was proposed by McCulloch and Pitts firstly [10]. This model is relatively simple but significant. In 1957, Rosenblatt came up with perceptron model, which was based on the M-P model [11]. Perceptron has fundamental principles of modern neural networks, while its structure is consistent with the real biological nerves. However, the linear perceptron has limited functions and cannot even solve the simple XOR problem. In 1986, with research based on multilayer neural networks, Rumelhart put forward the backpropagation algorithm (BP) for weight correction of a multilayer neural network [12]. It solved the learning problem of a multilayer feedforward neural network and proved that a multilayer neural network

has a strong learning capacity, which can complete lots of learning tasks and solve a large number of practical problems.

With the deepening of research, engineering problems become more and more complex. The field of machine learning is also quickly evolving, and many scholars have proposed many different neural networks, such as a cyclic neural network [13], convolutional neural network [14], feedforward neural network [15] and dendritic neural model [16–18].

The artificial neural network is originally constructed in the form of a perceptron [19]; almost all modern artificial neural networks use this kind of structure, but experiments in the field of physiology have found that biological neurons are far more complex than the above model. Studies have shown that dendritic structures contain a large number of active ion channels, and synaptic input may have a linear effect on its adjacent synaptic input [20]. In addition, many studies have shown that the biological plasticity mechanism also plays a local role in dendrites [21]. These characteristics greatly promote the role of local nonlinear components in neuron output and endow neural networks with higher information processing capabilities [22]. Different from other neural models, a dendritic neural model considers the nonlinearity of synapse, and it simulates the process of the information transmission of a neuron [17]. Because of its easy explanation and simple implementation, it has been used by many scholars to solve various complex problems, for example, tourism economic forecast [23], bankruptcy prediction [24], breast cancer classification [25], liver disorders [26] and so on [27–33]. References [25,26] use traditional dendritic neural models with backpropagation algorithm to optimize weights and thresholds. Backpropagation is gradient descent in essence; it has poor robustness, and it is extremely easy to fall into local traps [34]. References [27,35], respectively, proposed to use particle swarm optimization and a states of matter search algorithm as the optimization algorithm. However, their comparative experiment is single, and there is a lack of systematic and complete research on the application of intelligent optimization algorithm in dendritic neural models.

With the continuous innovation of evolutionary computation, intelligent algorithm has a rapid development and a wide range of practical applications in various fields such as model symmetry/asymmetry, model architecture and hyper-parameters, clustering and prediction, becoming a novel method to solve traditional optimization problems in machine learning.

Intelligent optimization algorithm is a cluster of algorithms. With continuous research and development, the algorithm cluster is growing, and a variety of algorithms arises at the historic moment, most of them inspired by biological evolution in nature. A example is the genetic algorithm (GA), which maintains and improves multiple candidate solutions based on population method and uses population characteristics to guide search [36,37], and another typical algorithm is the differential evolution algorithm (DE) of heuristic search based on population [38,39]. Moreover, inspired by biogeography theory, a biogeographic optimization algorithm (BBO) is proposed [40,41], which based on study of mathematical model of biological species migration. As another type of intelligent optimization algorithm, based on mathematical statistics theory, the estimation of distribution algorithm realizes population search and evolution by constructing probability models. Population-based incremental learning (PBIL) is a classical estimation of distribution algorithm [42,43]. Another kind of intelligent optimization algorithm are swarm intelligence algorithms, inspired by natural phenomena such as particle swarm optimization (PSO) [44,45], ant colony optimization (ACO) [46,47], artificial bee colony algorithm (ABC) [48,49] and whale optimization algorithm (WOA) [50,51]. In recent years, the Harris hawks optimization algorithm (HHO) inspired by the group cooperation behavior of the Harris eagle and chimp optimization algorithm (ChOA) inspired by chimpanzee hunting behavior in its group are new swarm intelligence optimization algorithms [52,53].

The contribution of this paper is as follows:

1. This paper proposes an intelligent dendritic neural model firstly, which uses an intelligent optimization algorithm to optimize the model instead of the traditional backpropagation algorithm. The experimental results show that the performance of

the intelligent dendritic neural model is superior to the traditional dendritic neural model, which provides an effective approach for the application of dendritic neural model in engineering classification problems.

- Through the comparison of scientific experiments, the effective intelligent learning algorithm of dendritic neural model for classification problems is determined. By comparing different types of intelligent optimization algorithms, the result shows that the biogeographic optimization algorithm has excellent performance, can quickly obtain high-quality solutions and has excellent convergence speed.

2. Dendritic Neural Model

There exists four layers in the dendritic neural model; they are the synaptic layer, dendritic layer, membrane layer and soma layer. Each layer has corresponding functions and characteristics. The detailed structure of the whole model is shown in Figure 1, where $X = \{x_1, x_2, x_3, \dots, x_n\}$ is the input data of model, m branches represent m dendritic layers and O is actual output of model. There are weights and thresholds in each synapse. In the training process of the model, the algorithm will continuously adjust the weights and thresholds in each synapse to optimize the performance of the model. The specific functions of each layer are introduced in detail below.

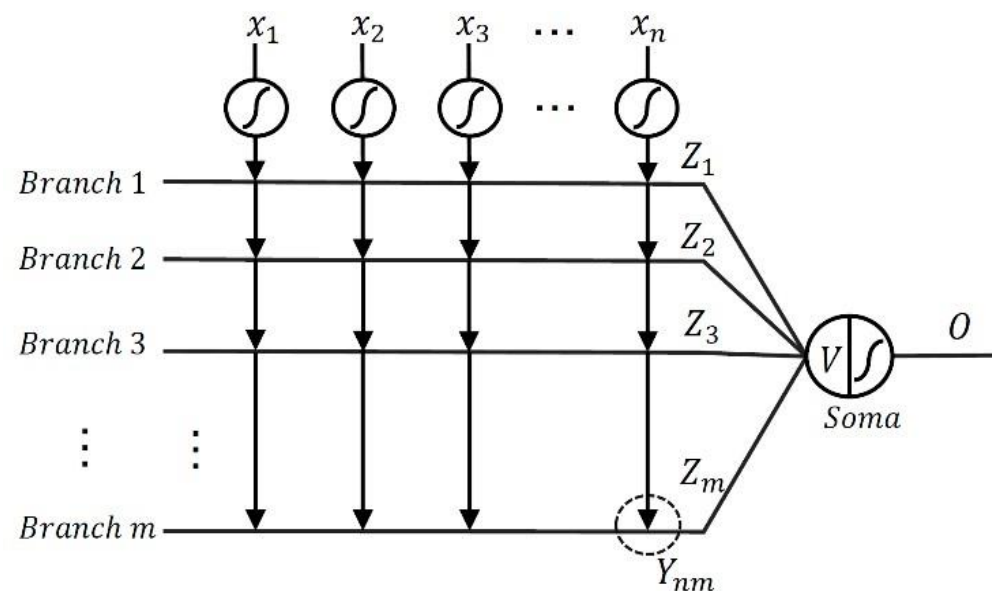


Figure 1. Structure in detail of dendritic neural model.

2.1. Synaptic Layer

Input vector $X = \{x_1, x_2, x_3, \dots, x_n\}$ inputs data from synapses, and the output of synapse layer is obtained by the activation of the sigmoid function. The nodes in the synaptic layer contain the weights and thresholds of the dendritic neural network. In the training process of the model, the intelligent optimization algorithm needs to optimize all the weights and thresholds in the synaptic layer. The expression of the synaptic layer is (1).

$$Y_{ij} = \frac{1}{e^{-k(\omega_{ij}x_i - \theta_{ij})}} \quad (1)$$

where k is a user-defined constant. ω_{ij} and θ_{ij} are the corresponding weight and threshold. According to different values of ω_{ij} and θ_{ij} , there are six situations: ① $0 < \theta_{ij} < \omega_{ij}$, ② $0 < \omega_{ij} < \theta_{ij}$, ③ $\omega_{ij} < \theta_{ij} < 0$, ④ $\theta_{ij} < \omega_{ij} < 0$, ⑤ $\theta_{ij} < 0 < \omega_{ij}$ and ⑥ $\omega_{ij} < 0 < \theta_{ij}$. In case ①, the output Y_{ij} is proportional to the output x_i , which called the Excitatory state; in case ③, the output Y_{ij} is inversely proportional to the output x_i , which is called the Inhibitory state; in cases ② and ⑥, the value of the output Y_{ij} is always close to 0, which called the Constant-0

state and in cases ④ and ⑤, the value of the output Y_{ij} is always close to 1, which called the Constant-1 state. Figure 2 shows the details of the four states corresponding to the six situations.

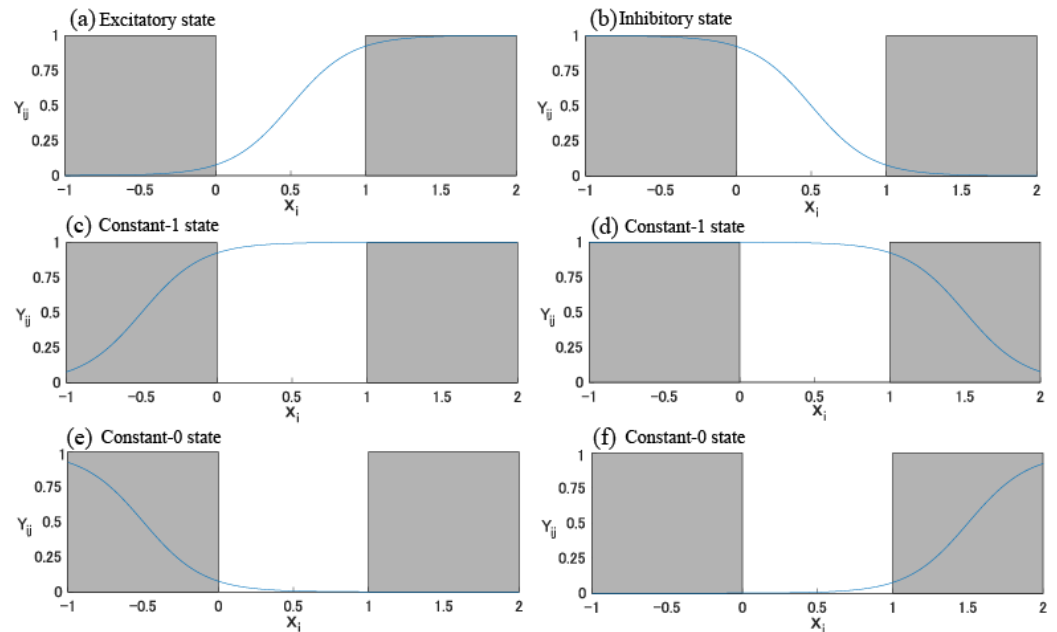


Figure 2. Details of the four states corresponding to the six situations.

2.2. Dendritic Layer

The dendrite layer multiplies results of n synaptic layers. The whole process can be expressed as (2).

$$Z_j = \prod_{i=1}^n Y_{ij} \quad (2)$$

2.3. Membrane Layer

The membrane layer is connected to m dendritic branches; the function of the membrane is to perform the sum operation over the results of all branches. The whole process can be expressed as (3).

$$V = \sum_{j=1}^m Z_j \quad (3)$$

2.4. Soma Layer

Finally, the final output of the soma layer is obtained by the sigmoid function, which can be expressed as (4).

$$O = \frac{1}{1 + e^{k_{soma}(V - \theta_{soma})}} \quad (4)$$

where k_{soma} and θ_{soma} are self-defined parameters.

3. Learning Algorithms

3.1. Backpropagation Algorithm

The traditional dendritic neural model uses the backpropagation algorithm to update weights and thresholds of the model, which adjusts the weight by backpropagation of the training error layer by layer through the chain rule. Firstly, the least squared error between the actual output O_p and the desired output T_p of the target is obtained as follows in (5).

$$E_p = \frac{1}{2} (T_p - O_p)^2 \quad (5)$$

The synaptic parameters ω_{ij} and θ_{ij} are updated along the negative gradient direction; the whole process can be described as (6) and (7).

$$\begin{aligned}\Delta\omega_{ij}(t) &= \sum_{p=1}^P \frac{\partial E_p}{\partial \omega_{ij}} \\ \Delta\theta_{ij}(t) &= \sum_{p=1}^P \frac{\partial E_p}{\partial \theta_{ij}}\end{aligned}\quad (6)$$

$$\begin{aligned}\omega_{ij}(t+1) &= \omega_{ij}(t) - \eta \Delta\omega_{ij}(t) \\ \theta_{ij}(t+1) &= \theta_{ij}(t) - \eta \Delta\theta_{ij}(t)\end{aligned}\quad (7)$$

where η represents the learning rate; it is usually set to 0.1. According to the structure of the dendritic neural model in Section 2, the whole process of partial differential derivation can be described as (8) and (9).

$$\begin{aligned}\frac{\partial E_p}{\partial \omega_{ij}} &= \frac{\partial E_p}{\partial O_p} \frac{\partial O_p}{\partial V} \frac{\partial V}{\partial Z_j} \frac{\partial Z_j}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial \omega_{ij}} \\ \frac{\partial E_p}{\partial \theta_{ij}} &= \frac{\partial E_p}{\partial O_p} \frac{\partial O_p}{\partial V} \frac{\partial V}{\partial Z_j} \frac{\partial Z_j}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial \theta_{ij}}\end{aligned}\quad (8)$$

$$\begin{aligned}\frac{\partial E_p}{\partial O_p} &= T_p - O_p \\ \frac{\partial O_p}{\partial V} &= \frac{k_s x_i e^{-k_s(V-\theta_s)}}{(1+e^{-k_s(V-\theta_s)})^2} \\ \frac{\partial V}{\partial Z_j} &= 1 \\ \frac{\partial Z_j}{\partial Y_{ij}} &= \prod_{l=1 \text{ and } l \neq i}^n Y_{lj} \\ \frac{\partial Y_{ij}}{\partial \omega_{ij}} &= \frac{k x_i e^{-k(\omega_{ij} x_i - \theta_{ij})}}{(1+e^{-k(\omega_{ij} x_i - \theta_{ij})})^2} \\ \frac{\partial Y_{ij}}{\partial \theta_{ij}} &= \frac{k e^{-k(\omega_{ij} x_i - \theta_{ij})}}{(1+e^{-k(\omega_{ij} x_i - \theta_{ij})})^2}\end{aligned}\quad (9)$$

This paper introduces the intelligent optimization algorithm into the training of the model. Different from the idea of backpropagation algorithm, the intelligent optimization algorithm updates ω and θ by iterating to find the optimization individual. Assuming that the input number of the dendritic neural model is n and m is the number of dendritic layers, if a vector represents a viable solution, then $X_i(i \in [1, Q])$ can be expressed as (10).

$$\begin{aligned}X_i &= \{x_i^1, x_i^2, \dots, x_i^n\} \\ &= \{\omega_{11}, \omega_{12}, \dots, \omega_{mn}, \theta_{11}, \theta_{12}, \dots, \theta_{mn}\}\end{aligned}\quad (10)$$

where X_i denotes the i th feasible solutions and ω and θ are weight and threshold in the dendritic layer. Generally, the intelligent optimization algorithm first initializes a certain number of feasible solutions randomly, and then iteratively optimizes all feasible solutions according to the characteristics of different optimization algorithms until meeting the termination conditions. The optimal solution X_{best} is the optimal solution of the weight and threshold of the dendritic neural model. In the iterative optimization process, the minimum mean squared error (MSE) is used as a loss function to assess the quality of each feasible solution. The minimum mean squared error is calculated by (11).

$$MSE(X_i) = \frac{1}{2P} \sum_{p=1}^P (T_p - O_p)^2 \quad (11)$$

where T_p is desired output of the p th sample, and O_p is the actual output of the p th sample.

3.2. Genetic Algorithm

Referring to the genetic evolution theory of Darwin and Mendel, the genetic algorithm was proposed by J. Holland in 1975 firstly. It was developed by simulating the biological evolution mechanism in nature. The individual of population is called the chromosome. The constant updating of chromosomes during iteration is called heredity. Inheritance consists of three parts—selection, crossover and mutation. Chromosome quality is usually evaluated by fitness function. At the beginning of the genetic algorithm is the generation of random individuals, according to a predetermined loss function to evaluate each individual, and giving a fitness value. Through the fitness function, the selected individuals produce the next generation. This operation inherits the idea of survival of the fittest in nature, and then the selected individuals combine to produce a new generation through crossover and mutation. The new generation is better than the previous generation because it inherits the excellent characteristics of the previous generation, and the whole population gradually moves towards the optimal solution. According to Algorithm 1, genetic algorithm is mainly divided into the following three parts.

- (a) Selection operator: based on the assessment of individual fitness, selection operator usually selects individuals with higher fitness and eliminates individuals with lower fitness. The common selection methods: fitness allocation method based on proportion or ranking, roulette selection method and so on.
- (b) Crossover operator: in the process of biological evolution in nature, two chromosomes form new chromosomes by gene recombination. Therefore, crossover is the core link of the whole process. The design of crossover operator needs to be analyzed for each specific question. The familiar crossover operators such as single point crossover, uniform crossover, multi-point crossover and so on.
- (c) Mutation operator: mutation changes genes inherited on chromosomes by random selection. Mutation itself can be seen as a random algorithm, strictly speaking, an auxiliary algorithm used to generate new individuals.

Algorithm 1: Genetic Algorithm.

Begin:

Randomly initialized population of chromosomes ($\{X_i\}, i \in [1, Q]$)

Evaluate the fitness value for each chromosome using Equation (11)

while Termination criterion

Selection the best chromosome by Roulette Wheel Selection

Generating new chromosomes through single-point crossover and mutation

Evaluate the fitness value of the new chromosome

Replace the population's worst chromosomes with the greatest new chromosomes

$t = t + 1$

end while

return the best solution

End

3.3. Differential Evolution Algorithm

In 1996, Rainer and Kenneth, to solve Chebyshev polynomials, proposed the differential evolution algorithm. The basic idea of DE is to randomly generate a group of the initial population and randomly select three individuals in the initial population; the new individual is generated through summing up the vector difference of two individuals with the third individual according to certain rules. Then, it would compare the new individual with another individual randomly selected in population. If the new individual is superior to the compared individual, it would retain the new individual and eliminate the compared old individual; if the new individual is inferior to the one compared with it, the algorithm would abandon the new individual, retain the old individual compared with it and reselect other individuals from population to generate new individuals. Repeated in this way, the individuals of the initial population are continuously updated until the population reaches

a certain optimal state. As shown in Algorithm 2, the process of the differential evolution algorithm includes mutation, crossover and selection operations, which is very similar to the genetic algorithm, but each process has a completely different meaning. It has a variety of mutation strategies. The strategy adopted in the experiment is DE/rand/1, and the formula is shown in (12).

$$V_i = X_{r1} + F \cdot (X_{r2} - X_{r3}) \quad (12)$$

Algorithm 2: Differential Evolution Algorithm.

Begin:

$F0$: Initial mutation operator; CR : Crossover operator

Q : Population; D : Dimension

Initialize the population and calculate the fitness of each population

while Termination criterion

Adaptive mutation operator $\lambda = \exp\left(1 - \frac{G}{G+1-t}\right)$, $F = F0 * 2^\lambda$

for $i = 1: Q$

Crossover operator: $V_i = X_{r1} + F \cdot (X_{r2} - X_{r3})$

Mutation operator: $u_{ij} = \begin{cases} v_{ij} & \text{if } rand < CR \text{ or } randi(1, D) = j \\ x_{ij} & \text{if } rand > CR \text{ or } randi(1, D) \neq j \end{cases}$

Selection operator: $x_i = \begin{cases} u_i & \text{if } fit(u_i) \text{ superior to } fit(x_i) \\ x_i & \text{else} \end{cases}$

end for

$t = t + 1$

end while

return the best solution

End

3.4. Population-Based Incremental Learning Algorithm

PBIL guides the evolution of the population by maintaining a probability vector. The algorithm selects the individual with the best fitness in each generation to update the probability vector, and the next generation population is generated by the updated probability vector sampling. Repeating these steps, the final optimal solution is obtained. The probability vector is the core part of a population-based incremental learning algorithm. According to Algorithm 3, the main operations related to the probability vector in the algorithm are described below.

Algorithm 3: Population-based Incremental Learning Algorithm.

Begin:

Q : population; LR : learning factor

p_m : mutation probability; MS : mutated offset value

Initialization probability vector P^t , $p_i^t = 0.5$, $i = 1, 2, \dots, L$

while Termination criterion

Generate population according to P^t sampling

Evaluate the fitness of each individual according to Equation (11)

Find the individual B^t with the best fitness in the population

Update P^t according to Equations (13) and (14)

$t = t + 1$

end while

return the best solution

End

Assuming that the optimal individual of the current generation is $B^t = (b_1^t, b_2^t, \dots, b_L^t)$, the probability vector P^t of the current generation is updated by the following Formula (13).

$$p_i^t = (1.0 - LR) \cdot p_i^{t-1} + LR \cdot b_i^t, \quad i = 1, 2, \dots, L \quad (13)$$

where LR indicates the learning factor, which is generally set to 0.01. For each p_i^t in the probability vector P^t , if $\text{random}(0, 1)$ is less than the mutation probability p_m , the vector value p_i^t is mutated; otherwise, it is not mutated. Formula is as follows in (14).

$$p_i^{t*} = (1.0 - MS) \cdot p_i^t + MS \cdot \text{random}(0 \text{ or } 1) \quad (14)$$

where p_i^{t*} represents the i th mutated vector value of P^t , and MS represents the mutated offset value.

3.5. Particle Swarm Optimization Algorithm

PSO is a stochastic search algorithm developed by simulating the foraging behavior of birds. The algorithm abstracts individual birds into massless, volumetric and informative particles. Since bird populations do not know the specific location of food at first, they can only get the information that food is within a certain range, so bird populations adopt a stable and simple method: search the area around the bird that is closest to the food.

Inspired by this, particle swarm optimization randomly initializes many particles in a solution space, and each particle is given random speed and position information. Then these particles begin to move or stay in place with the initialized speed. After sharing information with the surrounding particles, the approximate orientation of the optimal solution and the position information of the particles closest to the optimal solution is obtained, and the particle moves progressively towards the optimal solution position. The whole process is shown in Algorithm 4. The mathematical formula of particle velocity update mode is described as follows in (15).

$$v_i^t = \omega v_i^{t-1} + c_1 r_1 (pbest_i - x_i^{t-1}) + c_2 r_2 (gbest - x_i^{t-1}) \quad (15)$$

where c_1 and c_2 are the learning factors of individual extreme and global extreme, r_1 and r_2 are the influence disturbance factors of individual extreme and global extreme and $pbest_i$ and $gbest$ are the individual extremum and global extreme in the t th iteration.

Algorithm 4: Particle Swarm Optimization Algorithm.

Begin:

Initialize population of particles ($\{X_i\}$, $i \in [1, Q]$)

Evaluate fitness for each particle using Equation (11)

Set $pbest_i = X_i$, $gbest = \min\{pbest\}$

while Termination criterion

for $i = 1: Q$

 Update the velocity and position of X_i by Equation (15)

 Evaluate the fitness of X_i

if $\text{fit}(X_i) < \text{fit}(pbest_i)$

$pbest_i = X_i$

if $\text{fit}(pbest_i) < \text{fit}(gbest)$

$gbest = pbest_i$

end for

$t = t + 1$

end while

return the best solution

End

3.6. Ant Colony Optimization Algorithm

ACO was proposed by M. Dorigo firstly, and inspired by real ant colony behavior. The study found that individual ants communicate with each other through a pheromone that allows them to collaborate on complex tasks. Ants tend to move towards the directions with high pheromone concentrations during movement. They not only leave pheromones on the path they pass but also sense the presence and concentration of pheromones during

movement to guide their movement direction and search for food. The ant colony algorithm simulates such optimization mechanisms to find the optimal solution through information exchange and cooperation among individuals. The whole simulation process is shown in Algorithm 5. According to the pheromone quantity and heuristic information on each path, the transition probability of the ant at point i at time t to select the next movable point j is (16).

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{s \in allowed_k} [\tau_{is}]^\alpha \cdot [\eta_{is}]^\beta}, & j \in allowed_k \\ 0, & otherwise \end{cases} \quad (16)$$

where $allowed_k = \{n - tanu_k\}$ represents the set of points that ant k at point i is allowed to choose to go next. α represents the influence of pheromone remaining on the path on the subsequent ants' choice of the path, β represents the expected heuristic factor, τ_{ij} is the pheromone concentration and η_{ij} is the heuristic function.

3.7. Artificial Bee Colony Algorithm

ABC simulates the honey-collecting behavior of honeybees. The location of a nectar source represents the solution, and the pollen quantity of the nectar source represents the value of fitness function. All bees are separated into three groups: the employed bees, the onlooker bees and the explored bees. The number of employed bees is the same as the number of nectar sources, and the number of onlooker bees is half of the number of nectar sources. Firstly, the employed bees take charge of the initial search for nectar sources and collects and shares information. Then the onlooker bees is in charge of staying in the hive and collecting nectar according to the information provided by the employed bees; last, the explored bees are responsible for randomly searching for new nectar sources to replace original ones after original ones are abandoned. According to Algorithm 6, each stage is described as follows.

Algorithm 5: Ant Colony Optimization Algorithm.

Begin:

keep: elitism parameter; ρ : local pheromone decay rate

Q : Population; D : Dimension

Initialize the population and calculate the fitness of each population

while *Termination criterion*

for $i = 1: Q$

for $j = 1: Q$

 Use each solution to update the pheromone for each parameter value:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{i=1}^Q Q \Delta \tau_{ij}^k$$

end for

end for

for $k = keep + 1: Q$

 Use the probability Equation (16) to generate new solutions

end for

$t = t + 1$;

end

return the best solution

End

- (a) Employed bees' stage: employed bees use Formula (17) to find new nectar sources.

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (17)$$

where x_{kj} represents the domain nectar source, and k is not equal to i . ρ_{ij} is a random constant between -1 and 1 . After obtaining the new nectar source through the above

formula, the fitness function values of new and old nectar sources are compared by using the greedy algorithm, and the superior one is selected.

- (b) Onlooker bees' stage: at this stage, employed bees share nectar information in the dance area. Then onlooker bees analyze information and adopt a roulette strategy to select nectar source tracking mining to ensure that the probability of nectar source mining with higher fitness value is greater.
- (c) Explored bees' stage: if a nectar source has not been renewed after several mining sessions, the nectar source should be abandoned and the explored bees stage starts. The explored bees use Formula (18) to randomly search for new nectar sources to replace abandoned ones.

$$x_{ij} = x_{minj} + rand[0, 1](x_{maxj} - x_{minj}) \quad (18)$$

where x_{minj} and x_{maxj} represent the minimum and maximum values of the j th dimension.

Algorithm 6: Artificial Bee Colony Algorithm.

Begin:

Q : The number of nectar sources or the employed bees;

D : Dimension; O : The number of the onlooker bees

Initialize the nectar sources and calculate the fitness of each nectar source

while *Termination criterion*

for $i = 1: Q$

for $j = 1: D$

 Update the location of employed bees by Equation (17)

 Store the best nectar sources in a greedy way, and record if not updated

end for

end for

for $l = 1: O$

 Check whether there is nectar source stagnant update. If so, update through Equation (18)

end for

 Complete generation update

$t = t + 1$;

end

return the best solution

End

3.8. Whale Optimizaton Algorithm

Mirjalili proposed a WOA-inspired predation mode of humpback whales. Humpback whale individuals can identify and surround prey. According to Algorithm 7, the whole process can be abstracted into three stages: searching prey, surrounding prey and bubble net attack.

- (a) Surrounding prey: humpback whales can recognize prey and continuously reduce their surrounding range. The optimal solution represents target prey or location close to target prey, and other whales will keep approaching it. The entire mathematical model can be interpreted as follows in (19) and (20).

$$D = |C \cdot X^*(t) - X(t)| \quad (19)$$

$$X(t+1) = X^*(t) - A \cdot D \quad (20)$$

where $X^*(t)$ is the current optimal whale position vector, t is iterations, $X(t)$ is the current humpback whale position vector, $A \cdot D$ is bounding step length A and C represent different coefficient vectors.

- (b) Bubble net attack: when a humpback whale surrounds prey, it spits out bubbles in a spiral way to surround the prey. When $|A| \leq 1$, the Formula (21) simulates the spiral hunting behavior of the humpback whale.

$$X_{t+1} = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \quad (21)$$

where $D' = |X^*(t) - X(t)|$, D' is the distance between current humpback whale position and prey, and b is the constant that determines shape of spiral. l is a random constant in $[-1, 1]$.

- (c) Searching prey: when $|A| > 1$, whale individuals are forced to stay away from the optimal whale location of the current generation so that whale individuals randomly search for prey, which is no longer affected by the current optimal whale. Its mathematical model can be described as follows in (22) and (23).

$$D = |C \cdot X_{rand}(t) - X(t)| \quad (22)$$

$$X(t+1) = X_{rand}(t) - A \cdot D \quad (23)$$

where $X_{rand}(t)$ represents the random whale location in the current whale population.

Algorithm 7: Whale Optimization Algorithm.

Begin:

Initialize related parameters, Q : Population

Initialize the population and calculate the fitness of each population

while Termination criterion

for $i = 1: Q$

if ($p < 0.5$)

if ($|A| < 1$)

 Update the individual by Equation (19)

else if ($|A| \geq 1$)

 Select a random individual to update by Equation (23)

else if ($p \geq 0.5$)

 Update the individual by Equation (21)

end for

Calculate individual fitness and update the optimal solution

$t = t + 1$;

end while

return the best solution

End

3.9. Harris Hawks Optimization Algorithm

Inspired by the process of Harris eagle predation, Heidari proposes HHO. The Harris eagle is a group predator, and all are members of the division of labor and coordinated action. The exploration phase is a global search process in which eagles track and detect prey from the air. When the target prey is determined, all members of the group gradually approach the location of the prey, find a suitable position around the prey and complete the encircling for the final attack preparation. The whole simulation process is shown in Algorithm 8. According to the escape behavior of prey and the chase strategy of Harris eagle, Harris Hawks Optimization uses four different strategies to simulate the attack.

- (a) When $|E| \geq 0.5$ & $r \geq 0.5$, in this solution, prey has enough physical strength to try to escape by jumping but is eventually captured; the formula is as follows (24) and (25).

$$X(t+1) = DX(t) - E \cdot |J \cdot X_{Prey}(t) - X(t)| \quad (24)$$

$$DX(t) = X_{Prey}(t) - X(t) \quad (25)$$

where $DX(t)$ represents position deviation between eagle group and prey after the t th iteration, and j is the jumping distance during the escape of the prey. X_{prey} is the position of the prey.

- (b) When $|E| < 0.5$ & $r \geq 0.5$, in this solution, prey does not have sufficient energy and is directly captured by the eagle, the formula can be described as follows (26).

$$X(t+1) = X_{prey}(t) - E \cdot |DX(t)| \quad (26)$$

- (c) When $|E| \geq 0.5$ & $r < 0.5$, in this solution, the prey has plenty of energy to escape and has the opportunity to escape. Therefore, the eagles form a more intelligent encirclement. The implementation strategies are as follows (27) and (28).

$$Y = X_{prey}(t) - E \cdot |J \cdot X_{prey}(t) - X(t)| \quad (27)$$

$$X(t+1) = \begin{cases} Y, & F(Y) < F(X(t)) \\ Z, & F(Z) < F(X(t)) \end{cases} \quad (28)$$

When $|E| < 0.5$ & $r < 0.5$, in this solution, there is a prey energy shortage but still a chance to escape. Therefore, when the eagles form a hard encirclement to narrow average distance from their prey, the formula can be interpreted as follows in (29) and (30).

$$Y = X_{prey}(t) - E \cdot |J \cdot X_{prey}(t) - X(t)| \quad (29)$$

$$X(t+1) = \begin{cases} Y, & F(Y) < F(X(t)) \\ Z, & F(Z) < F(X(t)) \end{cases} \quad (30)$$

3.10. Chimp Optimization Algorithm

During chimpanzee hunting, any chimpanzee can randomly change its position in the space around its prey. The mathematical description is as follows in (31) and (32).

$$d = |cX_{prey}(t) - mX_{chimp}(t)| \quad (31)$$

$$X_{chimp}(t+1) = X_{prey}(t) - ad \quad (32)$$

where d is the distance between chimpanzees and prey; t is the current number of iterations; $X_{prey}(t)$ is the prey position vector; $X_{chimp}(t)$ is the chimpanzee position vector and a , m , and c are coefficient vectors.

In the chimpanzee community, according to the diversity of intelligence and ability shown by individuals in the process of hunting, chimpanzee groups are classified as “driver”, “barrier”, “chaser” and “attacker”. Each type of chimpanzee has its own ability to think independently and use its own search strategy to explore and predict the location of prey. While they have their own tasks, they also exhibit chaotic individual hunting behavior at the end of the hunt due to social incentives to obtain sexual behavior and benefits. The Chimp Optimization Algorithm solves the problem by simulating the co-hunting behavior of the four chimpanzee species. According to Algorithm 9, the formula for updating the positions of the four chimpanzee groups is described as follows (33)–(35).

$$\begin{cases} D_{Attacker} = |C_1 * X_{Attacker} - m * X| \\ D_{Barrier} = |C_2 * X_{Barrier} - m * X| \\ D_{Chaser} = |C_3 * X_{Chaser} - m * X| \\ D_{Driver} = |C_4 * X_{Driver} - m * X| \end{cases} \quad (33)$$

$$\begin{cases} X_1 = X_{Attacker} - A_1 * D_{Attacker} \\ X_2 = X_{Barrier} - A_2 * D_{Barrier} \\ X_3 = X_{Chaser} - A_3 * D_{Chaser} \\ X_4 = X_{Driver} - A_4 * D_{Driver} \end{cases} \quad (34)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3 + X_4}{4} \quad (35)$$

Algorithm 8: Harris Hawks Optimization Algorithm.

Begin:

Initialize related parameters, Q : Population

Initialize the population

while *Termination criterion*

Calculate the fitness of each population

Set X_{prey} as the best individual

for $i = 1: Q$

Update the energy E and jump strength J

if ($|E| \geq 1$)

Update by $X(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X(t)| & q \geq 0.5 \\ X_{prey}(t) - X_m(t) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases}$

if ($|E| < 1$)

if ($|E| \geq 0.5 \ \& \ r \geq 0.5$)

Update by formulae (24) and (25)

else if ($|E| < 0.5 \ \& \ r \geq 0.5$)

Update by Formula (26)

else if ($|E| \geq 0.5 \ \& \ r < 0.5$)

Update by formulae (27) and (28)

else if ($|E| < 0.5 \ \& \ r < 0.5$)

Update by formulae (29) and (30)

end for

$t = t + 1$

end while

return the best solution

End

3.11. Biogeography-Based Optimization Algorithm

BBO is an intelligence algorithm that uses biogeographic theory to solve optimization problems. In a biogeographic optimization algorithm, habitat is used to represent individuals in intelligent optimization algorithm, the suitable index variable (SIV) represents variables in individuals and the habitat suitability index (HSI) represents individual fitness. In nature, the habitat suitability index of each habitat for biological population is different. Habitats with high HSI can accommodate more species and have high species migration rates and low species migration rates. Individual migration can share excellent SIV between habitats. The migration models in biogeography-based optimization mainly include the linear migration model, trapezoidal migration model, secondary migration model and cosine migration model. The linear migration model is used in this paper. Formula (36) is the calculation of immigration rate $\lambda(S)$; Formula (37) is emigration rate $\mu(S)$. The entire migration operation is shown in Algorithm 10.

$$\mu(S) = E \times \frac{S}{S_{max}} \quad (36)$$

$$\lambda(S) = I \times \left(1 - \frac{S}{S_{max}}\right) \quad (37)$$

where E represents the probability of maximum emigration, I represents the maximum immigration probability.

Algorithm 9: Chimp Optimization Algorithm.**Begin:**Initialize f , m , a , and c ; Q : Population

Initialize the population and calculate the fitness of each chimp

 $X_{Attacker}$ represents the best search agent $X_{Barrier}$ represents the second best search agent X_{Chaser} represents the third best search agent X_{Driver} represents the fourth best search agent**while** Termination criterion**for** $i = 1: Q$ Update f , m , a and c according to different group strategies**if** ($\mu < 0.5$)**if** ($|a| < 1$)

Update the position by (32)

else if ($|a| \geq 1$)

Select a search agent randomly

else if ($\mu \geq 0.5$)Update the position by $X_{chimp}(t+1) = \begin{cases} X_{prey}(t) - a.d & \text{if } \mu < 0.5 \\ Chaotic_value & \text{if } \mu \geq 0.5 \end{cases}$ **end for**Update $X_{Attacker}, X_{Barrier}, X_{Chaser}, X_{Driver}$ $t = t + 1$ **end while****return** the best solution**End**

Mutation operation simulates the phenomenon that diseases, natural disasters and other factors change the living environment of habitat and lead to the deviation of habitat population from the equilibrium point. According to Algorithm 11, the mutation probability M_s of the species is calculated as follows (38).

$$M_s = M_{max} \times \left(1 - \frac{P_s}{P_{Smax}}\right) \quad (38)$$

where P_s is probability of S species in the habitat; P_{Smax} is the maximum value of P_s , and M_{max} represent the maximum mutation rate.

The above introduces how the intelligent optimization algorithm optimizes the weight and threshold of the dendritic neural network. At the same time, the principle, characteristics, important formulas and pseudocode of each algorithm are introduced in detail. Each algorithm has its own characteristics, which also determines the adaptability of the algorithm.

Algorithm 10: Migration Operation(BBO).**For** $d = 1: D$ **if** $\text{rand}() < \lambda_i$ select another habitat h from the population with the migration probability μ_j $H_i(d) \leftarrow H_j(d)$ **end if****End for****Algorithm 11:** Mutation Operation(BBO).**For** $d = 1: D$ **if** $\text{rand}() < \pi_i$ $H_i(d) \leftarrow l_d + \text{rand}() \cdot (\mu_d - l_d)$ end if**end if****End for**

4. Experiment

For the classification problem of DNM, the experiment compared the results of ten intelligent optimization algorithms and traditional backpropagation algorithms on six classical classification datasets. The software environment of the experiment is MATLAB2018a, and the hardware environment is Intel(R) Core(TM) I5-9500 CPU @ 3.00 GHz, 8.00GB.

All datasets are from the UCI database [54]. Table 1 shows the attributes, number and classes of data in detail. All six data sets are binary classification problems. All datasets are randomly divided into two groups: 70% training and 30% testing. These six datasets contain common classification problems in the real world. For example, the Australian Credit Approval dataset and Banknote Authentication dataset are related to finance, and the Bank Credit Approval dataset concerns credit card applications and is used to evaluate the credit of the new applicant, which can effectively identify the good or bad of the customer, provide the basis for issuing the card and help the bank to establish the first line of defense against credit card risk. The Banknote Authentication dataset is to classify the data extracted from the images of real and counterfeit similar banknote samples. The Breast Cancer dataset and Diabetic Retinopathy dataset are classified datasets about common breast cancer and diabetes. By classifying and predicting different influence attributes of diseases, it has a positive effect on medical treatment activities. The last two data are on actual production life; the Car Evaluation dataset is a classified data set to assess car safety, and the Glass Identification dataset is a classified data set to classify the type of glass.

Table 1. Details of the dataset.

Classification Datasets	Attributes	Samples	Classes
Australian Credit Approval	15	690	2
Banknote authentication	5	1372	2
Breast cancer	10	699	2
Car Evaluation	7	1728	2
Diabetic Retinopathy	17	520	2
Glass Identification	10	214	2

To eliminate interference, all experimental results are the average results of 30 independent experiments. The experiment compares the performance of eleven algorithms, including ten intelligent algorithms (GA, DE, PBIL, PSO, ACO, ABC, WOA, HHO, ChOA and BBO) and a traditional back propagation algorithm. The population size of ten intelligent optimization algorithms is 50, and the number of iterations is 250. Other parameters intelligent optimization algorithms of are set according to the experience and characteristics of the algorithm. The initialization parameters of intelligent optimization algorithm are shown in Table 2. The learning rate of the backpropagation algorithm is 0.01, and the number of iterations is 250.

There are four user-defined parameters in DNM, which are the number of synaptic layers M , k of synaptic layer, k_{soma} and θ_{soma} in the activation function of the output layer. The experiment uses Taguchi's method [55] to obtain a reasonable combination of four parameters of DNM. Taguchi's method adopts the orthogonal experimental design method, which is a scientific experimental design method to select appropriate and representative points from a large number of experimental points. The four factors of the number of levels are set as follows: $M \in \{1, 5, 10, 15, 20\}$, $k \in \{1, 5, 10, 15, 25\}$, $k_{soma} \in \{1, 5, 10, 15, 25\}$, $\theta_{soma} \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Full-factor analysis contains $5^4 = 625$ experiments, orthogonal array $L_{25}(5^4)$ only contains 25 experiments, it greatly reduces the number of test runs, cost time, manpower, and materials. Table 3 shows the optimal parameters of six datasets obtained through the test of orthogonal array $L_{25}(5^4)$. The orthogonal matrix experimental results of eleven algorithms in six datasets are recorded in detail in the Appendix A.

Table 2. Initialize parameter Settings of intelligent algorithms.

Algorithm	Parameter	Value
GA	Selection mechanism	Roulette wheel
	Crossover probability	0.9
	Mutation probability	0.1
DE	Crossover probability	0.9
	Differential weight	0.5
PBIL	Learning rate	0.05
	Good population member	1
	Bad population member	0
	Elitism parameter	1
	Mutational probability	0.1
PSO	Acceleration constants	[2,2]
	Inertia weights	[0.9,0.5]
ACO	Initial pheromone	1e-6
	Pheromone update constant	20
	Exploration constant	1
	Global pheromone decay rate	0.9
	Local pheromone decay rate	0.5
ABC	Number of employed bee	50
	Number of onlooker bee	25
WOA	Convergence factor a	$a = 2 - \frac{2 \cdot t}{t_{\max}}$
	Coefficient vectors A	$A = 2a \cdot \text{rand}(0,1) - a$
HHO	Initial escape energy	inf
ChOA	Initial attacker_score	inf
	Initial barrier_score	inf
	Initial chaser_score	inf
	Initial driver_score	inf
BBO	Habitat modification probability	1
	Mutation probability	0.005
	bound for immigration probability per gene	[0,1]

In the process of model training optimization, MSE is the loss function of the model, which is the evaluated fitness value of each solution. The accuracy of the classification is to evaluate the performance of the model during the test. Formula (11) is the calculation of MSE, and the calculation method of accuracy is as follows (39).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \quad (39)$$

where TP , TN , FP , and FN repent true positives, true negatives, false positives, and false negatives respectively.

Table 3. Optimal parameters of the six datasets in the dendrite neural model.

Classification Datasets	M	k	k_{soma}	θ_{soma}
Australian Credit Approval	10	1	10	0.5
Banknote Authentication	10	1	10	0.5
Breast Cancer	10	1	10	0.5
Car Evaluation	20	1	25	0.9
Diabetic Retinopathy	10	1	10	0.5
Glass Identification	10	1	10	0.5

5. Result

Figure 3 shows the convergence diagram of the algorithm in the iterative process of optimizing the weights and thresholds of the model, and Figure 4 shows the boxplot of the algorithm after iterative optimization, which can intuitively see the distribution of the optimal solution obtained by the algorithm. At this time, the user-defined parameters of the model adopt the optimal parameters in Table 3. Table 4 summarizes the classification accuracy and minimum mean squared error of each algorithm under the optimized parameters.

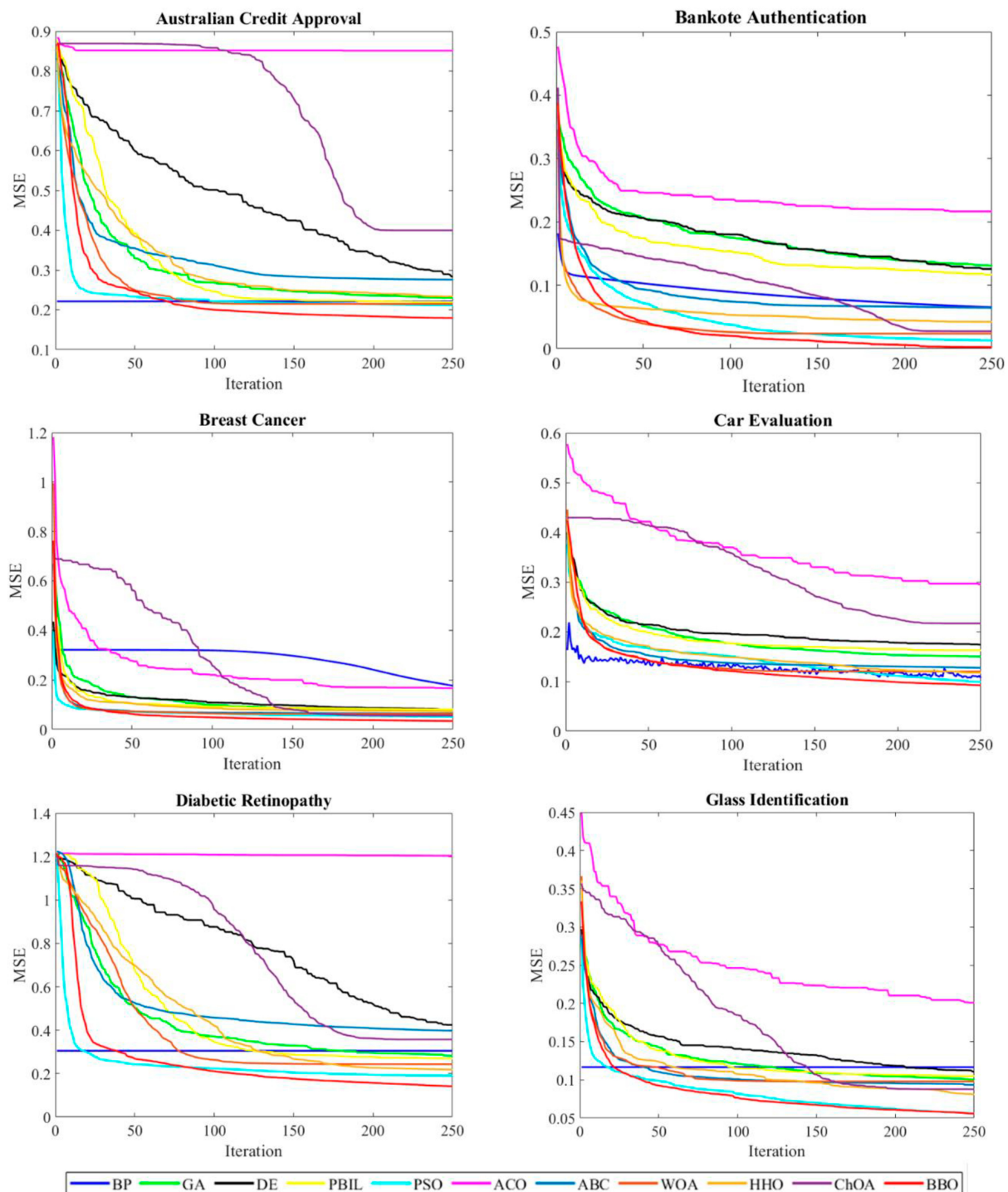


Figure 3. Iterative convergence graphs of eleven algorithms for six datasets.

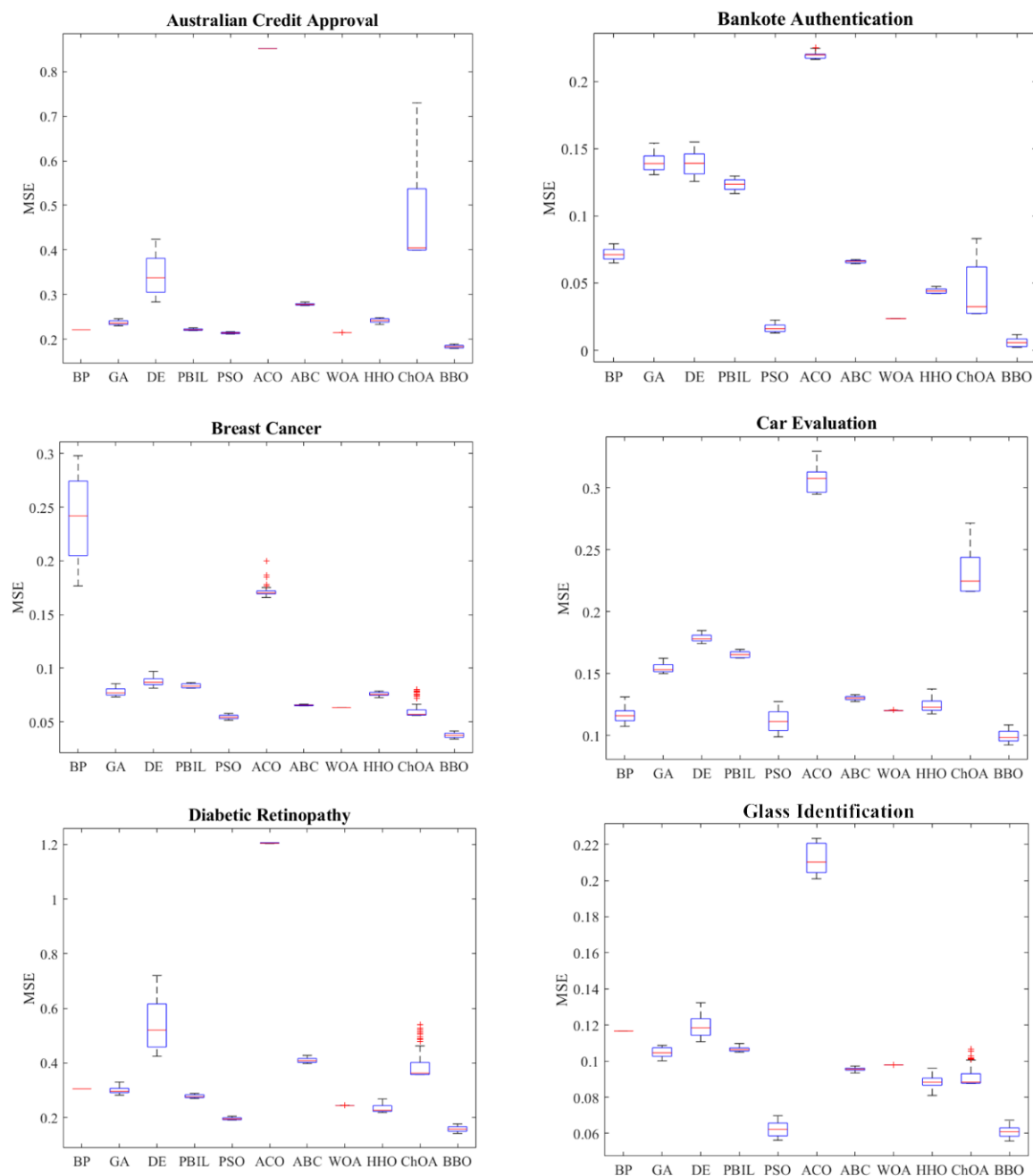


Figure 4. Box plots of eleven algorithms for six datasets.

As we can see, the BP algorithm converges very slowly in the iterative process. In Australian Credit Approval dataset, Diabetic Retinopathy dataset, and Glass Identification dataset, the algorithm falls into local optimization and loses its function. Its robustness is very poor. ACO and ChOA also have slow convergence speed. In the Australian Credit Approval dataset and Diabetic Retinopathy dataset, it is the same as the BP algorithm, which falls into local traps and has poor overall performance. In the six datasets, BBO has the best convergence effect, followed by PSO. In the first 50 iterations of Australian Credit Approval dataset and Diabetic Retinopathy dataset, the convergence speed of PSO is better than that of BBO, but after 50 iterations, BBO's convergence value is lower than that of PSO, indicating that BBO has better global search ability than PSO. Unlike the DE algorithm, which has a good effect on some datasets (Breast Cancer, Car Evaluation) and

poor effect on some datasets (Australian Credit Approval, Diabetic Retinopathy), BBO has good robustness.

From the distribution of optimal solutions, DNM + BBO still has obvious advantages, with smaller error and stable performance, followed by is DNM + PSO. DNM + ACO and DNM + ChOA have many outliers, which means that the algorithm has poor performance and cannot find a good feasible solution.

As can be seen from Table 4, the classification accuracy of BP in the six datasets is very low, where in the Diabetic Retinopathy dataset it is as low as 39% and the highest at 86.5%. BBO has the highest classification accuracy in all six datasets, reaching 99.5% in the Banknote Authentication dataset. PSO is inferior to BBO. The performance of ACO and ChOA is unstable. For example, in the Diabetic Retinopathy dataset, ACO is as low as 38.7%, and ChOA is as low as 69.6%. Other data sets such as WOA, GA, PBIL and so on have good performance, but the overall performance is worse than BBO. In addition, MSE is the loss function value of the model, and BBO is the smallest in six data sets, which shows that BBO has good performance.

Among the ten intelligent optimization algorithms, DNM + BBO has the fastest convergence speed and the highest accuracy, followed by DNM + PSO. Otherwise DNM + WOA, DNM + GA and so on have good convergence speeds and higher classification, but they are still lower than BBO as a whole. Meanwhile, just like the ideal of “no free lunch”, not all intelligent optimization algorithms are suitable for the classification of dendritic neural models. For example, DNM + ACO and DNM + ChOA have poor performance.

In contrast, intelligent optimization algorithms have obvious advantages, the performance of the intelligent dendrite neural model is far better than that of the traditional dendrite neural model.

Table 4. Optimal parameters of the six datasets in the dendrite neural model.

Learning Algorithm	Datasets	M	k	k_{soma}	θ_{soma}	Average Accuracy (%) $\pm$ Standard Deviation	MSE
BP	Australian Credit Approval	10	1	10	0.5	56.34 ± 2.97	2.21×10^{-1}
	Banknote Authentication	10	1	10	0.5	86.50 ± 2.78	6.51×10^{-2}
	Breast Cancer	10	1	10	0.5	58.13 ± 25.88	1.77×10^{-1}
	Car Evaluation	20	1	25	0.9	75.08 ± 8.89	1.07×10^{-1}
	Diabetic Retinopathy	10	1	10	0.5	39.08 ± 2.91	3.05×10^{-1}
	Glass Identification	10	1	10	0.5	75.88 ± 4.92	1.17×10^{-1}
GA	Australian Credit Approval	10	1	10	0.5	84.21 ± 3.30	2.30×10^{-1}
	Banknote Authentication	10	1	10	0.5	91.62 ± 2.45	1.31×10^{-1}
	Breast Cancer	10	1	10	0.5	94.98 ± 1.62	7.34×10^{-2}
	Car Evaluation	20	1	25	0.9	89.34 ± 1.64	1.50×10^{-1}
	Diabetic Retinopathy	10	1	10	0.5	80.66 ± 4.59	2.82×10^{-1}
	Glass Identification	10	1	10	0.5	91.98 ± 2.92	1.00×10^{-1}
DE	Australian Credit Approval	10	1	10	0.5	82.64 ± 4.09	2.84×10^{-1}
	Banknote Authentication	10	1	10	0.5	91.50 ± 2.25	1.26×10^{-1}
	Breast Cancer	10	1	10	0.5	94.00 ± 1.33	8.15×10^{-2}
	Car Evaluation	20	1	25	0.9	86.63 ± 2.10	1.74×10^{-1}
	Diabetic Retinopathy	10	1	10	0.5	69.06 ± 9.17	4.24×10^{-1}
	Glass Identification	10	1	10	0.5	92.34 ± 3.61	1.11×10^{-1}
PBIL	Australian Credit Approval	10	1	10	0.5	83.11 ± 2.00	2.20×10^{-1}
	Banknote Authentication	10	1	10	0.5	92.67 ± 2.00	1.17×10^{-1}
	Breast Cancer	10	1	10	0.5	94.35 ± 1.50	8.15×10^{-2}
	Car Evaluation	20	1	25	0.9	87.71 ± 1.72	1.63×10^{-1}
	Diabetic Retinopathy	10	1	10	0.5	81.54 ± 3.27	2.69×10^{-1}
	Glass Identification	10	1	10	0.5	90.26 ± 2.98	1.05×10^{-1}

Table 4. Cont.

Learning Algorithm	Datasets	M	k	k_{soma}	θ_{soma}	Average Accuracy (%) $\pm$ Standard Deviation	MSE
PSO	Australian Credit Approval	10	1	10	0.5	84.34 ± 3.51	2.12×10^{-1}
	Banknote Authentication	10	1	10	0.5	98.45 ± 0.89	1.28×10^{-2}
	Breast Cancer	10	1	10	0.5	95.06 ± 1.25	5.16×10^{-2}
	Car Evaluation	20	1	25	0.9	92.94 ± 2.44	9.89×10^{-2}
	Diabetic Retinopathy	10	1	10	0.5	87.16 ± 2.67	1.90×10^{-1}
	Glass Identification	10	1	10	0.5	93.00 ± 2.94	5.62×10^{-2}
ACO	Australian Credit Approval	10	1	10	0.5	56.10 ± 3.80	8.51×10^{-1}
	Banknote Authentication	10	1	10	0.5	88.10 ± 1.31	2.17×10^{-1}
	Breast cancer	10	1	10	0.5	90.05 ± 2.75	1.66×10^{-1}
	Car evaluation	20	1	25	0.9	82.28 ± 4.68	2.95×10^{-1}
	Diabetic retinopathy	10	1	10	0.5	38.70 ± 2.93	1.20×10^0
	Glass Identification	10	1	10	0.5	88.07 ± 3.88	2.01×10^{-1}
ABC	Australian Credit Approval	10	1	10	0.5	80.34 ± 6.40	2.76×10^{-1}
	Banknote Authentication	10	1	10	0.5	93.23 ± 6.25	6.44×10^{-2}
	Breast Cancer	10	1	10	0.5	89.97 ± 15.17	6.50×10^{-2}
	Car Evaluation	20	1	25	0.9	87.57 ± 6.31	1.27×10^{-1}
	Diabetic Retinopathy	10	1	10	0.5	74.29 ± 10.42	3.97×10^{-1}
	Glass Identification	10	1	10	0.5	91.72 ± 3.48	9.34×10^{-2}
WOA	Australian Credit Approval	10	1	10	0.5	83.96 ± 3.34	2.15×10^{-1}
	Banknote Authentication	10	1	10	0.5	98.67 ± 0.80	2.35×10^{-2}
	Breast Cancer	10	1	10	0.5	95.08 ± 1.35	6.33×10^{-2}
	Car Evaluation	20	1	25	0.9	91.22 ± 2.57	1.20×10^{-1}
	Diabetic Retinopathy	10	1	10	0.5	82.82 ± 6.25	2.43×10^{-1}
	Glass Identification	10	1	10	0.5	92.50 ± 2.61	9.79×10^{-2}
HHO	Australian Credit Approval	10	1	10	0.5	83.17 ± 5.35	2.34×10^{-1}
	Banknote Authentication	10	1	10	0.5	97.53 ± 1.13	4.23×10^{-2}
	Breast Cancer	10	1	10	0.5	94.43 ± 2.78	7.27×10^{-2}
	Car Evaluation	20	1	25	0.9	91.47 ± 2.14	1.18×10^{-1}
	Diabetic Retinopathy	10	1	10	0.5	85.15 ± 5.34	2.18×10^{-1}
	Glass Identification	10	1	10	0.5	91.67 ± 3.39	8.10×10^{-2}
ChOA	Australian Credit Approval	10	1	10	0.5	71.40 ± 10.97	4.00×10^{-1}
	Banknote Authentication	10	1	10	0.5	98.73 ± 0.71	2.73×10^{-2}
	Breast Cancer	10	1	10	0.5	95.71 ± 1.39	5.62×10^{-2}
	Car Evaluation	20	1	25	0.9	84.43 ± 7.46	2.16×10^{-1}
	Diabetic Retinopathy	10	1	10	0.5	69.64 ± 9.35	3.57×10^{-1}
	Glass Identification	10	1	10	0.5	93.05 ± 3.53	8.77×10^{-2}
BBO	Australian Credit Approval	10	1	10	0.5	84.38 ± 1.91	1.79×10^{-1}
	Banknote Authentication	10	1	10	0.5	99.50 ± 0.37	2.28×10^{-3}
	Breast Cancer	10	1	10	0.5	95.94 ± 1.15	3.42×10^{-2}
	Car Evaluation	20	1	25	0.9	93.32 ± 1.98	9.23×10^{-2}
	Diabetic Retinopathy	10	1	10	0.5	89.91 ± 3.19	1.41×10^{-1}
	Glass Identification	10	1	10	0.5	93.07 ± 3.30	5.57×10^{-2}

6. Conclusions

In this paper, an intelligent dendritic neural model is proposed for the first time, which uses intelligent optimization algorithm instead of the traditional BP algorithm to train the model. In the experiment, ten intelligent optimization algorithms including GA, DE, PBIL, PSO, ACO, ABC, WOA, HHO, ChOA, BBO, and traditional BP algorithm were selected to train and test the model on six datasets (Australian Credit Approval, Banknote Authentication, Breast Cancer, Car Evaluation, Diabetic Retinopathy, Glass Identification). These ten algorithms are representative intelligent algorithms, such as GA, DE and BBO

based on evolutionary ideas, PBIL based on mathematical statistics, ABC, ACO, PSO, and WOA based on swarm intelligence, and new algorithms HHO, ChOA. The experiment uses Taguchi's method to obtain a reasonable combination of four parameters of DNM. The experiments compare and analyze effectiveness, convergence speed, and classification accuracy of the algorithm. The experimental results show that the intelligent dendritic neural model (DNM-BBO) is obviously superior to the traditional dendritic neural model. At the same time, through the comparison of intelligent optimization algorithms, the result shows that BBO algorithm has excellent performance, and its robustness, accuracy and convergence speed are the best. The intelligent dendritic neural model established in this study is a powerful tool for solving classification problems and provides more choices in practical engineering applications.

Author Contributions: Conceptualization, D.J., W.X.; methodology, D.J., W.X.; software, W.X.; validation, W.X., D.J., Z.Z., C.L. and Z.X.; formal analysis, D.J.; investigation, W.X.; resources, W.X.; data curation, W.X.; writing—original draft preparation, W.X.; writing—review and editing, D.J., W.X., Z.Z., C.L. and Z.X.; visualization, W.X.; supervision, D.J., Z.Z. and C.L.; project administration, D.J.; funding acquisition, D.J. and Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (NSFC) under Grants 12105120 and 72174079, the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 19KJB160001, the Lianyungang City Haiyan Project under Grant 2019-QD-004, the MOE Key Laboratory of TianQin Project, Sun Yat-sen University, and the Open Fund Project of the Jiangsu Institute of Marine Resources Development under Grant JSIMR202018.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Six datasets of this paper are obtained from the following website: <https://archive.ics.uci.edu/ml/index.php> accessed on 22 October 2021.

Acknowledgments: The authors would like to thank all collaborators for their time and all reviewers for their valuable suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The appendix shows results of eleven learning algorithms (BP, GA, DE, PBIL, PSO, ACO, ABC, WOA, HHO, ChOA and BBO) in Taguchi's method on six datasets. Tables A1–A6 describe the parameter sensitivity results of each learning algorithm based on the orthogonal array for each dataset. Tables A7–A12 shows the complete parameter combination in each dataset and the average classification accuracy of 30 independent experiments of each learning algorithm.

Table A1. Parameter sensitivity results of Australian Credit Approval dataset based on orthogonal arrangement.

No.	<i>M</i>	<i>k</i>	<i>k_{soma}</i>	<i>θ_{soma}</i>	MSE										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	1.24×10^{-1}	4.01×10^{-1}	4.05×10^{-1}	3.95×10^{-1}	4.07×10^{-1}	4.95×10^{-1}	4.16×10^{-1}	4.09×10^{-1}	3.93×10^{-1}	4.88×10^{-1}	3.93×10^{-1}
2	1	5	10	0.9	2.14×10^{-1}	2.90×10^{-1}	3.09×10^{-1}	2.66×10^{-1}	6.79×10^{-1}	8.87×10^{-1}	3.88×10^{-1}	2.75×10^{-1}	2.38×10^{-1}	7.59×10^{-1}	2.63×10^{-1}
3	1	10	15	0.3	2.15×10^{-1}	3.00×10^{-1}	3.83×10^{-1}	2.87×10^{-1}	7.94×10^{-1}	8.71×10^{-1}	3.45×10^{-1}	2.36×10^{-1}	2.43×10^{-1}	7.99×10^{-1}	3.37×10^{-1}
4	1	15	25	0.5	2.23×10^{-1}	3.41×10^{-1}	4.27×10^{-1}	3.62×10^{-1}	8.11×10^{-1}	8.91×10^{-1}	4.14×10^{-1}	2.53×10^{-1}	2.60×10^{-1}	8.39×10^{-1}	4.36×10^{-1}
5	1	25	5	0.7	2.07×10^{-1}	2.70×10^{-1}	4.35×10^{-1}	3.52×10^{-1}	7.59×10^{-1}	8.31×10^{-1}	4.33×10^{-1}	2.44×10^{-1}	2.68×10^{-1}	7.65×10^{-1}	3.98×10^{-1}
6	5	1	5	0.3	1.24×10^{-1}	2.35×10^{-1}	2.52×10^{-1}	2.24×10^{-1}	2.27×10^{-1}	6.27×10^{-1}	2.64×10^{-1}	2.31×10^{-1}	2.17×10^{-1}	4.73×10^{-1}	2.07×10^{-1}
7	5	5	25	0.7	2.12×10^{-1}	2.62×10^{-1}	3.24×10^{-1}	2.85×10^{-1}	3.99×10^{-1}	8.72×10^{-1}	3.87×10^{-1}	2.84×10^{-1}	2.63×10^{-1}	7.48×10^{-1}	2.78×10^{-1}
8	5	10	1	0.5	1.26×10^{-1}	3.53×10^{-1}	3.66×10^{-1}	3.56×10^{-1}	4.00×10^{-1}	5.01×10^{-1}	3.80×10^{-1}	3.45×10^{-1}	3.42×10^{-1}	4.95×10^{-1}	3.47×10^{-1}
9	5	15	15	0.9	2.23×10^{-1}	2.46×10^{-1}	3.17×10^{-1}	2.46×10^{-1}	4.58×10^{-1}	8.87×10^{-1}	3.42×10^{-1}	2.41×10^{-1}	2.36×10^{-1}	7.56×10^{-1}	2.38×10^{-1}
10	5	25	10	0.1	1.39×10^{-1}	2.81×10^{-1}	3.16×10^{-1}	2.56×10^{-1}	4.34×10^{-1}	5.56×10^{-1}	3.20×10^{-1}	2.52×10^{-1}	2.84×10^{-1}	5.50×10^{-1}	2.89×10^{-1}
11	10	1	10	0.5	2.21×10^{-1}	2.30×10^{-1}	2.84×10^{-1}	2.20×10^{-1}	2.12×10^{-1}	8.51×10^{-1}	2.76×10^{-1}	2.15×10^{-1}	2.34×10^{-1}	4.00×10^{-1}	1.79×10^{-1}
12	10	5	15	0.1	1.25×10^{-1}	2.48×10^{-1}	2.86×10^{-1}	2.50×10^{-1}	2.30×10^{-1}	6.27×10^{-1}	2.78×10^{-1}	2.13×10^{-1}	2.32×10^{-1}	5.54×10^{-1}	2.29×10^{-1}
13	10	10	5	0.9	2.19×10^{-1}	2.69×10^{-1}	2.98×10^{-1}	2.67×10^{-1}	3.26×10^{-1}	8.66×10^{-1}	3.66×10^{-1}	2.91×10^{-1}	2.61×10^{-1}	6.49×10^{-1}	2.27×10^{-1}
14	10	15	1	0.7	1.30×10^{-1}	3.42×10^{-1}	3.65×10^{-1}	3.40×10^{-1}	3.74×10^{-1}	5.20×10^{-1}	3.76×10^{-1}	3.26×10^{-1}	3.20×10^{-1}	4.97×10^{-1}	3.12×10^{-1}
15	10	25	25	0.3	2.24×10^{-1}	2.85×10^{-1}	3.26×10^{-1}	2.61×10^{-1}	3.96×10^{-1}	8.74×10^{-1}	2.97×10^{-1}	2.42×10^{-1}	2.96×10^{-1}	7.93×10^{-1}	3.04×10^{-1}
16	15	1	15	0.7	2.22×10^{-1}	2.34×10^{-1}	3.43×10^{-1}	2.29×10^{-1}	2.67×10^{-1}	8.82×10^{-1}	3.51×10^{-1}	2.39×10^{-1}	2.22×10^{-1}	6.12×10^{-1}	1.89×10^{-1}
17	15	5	5	0.5	1.45×10^{-1}	2.34×10^{-1}	2.68×10^{-1}	2.18×10^{-1}	2.52×10^{-1}	7.67×10^{-1}	2.88×10^{-1}	2.22×10^{-1}	2.44×10^{-1}	5.38×10^{-1}	2.01×10^{-1}
18	15	10	25	0.1	1.83×10^{-1}	2.59×10^{-1}	2.97×10^{-1}	2.44×10^{-1}	2.44×10^{-1}	7.64×10^{-1}	2.81×10^{-1}	2.03×10^{-1}	2.48×10^{-1}	6.35×10^{-1}	2.38×10^{-1}
19	15	15	10	0.3	2.04×10^{-1}	2.72×10^{-1}	3.04×10^{-1}	2.27×10^{-1}	3.09×10^{-1}	8.02×10^{-1}	2.72×10^{-1}	2.34×10^{-1}	2.57×10^{-1}	6.67×10^{-1}	2.31×10^{-1}
20	15	25	1	0.9	1.36×10^{-1}	3.41×10^{-1}	3.72×10^{-1}	3.51×10^{-1}	3.76×10^{-1}	5.40×10^{-1}	3.76×10^{-1}	3.37×10^{-1}	3.15×10^{-1}	5.05×10^{-1}	3.10×10^{-1}
21	20	1	25	0.9	2.22×10^{-1}	2.33×10^{-1}	3.92×10^{-1}	2.25×10^{-1}	2.36×10^{-1}	8.83×10^{-1}	3.85×10^{-1}	3.97×10^{-1}	2.13×10^{-1}	6.51×10^{-1}	1.88×10^{-1}
22	20	5	1	0.3	1.18×10^{-1}	3.57×10^{-1}	3.79×10^{-1}	3.67×10^{-1}	3.58×10^{-1}	4.92×10^{-1}	3.86×10^{-1}	3.59×10^{-1}	3.46×10^{-1}	4.69×10^{-1}	3.24×10^{-1}
23	20	10	10	0.7	2.21×10^{-1}	2.48×10^{-1}	2.88×10^{-1}	2.66×10^{-1}	2.63×10^{-1}	8.82×10^{-1}	3.53×10^{-1}	2.56×10^{-1}	2.46×10^{-1}	6.40×10^{-1}	2.43×10^{-1}
24	20	15	5	0.1	1.24×10^{-1}	3.17×10^{-1}	3.27×10^{-1}	3.02×10^{-1}	3.33×10^{-1}	5.03×10^{-1}	3.20×10^{-1}	3.00×10^{-1}	3.07×10^{-1}	4.69×10^{-1}	3.03×10^{-1}
25	20	25	15	0.5	2.21×10^{-1}	2.74×10^{-1}	3.26×10^{-1}	2.34×10^{-1}	3.17×10^{-1}	8.85×10^{-1}	3.22×10^{-1}	2.45×10^{-1}	2.81×10^{-1}	7.27×10^{-1}	2.59×10^{-1}

Table A2. Parameter sensitivity results of Banknote Authentication dataset based on orthogonal arrangement.

No.	<i>M</i>	<i>k</i>	<i>k_{soma}</i>	<i>θ_{soma}</i>	MSE										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	1.23×10^{-1}	3.73×10^{-1}	3.72×10^{-1}	3.73×10^{-1}	3.68×10^{-1}	3.94×10^{-1}	3.71×10^{-1}	3.76×10^{-1}	3.72×10^{-1}	3.72×10^{-1}	3.72×10^{-1}
2	1	5	10	0.9	1.13×10^{-1}	2.10×10^{-1}	1.96×10^{-1}	2.16×10^{-1}	2.00×10^{-1}	3.16×10^{-1}	1.96×10^{-1}	1.86×10^{-1}	1.89×10^{-1}	1.87×10^{-1}	2.03×10^{-1}
3	1	10	15	0.3	2.04×10^{-1}	2.07×10^{-1}	1.90×10^{-1}	2.06×10^{-1}	1.93×10^{-1}	3.17×10^{-1}	1.78×10^{-1}	1.21×10^{-1}	1.28×10^{-1}	1.75×10^{-1}	1.91×10^{-1}
4	1	15	25	0.5	2.21×10^{-1}	2.15×10^{-1}	1.96×10^{-1}	2.12×10^{-1}	1.94×10^{-1}	3.16×10^{-1}	1.90×10^{-1}	1.56×10^{-1}	1.58×10^{-1}	1.91×10^{-1}	2.04×10^{-1}
5	1	25	5	0.7	2.08×10^{-1}	1.97×10^{-1}	1.85×10^{-1}	1.98×10^{-1}	1.84×10^{-1}	2.91×10^{-1}	1.85×10^{-1}	1.74×10^{-1}	1.74×10^{-1}	1.81×10^{-1}	1.90×10^{-1}
6	5	1	5	0.3	1.23×10^{-1}	1.35×10^{-1}	1.28×10^{-1}	1.26×10^{-1}	5.39×10^{-2}	2.32×10^{-1}	1.10×10^{-1}	8.27×10^{-2}	7.37×10^{-2}	8.64×10^{-2}	5.14×10^{-2}
7	5	5	25	0.7	2.18×10^{-1}	1.53×10^{-1}	1.52×10^{-1}	1.32×10^{-1}	3.87×10^{-2}	3.03×10^{-1}	1.25×10^{-1}	4.93×10^{-2}	6.85×10^{-2}	5.00×10^{-2}	1.49×10^{-2}
8	5	10	1	0.5	1.10×10^{-1}	2.60×10^{-1}	2.78×10^{-1}	2.62×10^{-1}	2.17×10^{-1}	3.44×10^{-1}	2.65×10^{-1}	2.24×10^{-1}	2.30×10^{-1}	2.18×10^{-1}	2.02×10^{-1}
9	5	15	15	0.9	2.09×10^{-1}	1.53×10^{-1}	1.52×10^{-1}	1.28×10^{-1}	4.33×10^{-2}	2.71×10^{-1}	1.17×10^{-1}	5.98×10^{-2}	7.74×10^{-2}	4.49×10^{-2}	2.60×10^{-2}
10	5	25	10	0.1	2.09×10^{-1}	1.87×10^{-1}	1.86×10^{-1}	1.67×10^{-1}	1.11×10^{-1}	2.76×10^{-1}	1.63×10^{-1}	1.34×10^{-1}	1.42×10^{-1}	1.29×10^{-1}	9.03×10^{-2}
11	10	1	10	0.5	6.51×10^{-2}	1.31×10^{-1}	1.26×10^{-1}	1.17×10^{-1}	1.28×10^{-2}	2.17×10^{-1}	6.44×10^{-2}	2.35×10^{-2}	4.23×10^{-2}	2.73×10^{-2}	2.28×10^{-3}
12	10	5	15	0.1	7.29×10^{-2}	1.56×10^{-1}	1.64×10^{-1}	1.51×10^{-1}	4.99×10^{-2}	2.17×10^{-1}	1.12×10^{-1}	8.24×10^{-2}	8.93×10^{-2}	8.55×10^{-2}	4.06×10^{-2}
13	10	10	5	0.9	1.90×10^{-1}	1.54×10^{-1}	1.55×10^{-1}	1.30×10^{-1}	3.58×10^{-2}	2.63×10^{-1}	1.12×10^{-1}	4.94×10^{-2}	6.15×10^{-2}	3.96×10^{-2}	1.18×10^{-2}
14	10	15	1	0.7	1.27×10^{-1}	2.37×10^{-1}	2.49×10^{-1}	2.34×10^{-1}	1.63×10^{-1}	3.18×10^{-1}	2.21×10^{-1}	1.83×10^{-1}	1.87×10^{-1}	1.73×10^{-1}	1.45×10^{-1}
15	10	25	25	0.3	2.44×10^{-1}	1.63×10^{-1}	1.59×10^{-1}	1.39×10^{-1}	1.60×10^{-2}	2.43×10^{-1}	1.13×10^{-1}	5.58×10^{-2}	5.92×10^{-2}	5.15×10^{-2}	1.12×10^{-2}
16	15	1	15	0.7	5.07×10^{-2}	1.42×10^{-1}	1.49×10^{-1}	1.22×10^{-1}	1.03×10^{-2}	2.19×10^{-1}	7.49×10^{-2}	1.67×10^{-2}	3.81×10^{-2}	2.10×10^{-2}	2.58×10^{-3}
17	15	5	5	0.5	8.13×10^{-2}	1.51×10^{-1}	1.61×10^{-1}	1.35×10^{-1}	1.81×10^{-2}	2.16×10^{-1}	1.16×10^{-1}	4.07×10^{-2}	8.16×10^{-2}	4.94×10^{-2}	1.08×10^{-2}
18	15	10	25	0.1	2.65×10^{-1}	1.68×10^{-1}	1.81×10^{-1}	1.48×10^{-1}	2.05×10^{-2}	2.19×10^{-1}	1.09×10^{-1}	7.18×10^{-2}	7.80×10^{-2}	5.31×10^{-2}	1.66×10^{-2}
19	15	15	10	0.3	2.28×10^{-1}	1.54×10^{-1}	1.74×10^{-1}	1.40×10^{-1}	1.54×10^{-2}	2.27×10^{-1}	1.10×10^{-1}	5.17×10^{-2}	7.46×10^{-2}	5.08×10^{-2}	6.52×10^{-3}
20	15	25	1	0.9	1.50×10^{-1}	2.16×10^{-1}	2.28×10^{-1}	2.08×10^{-1}	1.28×10^{-1}	3.21×10^{-1}	2.03×10^{-1}	1.53×10^{-1}	1.66×10^{-1}	1.44×10^{-1}	1.08×10^{-1}
21	20	1	25	0.9	3.62×10^{-2}	1.36×10^{-1}	1.48×10^{-1}	1.41×10^{-1}	6.87×10^{-3}	2.35×10^{-1}	7.21×10^{-2}	1.90×10^{-2}	3.65×10^{-2}	2.38×10^{-2}	2.97×10^{-3}
22	20	5	1	0.3	7.96×10^{-2}	2.92×10^{-1}	3.03×10^{-1}	2.92×10^{-1}	2.20×10^{-1}	3.46×10^{-1}	2.87×10^{-1}	2.41×10^{-1}	2.51×10^{-1}	2.42×10^{-1}	2.08×10^{-1}
23	20	10	10	0.7	2.09×10^{-1}	1.71×10^{-1}	2.05×10^{-1}	1.66×10^{-1}	8.99×10^{-3}	2.42×10^{-1}	1.26×10^{-1}	4.13×10^{-2}	5.65×10^{-2}	3.63×10^{-2}	1.59×10^{-1}
24	20	15	5	0.1	2.71×10^{-1}	2.63×10^{-1}	2.79×10^{-1}	2.58×10^{-1}	1.65×10^{-1}	2.84×10^{-1}	2.44×10^{-1}	2.01×10^{-1}	2.14×10^{-1}	1.95×10^{-1}	1.59×10^{-1}
25	20	25	15	0.5	2.41×10^{-1}	1.86×10^{-1}	2.20×10^{-1}	1.68×10^{-1}	1.27×10^{-2}	2.45×10^{-1}	1.42×10^{-1}	4.85×10^{-2}	7.32×10^{-2}	4.86×10^{-2}	7.02×10^{-3}

Table A3. Parameter sensitivity results of Breast Cancer dataset based on orthogonal arrangement.

No.	<i>M</i>	<i>k</i>	<i>k_{soma}</i>	<i>θ_{soma}</i>	MSE										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	1.29×10^{-1}	2.94×10^{-1}	2.92×10^{-1}	2.91×10^{-1}	2.82×10^{-1}	3.29×10^{-1}	2.90×10^{-1}	3.05×10^{-1}	2.94×10^{-1}	2.87×10^{-1}	2.84×10^{-1}
2	1	5	10	0.9	2.73×10^{-1}	1.73×10^{-1}	1.60×10^{-1}	1.66×10^{-1}	1.50×10^{-1}	2.31×10^{-1}	1.56×10^{-1}	1.58×10^{-1}	1.59×10^{-1}	1.46×10^{-1}	1.41×10^{-1}
3	1	10	15	0.3	2.81×10^{-1}	1.01×10^{-1}	9.33×10^{-2}	8.45×10^{-2}	1.52×10^{-1}	1.76×10^{-1}	7.48×10^{-2}	7.02×10^{-2}	9.91×10^{-2}	5.60×10^{-2}	6.61×10^{-2}
4	1	15	25	0.5	3.28×10^{-1}	1.04×10^{-1}	9.54×10^{-2}	8.78×10^{-2}	3.20×10^{-1}	1.63×10^{-1}	7.78×10^{-2}	7.10×10^{-2}	1.02×10^{-1}	5.75×10^{-2}	6.56×10^{-2}
5	1	25	5	0.7	3.09×10^{-1}	1.27×10^{-1}	1.19×10^{-1}	1.18×10^{-1}	4.46×10^{-1}	1.86×10^{-1}	1.08×10^{-1}	1.08×10^{-1}	1.17×10^{-1}	9.62×10^{-2}	1.01×10^{-1}
6	5	1	5	0.3	1.28×10^{-1}	8.00×10^{-2}	9.25×10^{-2}	8.31×10^{-2}	7.21×10^{-2}	1.48×10^{-1}	8.06×10^{-2}	8.43×10^{-2}	8.19×10^{-2}	7.33×10^{-2}	5.26×10^{-2}
7	5	5	25	0.7	9.98×10^{-2}	9.03×10^{-2}	9.93×10^{-2}	9.93×10^{-2}	6.56×10^{-2}	1.93×10^{-1}	7.44×10^{-2}	7.48×10^{-2}	8.38×10^{-2}	5.73×10^{-2}	5.18×10^{-2}
8	5	10	1	0.5	1.41×10^{-1}	1.94×10^{-1}	2.27×10^{-1}	2.06×10^{-1}	2.48×10^{-1}	3.07×10^{-1}	2.27×10^{-1}	2.64×10^{-1}	2.71×10^{-1}	3.02×10^{-1}	1.43×10^{-1}
9	5	15	15	0.9	3.28×10^{-1}	1.07×10^{-1}	1.22×10^{-1}	1.16×10^{-1}	1.15×10^{-1}	2.16×10^{-1}	1.01×10^{-1}	1.01×10^{-1}	1.11×10^{-1}	8.98×10^{-2}	4.91×10^{-2}
10	5	25	10	0.1	1.78×10^{-1}	1.06×10^{-1}	1.14×10^{-1}	1.08×10^{-1}	9.13×10^{-2}	1.63×10^{-1}	1.01×10^{-1}	9.84×10^{-2}	9.67×10^{-2}	8.91×10^{-2}	8.46×10^{-2}
11	10	1	10	0.5	1.77×10^{-1}	7.34×10^{-2}	8.15×10^{-2}	8.15×10^{-2}	5.16×10^{-2}	1.66×10^{-1}	6.50×10^{-2}	6.33×10^{-2}	7.27×10^{-2}	5.62×10^{-2}	3.42×10^{-2}
12	10	5	15	0.1	1.83×10^{-2}	8.33×10^{-2}	8.97×10^{-2}	9.20×10^{-2}	7.14×10^{-2}	1.49×10^{-1}	7.41×10^{-2}	6.96×10^{-2}	7.45×10^{-2}	6.48×10^{-2}	5.30×10^{-2}
13	10	10	5	0.9	2.45×10^{-1}	9.22×10^{-2}	1.23×10^{-1}	9.54×10^{-2}	6.48×10^{-2}	1.75×10^{-1}	1.01×10^{-1}	1.20×10^{-1}	1.08×10^{-1}	8.60×10^{-2}	4.45×10^{-2}
14	10	15	1	0.7	1.66×10^{-1}	1.70×10^{-1}	2.16×10^{-1}	1.88×10^{-1}	1.74×10^{-1}	2.35×10^{-1}	1.81×10^{-1}	2.19×10^{-1}	2.41×10^{-1}	1.90×10^{-1}	1.10×10^{-1}
15	10	25	25	0.3	2.88×10^{-1}	8.28×10^{-2}	9.90×10^{-2}	8.38×10^{-2}	6.18×10^{-2}	1.76×10^{-1}	7.29×10^{-2}	7.23×10^{-2}	7.98×10^{-2}	5.94×10^{-2}	5.67×10^{-2}
16	15	1	15	0.7	1.66×10^{-1}	7.08×10^{-2}	8.98×10^{-2}	8.68×10^{-2}	5.84×10^{-2}	1.67×10^{-1}	6.76×10^{-2}	6.91×10^{-2}	6.38×10^{-2}	5.81×10^{-2}	3.46×10^{-2}
17	15	5	5	0.5	2.63×10^{-2}	7.00×10^{-2}	9.02×10^{-2}	8.88×10^{-2}	5.53×10^{-2}	1.57×10^{-1}	6.89×10^{-2}	7.07×10^{-2}	7.43×10^{-2}	6.17×10^{-2}	3.79×10^{-2}
18	15	10	25	0.1	7.14×10^{-2}	7.24×10^{-2}	9.11×10^{-2}	8.34×10^{-2}	5.56×10^{-2}	1.54×10^{-1}	6.35×10^{-2}	6.50×10^{-2}	6.76×10^{-2}	5.61×10^{-2}	4.23×10^{-2}
19	15	15	10	0.3	1.41×10^{-1}	7.62×10^{-2}	9.23×10^{-2}	8.64×10^{-2}	5.78×10^{-2}	1.51×10^{-1}	6.70×10^{-2}	5.95×10^{-2}	6.41×10^{-2}	5.65×10^{-2}	4.70×10^{-2}
20	15	25	1	0.9	1.80×10^{-1}	1.60×10^{-1}	2.04×10^{-1}	1.80×10^{-1}	1.41×10^{-1}	2.41×10^{-1}	1.84×10^{-1}	1.91×10^{-1}	1.99×10^{-1}	1.54×10^{-1}	9.11×10^{-2}
21	20	1	25	0.9	2.02×10^{-1}	7.64×10^{-2}	9.92×10^{-2}	8.30×10^{-2}	5.00×10^{-2}	1.88×10^{-1}	6.95×10^{-2}	1.16×10^{-1}	6.73×10^{-2}	5.81×10^{-2}	3.63×10^{-2}
22	20	5	1	0.3	6.27×10^{-2}	1.83×10^{-1}	2.12×10^{-1}	1.97×10^{-1}	1.58×10^{-1}	2.28×10^{-1}	1.97×10^{-1}	1.97×10^{-1}	1.95×10^{-1}	1.89×10^{-1}	1.44×10^{-1}
23	20	10	10	0.7	1.93×10^{-1}	1.36×10^{-1}	9.19×10^{-2}	9.09×10^{-2}	1.25×10^{-1}	1.74×10^{-1}	6.88×10^{-2}	6.69×10^{-2}	7.90×10^{-2}	5.69×10^{-2}	4.44×10^{-2}
24	20	15	5	0.1	6.66×10^{-2}	1.37×10^{-1}	1.46×10^{-1}	1.44×10^{-1}	1.24×10^{-1}	1.80×10^{-1}	1.32×10^{-1}	1.33×10^{-1}	1.39×10^{-1}	1.27×10^{-1}	1.19×10^{-1}
25	20	25	15	0.5	2.87×10^{-1}	7.69×10^{-2}	8.89×10^{-2}	8.28×10^{-2}	5.39×10^{-2}	2.43×10^{-1}	1.32×10^{-1}	6.83×10^{-2}	8.01×10^{-2}	5.72×10^{-2}	4.73×10^{-2}

Table A4. Parameter sensitivity results of Car Evaluation dataset based on orthogonal arrangement.

No.	M	k	k_{soma}	θ_{soma}	MSE										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	1.20×10^{-1}	4.32×10^{-1}	4.29×10^{-1}	4.32×10^{-1}	4.24×10^{-1}	4.65×10^{-1}	4.34×10^{-1}	4.50×10^{-1}	4.33×10^{-1}	4.66×10^{-1}	4.28×10^{-1}
2	1	5	10	0.9	9.21×10^{-2}	1.81×10^{-1}	1.72×10^{-1}	1.99×10^{-1}	2.22×10^{-1}	3.23×10^{-1}	1.81×10^{-1}	1.76×10^{-1}	1.76×10^{-1}	3.18×10^{-1}	1.78×10^{-1}
3	1	10	15	0.3	1.47×10^{-1}	2.25×10^{-1}	2.05×10^{-1}	2.20×10^{-1}	2.72×10^{-1}	3.72×10^{-1}	1.74×10^{-1}	1.70×10^{-1}	1.81×10^{-1}	4.79×10^{-1}	1.54×10^{-1}
4	1	15	25	0.5	1.50×10^{-1}	2.30×10^{-1}	2.24×10^{-1}	2.04×10^{-1}	2.90×10^{-1}	3.47×10^{-1}	1.89×10^{-1}	1.86×10^{-1}	1.89×10^{-1}	4.80×10^{-1}	1.58×10^{-1}
5	1	25	5	0.7	1.40×10^{-1}	1.93×10^{-1}	1.83×10^{-1}	1.96×10^{-1}	2.73×10^{-1}	3.10×10^{-1}	1.76×10^{-1}	1.71×10^{-1}	1.76×10^{-1}	4.24×10^{-1}	1.71×10^{-1}
6	5	1	5	0.3	1.21×10^{-1}	2.05×10^{-1}	2.14×10^{-1}	2.10×10^{-1}	1.64×10^{-1}	3.09×10^{-1}	1.86×10^{-1}	2.20×10^{-1}	1.94×10^{-1}	2.70×10^{-1}	1.48×10^{-1}
7	5	5	25	0.7	1.35×10^{-1}	2.16×10^{-1}	2.10×10^{-1}	2.28×10^{-1}	1.65×10^{-1}	3.67×10^{-1}	1.67×10^{-1}	1.71×10^{-1}	1.70×10^{-1}	3.61×10^{-1}	1.82×10^{-1}
8	5	10	1	0.5	1.03×10^{-1}	3.30×10^{-1}	3.43×10^{-1}	3.43×10^{-1}	3.41×10^{-1}	3.65×10^{-1}	3.39×10^{-1}	3.37×10^{-1}	3.32×10^{-1}	4.01×10^{-1}	3.16×10^{-1}
9	5	15	15	0.9	1.49×10^{-1}	1.92×10^{-1}	1.85×10^{-1}	1.99×10^{-1}	1.69×10^{-1}	3.15×10^{-1}	1.65×10^{-1}	1.63×10^{-1}	1.71×10^{-1}	3.07×10^{-1}	1.76×10^{-1}
10	5	25	10	0.1	1.19×10^{-1}	2.65×10^{-1}	2.60×10^{-1}	2.60×10^{-1}	2.42×10^{-1}	3.38×10^{-1}	2.43×10^{-1}	2.56×10^{-1}	2.47×10^{-1}	3.77×10^{-1}	2.53×10^{-1}
11	10	1	10	0.5	5.44×10^{-2}	1.61×10^{-1}	1.79×10^{-1}	1.84×10^{-1}	1.21×10^{-1}	3.82×10^{-1}	1.39×10^{-1}	1.54×10^{-1}	1.22×10^{-1}	2.67×10^{-1}	9.65×10^{-2}
12	10	5	15	0.1	4.67×10^{-2}	2.18×10^{-1}	2.24×10^{-1}	2.37×10^{-1}	1.86×10^{-1}	3.85×10^{-1}	1.84×10^{-1}	1.99×10^{-1}	2.00×10^{-1}	3.11×10^{-1}	1.62×10^{-1}
13	10	10	5	0.9	1.26×10^{-1}	1.85×10^{-1}	1.91×10^{-1}	1.86×10^{-1}	1.69×10^{-1}	3.37×10^{-1}	1.78×10^{-1}	1.64×10^{-1}	1.61×10^{-1}	2.65×10^{-1}	1.36×10^{-1}
14	10	15	1	0.7	1.05×10^{-1}	2.95×10^{-1}	3.16×10^{-1}	3.12×10^{-1}	2.96×10^{-1}	3.68×10^{-1}	3.13×10^{-1}	2.96×10^{-1}	2.95×10^{-1}	3.64×10^{-1}	2.73×10^{-1}
15	10	25	25	0.3	1.94×10^{-1}	2.25×10^{-1}	2.25×10^{-1}	2.21×10^{-1}	1.99×10^{-1}	4.38×10^{-1}	1.81×10^{-1}	1.92×10^{-1}	1.88×10^{-1}	3.93×10^{-1}	1.98×10^{-1}
16	15	1	15	0.7	4.40×10^{-2}	1.57×10^{-1}	1.73×10^{-1}	1.73×10^{-1}	1.24×10^{-1}	3.71×10^{-1}	1.28×10^{-1}	1.22×10^{-1}	1.19×10^{-1}	2.31×10^{-1}	8.71×10^{-2}
17	15	5	5	0.5	4.04×10^{-2}	1.86×10^{-1}	1.86×10^{-1}	1.90×10^{-1}	1.47×10^{-1}	3.48×10^{-1}	1.56×10^{-1}	1.60×10^{-1}	1.57×10^{-1}	3.07×10^{-1}	1.08×10^{-1}
18	15	10	25	0.1	1.17×10^{-1}	2.05×10^{-1}	2.13×10^{-1}	2.27×10^{-1}	1.72×10^{-1}	4.12×10^{-1}	1.69×10^{-1}	1.81×10^{-1}	1.76×10^{-1}	3.29×10^{-1}	1.69×10^{-1}
19	15	15	10	0.3	1.34×10^{-1}	2.14×10^{-1}	2.15×10^{-1}	2.03×10^{-1}	1.74×10^{-1}	4.11×10^{-1}	1.70×10^{-1}	1.59×10^{-1}	1.75×10^{-1}	3.21×10^{-1}	1.55×10^{-1}
20	15	25	1	0.9	1.04×10^{-1}	2.68×10^{-1}	2.93×10^{-1}	2.96×10^{-1}	2.54×10^{-1}	3.60×10^{-1}	2.82×10^{-1}	2.71×10^{-1}	2.69×10^{-1}	3.34×10^{-1}	2.36×10^{-1}
21	20	1	25	0.9	1.07×10^{-1}	1.50×10^{-1}	1.74×10^{-1}	1.63×10^{-1}	9.89×10^{-2}	2.95×10^{-1}	1.28×10^{-1}	1.20×10^{-1}	1.18×10^{-1}	2.16×10^{-1}	9.23×10^{-2}
22	20	5	1	0.3	9.22×10^{-2}	3.68×10^{-1}	3.78×10^{-1}	3.77×10^{-1}	3.62×10^{-1}	4.13×10^{-1}	3.71×10^{-1}	3.77×10^{-1}	3.72×10^{-1}	4.10×10^{-1}	3.38×10^{-1}
23	20	10	10	0.7	1.37×10^{-1}	2.11×10^{-1}	2.06×10^{-1}	2.04×10^{-1}	1.62×10^{-1}	3.85×10^{-1}	1.60×10^{-1}	1.51×10^{-1}	1.59×10^{-1}	3.11×10^{-1}	1.72×10^{-1}
24	20	15	5	0.1	1.07×10^{-1}	3.23×10^{-1}	3.22×10^{-1}	3.19×10^{-1}	3.04×10^{-1}	3.84×10^{-1}	3.02×10^{-1}	3.13×10^{-1}	3.18×10^{-1}	4.03×10^{-1}	3.01×10^{-1}
25	20	25	15	0.5	1.54×10^{-1}	2.39×10^{-1}	2.21×10^{-1}	2.00×10^{-1}	1.88×10^{-1}	3.57×10^{-1}	1.75×10^{-1}	1.58×10^{-1}	1.70×10^{-1}	4.20×10^{-1}	1.80×10^{-1}

Table A5. Parameter sensitivity results of Diabetic Retinopathy dataset based on orthogonal arrangement.

No.	<i>M</i>	<i>k</i>	<i>k_{soma}</i>	<i>θ_{soma}</i>	MSE										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	1.28×10^{-1}	3.80×10^{-1}	3.90×10^{-1}	3.57×10^{-1}	3.76×10^{-1}	5.12×10^{-1}	3.91×10^{-1}	3.88×10^{-1}	3.75×10^{-1}	4.05×10^{-1}	3.51×10^{-1}
2	1	5	10	0.9	3.09×10^{-1}	4.19×10^{-1}	4.51×10^{-1}	4.23×10^{-1}	8.41×10^{-1}	1.23×10	4.76×10^{-1}	4.49×10^{-1}	3.59×10^{-1}	3.74×10^{-1}	4.22×10^{-1}
3	1	10	15	0.3	3.02×10^{-1}	4.17×10^{-1}	5.77×10^{-1}	3.63×10^{-1}	8.21×10^{-1}	1.21×10	4.06×10^{-1}	4.28×10^{-1}	3.93×10^{-1}	3.51×10^{-1}	4.61×10^{-1}
4	1	15	25	0.5	3.09×10^{-1}	4.73×10^{-1}	8.02×10^{-1}	6.86×10^{-1}	1.05×10	1.22×10	5.51×10^{-1}	6.48×10^{-1}	4.40×10^{-1}	3.68×10^{-1}	7.62×10^{-1}
5	1	25	5	0.7	2.91×10^{-1}	4.26×10^{-1}	7.40×10^{-1}	4.64×10^{-1}	1.10×10	1.16×10	5.93×10^{-1}	6.20×10^{-1}	3.88×10^{-1}	3.67×10^{-1}	4.55×10^{-1}
6	5	1	5	0.3	1.28×10^{-1}	2.66×10^{-1}	3.18×10^{-1}	2.53×10^{-1}	2.38×10^{-1}	8.16×10^{-1}	3.19×10^{-1}	2.61×10^{-1}	2.56×10^{-1}	3.58×10^{-1}	1.75×10^{-1}
7	5	5	25	0.7	2.94×10^{-1}	3.79×10^{-1}	4.77×10^{-1}	3.68×10^{-1}	5.52×10^{-1}	1.21	5.08×10^{-1}	5.25×10^{-1}	3.21×10^{-1}	3.50×10^{-1}	2.87×10^{-1}
8	5	10	1	0.5	1.47×10^{-1}	3.75×10^{-1}	4.01×10^{-1}	3.71×10^{-1}	4.53×10^{-1}	5.84×10^{-1}	4.08×10^{-1}	4.06×10^{-1}	3.93×10^{-1}	4.24×10^{-1}	3.56×10^{-1}
9	5	15	15	0.9	3.08×10^{-1}	3.80×10^{-1}	5.74×10^{-1}	3.74×10^{-1}	6.83×10^{-1}	1.21×10	4.52×10^{-1}	4.01×10^{-1}	3.70×10^{-1}	3.74×10^{-1}	3.52×10^{-1}
10	5	25	10	0.1	1.79×10^{-1}	2.85×10^{-1}	3.94×10^{-1}	2.59×10^{-1}	5.39×10^{-1}	6.93×10^{-1}	3.68×10^{-1}	3.22×10^{-1}	3.42×10^{-1}	3.55×10^{-1}	2.85×10^{-1}
11	10	1	10	0.5	3.05×10^{-1}	2.82×10^{-1}	4.24×10^{-1}	2.69×10^{-1}	1.90×10^{-1}	1.20	3.14×10^{-1}	2.43×10^{-1}	2.18×10^{-1}	3.57×10^{-1}	1.41×10^{-1}
12	10	5	15	0.1	1.93×10^{-1}	2.73×10^{-1}	3.32×10^{-1}	2.67×10^{-1}	2.34×10^{-1}	7.86×10^{-1}	3.14×10^{-1}	2.79×10^{-1}	2.66×10^{-1}	3.57×10^{-1}	1.92×10^{-1}
13	10	10	5	0.9	3.02×10^{-1}	4.06×10^{-1}	4.88×10^{-1}	4.01×10^{-1}	4.62×10^{-1}	1.19×10	3.68×10^{-1}	4.61×10^{-1}	3.98×10^{-1}	3.29×10^{-1}	2.49×10^{-1}
14	10	15	1	0.7	1.59×10^{-1}	3.88×10^{-1}	4.34×10^{-1}	4.01×10^{-1}	4.24×10^{-1}	6.31×10^{-1}	4.29×10^{-1}	4.06×10^{-1}	3.91×10^{-1}	4.30×10^{-1}	3.38×10^{-1}
15	10	25	25	0.3	3.09×10^{-1}	3.96×10^{-1}	5.56×10^{-1}	2.84×10^{-1}	7.47×10^{-1}	1.23	4.23×10^{-1}	4.34×10^{-1}	4.66×10^{-1}	4.08×10^{-1}	3.19×10^{-1}
16	15	1	15	0.7	3.06×10^{-1}	2.79×10^{-1}	5.86×10^{-1}	2.78×10^{-1}	1.91×10^{-1}	1.21	5.74×10^{-1}	6.41×10^{-1}	2.28×10^{-1}	3.60×10^{-1}	1.26×10^{-1}
17	15	5	5	0.5	2.17×10^{-1}	3.10×10^{-1}	3.92×10^{-1}	2.95×10^{-1}	2.93×10^{-1}	1.00	3.67×10^{-1}	2.82×10^{-1}	2.98×10^{-1}	3.40×10^{-1}	1.75×10^{-1}
18	15	10	25	0.1	2.63×10^{-1}	3.16×10^{-1}	3.84×10^{-1}	2.99×10^{-1}	2.44×10^{-1}	8.96×10^{-1}	3.15×10^{-1}	3.38×10^{-1}	3.00×10^{-1}	3.57×10^{-1}	2.30×10^{-1}
19	15	15	10	0.3	2.69×10^{-1}	3.58×10^{-1}	4.66×10^{-1}	2.74×10^{-1}	3.10×10^{-1}	1.08×10	3.46×10^{-1}	3.22×10^{-1}	3.46×10^{-1}	3.83×10^{-1}	2.37×10^{-1}
20	15	25	1	0.9	1.71×10^{-1}	4.16×10^{-1}	4.73×10^{-1}	4.34×10^{-1}	5.20×10^{-1}	6.79×10^{-1}	4.68×10^{-1}	4.47×10^{-1}	4.41×10^{-1}	4.10×10^{-1}	3.63×10^{-1}
21	20	1	25	0.9	3.07×10^{-1}	3.06×10^{-1}	9.60×10^{-1}	3.01×10^{-1}	2.71×10^{-1}	1.21×10	4.78×10^{-1}	1.13	2.98×10^{-1}	4.19×10^{-1}	1.27×10^{-1}
22	20	5	1	0.3	1.30×10^{-1}	3.46×10^{-1}	3.91×10^{-1}	3.55×10^{-1}	3.20×10^{-1}	5.37×10^{-1}	3.76×10^{-1}	3.76×10^{-1}	3.54×10^{-1}	4.03×10^{-1}	2.62×10^{-1}
23	20	10	10	0.7	3.03×10^{-1}	3.81×10^{-1}	4.79×10^{-1}	2.99×10^{-1}	2.70×10^{-1}	1.22	3.98×10^{-1}	3.89×10^{-1}	3.06×10^{-1}	3.41×10^{-1}	2.08×10^{-1}
24	20	15	5	0.1	1.40×10^{-1}	2.75×10^{-1}	3.21×10^{-1}	2.61×10^{-1}	2.87×10^{-1}	5.87×10^{-1}	2.95×10^{-1}	2.94×10^{-1}	3.08×10^{-1}	3.30×10^{-1}	2.30×10^{-1}
25	20	25	15	0.5	3.07×10^{-1}	3.99×10^{-1}	4.76×10^{-1}	3.02×10^{-1}	4.50×10^{-1}	$1.22 \times 10+00$	3.70×10^{-1}	4.57×10^{-1}	3.75×10^{-1}	3.79×10^{-1}	2.66×10^{-1}

Table A6. Parameter sensitivity results of Glass Identification dataset based on orthogonal arrangement.

No.	M	k	k_{soma}	θ_{soma}	MSE										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	1.19×10^{-1}	4.18×10^{-1}	4.18×10^{-1}	4.20×10^{-1}	4.11×10^{-1}	4.36×10^{-1}	4.19×10^{-1}	4.27×10^{-1}	4.21×10^{-1}	4.19×10^{-1}	4.20×10^{-1}
2	1	5	10	0.9	1.15×10^{-1}	1.25×10^{-1}	1.09×10^{-1}	1.18×10^{-1}	2.06×10^{-1}	2.15×10^{-1}	1.15×10^{-1}	1.22×10^{-1}	1.35×10^{-1}	1.26×10^{-1}	1.12×10^{-1}
3	1	10	15	0.3	1.16×10^{-1}	1.17×10^{-1}	1.11×10^{-1}	1.15×10^{-1}	1.86×10^{-1}	2.02×10^{-1}	8.84×10^{-2}	9.72×10^{-2}	1.06×10^{-1}	9.60×10^{-2}	1.44×10^{-1}
4	1	15	25	0.5	1.20×10^{-1}	1.22×10^{-1}	1.20×10^{-1}	1.23×10^{-1}	2.32×10^{-1}	2.09×10^{-1}	1.17×10^{-1}	1.48×10^{-1}	1.29×10^{-1}	1.47×10^{-1}	1.63×10^{-1}
5	1	25	5	0.7	1.14×10^{-1}	1.06×10^{-1}	1.15×10^{-1}	1.08×10^{-1}	2.88×10^{-1}	2.01×10^{-1}	1.09×10^{-1}	1.71×10^{-1}	1.22×10^{-1}	1.18×10^{-1}	1.18×10^{-1}
6	5	1	5	0.3	1.19×10^{-1}	1.37×10^{-1}	1.50×10^{-1}	1.40×10^{-1}	1.11×10^{-1}	2.04×10^{-1}	1.32×10^{-1}	1.49×10^{-1}	1.35×10^{-1}	1.32×10^{-1}	1.12×10^{-1}
7	5	5	25	0.7	9.39×10^{-2}	1.04×10^{-1}	1.21×10^{-1}	1.18×10^{-1}	8.48×10^{-2}	2.34×10^{-1}	9.09×10^{-2}	1.02×10^{-1}	9.89×10^{-2}	9.22×10^{-2}	7.45×10^{-2}
8	5	10	1	0.5	9.89×10^{-2}	2.93×10^{-1}	3.14×10^{-1}	3.02×10^{-1}	2.97×10^{-1}	3.40×10^{-1}	3.05×10^{-1}	3.12×10^{-1}	3.17×10^{-1}	3.12×10^{-1}	2.77×10^{-1}
9	5	15	15	0.9	1.18×10^{-1}	9.22×10^{-2}	1.10×10^{-1}	1.05×10^{-1}	1.18×10^{-1}	2.23×10^{-1}	9.13×10^{-2}	1.03×10^{-1}	1.13×10^{-1}	1.08×10^{-1}	9.61×10^{-2}
10	5	25	10	0.1	8.84×10^{-2}	1.76×10^{-1}	1.92×10^{-1}	1.88×10^{-1}	1.91×10^{-1}	2.58×10^{-1}	1.87×10^{-1}	1.81×10^{-1}	2.00×10^{-1}	2.01×10^{-1}	1.74×10^{-1}
11	10	1	10	0.5	1.17×10^{-1}	1.00×10^{-1}	1.11×10^{-1}	1.05×10^{-1}	5.62×10^{-2}	2.01×10^{-1}	9.34×10^{-2}	9.79×10^{-2}	8.10×10^{-2}	8.77×10^{-2}	5.57×10^{-2}
12	10	5	15	0.1	5.73×10^{-2}	1.23×10^{-1}	1.42×10^{-1}	1.37×10^{-1}	1.04×10^{-1}	2.09×10^{-1}	1.21×10^{-1}	1.21×10^{-1}	1.14×10^{-1}	1.31×10^{-1}	1.01×10^{-1}
13	10	10	5	0.9	1.08×10^{-1}	9.84×10^{-2}	1.31×10^{-1}	1.23×10^{-1}	8.44×10^{-2}	2.29×10^{-1}	1.09×10^{-1}	1.05×10^{-1}	1.22×10^{-1}	1.21×10^{-1}	5.99×10^{-2}
14	10	15	1	0.7	9.59×10^{-2}	2.49×10^{-1}	2.75×10^{-1}	2.63×10^{-1}	2.53×10^{-1}	2.98×10^{-1}	2.62×10^{-1}	2.67×10^{-1}	2.70×10^{-1}	2.80×10^{-1}	2.40×10^{-1}
15	10	25	25	0.3	1.18×10^{-1}	9.77×10^{-2}	1.22×10^{-1}	1.17×10^{-1}	9.47×10^{-2}	2.24×10^{-1}	9.57×10^{-2}	9.80×10^{-2}	1.09×10^{-1}	9.67×10^{-2}	8.62×10^{-2}
16	15	1	15	0.7	1.17×10^{-1}	9.19×10^{-2}	1.15×10^{-1}	1.01×10^{-1}	6.00×10^{-2}	2.24×10^{-1}	9.42×10^{-2}	1.04×10^{-1}	8.88×10^{-2}	9.29×10^{-2}	5.49×10^{-2}
17	15	5	5	0.5	4.48×10^{-2}	9.35×10^{-2}	1.00×10^{-1}	1.12×10^{-1}	7.56×10^{-2}	2.07×10^{-1}	8.71×10^{-2}	8.79×10^{-2}	9.05×10^{-2}	8.94×10^{-2}	5.42×10^{-2}
18	15	10	25	0.1	5.94×10^{-2}	9.94×10^{-2}	1.11×10^{-1}	1.08×10^{-1}	7.37×10^{-2}	2.15×10^{-1}	8.73×10^{-2}	9.53×10^{-2}	1.03×10^{-1}	8.61×10^{-2}	6.23×10^{-2}
19	15	15	10	0.3	9.17×10^{-2}	9.90×10^{-2}	1.15×10^{-1}	9.92×10^{-2}	6.83×10^{-2}	2.34×10^{-1}	8.84×10^{-2}	8.88×10^{-2}	9.14×10^{-2}	9.11×10^{-2}	6.27×10^{-2}
20	15	25	1	0.9	9.17×10^{-2}	2.20×10^{-1}	2.38×10^{-1}	2.30×10^{-1}	2.03×10^{-1}	2.87×10^{-1}	2.35×10^{-1}	2.34×10^{-1}	2.41×10^{-1}	2.53×10^{-1}	1.84×10^{-1}
21	20	1	25	0.9	1.20×10^{-1}	9.36×10^{-2}	1.19×10^{-1}	1.11×10^{-1}	5.80×10^{-2}	2.29×10^{-1}	9.66×10^{-2}	1.01×10^{-1}	8.64×10^{-2}	9.72×10^{-2}	5.20×10^{-2}
22	20	5	1	0.3	1.02×10^{-1}	3.36×10^{-1}	3.55×10^{-1}	3.49×10^{-1}	3.20×10^{-1}	3.76×10^{-1}	3.43×10^{-1}	9.04×10^{-1}	3.45×10^{-1}	3.53×10^{-1}	3.11×10^{-1}
23	20	10	10	0.7	1.15×10^{-1}	9.16×10^{-2}	1.14×10^{-1}	1.07×10^{-1}	6.69×10^{-2}	2.38×10^{-1}	8.79×10^{-2}	8.13×10^{-2}	8.83×10^{-2}	8.06×10^{-2}	6.12×10^{-2}
24	20	15	5	0.1	8.52×10^{-2}	2.65×10^{-1}	2.79×10^{-1}	2.78×10^{-1}	2.51×10^{-1}	3.45×10^{-1}	2.68×10^{-1}	2.75×10^{-1}	1.07×10^{-1}	2.71×10^{-1}	2.55×10^{-1}
25	20	25	15	0.5	1.19×10^{-1}	9.18×10^{-2}	1.15×10^{-1}	1.10×10^{-1}	7.40×10^{-2}	2.50×10^{-1}	9.03×10^{-2}	7.62×10^{-2}	1.05×10^{-1}	9.30×10^{-2}	6.92×10^{-2}

Table A7. Average accuracy of testing of each learning algorithm in Australian Credit Approval dataset.

No.	M	k	k_{soma}	θ_{soma}	Accuracy										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	54.33	82.56	84.09	85.28	80.19	55.88	77.04	84.11	85.19	56.55	85.39
2	1	5	10	0.9	55.31	83.17	82.33	84.64	66.33	55.01	70.82	82.13	84.06	58.12	84.67
3	1	10	15	0.3	54.33	83.93	79.44	80.85	59.40	56.01	76.51	83.80	84.22	58.79	78.62
4	1	15	25	0.5	55.81	81.79	78.36	76.47	59.15	56.12	71.51	84.30	83.08	56.73	73.72
5	1	25	5	0.7	54.33	83.27	75.96	77.95	59.71	55.60	69.60	83.85	83.46	56.84	77.33
6	5	1	5	0.3	55.76	84.41	83.69	84.59	84.35	55.01	82.06	83.98	85.39	63.67	85.04
7	5	5	25	0.7	57.65	83.80	83.66	85.23	76.59	55.85	75.04	81.82	81.85	60.85	83.64
8	5	10	1	0.5	55.76	84.04	83.35	84.57	73.99	54.98	78.99	84.40	83.90	57.12	83.59
9	5	15	15	0.9	55.76	85.52	82.14	85.99	74.51	56.02	79.15	84.11	84.30	59.15	84.48
10	5	25	10	0.1	55.76	82.35	80.45	84.72	66.36	55.62	77.71	84.06	81.30	56.76	80.66
11	10	1	10	0.5	56.34	84.21	82.64	83.11	84.34	56.10	80.34	83.96	83.17	71.40	84.38
12	10	5	15	0.1	65.64	84.61	82.46	84.25	85.06	55.28	79.94	84.81	84.52	59.77	83.62
13	10	10	5	0.9	56.34	85.67	83.64	85.31	81.00	55.41	78.90	83.11	83.64	62.19	84.51
14	10	15	1	0.7	55.07	83.88	83.17	83.41	78.33	56.10	79.68	83.78	84.25	58.18	84.86
15	10	25	25	0.3	56.34	83.53	82.38	83.59	76.54	57.00	79.82	84.48	81.34	59.29	82.27
16	15	1	15	0.7	55.48	84.40	78.78	84.62	82.33	55.78	78.65	83.24	84.27	64.17	85.97
17	15	5	5	0.5	64.38	84.30	83.37	81.32	81.88	55.43	76.28	85.43	83.12	66.12	84.86
18	15	10	25	0.1	56.97	84.04	83.54	84.48	83.41	55.41	77.62	85.89	82.46	61.71	85.10
19	15	15	10	0.3	56.36	83.88	83.24	82.93	79.13	55.17	81.00	83.99	82.24	62.11	83.74
20	15	25	1	0.9	55.48	84.73	82.85	84.94	79.18	54.78	78.47	82.09	83.86	61.45	83.45
21	20	1	25	0.9	55.14	84.51	76.97	83.96	83.09	56.33	75.43	77.00	85.65	62.01	85.01
22	20	5	1	0.3	61.14	84.52	82.91	84.96	83.14	55.73	77.12	83.82	84.56	59.32	84.65
23	20	10	10	0.7	55.14	84.20	84.90	85.15	82.29	55.09	78.62	83.19	83.49	64.30	84.73
24	20	15	5	0.1	58.12	82.25	80.45	84.06	78.57	55.70	79.98	83.78	84.25	60.34	83.24
25	20	25	15	0.5	55.14	84.80	82.72	81.95	81.01	55.93	79.05	83.85	81.72	59.89	82.75

Table A8. Average accuracy of testing of each learning algorithm in Banknote Authentication dataset.

No.	M	k	k_{soma}	θ_{soma}	Accuracy										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	57.45	88.88	89.04	89.28	89.00	84.94	82.43	88.67	89.10	89.09	88.55
2	1	5	10	0.9	75.02	88.11	88.57	87.62	87.28	82.85	78.79	89.26	88.88	89.25	87.73
3	1	10	15	0.3	57.89	88.79	88.91	88.62	88.92	83.60	80.65	91.82	91.29	89.90	88.67
4	1	15	25	0.5	54.93	88.68	88.96	88.88	88.95	83.86	83.67	90.56	90.00	88.86	89.35
5	1	25	5	0.7	54.93	88.11	88.41	88.62	88.33	83.57	78.91	88.79	89.24	88.75	88.30
6	5	1	5	0.3	67.58	92.40	93.12	92.94	98.10	85.50	87.79	96.80	97.22	96.63	98.94
7	5	5	25	0.7	57.91	91.31	91.48	92.45	96.89	84.17	83.60	96.42	95.43	96.46	98.03
8	5	10	1	0.5	69.01	90.44	89.40	90.96	95.35	84.23	85.95	94.74	94.13	95.24	96.34
9	5	15	15	0.9	57.56	90.57	90.76	93.20	97.09	84.82	86.69	96.08	95.13	97.16	98.00
10	5	25	10	0.1	52.59	91.38	90.96	92.47	96.59	83.16	86.80	94.59	94.75	95.11	98.04
11	10	1	10	0.5	86.50	91.62	91.50	92.67	98.45	88.10	93.23	98.67	97.53	98.73	99.50
12	10	5	15	0.1	81.17	91.20	91.37	91.22	97.94	88.17	92.68	96.43	95.37	96.44	99.26
13	10	10	5	0.9	61.35	89.69	89.79	90.66	96.25	86.33	91.58	96.46	96.31	97.54	98.66
14	10	15	1	0.7	63.06	89.89	89.23	90.73	95.74	85.02	90.59	94.45	94.01	96.11	97.57
15	10	25	25	0.3	50.11	91.11	91.83	92.42	97.86	87.66	92.69	96.29	95.60	96.50	98.55
16	15	1	15	0.7	88.16	91.33	90.89	92.52	98.69	88.24	94.33	98.96	97.68	99.01	99.57
17	15	5	5	0.5	80.88	90.68	89.89	91.67	98.11	87.97	91.01	97.27	95.39	96.96	98.88
18	15	10	25	0.1	47.85	90.69	89.22	91.82	97.97	88.18	91.86	95.31	95.50	96.09	98.75
19	15	15	10	0.3	51.42	91.25	90.40	91.65	97.99	87.65	91.33	96.67	94.85	96.51	98.82
20	15	25	1	0.9	55.69	87.67	87.86	89.94	95.53	83.22	88.25	94.16	93.01	96.13	97.93
21	20	1	25	0.9	92.48	91.51	91.33	91.75	98.93	87.31	94.91	98.94	97.52	98.35	99.42
22	20	5	1	0.3	96.19	90.03	88.26	90.19	97.83	88.04	90.01	95.19	94.78	95.53	98.50
23	20	10	10	0.7	55.22	90.15	88.54	90.66	98.16	87.66	91.15	97.18	96.44	97.12	99.07
24	20	15	5	0.1	46.21	89.18	87.34	90.17	98.28	87.54	91.24	95.55	94.60	95.93	98.97
25	20	25	15	0.5	51.43	89.35	88.58	90.67	98.21	88.20	88.44	96.86	95.19	96.59	98.81

Table A9. Average accuracy of testing of each learning algorithm in Breast Cancer dataset.

No.	M	k	k_{soma}	θ_{soma}	Accuracy										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	34.71	93.59	94.25	94.97	95.17	89.48	82.37	92.65	93.63	94.48	94.78
2	1	5	10	0.9	46.03	92.89	94.19	94.1	94.84	90.06	85.33	94.32	94.48	95.05	95.43
3	1	10	15	0.3	42.46	93.87	94.19	93.92	90.52	90.67	90.78	94.16	92.84	95.68	95.33
4	1	15	25	0.5	34.71	93.97	94.37	94.05	82.79	90.08	83.84	94.41	93.16	95.32	94.56
5	1	25	5	0.7	34.71	93.57	94.19	93.43	76.76	90.19	80.68	94.54	93.86	94.81	94.56
6	5	1	5	0.3	34.54	95.06	94.92	94.49	94.84	91.11	94.76	95.38	94.54	95.30	96.22
7	5	5	25	0.7	78.95	93.75	94.41	94.02	95.10	90.67	94.56	94.95	94.19	94.84	95.14
8	5	10	1	0.5	42.71	93.41	90.84	92.76	94.97	89.68	94.19	94.33	94.75	95.38	95.35
9	5	15	15	0.9	34.54	94.05	93.79	93.73	92.49	89.24	92.46	94.29	94.29	95.57	95.83
10	5	25	10	0.1	38.29	94.19	93.92	94.32	94.59	89.84	92.29	94.67	95.05	95.14	94.76
11	10	1	10	0.5	58.13	94.98	94.00	94.35	95.06	90.05	89.97	95.08	94.43	95.71	95.94
12	10	5	15	0.1	95.95	94.86	94.11	94.13	95.11	90.46	92.87	95.13	94.9	95.57	95.90
13	10	10	5	0.9	50.84	93.19	92.35	93.78	95.43	90.52	91.75	95.00	94.67	95.48	95.60
14	10	15	1	0.7	35.35	92.84	91.94	92.1	94.89	89.98	89.59	94.44	93.98	94.95	95.38
15	10	25	25	0.3	42.19	94.21	93.65	94.25	95.06	90.76	94.32	94.94	94.16	95.32	95.08
16	15	1	15	0.7	58.78	95.43	93.92	94.32	95.37	90.05	94.46	95.00	95.03	95.32	96.16
17	15	5	5	0.5	93.30	94.56	93.97	94.24	95.6	90.67	92.49	95.02	95.14	95.75	96.13
18	15	10	25	0.1	83.17	94.83	94.10	94.25	95.51	90.76	92.65	95.56	95.22	95.68	95.51
19	15	15	10	0.3	67.75	94.21	93.73	94.46	94.98	91.14	88.63	95.08	94.65	95.51	95.71
20	15	25	1	0.9	33.84	92.44	92.38	91.81	94.81	88.46	89.14	94.33	94.76	95.02	95.46
21	20	1	25	0.9	54.52	94.67	93.63	94.41	95.37	89.17	91.27	92.94	95.56	95.44	96.00
22	20	5	1	0.3	95.41	93.43	93.52	93.78	94.86	89.41	94.10	94.27	94.70	95.11	95.30
23	20	10	10	0.7	60.48	94.57	93.43	94.11	95.19	91.08	94.25	94.92	93.92	95.08	96.21
24	20	15	5	0.1	78.52	94.49	94.35	94.25	95.38	90.92	94.10	94.87	94.46	95.10	95.46
25	20	25	15	0.5	42.37	94.16	94.24	93.84	94.71	80.38	94.10	95.00	94.06	95.25	95.83

Table A10. Average accuracy of testing of each learning algorithm in Car Evaluation dataset.

No.	M	k	k_{soma}	θ_{soma}	Accuracy										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	70.1	86.78	87.52	85.22	86.24	75.53	81.6	81.5	86.65	75.88	87.78
2	1	5	10	0.9	79.63	86.49	87.89	85.82	83.73	76.44	81.94	87.77	87.81	74.79	86.74
3	1	10	15	0.3	70.1	86.37	86.97	86.5	82.99	80.06	84.83	88.4	87.94	71.06	88.93
4	1	15	25	0.5	70.1	87.73	87.20	83.73	82.89	77.86	83.42	87.95	87.32	72.37	85.21
5	1	25	5	0.7	70.43	86.43	86.4	86.00	81.10	78.5	81.71	88.11	86.89	71.62	88.42
6	5	1	5	0.3	69.43	88.74	86.85	87.42	90.93	77.86	88.62	88.65	88.58	81.63	93.8
7	5	5	25	0.7	71.2	87.53	86.28	87.79	88.96	81.16	83.93	88.44	88.38	77.62	89.36
8	5	10	1	0.5	72.95	87.41	87.68	86.85	86.17	82.07	85.97	87.03	87.86	75.51	89.66
9	5	15	15	0.9	69.43	86.45	86.02	88.27	87.67	80.89	84.83	87.83	87.76	78.71	87.57
10	5	25	10	0.1	70.42	85.93	86.82	87.12	87.61	81.10	86.34	87.11	88.31	73.91	86.64
11	10	1	10	0.5	85.45	88.91	87.64	86.53	91.53	75.97	88.82	89.36	91.45	79.58	92.94
12	10	5	15	0.1	88.55	86.72	86.39	87.32	89.56	76.61	87.99	88.98	88.85	77.77	91.6
13	10	10	5	0.9	74.4	86.1	86.38	86.04	86.89	69.8	85.44	88.38	88.52	78.71	89.36
14	10	15	1	0.7	71.08	87.93	87.32	87.05	88.98	76.55	85.36	87.88	87.69	77.95	89.49
15	10	25	25	0.3	61.13	87.3	87.07	87.79	87.37	77.10	87.34	87.73	86.56	77.29	89.03
16	15	1	15	0.7	88.44	88.63	87.48	87.99	90.72	77.77	89.85	91.46	91.51	82.46	93.8
17	15	5	5	0.5	88.59	87.21	87.04	84.09	88.97	78.68	86.24	89.89	88.69	77.59	92.87
18	15	10	25	0.1	73.54	87.07	87.10	87.77	88.75	76.94	87.64	87.7	88.53	76.71	89.95
19	15	15	10	0.3	71.83	87.42	86.3	87.7	88.24	76.81	86.96	89.18	87.73	77.90	89.96
20	15	25	1	0.9	70.35	86.55	85.51	86.43	88.7	75.33	86.35	87.92	86.54	81.06	90.15
21	20	1	25	0.9	88.77	89.34	86.63	87.71	92.94	82.28	87.57	91.22	91.47	84.43	93.32
22	20	5	1	0.3	89.27	88.42	88.07	87.66	88.97	80.37	87.05	87.39	87.96	79.46	90.74
23	20	10	10	0.7	72.05	86.18	86.81	87.86	88.2	78.31	85.53	89.56	88.13	78.88	89.21
24	20	15	5	0.1	80.05	86.03	86.17	86.74	89.4	79.51	87.09	88.07	87.57	73.93	88.44
25	20	25	15	0.5	69.49	87.07	86.96	84.17	87.92	79.39	86.8	88.81	88.15	74.76	88.65

Table A11. Average accuracy of testing of each learning algorithm in Diabetic Retinopathy dataset.

No.	M	k	k_{soma}	θ_{soma}	Accuracy										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	39.04	75.62	77.05	81.45	81.56	37.07	65.88	75.43	74.32	67.20	81.11
2	1	5	10	0.9	39.04	70.32	70.53	80.6	56.94	38.65	59.57	72.35	74.40	72.88	76.73
3	1	10	15	0.3	39.04	77.8	70.00	78.68	56.82	39.29	74.42	73.61	74.00	72.24	72.31
4	1	15	25	0.5	39.04	75.24	58.95	61.05	47.35	37.48	67.65	63.70	72.01	72.22	59.04
5	1	25	5	0.7	39.04	75.28	57.5	72.26	40.00	38.93	58.63	61.92	69.98	69.87	71.67
6	5	1	5	0.3	38.68	81.92	78.7	81.45	83.35	38.65	77.20	82.09	81.37	71.71	88.27
7	5	5	25	0.7	41.11	78.29	74.29	81.52	69.38	38.16	67.67	67.37	76.28	75.13	83.08
8	5	10	1	0.5	38.68	80.00	77.46	80.96	64.51	38.95	72.44	73.1	75.45	68.29	81.05
9	5	15	15	0.9	38.68	81.62	67.33	82.29	63.14	39.06	69.89	71.79	74.17	72.48	82.33
10	5	25	10	0.1	38.68	80.17	68.59	81.62	55.56	38.76	67.03	75.26	73.70	70.17	77.95
11	10	1	10	0.5	39.08	80.66	69.06	81.54	87.16	38.7	74.29	82.82	85.15	70.38	89.91
12	10	5	15	0.1	45.21	81.54	77.39	82.65	83.70	46.13	77.63	79.32	80.21	69.64	86.22
13	10	10	5	0.9	39.08	78.25	66.94	81.54	75.32	40.24	67.03	70.49	72.54	74.59	84.51
14	10	15	1	0.7	38.89	80.53	75.00	82.01	72.29	37.95	74.55	76.18	75.66	71.71	82.69
15	10	25	25	0.3	39.08	80.00	71.20	81.90	59.83	39.53	74.94	74.59	69.53	72.52	80.85
16	15	1	15	0.7	37.91	80.15	62.48	81.43	86.13	38.59	68.12	62.52	83.63	70.06	90.79
17	15	5	5	0.5	48.48	78.68	77.35	78.91	81.03	35.60	71.79	79.12	77.88	72.97	86.28
18	15	10	25	0.1	37.88	80.34	76.79	81.71	82.71	45.53	77.35	76.11	77.74	72.86	84.27
19	15	15	10	0.3	41.30	79.59	74.42	83.03	79.83	38.61	74.44	77.50	74.64	70.21	84.70
20	15	25	1	0.9	37.91	79.36	69.62	81.37	64.08	36.82	68.23	72.39	72.01	74.81	81.62
21	20	1	25	0.9	38.18	78.03	48.38	80.38	81.05	38.12	67.78	43.48	80.09	65.64	89.66
22	20	5	1	0.3	45.26	81.82	74.87	81.45	82.71	37.71	76.71	75.88	75.79	68.97	85.94
23	20	10	10	0.7	38.8	77.99	72.76	80.13	81.77	39.29	69.49	72.93	76.84	74.36	85.92
24	20	15	5	0.1	42.39	79.51	73.93	81.39	77.03	40.06	74.81	78.57	75.96	75.24	83.59
25	20	25	15	0.5	38.18	78.97	75.45	76.37	73.59	37.63	73.97	71.97	74.83	69.76	81.50

Table A12. Average accuracy of testing of each learning algorithm in Diabetic Retinopathy dataset.

No.	M	k	k_{soma}	θ_{soma}	Accuracy										
					BP	GA	DE	PBIL	PSO	ACO	ABC	WOA	HHO	ChOA	BBO
1	1	1	1	0.1	76.72	90.73	91.88	90.89	90.10	88.44	87.66	89.95	91.04	92.29	91.67
2	1	5	10	0.9	77.24	92.19	91.51	91.61	86.30	87.24	88.39	92.24	91.61	92.86	91.25
3	1	10	15	0.3	76.98	92.19	91.61	90.52	88.23	89.06	87.55	92.92	91.56	93.07	90.26
4	1	15	25	0.5	76.72	91.46	91.30	91.88	86.98	85.52	87.66	89.38	90.21	90.68	89.53
5	1	25	5	0.7	76.72	91.41	92.19	92.55	82.97	87.40	87.86	88.80	90.63	92.60	92.14
6	5	1	5	0.3	75.63	92.34	91.93	91.15	92.45	88.07	89.53	90.94	92.19	93.13	91.88
7	5	5	25	0.7	78.70	92.40	90.99	92.71	92.76	86.82	91.30	90.78	90.31	92.76	92.08
8	5	10	1	0.5	76.61	92.60	90.42	92.45	91.15	86.77	91.41	90.26	91.46	92.92	91.82
9	5	15	15	0.9	75.63	90.68	92.14	91.04	92.19	86.41	91.25	90.26	91.35	93.23	90.73
10	5	25	10	0.1	76.35	92.08	91.35	90.73	89.90	86.25	89.69	90.31	90.94	89.95	90.99
11	10	1	10	0.5	75.89	91.98	92.34	90.26	93.01	88.07	91.72	92.50	91.67	93.05	93.07
12	10	5	15	0.1	84.58	91.77	90.36	91.88	92.40	85.83	90.10	90.47	91.51	92.50	91.30
13	10	10	5	0.9	77.40	90.57	90.26	91.56	91.72	86.61	90.94	92.45	91.15	92.86	92.08
14	10	15	1	0.7	78.39	91.56	90.57	91.72	93.39	88.02	90.99	90.63	91.04	94.43	92.60
15	10	25	25	0.3	75.89	91.61	91.20	91.98	92.40	87.86	90.99	92.08	90.94	92.60	91.25
16	15	1	15	0.7	75.36	90.16	91.09	90.36	93.18	87.86	92.14	90.57	91.98	93.59	93.49
17	15	5	5	0.5	86.04	91.51	90.78	92.76	92.97	85.47	91.25	91.20	92.08	93.39	90.89
18	15	10	25	0.1	83.75	91.46	90.89	92.19	92.76	86.77	89.79	91.82	91.67	92.29	91.15
19	15	15	10	0.3	78.29	91.61	91.41	91.35	93.49	86.41	92.08	91.25	92.76	94.06	92.81
20	15	25	1	0.9	75.36	92.08	89.84	91.35	91.82	85.21	90.73	92.45	92.29	92.71	92.66
21	20	1	25	0.9	76.35	92.50	91.35	92.14	93.07	87.86	91.51	90.94	92.08	93.65	92.40
22	20	5	1	0.3	82.86	90.73	89.11	89.95	92.71	85.26	90.10	90.36	91.46	93.54	92.40
23	20	10	10	0.7	76.88	92.03	91.61	91.56	93.07	86.35	90.83	92.71	93.39	92.97	92.66
24	20	15	5	0.1	83.91	90.31	89.53	91.46	89.90	84.74	88.65	89.90	92.50	91.46	90.89
25	20	25	15	0.5	76.30	92.08	91.25	89.38	92.71	85.00	90.89	93.07	90.78	93.39	91.46

References

- Novaković, J.D.; Veljović, A. Evaluation of classification models in machine learning. *Theory Appl. Math. Comput. Sci.* **2017**, *7*, 39–46.
- Awad, W.A.; ELseiofi, S.M. Machine learning methods for spam e-mail classification. *IJCSIT* **2011**, *3*, 173–184. [\[CrossRef\]](#)
- Ossama, A.H.; Mohamed, A.R. Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Trans.* **2014**, *22*, 1533–1545.
- Argentiero, P.; Chin, R. An automated approach to the design of decision tree classifiers. *IEEE Trans.* **1982**, *PAMI-4*, 51–57. [\[CrossRef\]](#)
- Rish, I. An empirical study of the naïve Bayes classifier. *IJCAI* **2001**, *3*, 41–46.
- Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [\[CrossRef\]](#)
- Schalkoff, R.J. *Artificial Neural Networks*; McGraw-Hill: New York, USA, 1997.
- Zouhal, L.M.; Dencœur, T. An evidence-theoretic k -NN rule with parameter optimization. *IEEE Trans.* **1998**, *28*, 263–271. [\[CrossRef\]](#)
- Dietterich, T.G. *The Handbook of Brain Theory and Neural Networks*; The MIT Press: Cambridge, MA, USA, 2002.
- McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biol.* **1990**, *52*, 99–115. [\[CrossRef\]](#)
- Rosenblatt, F. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Rev.* **1958**, *65*, 386. [\[CrossRef\]](#)
- Rumelhart, D.E.; Geoffrey, E.H. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [\[CrossRef\]](#)
- Riess, J. Adaptive neural network control of cyclic movements using functional neuromuscular stimulation. *IEEE Trans.* **2000**, *8*, 42–52. [\[CrossRef\]](#) [\[PubMed\]](#)
- Albawi, S.; Mohammed, T.A.; Al-zawi, S. *Understanding of a Convolutional Neural Network*; ICET2017: Antalya, Turkey, 2017.
- Terrence, L.F. *Feedforward Neural Network Methodology*; Springer Science & Business Media: Berlin, Germany, 2006.
- Gao, S.C.; Zhou, M.C. Dendritic neuron with effective learning algorithms for classification, approximation, and prediction. *IEEE Trans.* **2019**, *30*, 601–614. [\[CrossRef\]](#)
- Todo, Y.; Tamura, H.; Yamashita, K.; Tang, Z. Unsupervised learnable neuron model with nonlinear interaction on dendrites. *Neural Netw.* **2014**, *60*, 96–103. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ji, J.K.; Gao, S.C.; Cheng, J. An approximate logic neuron model with a dendritic structure. *Neurocomputing* **2016**, *173*, 1775–1783. [\[CrossRef\]](#)
- Gardner, M.W.; Dorling, S.R. Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. *Atmos. Environ.* **1998**, *32*, 2627–2636. [\[CrossRef\]](#)
- Losonczy, A.; Makara, J. Compartmentalized dendritic plasticity and input feature storage in neurons. *Nature* **2008**, *452*, 436–441. [\[CrossRef\]](#) [\[PubMed\]](#)
- Losonczy, A.; Magee, J. Integrative properties of radial oblique dendrites in hippocampal CA1 pyramidal neurons. *Neuron* **2006**, *50*, 291–307. [\[CrossRef\]](#)
- Branco, T.; Häusser, M. The single dendritic branch as a fundamental functional unit in the nervous system. *Curr. Opin. Neurobiol.* **2010**, *20*, 494–502. [\[CrossRef\]](#)
- Yu, Y.; Wang, Y.R.; Gao, S.C.; Tang, Z. Statistical modeling and prediction for tourism economy using dendritic neural network. *Comput. Intell. Neurosci.* **2017**, *2017*, 9. [\[CrossRef\]](#) [\[PubMed\]](#)
- Tang, Y.J.; Ji, J.K.; Zhu, Y.L.; Gao, S.C.; Tang, Z.; Todo, Y. A differential evolution-oriented pruning neural network model for bankruptcy prediction. *Complexity* **2019**, *2019*, 21. [\[CrossRef\]](#)
- Sha, Z.J.; Hu, L.; Todo, Y.; Ji, J.K.; Gao, S.C.; Tang, Z. A breast cancer classifier using a neuron model with dendritic nonlinearity. *IEICE Trans. Inf. Syst.* **2015**, *98*, 1365–1376. [\[CrossRef\]](#)
- Jiang, T.; Gao, S.C.; Wang, D.Z.; Ji, J.K.; Todo, Y.; Tang, Z. A neuron model with synaptic nonlinearities in a dendritic tree for liver disorders. *IEEE Trans. Electr. Electron. Eng.* **2017**, *12*, 105–115. [\[CrossRef\]](#)
- Jia, D.B.; Yuka, F. Validation of large-scale classification problem in dendritic neuron model using particle antagonism mechanism. *Electronics* **2020**, *9*, 792. [\[CrossRef\]](#)
- Gao, S.C.; Zhou, M.C.; Wang, Z. Fully Complex-valued Dendritic Neuron Model. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**. [\[CrossRef\]](#) [\[PubMed\]](#)
- Jia, D.B.; Li, C.H. Application and evolution for neural network and signal processing in large-scale systems. *Complexity* **2021**, *2021*, 7. [\[CrossRef\]](#)
- Luo, X.D.; Wen, X.H.; Zhou, M.C.; Abusorrah, A. Decision-Tree-Initialized Dendritic Neuron Model for Fast and Accurate Data Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**. [\[CrossRef\]](#)
- Jia, D.B.; Dai, H.W. EEG processing in Internet of medical things using non-harmonic analysis: Application and evolution for SSVEP responses. *IEEE Access* **2019**, *7*, 11318–11327. [\[CrossRef\]](#)
- Jia, D.B.; Zheng, S.X. *A Dendritic Neuron Model with Nonlinearity Validation on Istanbul Stock and Taiwan Futures Exchange Indexes Prediction*; IEEE CCIS: Nanjing, China, 2018.
- Xu, W.X.; Li, C.H. *Optimizing the Weights and Thresholds in Dendritic Neuron Model Using the Whale Optimization Algorithm*; Journal of Physics: Conference Series; IOP Publishing: Beijing, China, 2021.
- Hecht-Nielsen, R. Theory of the backpropagation neural network. *Neural Netw. Percept.* **1992**, 65–93. [\[CrossRef\]](#)

35. Ji, J.K.; Song, S.B.; Tang, Y.J.; Gao, S.C.; Tang, Z.; Todo, Y. Approximate logic neuron model trained by states of matter search algorithm. *Knowl.-Based Syst.* **2019**, *163*, 120–130. [\[CrossRef\]](#)
36. Mirjalili, S. Genetic algorithm. In *Evolutionary Algorithms and Neural Networks*; Springer: Cham, Switzerland; New York, NY, USA, 2019; pp. 43–55.
37. Whitley, D. A genetic algorithm tutorial. *Stat. Comput.* **1994**, *4*, 65–85. [\[CrossRef\]](#)
38. Ali, K.; Neda, F. A new optimization method: Dolphin echolocation. *Adv. Eng. Softw.* **2013**, *59*, 53–70.
39. Soto, R.; Crawford, B.; Olivares, R. A reactive population approach on the dolphin echolocation algorithm for solving cell manufacturing systems. *Mathematics* **2020**, *8*, 1389. [\[CrossRef\]](#)
40. Dan, S. Biogeography-based optimization. *IEEE Trans.* **2008**, *12*, 702–713.
41. Zhang, Y.; Gu, X. Biogeography-based optimization algorithm for large-scale multistage batch plant scheduling. *Expert Syst. Appl.* **2020**, *162*, 113776. [\[CrossRef\]](#)
42. Höhfeld, M.; Rudolph, G. Towards a theory of population-based incremental learning. In Proceedings of the IEEE Conference on Evolutionary Computation, Indianapolis, IN, USA, 13–16 April 1997; pp. 1–5.
43. Li, Y.; Feng, X.; Wang, G. Application of Population Based Incremental Learning Algorithm in Satellite Mission Planning. In Proceedings of the International Conference on Wireless and Satellite Systems, Nanjing, China, 17–18 September 2020; Springer: Cham, Switzerland, 2020.
44. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; IEEE: Nagoya, Japan, 2002.
45. Wang, F.; Zhang, H.; Zhou, A. A particle swarm optimization algorithm for mixed-variable optimization problems. *Swarm Evol. Comput.* **2021**, *60*, 100808. [\[CrossRef\]](#)
46. Dorigo, M.; Birattari, M. Ant colony optimization. *IEEE Commun. Intell. Mag.* **2006**, *1*, 28–39. [\[CrossRef\]](#)
47. Paniri, M.; Dowlatabadi, M.B.; Nezamabadi-pour, H. MLACO: A multi-label feature selection algorithm based on ant colony optimization. *Knowl.-Based Syst.* **2020**, *192*, 105285. [\[CrossRef\]](#)
48. Karaboga, D. Artificial bee colony algorithm. *Scholarpedia* **2010**, *5*, 6915. [\[CrossRef\]](#)
49. Wang, H.; Wang, W.; Xiao, S.; Cui, Z.; Xu, M. Improving artificial bee colony algorithm using a new neighborhood selection mechanism. *Inf. Sci.* **2020**, *527*, 227–240. [\[CrossRef\]](#)
50. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
51. Jafari-Asl, J.; Seghier, M.E.; Ohadi, S. Efficient method using Whale Optimization Algorithm for reliability-based design optimization of labyrinth spillway. *Appl. Soft Comput.* **2021**, *101*, 107036. [\[CrossRef\]](#)
52. Heidari, A.A.; Mirjalili, S. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [\[CrossRef\]](#)
53. Khishe, M.; Mosavi, M.R. Chimp optimization algorithm. *Expert Syst. Appl.* **2020**, *149*, 113338. [\[CrossRef\]](#)
54. UCI Machine Learning Repository. Available online: <https://archive.ics.uci.edu/ml/index.php> (accessed on 23 October 2021).
55. Jugulum, R.; Taguchi, S. *Computer-Based Robust Engineering: Essentials for DFSS*; ASQ Quality Press: Milwaukee, WI, USA, 2004.