Hindawi
Discrete Dynamics in Nature and Society
Volume 2022, Article ID 3259222, 14 pages
https://doi.org/10.1155/2022/3259222

Research Article

Verification of Classification Model and Dendritic Neuron Model Based on Machine Learning

Dongbao Jia,[1,2] **Weixiang Xu**,[1] **Dengzhi Liu**,[1] **Zhongxun Xu**,[1] **Zhaoman Zhong**,[1] **and Xinxin Ban**[3]

[1]*School of Computer Engineering, Jiangsu Ocean University, Lianyungang 222005, China*
[2]*The MOE Key Laboratory of TianQin Project, Sun Yat-Sen University, Zhuhai 519082, China*
[3]*School of Environmental and Chemical Engineering, Jiangsu Ocean University, Lianyungang 222005, China*

Correspondence should be addressed to Zhaoman Zhong; zmzhong@jou.edu.cn and Xinxin Ban; banxx@jou.edu.cn

Received 19 March 2022; Revised 24 May 2022; Accepted 31 May 2022; Published 4 July 2022

Academic Editor: Shi Cheng

Artificial neural networks have achieved a great success in simulating the information processing mechanism and process of neuron supervised learning, such as classification. However, traditional artificial neurons still have many problems such as slow and difficult training. This paper proposes a new dendrite neuron model (DNM), which combines metaheuristic algorithm and dendrite neuron model effectively. Eight learning algorithms including traditional backpropagation, classic evolutionary algorithms such as biogeography-based optimization, particle swarm optimization, genetic algorithm, population-based incremental learning, competitive swarm optimization, differential evolution, and state-of-the-art jSO algorithm are used for training of dendritic neuron model. The optimal combination of user-defined parameters of model has been systemically investigated, and four different datasets involving classification problem are investigated using proposed DNM. Compared with common machine learning methods such as decision tree, support vector machine, *k*-nearest neighbor, and artificial neural networks, dendritic neuron model trained by biogeography-based optimization has significant advantages. It has the characteristics of simple structure and low cost and can be used as a neuron model to solve practical problems with a high precision.

1. Introduction

In the human brain, about tens of billions of interconnected neurons transmit signals through synapses to form a complex neural network to guide human behavior in the network. Neurons are composed of cell bodies with branched dendritic structures, cell membranes, and axons responsible for transmitting nerve signals. The first truly dominant concept of neural network established by scholars has only one neuron unit, known as binary McCulloch-Pitts neuron, which was proposed by McCulloch and Pitts in 1943 [1]. However, it does not consider the nonlinear transmission of cellular signals in dendritic neuron networks. Moreover, as the single-layer McCulloch Pitts neuron model cannot solve the basic nonlinear operation problem [2], it is also criticized as too simple.

Traditional neural networks generally believe that the connection between neurons is very complex, and the brain has strong computing and thinking ability. A single neuron does not need strong computing ability and only needs simple linear summation or nonlinear threshold operation in the process of signal transmission. As a result, the computational potential of individual neurons and their dendrites has been neglected for a long time.

Some researchers have proposed that dendrites perform more complex nonlinear operations in the process of signal transmission, which can improve the computing power of a single neuron [3, 4]. Koch et al. [5] hypothesized that the synaptic interaction at the branch turning point can be realized by Boolean logic operation, which means that the dendritic branch point is responsible for summarizing the current signals from the dendritic branch, its output is the input logic or, and each branch performs logic AND

operation on their synaptic input. However, as small differences in individual neuron morphology can lead to great changes in function, Koch model is difficult to distinguish different synaptic and dendritic morphology to solve specific complex problems. Therefore, the structure of synapses and dendrites requires a plasticity mechanism. Subsequent studies have found the phenomenon of neuronal plasticity, and an important progress in neuronal cell structure has been achieved by proposing neuronal pruning method [6, 7].

Compared with the widely used neural network model under the current mainstream view, the single dendritic neuron network model can deal with more complex nonlinear operations. The dendritic neural network model can carry out more complex nonlinear operation and obtain more accurate results with the same number of neurons. At present, the dendritic neural network model has been applied to many fields and achieved good results, such as live disorders [8], financial time series prediction [9, 10], and breast cancer classification [11–24].

In recent years, the research method of combining metaheuristic algorithm and neural network is more and more widely used. Its basic idea is to use metaheuristic algorithm to continuously adjust the corresponding parameters in neural network, guide the output after obtaining the optimal value, and then replace the obtained parameters into neural network for classification and prediction [25, 26]. The traditional dendritic neural network is optimized by backpropagation algorithm. The backpropagation algorithm is based on the chain derivation rule, which has the problems of falling into local traps, gradient disappearance, and so on [27–29]. This paper proposes a new dendrite neuron model (DNM), which combines metaheuristic algorithm and dendrite neuron model effectively.

Seven different metaheuristic algorithms are compared in the experiment, each of which has its own characteristics. Genetic algorithm (GA) [30–32] is an algorithm of finding the optimal solution based on the simulation of natural selection and genetic mechanism of biological evolution, whose main characteristic is to directly operate on the structure object and free from the restriction of derivation and function continuity. Biogeography-based optimization (BBO) [33–35] has been widely applied to simulate ecological concepts, well-known as its high prevision and strong stability using the representative metaheuristics. Particle swarm optimization (PSO) [36, 37] has been applied to train neural network instead of BP, whose whole searching and updating process follows the current optimal solution. Unlike genetic algorithm, all particles may converge to the optimal solution faster in most cases, and its advantage of evolutionary computation can deal with some problems of nondifferentiable node transfer function or no gradient information. Competitive swarm optimizer (CSO) [38, 39] is a simplified metaheuristic method and, which as a variant of PSO, is not only suitable for multi-point search, but also for local search. On this basis, the competition mechanism is applied, and it is not necessary to update the individual and global optimal value of position. Thus, CSO can balance the local-minimum trapping and convergence rate. Population-based incremental learning (PBIL) [40–42] selects the individuals with the highest fitness in each generation of the group to modify the learning probability and guides the generation of new individuals. Differential evolution (DE) [43, 44] is a stochastic model simulating biological evolution. As with other evolutionary algorithms, DE remains a global search strategy based on population, and for a further step, the process of genetic operation is simplified using real encoding, simple mutation operator, and competitive optimization mechanism. jSO algorithm is a new variant of DE algorithm, which is a state-of-the-art algorithm for single objective real-parameter optimization [45], and we, for the first time, introduce it to learn DNM.

The learning algorithm using BP is called DNM + BP, and DNM + BBO, DNM + PSO, DNM + GA, DNM + PBIL, DNM + CSO, DNM + DE, and DNM + jSO are similarly named. The effectiveness, accuracy, and convergence of each algorithm in classification problems are explored and demonstrated using four datasets. The experimental results show that DNM + BBO has fast convergence and high accuracy. At the same time, the classification accuracy of decision tree, KNN, support vector machine (SVM), MLP, and DNM + BBO is compared, and the results show that DNM + BBO has the highest classification accuracy. This paper effectively combines metaheuristic algorithm with dendritic neuron model to establish DNM + BBO, which provides an effective method to solve the classification problem.

2. Model and Learning Algorithms

2.1. Dendritic Neuron Model. The DNM is composed of four layers based on dendrite structure. In the synaptic layer, inputs $x_1, x_2, \ldots, x_n$ of each dendrite are firstly transformed using a sigmoid function. Secondly, in the dendrite layer, the outputs of the first layer are transmitted to a function of multiplication. Thirdly, membrane layer processes the received inputs from the dendrite layer. Finally, the signal from the membrane layer is transformed using another sigmoid function to accomplish the whole process [46]. Figure 1 shows the complete structure of DNM, and below are the details of this model.

2.1.1. Synaptic Layer. A synapse is the connection between neurons. Statistics flow from a synaptic neuron to another, which exhibits a feedforward pattern. The synapse has four connection states, namely, the excitatory connection, the inhibitory connection, constant 0 connection, and constant 1 connection. It depends on changes in the potential of the accepting neuron arising from ionotropic phenomena. The connecting function from the ith ($i = 1, 2, \ldots, n$) synaptic input to the jth ($j = 1, 2, \ldots, m$) synaptic layer is described as follows:

$$Y_{ij} = \frac{1}{1 + e^{-k\left(w_{ij}x_i - \theta_{ij}\right)}}. \tag{1}$$

Equation (1) expresses the transfer function Y_{ij} of the ith input of a synapse x_i varying from 0 to 1. k is constant. w_{ij} and θ_{ij} respectively denote the weight and threshold in synapse.

As for the values of w_{ij} and θ_{ij}, there are four different connections discussed in Figure 2. The X-axis indicates the

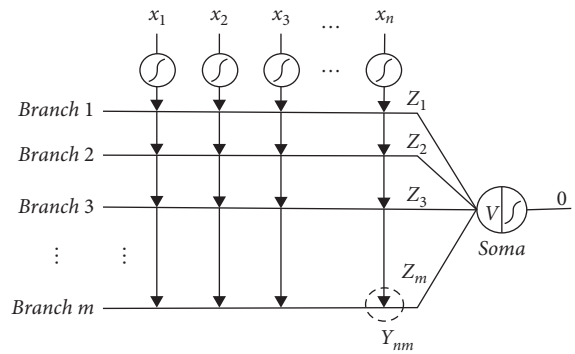

Figure 1: Structure of the dendritic neuron model.

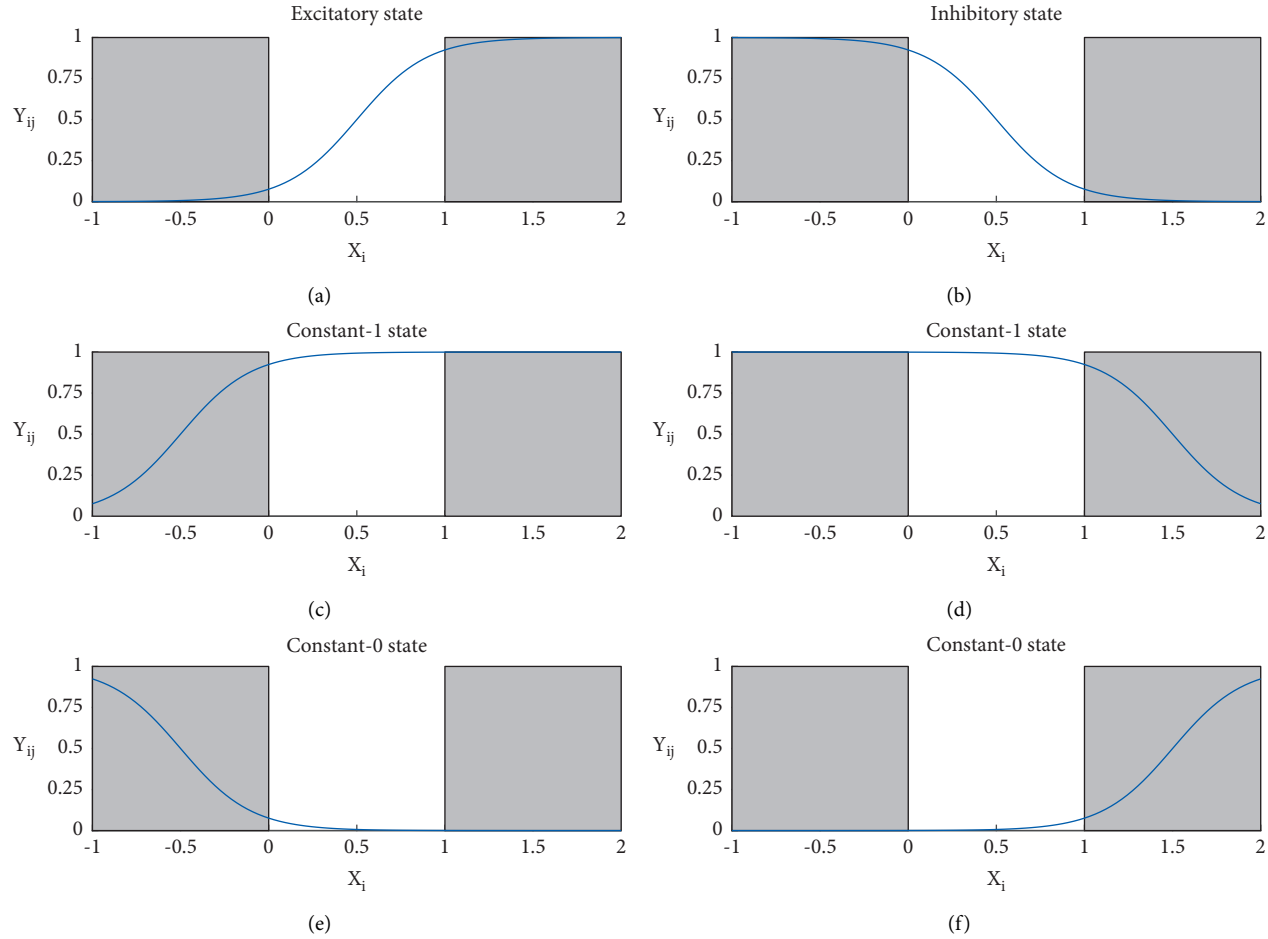

Figure 2: Four different connections in the synaptic layer: (a) excitatory connection, (b) inhibitory connection, (c) constant 1 connection, (d) constant 1 connection, (e) constant 0 connection, and (f) constant 0 connection.

input of the DNM, and the Y-axis indicates the output of the synaptic layer. Since the value of input x is from 0 to 1, only the blank part of each illustration is required to be focused on. The four connections include the following: Figure 2(a) presents excitatory connection, and when $0 < \theta_{ij} < w_{ij}$, the output is proportional to the input. On the contrary, Figure 2(b) depicts the inhibitory connection. As for $w_{ij} < \theta_{ij} < 0$, the output is inversely proportional to the input. Figures 2(c) and 2(d) present constant 1 connection, and when $\theta_{ij} < 0 < w_{ij}$ or $\theta_{ij} < w_{ij} < 0$, regardless of the value of the input, x varies between 0 and 1, and the output is always 1; Figures 2(e) and 2(f) present constant 0 connection, and when $w_{ij} < 0 < \theta_{ij}$ or $0 < w_{ij} < \theta_{ij}$, regardless of the value of the input, x varies between 0 and 1, and the output is always 0.

2.1.2. Dendrite Layer. In this layer, outputs from the synapses are multiplicated altogether. As a method of describing nonlinearity features, multiplication is the first selection due

to its simplicity. In addition, if we take constant 0 or 1 connection as an example, this function is equivalent to the logical AND operator for their similar output values. The output formula for the jth dendrite is as follows:

$$Z_j = \prod_{i=1}^{n} Y_{ij}. \tag{2}$$

2.1.3. Membrane Layer. This layer represents the summary of signals coming from each dendritic branch. The input of the next layer is obtained by a sum function, which resembles a logical OR operator. Then, the processed signal will be transmitted to the soma body. Thus, the output of the membrane layer is as follows:

$$V = \sum_{j=1}^{m} Z_j. \tag{3}$$

2.1.4. Soma Layer. Finally, the received signal in the soma layer is taken as the input of another sigmoid function. The detailed formula for this layer is as follows:

$$O = \frac{1}{1 + e^{-k_s(V-\theta_s)}}, \tag{4}$$

where k_s is a positive constant, and the range of threshold θ_s is [0, 1].

2.2. Backpropagation. Backpropagation (BP) is the gradient descent method [47, 48]. This algorithm contributes to reducing the error between the target output and its real value through neural network training. The error can be expressed as follows:

$$E = \frac{1}{2}(T - O)^2, \tag{5}$$

where T is the target output vector, and O is the actual output vector. By modifying the parameters w_{ij} and θ_{ij} of the DNM model in the process of learning, the error can be decreased. The updated expressions are set as follows:

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

where E_p is the mean square error. After computing these two increments, we can get values of w_{ij} and θ_{ij} at the next moment through the following:

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

where η denotes the learning rate defined by users. t is a characterization of learning times. Furthermore, the partial differentials of E_p regarding w_{ij} and θ_{ij} are calculated as follows:

$$\begin{aligned} \frac{\partial E_p}{\partial w_{ij}} &= \frac{\partial E_p}{\partial O_p} \frac{\partial O_p}{\partial V} \frac{\partial V}{\partial Z_j} \frac{\partial Z_j}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial w_{ij}}, \\ \frac{\partial E_p}{\partial \theta_{ij}} &= \frac{\partial E_p}{\partial O_p} \frac{\partial O_p}{\partial V} \frac{\partial V}{\partial Z_j} \frac{\partial Z_j}{\partial Y_{ij}} \frac{\partial Y_{ij}}{\partial \theta_{ij}}. \end{aligned} \tag{8}$$

The detailed results of the above partial differentials in DNM are given as follows:

$$\begin{aligned} \frac{\partial E_p}{\partial O_p} &= T_p - O_p \\ \frac{\partial O_p}{\partial V} &= \frac{k_s x_i e^{-k_s(w_{ij}x_i - \theta_s)}}{\left(1 + e^{-k_s(w_{ij}x_i - \theta_s)}\right)^2} \\ \frac{\partial V}{\partial Z_j} &= 1 \\ \frac{\partial Z_j}{\partial Y_{ij}} &= \prod_{l=1 an\ dl\ i}^{n} Y_{lj} \\ \frac{\partial Y_{ij}}{\partial w_{ij}} &= \frac{k x_i e^{-k(w_{ij}x_i - \theta_{ij})}}{\left(1 + e^{-k(w_{ij}x_i - \theta_{ij})}\right)^2} \\ \frac{\partial Y_{ij}}{\partial \theta_{ij}} &= \frac{k e^{-k(w_{ij}x_i - \theta_{ij})}}{\left(1 + e^{-k(w_{ij}x_i - \theta_{ij})}\right)^2}. \end{aligned} \tag{9}$$

When computing $\Delta w_{ij}(t)$ and $\Delta\theta_{ij}(t)$, the chain rule is applied, and layer-by-layer calculation is performed throughout the DNM.

2.3. Biogeography-Based Optimization. Biogeography-based optimization derives from biogeography, which investigates the speciation, extinction, and geographical distribution in nature. Each habitat is regarded as a solution and the principal task is to obtain the best one. For convenience, the model introduces mathematics, using high habitat suitability index (HSI) as the degree of fitness among species. And the suitability index variables (SIV) are utilized for describing various aspects of HSI [49]. Procedures using BBO are implemented as follows:

(1) Initializing the integer sequence SIV based on the current habitat H_i ($i = 1, 2, \ldots, n$).

(2) Calculating the HSI of each habitat according to the following formula.

$$HSI(H_i) = \frac{1}{2P} \sum_{p=1}^{P} \left(T_p - O_p\right)^2, \tag{10}$$

where P represents the total number of training samples. T_p is the target vector of the pth sample, and O_p is the actual output vector determined by H_i.

(3) Implementing random selection on the SIV and migration among habitats occurs in the case that the emigration rate and immigration rate are μ_i and λ_i respectively.

$$\mu_i = \frac{E_i}{m}$$
$$\lambda_i = I\left(1 - \frac{i}{m}\right), \tag{11}$$

where E is the emigration rate, I is the maximum immigration rate, and m is the rank of habitat. These two parameters are set as $E = I = 1$ in this research, and λ and μ are constrained as follows:

$$\lambda_i + \mu_i = E. \tag{12}$$

(4) For each habitat H_i, the immigrated his, and the probability Ps_i, it contains the Sth species of habitat that are updated:

$$Ps_i(t + \Delta t) = Ps_i(t)(1 - \lambda_i \Delta t - \mu_i \Delta t) + Ps_{i-1}\mu_{i-1}\Delta t + Ps_{i+1}\mu_{i+1}\Delta t. \tag{13}$$

If t is small enough to be considered as 0, the following equation can be approximated:

$$Ps_i = \begin{cases} -(\lambda_i + \mu_i)Ps_i + \mu_{i+1}Ps_{i+1} & i = 0 \\ -(\lambda_i + \mu_i)Ps_i + \lambda_{i-1}Ps_{i-1} + \mu_{i+1}Ps_{i+1} & 1 \le i \le n-1 \\ -(\lambda_i + \mu_i)Ps_i + \lambda_{i-1}Ps_{i-1} & i = n \end{cases}. \tag{14}$$

(5) Mutating nonelite habitats according to the mutation rate Pm_i:

$$Pm_i = Pm_{\max}\frac{1 - Ps_i}{Ps_{\max}}, \tag{15}$$

where $Ps_{\max}$ is the maximum value of Ps_i, and $Pm_{\max}$ is the parameter.

(6) Go to Step 2 again and perform the next iteration if needed. Not until the termination criterion is met does this procedure end.

2.4. Particle Swarm Optimization. Particle swarm optimization mimics the search behavior of a flock of birds and consists of particles, each of which represents a possible solution [50]. The solution includes two attributes: speed and position. The former indicates moving rate of each particle, and the latter indicates its moving direction. Each particle moves to the optimal value separately and spontaneously and memorizes the current value for the particle itself (*pbest*). Then, it shares the individual optimal solution with other particles and obtains the global extreme value (*gbest*). All particles update their two attributes according to these two extreme values. PSO is widely adopted as its simple operating process, for instance, MLP [51]. The whole process of PSO being used to search for the optimal values of weights and thresholds in the synaptic layer of DNM can be described as follows:

$$X_i = \left\{x_i^1, x_i^2, \ldots, x_i^m\right\} = \left\{\omega_{11}, \omega_{12}, \ldots \omega_{MN}, \theta_{11}, \theta_{12}, \ldots \theta_{MN}\right\}, \tag{16}$$

where X_i ($i = 1, 2, \ldots, Q$) indicates the ith individual in the swarm, and Q denotes the number of particles. Besides, we use the mean square error (MSE) (X_i) to calculate the error of the last layer with output X_i as follows:

$$MSE(X_i) = \frac{1}{2P}\sum_{p=1}^{P}\left(T_p - O_p\right)^2. \tag{17}$$

2.5. Genetic Algorithm. Genetic Algorithm is inspired by natural selection. According to the previous study [30], a set of probable solutions is meant as individuals in optimization study. Good individuals tend to reproduce at a relatively high rate, while poor individuals have a relatively low reproductive rate. With the evolution of population, individuals develop in a healthier direction. However, as GA is a random searching algorithm, chances are that worse individuals are generated from fitter ones in the existing framework. Thus, we adopt the elite strategy to maintain optimized individuals in this scheme. Training DNM can also use GA. Similar to PSO, a chromosome in GA for training DNM can be exhibited using equation (16), where X_i ($i = 1, 2, \ldots, Q$) indicates the ith chromosome in the population, and Q represents the population size. We use single point crossover to update individuals. Moreover, the fitness function is the same as equation (17).

2.6. Population-Based Incremental Learning. Estimation of distribution algorithm (EDA) is a population-based strategy that tracks the statistical information of the candidate solution population for optimization [30, 41, 52]. It uses a solution of discarding at least a part of the population in each generation and using a sample according to the statistic quantity of high-fitness individuals in the current population to generate new populations, and the process is repeated from generation to generation.

Similar to EDA, PBIL is an extension of a univariate marginal distribution algorithm. When it functions as the optimizer of an m-dimensional binary problem, PBIL uses a probability vector p with m dimensions, whose kth value of p describes how likely this element is equal to one. In each generation, a random population is generated using the probability vector p probabilistically. Then, the fitness of each candidate solution is calculated. By adjusting the probability vector, the next generation is more likely to resemble the most suitable individual. After getting this new probability vector, using p to create another candidate solution to the random population, continue this process until the termination requirement is met.

Table 1: Details of the classification datasets.

Classification datasets	# of attributes	# of training samples	# of test samples	# of classes
Banknote authentication	5	960	412	2
Breast cancer	10	489	210	2
Car evaluation	7	1210	518	2
Diabetic retinopathy	17	364	156	2

2.7. Competitive Swarm Optimization. Competitive swarm optimization is a population algorithm used to solve large-scale classification problems, and the velocity of individual movement is not eliminated, same as PSO. It introduces a competitive strategy to make a comparison between two selected particles according to their evaluated results. As only the lost particles can learn to participate in iterating, except for reducing the number of updated particles to $2/N$ [53], it is not necessary to save the excellent solution in search, which can be applied to the effective solution of large-scale classification problems. Below are the operation steps:

(1) N represents the solution at the beginning, and the particle position x_{i} ($i = 1, 2, \ldots, N$) and velocity v_i ($i = 1, 2, \ldots, N$) of generated particles are initialized.

(2) All solutions are evaluated.

(3) The kth ($k = 1, 2, \ldots, 2/N$) competition for generation t occurs as follows:

(a) Nonrepeating particles N_{k1} and N_{k2} are selected from the undecided particles randomly.

(b) Positions of selected particles of N_{k1} and N_{k2} are compared and evaluated to determine the won particle and the failed particle.

(c) The $x_{\mathrm{l,k}}$ of failed particle is updated by the application of its velocity $v_{i,k}$.

$$\begin{aligned} v_{l,k}(t+1) &= R_1(k,t)v_{l,k}(t) \\ &\quad + R_2(k,t)\left[x_{w,k}(t) - x_{l,k}(t)\right] \\ &\quad + \varphi R_3(k,t)\left[\overline{x_k}(t) - x_{l,k}(t)\right] \\ x_{l,k}(t+1) &= x_{l,k}(t) + v_{l,k}(t+1), \end{aligned} \tag{18}$$

where $R_1(k, t)$, $R_2(k, t)$, and $R_3(k, t)$ are vectors with their elements varying from 0 to 1. $\overline{x_k}(t)$ represents the mean position of all particles, and φ represents at which degree the influence of the mean quantity takes effect, which has been advised as the following equation in reference to the existing researches:

$$\begin{cases} \varphi = 0 \; N \le 100 \\ \varphi \in [0.14\log(N) - 0.3,\ 0.27\log(N) - 0.51] \text{ otherwise} \end{cases}. \tag{19}$$

(d) Repeat the three steps above until all particles are identified.

(4) The next iteration starting with step 2 is operated until the parameter t reaches the maximum.

Table 2: Experimental environment of the DNM.

Item	Computing environment
CPU	3.00 GHz intel (R) core (TM) i5-8500
OS	Windows 10 education
RAM	16.0 GB
Software	MATLAB R2018b

2.8. Differential Evolution. Differential evolution is a random search method inspired by biological evolution, and highly fit individuals are preserved through iterations. As a variant of genetic algorithm, it is a global search strategy based on population and adopts real coding, basic mutation from one-to-one difference to simplify the genetic operation. Furthermore, its memory ability enables DE to track the real-time situation and modify the search strategy dynamically. Owing to the prominent global convergence ability and stability, DE is well applied to the solution of complicated optimization problems, which are difficult to resolve using traditional mathematical programming methods. At present, DE has been used in artificial neural network, signal processing, biological information, and other fields.

2.9. jSO. jSO is the latest improved algorithm of differential evolution, which is based on iL-SHADE algorithm [54]. It keeps parameter strategy based on historical memory and linear population size reduction strategy of iL-SHADE. jSO adopts adaptive strategy to improve the mutation coefficient M_{F} and crossover probability M_{CR}, which are the key parameters of differential evolution algorithm, and the effect is obvious. The mutation strategy of jSO is as follows:

$$\vec{v}_{i,g} = \vec{x}_{i,g} + F_w\left(\vec{x}_{p\mathrm{Best},g} - \vec{x}_{i,g}\right) + F\left(\vec{x}_{r_1,g} - \vec{x}_{r_2,g}\right), \tag{20}$$

$$F_w = \begin{cases} 0.7 * F, nfes < 0.2\mathrm{max\_}nfes \\ 0.8 * F, nfes < 0.4\mathrm{max\_}nfes \\ 1.2 * F, \text{otherwise} \end{cases}, \tag{21}$$

where *nfes* is the current population iteration, and *max_nfes* is the max population iteration.

3. Experiment

In this experiment, four classification problems are used to verify the performance of DNM with the eight learning algorithms mentioned above. Table 1 shows their attributes, number of training samples, number of test samples, and

Table 3: Initial parameters used in learning algorithms.

Algorithm	Parameter	Value
BP	Learning rate	0.01
	Maximum number of generations	1000
BBO	Habitat modification probability	1
	Immigration probability bounds per gene	[0, 1]
	Step size for numerical integration of probabilities	1
	Max immigration and max emigration	1
	Mutation probability	0.005
	Population size	50
	Maximum number of generations	1000
PSO	Acceleration constants	[2]
	Inertia weights	[0.9, 0.5]
	Population size	50
	Maximum number of generations	1000
GA	Selection mechanism	Roulette wheel
	Crossover probability	0.9
	Mutation probability	0.1
	Population size	50
	Maximum number of generations	1000
PBIL	Learning rate	0.05
	Good population member	1
	Bad population member	0
	Elitism parameter	1
	Mutational probability	0.1
	Population size	50
	Maximum number of generations	1000
CSO	The swarm size m	$m\begin{cases}1500\,(d \geq 5000)\\1000\,(d \geq 2000)\\500\,(d \geq 1000)\\250\,(d \geq 500)\\100\text{otherwise}\end{cases}$
	The social factor φ	$\varphi = \begin{cases}0.2\,(d \geq 2000)\\0.15\,(d \geq 1000)\\0.1\,(d \geq 500)\\0\text{otherwise}\end{cases}$
	Population size	50
	Maximum number of generations	1000
DE	Crossover probability	0.9
	Differential weight	0.5
	Population size	50
	Maximum number of generations	1000
jSO	Mutation coefficient M_F	0.3
	Crossover probability M_{CR}	0.5
	Historical memory size H	5
	Population size	50
	Maximum number of generations	1000

Table 4: Reasonable combination of four DNM parameters for four tested problems.

Datasets	M	k	k_s	θ_s
Banknote authentication	20	15	15	0.5
Breast cancer	20	10	5	0.3
Car evaluation	20	15	15	0.5
Diabetic retinopathy	15	5	1	0.1

Table 5: Average accuracy and standard deviation of each learning algorithm.

Datasets	Learning algorithm	Learning's average accuracy (%) ± standard deviation	Test's average accuracy (%) ± standard deviation
Banknote authentication	DNM + BP	63.06 ± 21.63	62.62 ± 21.43
	DNM + BBO	**100 ± 0.00**	**99.43 ± 0.39**
	DNM + PSO	99.84 ± 0.21	98.45 ± 0.79
	DNM + GA	94.11 ± 1.95	92.95 ± 2.42
	DNM + PBIL	93.11 ± 1.06	92.43 ± 2.01
	DNM + CSO	99.94 ± 0.08	99.03 ± 0.42
	DNM + DE	92.05 ± 1.27	91.25 ± 1.86
	DNM + jSO	99.94 ± 0.29	99.39 ± 0.36
Breast cancer	DNM + BP	90.67 ± 3.00	89.84 ± 3.12
	DNM + BBO	**98.92 ± 0.35**	95.97 ± 1.42
	DNM + PSO	97.61 ± 0.58	95.25 ± 1.81
	DNM + GA	97.25 ± 0.64	95.71 ± 1.38
	DNM + PBIL	96.19 ± 0.64	95.10 ± 1.42
	DNM + CSO	98.48 ± 0.34	**96.14 ± 1.23**
	DNM + DE	95.75 ± 0.61	95.25 ± 1.67
	DNM + jSO	97.73 ± 0.68	95.83 ± 1.21
Car evaluation	DNM + BP	63.06 ± 21.63	62.62 ± 21.43
	DNM + BBO	**100 ± 0.00**	**99.43 ± 0.39**
	DNM + PSO	99.84 ± 0.20	98.45 ± 0.79
	DNM + GA	94.11 ± 1.95	92.95 ± 2.42
	DNM + PBIL	93.11 ± 1.06	92.43 ± 2.01
	DNM + CSO	99.94 ± 0.08	99.03 ± 0.42
	DNM + DE	92.05 ± 1.27	91.25 ± 1.86
	DNM + jSO	96.93 ± 1.07	96.49 ± 1.21
Diabetic retinopathy	DNM + BP	38.66 ± 1.05	37.99 ± 2.44
	DNM + BBO	**97.49 ± 0.81**	**94.85 ± 1.75**
	DNM + PSO	90.63 ± 2.30	88.72 ± 3.84
	DNM + GA	87.72 ± 2.84	85.64 ± 3.08
	DNM + PBIL	85.42 ± 2.33	84.21 ± 3.22
	DNM + CSO	94.29 ± 1.41	92.22 ± 2.21
	DNM + DE	82.96 ± 2.48	83.10 ± 2.79
	DNM + jSO	87.49 ± 3.89	86.28 ± 4.90

Table 6: Accuracy of DNM + BBO and other machine learning methods for the datasets of banknote authentication, breast cancer, car evaluation, and diabetic retinopathy.

Datasets	DNM + BBO (%)	Decision tree (%)	SVM (%)	KNN (%)	MLP (%)
Banknote authentication	**99.4**	91.6	97.9	97.2	87.1
Breast cancer	96.0	93.4	**96.6**	94.7	93.8
Car evaluation	**99.4**	92.5	89.8	93.1	82.6
Diabetic retinopathy	**94.9**	89.4	88.3	81.7	87.8

Table 7: Average running time (sec) of each algorithm in different datasets.

Learning algorithm	Banknote authentication	Breast cancer	Car evaluation	Diabetic retinopathy
BP	10	20	3	15
BBO	30	70	8	60
PSO	54	128	9	92
GA	58	133	12	93
PBIL	60	142	12	104
CSO	28	63	6	45
DE	112	251	22	181
jSO	109	113	181	115

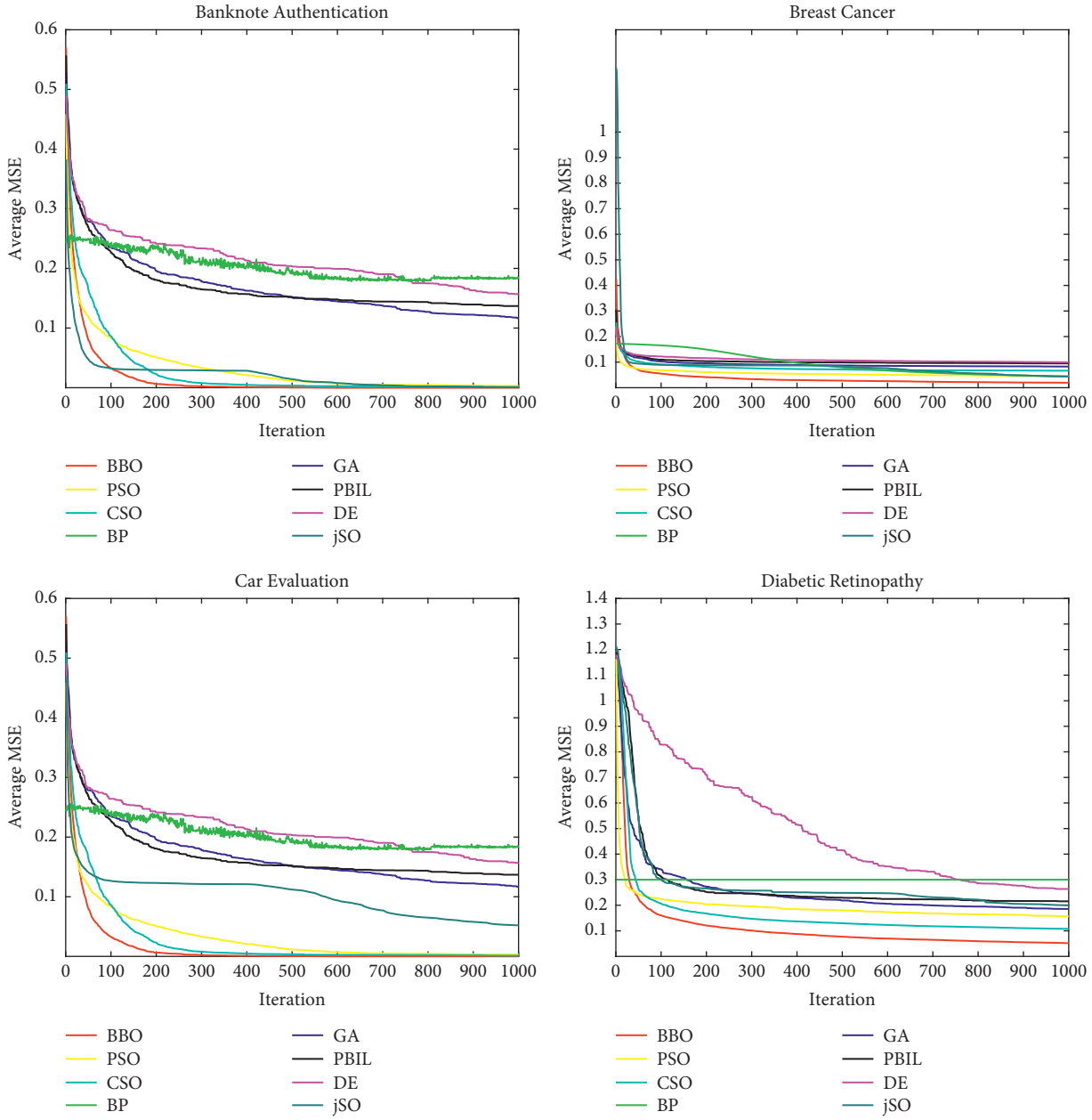

Figure 3: Convergence graphs of the learning algorithms for the datasets of banknote authentication, breast cancer, car evaluation, and diabetic retinopathy, respectively.

number of classes. The classification datasets are acquired from the open datasets of the UCI Machine Learning Repository in various aspects [55].

For each learning algorithm, the maximum generation number is set as 1000. Each data set includes two parts, with learning data accounting for 70% and testing data accounting for 30%. In addition, the characteristics of any classification problem are expressed by numbers with no data error that contain negative numbers and decimal numbers. As the input x of DNM varies from 0 to 1, each characteristic data set is normalized in the corresponding range for the experiment.

All the experimental results are averaged from 30 independent experiments, and the accuracy of the expected output is calculated according to the classification results. Equation (22) is used to calculate the accuracy with true positivity (TP), false positive (FP), true negative (TN), and false negative (FN). Besides, MSE is utilized as the evaluation function using equation (17) for each learning algorithm.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100\%. \tag{22}$$

As for the experimental equipment and operating rate, the experimental environment is shown in Table 2. The design of experiment adopts a statistical strategy for the effective analysis of large combinations using orthogonal arrays based on Latin square. The mentioned eight learning algorithms are tested under above situations, and owing to

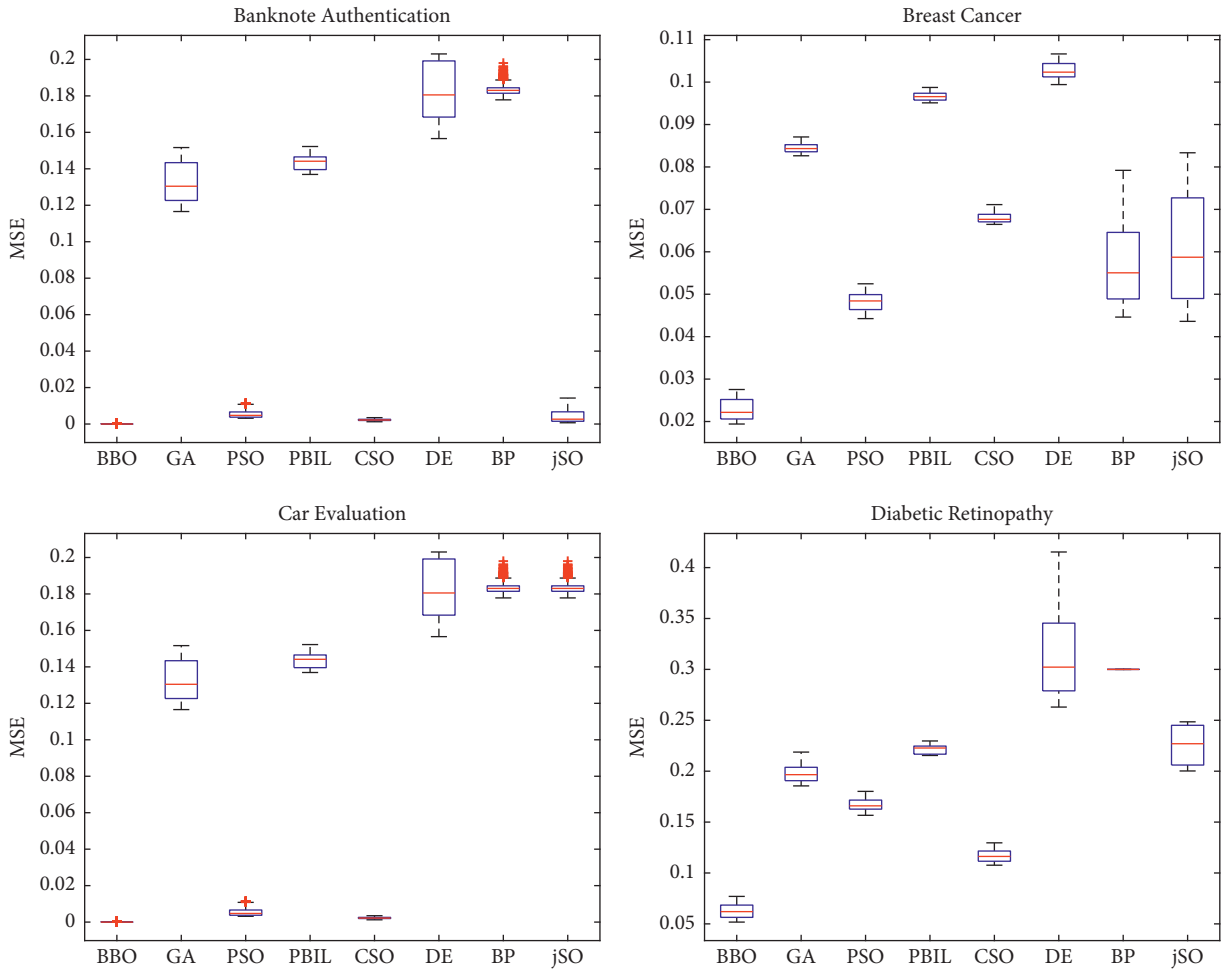

Figure 4: Solution distribution of DNM for the datasets of banknote authentication, breast cancer, car evaluation, and diabetic retinopathy, respectively.

the adoption of orthogonal arrays, the number of experiments can be greatly reduced by the relationship between the factors and levels.

From the previous research, the performance of the learning algorithm can be significantly enhanced by carefully selecting parameter values in some preliminary experiments [56–58]. According to experience and algorithm characteristics, Table 3 shows parameters setting for each algorithm. The population size is set to 50, and the maximum number of generations of each algorithm is 1000 uniformly, while other parameters are set empirically according to characteristics of each algorithm.

For obtaining the optimal performance of DNM, user-defined parameters are well worth investigating. There are four key parameters in DNM, that is, the number of dendrites in the model (M), the synaptic parameter in the connecting sigmoid function (k), and two soma parameters (k_s and θ_s) in the output sigmoid function.

The reasonable combination of four DNM parameters is obtained by Taguchi method [59–61]. It scans a portion of possible combinations among factors rather than the whole combination, resulting in minimal experimental runs and optimal estimation of factors during execution [62, 63]. Referring to relevant previous study and research experience, the hierarchy number of four factors is set as follows: five levels for $M \in \{3, 5, 10, 15, 20\}$; five levels for $k \in \{1, 5, 10, 15, 20\}$; five levels for $k_s \in \{1, 5, 10, 15, 25\}$; and five levels for $\theta_s \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Different from full factor analysis, which requires $5^4 = 625$ trials, the orthogonal array can obviously reduce the number of experiments and time cost. Hence, the orthogonal array $L_{25}(5^4)$, which involves only 25 experiments, is utilized in this time.

The supplementary material (available here) summarizes the experimental results, where MSE represents the mean square error values of the eight learning algorithms (i.e., BP, BBO, PSO, GA, PBIL, CSO, DE, and jSO) for four datasets. According to the experimental results of each dataset, we obtained acceptable user-defined parameter settings respectively, as shown in Table 4.

4. Result

The average accuracy and standard deviation of learning algorithms in each dataset are summarized in Table 5. It is obvious that regardless of which dataset, DNM + BBO achieves the highest accuracy among all the comparison

Table 8: Average accuracy, standard deviation, average MSE, and corresponding optimal parameters of each learning algorithm in different datasets.

Learning algorithm	Datasets	M	k	k_s	θ_s	Average accuracy (%) ± standard deviation	Average MSE
BP	Banknote authentication	5	5	1	0.1	96.93 ± 2.05	3.33E − 02
	Breast cancer	20	10	5	0.3	89.84 ± 3.13	4.46E − 02
	Car evaluation	15	5	1	0.1	98.31 ± 1.03	2.80E − 02
	Diabetic retinopathy	20	10	5	0.3	66.32 ± 23.63	1.13E − 01
BBO	Banknote authentication	20	15	15	0.5	99.43 ± 0.39	3.44E − 07
	Breast cancer	20	10	5	0.3	95.97 ± 1.42	1.94E − 02
	Car evaluation	20	15	15	0.5	99.43 ± 0.39	3.44E − 07
	Diabetic retinopathy	15	5	1	0.1	94.85 ± 1.75	5.18E − 02
PSO	Banknote authentication	20	1	25	0.9	98.79 ± 0.75	2.76E − 03
	Breast cancer	20	1	25	0.9	95.24 ± 1.64	4.13E − 02
	Car evaluation	20	1	25	0.9	98.79 ± 0.75	2.76E − 03
	Diabetic retinopathy	15	5	1	0.1	88.72 ± 3.84	1.57E − 01
GA	Banknote authentication	10	20	15	0.7	94.89 ± 2.03	7.82E − 02
	Breast cancer	20	15	15	0.5	95.22 ± 1.52	5.54E − 02
	Car evaluation	10	20	15	0.7	94.90 ± 2.03	7.82E − 02
	Diabetic retinopathy	15	5	1	0.1	85.64 ± 3.08	1.86E − 01
PBIL	Banknote authentication	5	1	25	0.9	94.09 ± 1.74	9.72E − 02
	Breast cancer	20	1	25	0.9	94.30 ± 1.27	7.41E − 02
	Car evaluation	5	15	15	0.5	94.09 ± 1.74	9.72E − 02
	Diabetic retinopathy	15	5	1	0.1	84.21 ± 3.22	2.07E − 01
CSO	Banknote authentication	20	1	25	0.9	98.91 ± 0.66	8.98E − 04
	Breast cancer	20	1	25	0.9	95.49 ± 1.19	3.14E − 02
	Car evaluation	20	1	25	0.9	98.91 ± 0.66	8.98E − 04
	Diabetic retinopathy	15	5	1	0.1	92.22 ± 2.21	1.14E − 01
DE	Banknote authentication	3	15	10	0.5	93.50 ± 1.70	1.10E − 01
	Breast cancer	3	10	5	0.3	94.75 ± 0.81	6.86E − 02
	Car evaluation	3	15	10	0.5	93.50 ± 1.69	1.10E − 01
	Diabetic retinopathy	15	10	5	0.3	90.89 ± 3.72	9.31E − 02
jSO	Banknote authentication	15	1	25	0.9	99.39 ± 0.36	6.95E − 04
	Breast cancer	3	1	25	0.9	95.83 ± 1.21	4.36E − 02
	Car evaluation	5	1	25	0.9	96.49 ± 1.21	5.21E − 02
	Diabetic retinopathy	20	15	15	0.5	86.28 ± 4.90	2.00E − 01

object, and some even reach 100%, while DNM + BP is the lowest. Otherwise, the accuracy of DNM + CSO is also higher, but inferior to that of DNM + BBO and higher than that of DNM + PSO. Moreover, the accuracy of DNM + GA, DNM + jSO, DNM + PBIL, and DNM + DE with little difference is relatively common.

Furthermore, the upper limit of m is set as 20 in this experiment, but both optimum parameters m of DNM + BBO and DNM + CSO are 15, three quarters of the upper limit, with a high accuracy of classification as shown in Table 5. As a result, for classification problems, DNM learning by metaheuristics may not require an extremely large number of m for problems with a small number of features.

Table 6 shows the accuracy comparison results of DNM + BBO and other common machine learning classification methods on four datasets, and DNM + BBO has obvious advantages. Using the Machine Learning Toolbox of MATLAB to realize decision tree, SVM, and KNN. Table 7 shows the average running time (sec) of each algorithm in four datasets. BP runs the fastest with the worst classification performance. Among metaheuristic algorithms, CSO is the best, and BBO is the second, while DE and jSO have bad performance.

Figure 3 presents the average convergence graph of each learning algorithm for the datasets of banknote authentication, breast cancer, car evaluation, and diabetic retinopathy, respectively. It is evident that the value of each learning algorithm converges to the final iteration time, but BBO converges the fastest in all cases. The multipoint search BBO maintains an elite habitat, changing the solution in each iteration, which produces a higher number of new candidate optimal solutions than any other learning algorithms.

By deriving a new solution from a candidate optimal solution at a certain time, the convergence rate of the high-quality solution can be accelerated. As a result, BBO has basically converged to the minimum value of MSE even in the 200 iterations in case of banknote authentication and car evaluation.

On the contrary, the MSE of BP is not significantly affected by the local solution in all datasets, indicating that BP is easily trapped in the local minima. Compared with the multipoint search method, BP has the disadvantages of insufficient problem orientation and easy deviation from the local solution. Therefore, different from other algorithms, we believe that BBO has the characteristics of obtaining smaller

Table 9: Overall statistical comparison obtained by Friedman test of seven learning algorithms for four datasets.

Algorithm	Average ranking	p_{Bonf}	$\alpha = 0.05$
BBO	1.53	—	—
CSO	2.46	0.088	Yes
PSO	4.38	0.002	Yes
GA	4.36	0.032	Yes
jSO	4.41	0.047	Yes
BP	5.1	0.057	Yes
PBIL	5.43	0.001	Yes
DE	5.58	0.006	Yes

MSE under fewer iterations, which emphasizes the unique advantages of DNM over the state-of-the-art methods.

Moreover, Figure 4 depicts the average solution distribution of 30 independent runs of each learning algorithm for four datasets, respectively. BBO has the best stability that often finds stable solutions, whereas GA, DE, jSO, or BP usually varies widely across different runs. In addition, it can be easily found that the minimum MSE for each dataset is BBO, and the maximum MSE for each dataset is BP.

According to the supplementary material (available here), the average accuracy, standard deviation, average MSE, and corresponding optimal parameter of each learning algorithm in different datasets are summarized in Table 8.

Table 9 shows the overall statistical results of eight learning algorithms for four problems via the Friedman test at the level of $\alpha = 0.05$ with the application of Bonferroni-Dunn procedures. Friedman test indicates whether there is a difference in the average rank between various ordinal variables. Post hoc test is Bonferroni-Dunn procedure, whose adjusted p value is p_{Bonf}. The result shows that BBO has excellent robustness in each problem and performs better than other methods in terms of average accuracy, standard deviation, and average MSE. This proves that BBO has higher stability and is not easily affected by the problem. Furthermore, CSO is the algorithm with the best results except BBO.

On the contrary, results of BP are the worst, which vary greatly with different datasets, indicating that its accuracy, reliability, and stability are not good and are easily affected by problems. Consequently, the accuracy of any learning algorithm will be biased according to the compatibility of the problem and the combination of parameters, and in terms of convergence and stability of MSE, especially for convergence speed, BBO is reliable and has obvious advantages.

5. Discussion

The experimental results show that DNM + BBO has significant advantages over the other seven optimization algorithms (BP, PSO, GA, PBIL, CSO, DE, and jSO) and other machine learning algorithms (decision tree, SVM, KNN, and MLP). This is due to the mechanism of BBO algorithm, through interspecies migration and intraspecies mutation, the feature information of different habitats has changed, and the dominant features of habitats have been shared; it avoids a large number of local solutions in DNM training. Although DNM + BBO shows great potential in classification problems, there are many challenges in practical applications, such as redundancy in data and more irrelevant information, which reduce the performance and increase the computational complexity of machine learning classification algorithms. For high-dimensional heterogeneous data, we will continue to solve the problems of DNM in future research.

6. Conclusion

With the advent of the era of big data, researches on high-precision models with simple structure and low cost are developing rapidly in solving complicated problems. In this paper, considering the synaptic nonlinearity, a new learning algorithm based on the DNM model is put forward to solve complex classification optimization problems. Adopting factor allocation and orthogonal array, eight learning algorithms (i.e., BP, BBO, PSO, GA, PBIL, CSO, DE, and jSO) are used to train the DNM systematically for four datasets. And the effectiveness, stability, classification accuracy, and convergence speed of these algorithms are compared and demonstrated for such problems.

The experimental results show that BP cannot find the global optimal weight and threshold since its inherent local minimum trap problem, and its effect and accuracy are extremely limited. And the performance of BBO is the most competitive. No matter what kind of dataset is used, the stability, accuracy, and convergence speed of BBO are the most excellent and obviously superior to those of other algorithms. Moreover, the comprehensive performance of CSO is the best except for BBO. The performance of PSO is second only to that of CSO. For GA, PBIL, DE, and state-of-the-art jSO, although they are slightly better than BP, the results vary greatly, while the datasets are different. In addition, compared with common machine learning methods such as decision tree, support vector machine, k-nearest neighbor, and artificial neural networks, dendritic neuron model trained by biogeography-based optimization has significant advantages.

Therefore, this paper combines metaheuristic algorithm and dendrite neuron model effectively, and DNM + BBO is established in this study, especially as a powerful algorithm to solve classification and optimization problems. In the future, we will further explore and try to expand the output range of DNM, apply it to a wider range of fields to solve other different problems, and continue to pursue improving its performance [64].

Data Availability

The datasets of this paper are obtained from the following website: https://archive.ics.uci.edu/ml/index.php.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (NSFC) under Grant nos. 12105120, 21805106, 62102169 and 72174079, the Natural Science Foundation of Jiangsu Province under Grant BK20181073, the MOE Key Laboratory of TianQin Project, Sun Yat-sen University, and the Open Fund Project of Jiangsu Institute of Marine Resources Development under Grant nos. JSIMR202018.

Supplementary Materials

Experimental results of eight learning algorithms (i.e., BP, BBO, PSO, GA, PBIL, CSO, DE, and jSO) over 30 independent runs for four datasets. Tables S1–S4 show the average accuracy of each learning algorithm for the full parameter combination in each dataset. Furthermore, Tables S5–S8 depict the parameter sensitivity results based on factor assignment and orthogonal array of each learning algorithm for each dataset. (*Supplementary Materials*)

References

[1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[2] S. Haykin, *Neural Networks*, A Comprehensive Foundation, Upper Saddle River, NJ, USA, 1994.

[3] Y. Todo, H. Tamura, K. Yamashita, and Z. Tang, "Unsupervised learnable neuron model with nonlinear interaction on dendrites," *Neural Networks*, vol. 60, pp. 96–103, 2014.

[4] Y. G. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: a review and roadmap," *Mechanical Systems and Signal Processing*, vol. 138, Article ID 106587, 2020.

[5] C. Koch, T. Poggio, and V. Torre, "Nonlinear interactions in a dendritic tree: localization, timing, and role in information processing," *Proceedings of the National Academy of Sciences*, vol. 80, no. 9, pp. 2799–2802, 1983.

[6] R. C. Paolicelli, G. Bolasco, F. Pagani et al., "Synaptic pruning by microglia is necessary for normal brain development," *Science*, vol. 333, no. 6048, pp. 1456–1458, 2011.

[7] L. K. Low and H. J. Cheng, "Axon pruning: an essential step underlying the developmental plasticity of neuronal connections," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 361, no. 1473, pp. 1531–1544, 2006.

[8] Z. Zhang, Z. Li, Y. Zhang, Y. Luo, and Y. Li, "Neural-dynamic-method based dual-arm CMG scheme with time-varying constraints applied to humanoid robots," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3251–3262, 2015.

[9] H. T. He, S. C. Gao, T. Jin, S. Sato, and X. Y. Zhang, "A seasonal-trend decomposition-based dendritic neuron model for financial time series prediction," *Applied Soft Computing*, vol. 108, Article ID 107488, 2021.

[10] W. Chen, J. Sun, S. Gao, J. J. Cheng, J. Wang, and Y. Todo, "Using a single dendritic neuron to forecast tourist arrivals to Japan," *IEICE - Transactions on Info and Systems*, vol. E100.D, no. 1, pp. 190–202, 2017.

[11] S. C. Gao, M. C. Zhou, Y. R. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–614, 2019.

[12] S. C. Gao, M. Z. Zhou, Z. Q. Wang et al., "Fully Complex-Valued Dendritic Neuron Model," *IEEE transactions on neural networks and learning systems*, pp. 1–14, 2021, In Press.

[13] Y. Yu, Z. Y. Lei, Y. R. Wang, T. Zhang, C. Peng, and S. Gao, "Improving dendritic neuron model with dynamic scale-free network-based differential evolution," *IEEE/CAA Journal of automatica sinica*, vol. 9, no. 1, pp. 99–110, 2022.

[14] Z. Xu, Z. Q. Wang, J. Y. Li, T. Jin, X. Meng, and S. Gao, "Dendritic neuron model trained by information feedback-enhanced differential evolution algorithm for classification," *Knowledge-Based Systems*, vol. 233, Article ID 107536, 2021.

[15] R. D. Cazé, S. Jarvis, A. J. Foust, and S. R. Schultz, "Dendrites enable a robust mechanism for neuronal stimulus selectivity," *Neural Computation*, vol. 29, no. 9, pp. 2511–2527, 2017.

[16] S. M. Almufti, R. Boya Marqas, and V. Ashqi Saeed, "Taxonomy of bio-inspired optimization algorithms," *Journal of Advanced Computer Science & Technology*, vol. 8, no. 2, 23 pages, 2019.

[17] Q. Kang, B. Huang, and M. Zhou, "Dynamic behavior of artificial Hodgkin–Huxley neuron model subject to additive noise," *IEEE Transactions on Cybernetics*, vol. 46, no. 9, pp. 2083–2093, 2016.

[18] Z. J. Sha, L. Hu, Y. Todo, J. Ji, S. Gao, and Z. Tang, "A breast cancer classifier using a neuron model with dendritic nonlinearity," *IEICE - Transactions on Info and Systems*, vol. E98.D, no. 7, pp. 1365–1376, 2015.

[19] R. Legenstein and W. Maass, "Branch-specific plasticity enables selforganization of nonlinear computation in single neurons," *Journal of Neuroscience*, vol. 31, no. 30, pp. 10787–10802, 2011.

[20] J. Bono, K. A. Wilmes, and C. Clopath, "Modelling plasticity in dendrites: from single cells to networks," *Current Opinion in Neurobiology*, vol. 46, pp. 136–141, 2017.

[21] D. B. Jia, K. Yanagisawa, Y. Ono et al., "Multiwindow non-harmonic analysis method for gravitational waves," *IEEE Access*, vol. 6, pp. 48645–48655, 2018.

[22] D. B. Jia, K. Yanagisawa, M. Hasegawa et al., "Time-frequency-based non-harmonic analysis to reduce line noise impact for LIGO observation system," *Astronomy and computing*, vol. 25, pp. 238–246, 2018.

[23] W. X. Xu, C. H. Li, Y. X. Dou et al., "Optimizing the weights and thresholds in dendritic neuron model using the whale optimization algorithm," *Journal of Physics: Conference Series*, vol. 2025, no. 1, Article ID 012037, 2021.

[24] D. B. Jia, H. W. Dai, Y. Takashima et al., "EEG Processing in internet of medical things using non-harmonic analysis: application and evolution for SSVEP responses," *IEEE Access*, vol. 7, pp. 11318–11327, 2019.

[25] M. M. Ghiasi, S. Zendehboudi, and A. A. Mohsenipour, "Decision tree-based diagnosis of coronary artery disease: CART model," *Computer Methods and Programs in Biomedicine*, vol. 192, Article ID 105400, 2020.

[26] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Let a biogeography based optimizer train your multi-layer perceptron," *Information sciences*, vol. 269, pp. 188–209, 2014.

[27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Internal Representations by Error Propagation*, University of California, San Diego, CA, USA, 1985.

[28] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.

[29] Y. Yu, S. C. Gao, Y. Wang, and Y. Todo, "Global optimum-based search differential evolution," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 2, pp. 379–394, 2019.

[30] T. V. Mathew, *Genetic Algorithm*, IIT Bombay, Adi Shankaracharya Marg, Powai, Mumbai, Maharashtra, India, 2012.

[31] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[32] B. Jafrasteh and N. Fathianpour, "A hybrid simultaneous perturbation artificial bee colony and back-propagation algorithm for training a local linear radial basis neural network on ore grade estimation," *Neurocomputing*, vol. 235, pp. 217–227, 2017.

[33] S. Dan, "Biogeography-based optimization," *IEEE transaction on evolutionary computation*, vol. 12, pp. 702–713, 2008.

[34] R. M. Li, Y. F. Huang, and J. Wang, "Long-term traffic volume prediction based on K-means Gaussian interval type-2 fuzzy sets," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, pp. 1–8, 2019.

[35] D. B. Jia, Y. Fujishita, C. H. Li, Y. Todo, and H. W. Dai, "Validation of large-scale classification problem in dendritic neuron model using particle antagonism mechanism," *Electronics*, vol. 9, no. 5, 792 pages, 2020.

[36] F. Wang, H. Zhang, and A. Zhou, "A particle swarm optimization algorithm for mixed-variable optimization problems," *Swarm and Evolutionary Computation*, vol. 60, Article ID 100808, 2021.

[37] D. B. Jia, C. H. Li, Q. Liu et al., "Application and evolution for neural network and signal processing in large-scale systems," *Complexity*, vol. 2021, no. 7, Article ID 6618833, 2021.

[38] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Transaction on cybernetics*, vol. 45, no. 2, pp. 191–204, 2014.

[39] A. H. Khan, X. W. Cao, S. Li, V. N. Katsikis, and L. Liao, "BAS-ADAM: an ADAM based approach to improve the performance of beetle antennae search optimizer," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 461–471, 2020.

[40] B. A. Norman and A. E Smith, "Random keys genetic algorithm with adaptive penalty function for optimization of constrained facility layout problems," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation*, pp. 407–411, Indianapolis, IN, USA, 1997.

[41] Y. Li, X. Feng, and G. Wang, "Application of population based incremental learning algorithm in satellite mission planning," in *International Conference on Wireless and Satellite Systems*Springer, Cham, 2020.

[42] L. Grippo, A. Manno, and M. Sciandrone, "Decomposition techniques for multilayer perceptron training," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 11, pp. 2146–2159, 2016.

[43] A. Kaveh and N. Farhoudi, "A new optimization method: dolphin echolocation," *Advances in Engineering Software*, vol. 59, pp. 53–70, 2013.

[44] R. Soto, B. Crawford, R. Olivares et al., "A reactive population approach on the dolphin echolocation algorithm for solving cell manufacturing systems," *Mathematics*, vol. 8, no. 9, 1389 pages, 2020.

[45] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: algorithm jSO," in *Proceedings of the 2017 IEEE congress on Evolutionary Computation (CEC)*, pp. 1311–1318, Donostia, Spain, June 2017.

[46] X. X. Qian, Y. R. Wang, S. C. Gao, and K. Todo, "Mr2DNM: A Novel Mutual Information-Based Dendritic Neuron Model," *Computational intelligence and neuroscience*, vol. 2019, Article ID 7362931, 2019.

[47] R. Chakraborty and N. R. Pal, "Feature selection using a neural framework with controlled redundancy," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 1, pp. 35–50, 2015.

[48] R. V. Gandhi and D. M. Adhyaru, "Takagi-Sugeno fuzzy regulator design for nonlinear and unstable systems using negative absolute eigenvalue approach," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 482–493, 2020.

[49] F. Gabbiani, H. G. Krapp, C. Koch, and G. Laurent, "Multiplicative computation in a visual neuron sensitive to looming," *Nature*, vol. 420, no. 6913, pp. 320–324, 2002.

[50] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory MHS'95," in *Proceedings of the Sixth International symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.

[51] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 2, pp. 997–1006, 2004.

[52] P. Larrañaga, "A Review on Estimation of Distribution Algorithms," *Estimation Of Distribution Algorithms*, vol. 2, pp. 57–100, 2002.

[53] W. X. Xu, C. H. Li, D. B. Jia et al., "A dendritic neuron model for disease prediction," in *Proceedings of the 2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications*, pp. 1118–1122, Dalian, China, August 2021.

[54] J. Brest, M. S. Maučec, and B. Bošković, "iL-SHADE: Improved L-SHADE Algorithm for Single Objective Real-Parameter Optimization," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1188–1195, Vancouver, BC, Canada, July 2016.

[55] Uci Machine Learning Repository, 2022, https://archive.ics.uci.edu/ml/index.php.

[56] J. J. Cheng, G. Y. Yuan, M. C. Zhou et al., "Accessibility analysis and modeling for IoV in an urban scene," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4246–4256, 2020.

[57] J. J. Cheng, G. Y. Yuan, M. C. Zhou, S. Gao, C. Liu, and H. Duan, "A fluid mechanics-based data flow model to estimate VANET capacity," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 2603–2614, 2020.

[58] J. J. Cheng, X. Wu, M. C. Zhou, S. C. Gao, Z. H. Huang, and C. Liu, "A novel method for detecting new overlapping community in complex evolving networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 9, pp. 1832–1844, 2019.

[59] G. Taguchi and R. Jugulum, *Computer-based Robust Engineering: Essentials for DFSS*, ASQ Quality Press, Mexico, 2004.

[60] J. J. Cheng, M. J. Chen, M. C. Zhou, S. C. Gao, C. N. Liu, and C Liu, "Overlapping community change-point detection in an evolving network," *IEEE transactions on big data*, vol. 6, no. 1, pp. 189–200, 2020.

[61] J. Sun, S. C. Gao, H. W. Dai, J. Cheng, M. Zhou, and J. Wang, "Bi-objective elite differential evolution algorithm for multivalued logic networks," *IEEE Transactions on Cybernetics*, vol. 50, no. 1, pp. 233–246, 2020.

[62] J. F. C. Khaw, B. S. Lim, and L. E. Lim, "Optimal design of neural networks using the Taguchi method," *Neurocomputing*, vol. 7, no. 3, pp. 225–245, 1995.

[63] S. C. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE transactions on systems, man, and cybernetics: Systems*, vol. 51, no. 6, pp. 3954–3967, 2021.

[64] W. X. Xu, D. B. Jia, Z. M. Zhong, C. H. Li, and Z. X. Xu, "Intelligent dendritic neural model for classification problems," *Symmetry*, vol. 14, no. 1, 11 pages, 2021.