

Research Article

Semisupervised SVM Based on Cuckoo Search Algorithm and Its Application

Ziping He , Kewen Xia , Wenjia Niu, Nelofar Aslam , and Jingzhong Hou

School of Electronics and Information Engineering, Hebei University of Technology, Tianjin 300401, China

Correspondence should be addressed to Kewen Xia; kwxia@hebut.edu.cn

Received 12 June 2018; Revised 15 July 2018; Accepted 14 August 2018; Published 13 September 2018

Academic Editor: Francesco Riganti-Fulginei

Copyright © 2018 Ziping He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Semisupervised support vector machine (S3VM) algorithm mainly depends on the predicted accuracy of unlabeled samples, if lots of misclassified unlabeled samples are added to the training will make the training model performance degrade. Thus, the cuckoo search algorithm (CS) is used to optimize the S3VM which also enhances the model performance of S3VM. Considering that the cuckoo search algorithm is limited to the local optimum problem, a new cuckoo search algorithm based on chaotic catfish effect optimization is proposed. First, use the chaotic mechanism with high randomness to initialize the nest for range expansion. Second, chaotic catfish nest is introduced into the effective competition coordination mechanism after falling into the local optimum, so that the candidate's nest can jump out of the local optimal solution and accelerate the convergence ability. In the experiment, results show that the improved cuckoo search algorithm is effective and better than the particle swarm optimization (PSO) algorithm and the cuckoo search algorithm on the benchmark functions. In the end, the improved cuckoo search algorithm is used to optimize semisupervised SVM which is applied into oil layer recognition. Results show that this optimization model is superior to the semisupervised SVM in terms of recognition rate and time.

1. Introduction

Semisupervised learning [1] studies how to improve learning performance by using labeled and unlabeled samples simultaneously. It has become a research focus and hotspot in pattern recognition and machine learning. In such a mixed data learning process, the sample distribution information of the unlabeled dataset is transferred to the final learning model, so that the trained learning model has better performance. Literature [2] proposes a novel graph for semisupervised discriminant analysis, which is called combined low-rank and k-nearest neighbor (LRKNN) graph, and map the data to the LR feature space and then the KNN is adopted to satisfy the algorithmic requirements of Semisupervised Discriminant Analysis (SDA). Semisupervised support vector machine is first proposed by Professor V. Vapnik [3] when the labeled samples are not enough, and it is difficult to achieve satisfactory performance; support vector machine (SVM) can use the transductive learning to improve the performance. It can be regarded as the generalization of the SVM in the unlabeled samples. S3VM has received extensive attention in recent

years. Qing Wu et al. [4] described a smooth piecewise function and research smooth piecewise semisupervised SVM and used a converging linear PSO to train semisupervised SVM to get better classification accuracy. Luo et al. [5] introduced semisupervised learning for the least squares support vector machine (LSSVM) algorithm to improve the accuracy of model predictions and is used to predict the distribution of porosity and sandstone in the Jingzhou study area. Literature [6] puts forward a method with the name of Ensemble S3VM which deals with the unknown distribution by ensemble learning and applies the algorithm to ground cover classification for polarimetric synthetic aperture radar images.

Cuckoo search [7] algorithm is a new nature-inspired metaheuristic based on the obligate brood parasitic behavior of cuckoo species [8] and the Levy flight search mechanism to effectively solve the optimization problem [9]. In order to further optimize the CS algorithm, many experts and scholars have studied it. When Srivastava et al. [10] investigate local search in Levy flights, they enter tabu search idea to avoid falling into local optimum, which is successfully applied

to solve the problem of automatic generation of software test data. Yang and Deb [11] proposed a multitarget cuckoo search, which was applied to engineering optimization and achieved good results. Zhang et al. [12] prescribed a modified adaptive cuckoo search (MACS) to improve the performance of CS. Wang et al. [13] used an evaluation strategy based on a dimension-by-dimension update for the progress of the iteration of the improved algorithm and proposed an enhanced CS algorithm. In literature [14], a new local-enhanced cuckoo search algorithm is designed aiming to deal with some multimodal numerical problems. Literature [15] considers utilizing multiple chaotic maps simultaneously to perform the local search within the neighborhood of the global best solution found by the chaotic cuckoo search algorithm.

The above-improved versions of CS algorithm jumped out of the local optimal or improve the convergence speed of the algorithm. In view of the fact that CS algorithm is coming to the end of the iteration, population groups tend to converge too early and lead to local optimum. First, we use chaos mapping instead of general randomization to initialize nest position, it makes the initial nest location not only has the distribution randomness but also strengthens the diversity of the bird's nest distribution. Second, according to the literature [16], they used catfish effect to optimize artificial bee colony (ABC) algorithm in order to obtain a good ability to jump out of local optimum, we applied the catfish effect to CS algorithm, added it to the nest, and then get the chaotic catfish nest. Consequently, it improves the convergence rate of the whole population and the shortage of the algorithm into a local optimum. Finally, the CS algorithm based on the above-improved strategy is used to optimize the SVM algorithm and apply it to the oil layer recognition and establish a new semisupervised oil layer recognition model, expecting a better recognition of oil layers.

2. Cuckoo Search Algorithm and Its Improvement

2.1. Cuckoo Search Algorithm Principle

2.1.1. Cuckoo Breeding Behavior. According to the long-term observations of an entomologist, cuckoo has adopted a special breeding strategy parasitic brood [8]. It lays eggs in the nests of other birds and allows other birds to hatch. In order to reduce the possibility of being discovered by host birds, the cuckoo will choose the host birds that are basically alike in eating habits and easy to imitate ovately and color. When it flies to a nest, it only produces one and removes the host's egg before spawning, or all out of the nest, forcing the host to lay eggs again. Once the cuckoo's hatchlings hatch, it has the habit of bringing the host chicks out of the nest, thus enjoying the host bird's tending. But when the host birds find their nests have foreign eggs, they also throw the parasitic eggs or abandon their nests and build a nest in other places.

2.1.2. Levy Flights. Various studies have shown that the flight behavior of many animals and insects has demonstrated

the typical characteristics of Levy flights [17–19]. A study by Reynolds and Frye shows that fruit flies or *Drosophila melanogaster* explore their landscape using a series of straight flight paths punctuated by a sudden 90° turn, leading to a Levy-flight-style intermittent scale-free search pattern. Studies on human behavior such as the Ju/'hoansi hunter-gatherer foraging patterns also show the typical feature of Levy flights. Even light can be related to Levy flights [20]. And when the target location is random and sparsely distributed, Levy flight is the best search strategy for M independent search seekers.

Levy flight belongs to one type of random walk, and the walking step satisfies a stable distribution of heavy-tailed. In this form of walking, short distance exploration and occasional long distance walk alternate. Levy flight in intelligent optimization algorithm can expand search scope, increase population diversity and make it easier to jump out of local optimum.

Subsequently, such behavior has been applied to optimization and optimal search, and preliminary results show its promising capability [18].

2.1.3. Cuckoo Search. The cuckoo search algorithm is based on the parasitic propagation mechanism of cuckoo bird and Levy flights search principle. It is mainly based on the following three ideal rules:

- (1) Each cuckoo lays one egg at one time and selects a nest randomly to hatch it.
- (2) The best nests will be preserved to the next generation in a randomly selected group of nests.
- (3) The number of nests available is fixed, and the probability that the host bird of a nest will find the exotic bird's egg is p_a , $p_a \in [0, 1]$.

On the basis of the above three ideal rules, the routing and location updating formula of cuckoo's nest is as follows:

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{levy}(\lambda) \quad (1)$$

In the case, α is the size of step, $\alpha=1$ in general. \oplus is point multiplication, and $\text{levy}(\lambda)$ is the search path.

The Levy flight essentially provides a random walk while the random step length is drawn from a levy distribution as shown in the following formula:

$$\text{Levy} \sim \mu = t^{-\lambda}, \quad 1 < \lambda \leq 3 \quad (2)$$

which has an infinite variance with an infinite mean. Here the steps essentially form a random walk process with a power-law step-length distribution with a heavy tail. Some of the new solutions should be generated by Levy walk around the best solution obtained so far; this will speed up the local search. However, a substantial fraction of the new solutions should be generated by far-field randomization and whose location should be far enough from the current best solution, this will make sure the system will not be trapped in a local optimum.

To sum up, the main steps of the CS algorithm can be described as follows.

Step 1. The objective function is $f(X)$, $X = (x_1, \dots, x_d)^T$, initialization group, and generates the initial position of N bird's nest X_i , ($i = 1, 2, \dots, n$) randomly.

Step 2. Calculate the objective function value of each nest and record the current best solution.

Step 3. Keep the location of the best nest in the previous generation, and update the position of other bird's nest according to formula (1).

Step 4. Comparing the existing bird's nest with the previous generation, if it is better, it takes it as the best position at present.

Step 5. Using a random number R compare with p_a , if $R > p_a$, then change the nest position randomly, and get a new set of nest position.

Step 6. If the end condition is not met, return Step 2.

Step 7. Output global optimal position.

2.2. The Cuckoo Search Algorithm Based on Chaotic Catfish Effect

2.2.1. Initial Chaotic Nest. The initial nest location of the bird's nest is an important link in the CS algorithm. It not only affects the convergence speed of the algorithm but also restricts the quality of the final solution of the problem. According to the randomness, regularity, and mutual correlation of chaotic systems, we use chaos mapping to extract and capture information in the solution space to enhance the diversity of the initial bird nest location. In this paper, the logistic chaotic mapping is adopted, and its mathematical iterative equation is as follows:

$$\lambda_{t+1} = \mu \times \lambda_t (1 - \lambda_t), \quad t = 0, 1, 2, \dots, T \quad (3)$$

Among them, T is a presupposed maximum number of chaotic iterations and λ_t is a random number distributed uniformly on the interval (0, 1), because the initial value cannot take any of the fixed points; otherwise it is a stable orbit and cannot generate chaos, so $\lambda_0 \notin \{0, 0.25, 0.5, 0.75, 1\}$. μ is chaos control parameters; the system will be in a state of complete chaos when $\mu=4$.

Firstly, a set of chaotic variables is produced by using formula (3); secondly, we map the location of n, d dimension nest X_i to the chaotic interval $[F_{min}, F_{max}]$ by the chaotic sequence once according to formula (4). F_{min} and F_{max} represent the largest and the smallest boundary of F_i , respectively. The average accuracy α_{k-cv} is the fitness function, which is based on the k-fold cross-validation approach. The distribution of nest is shown in Figure 1.

Finally, the fitness of each nest was calculated.

$$f_{i,j} = f_{min,j} + \lambda_{i,t} (f_{max,j} - f_{min,j}) \quad (4)$$

2.2.2. Updating the Catfish New Nest. The catfish effect is derived from a few catfish placed in the sardine sink by Norway merchants to drive and stir the sardine group, making the sardines more likely to survive. In other words, the catfish nests can guide nests trapped in a local optimum onto a new region of the search space and thus to potentially better nest solutions. In literature [21], the catfish effect is applied to the particle swarm optimization, which improves the algorithm's ability to solve. In literature [16], the catfish effect is fused into the artificial bee colony algorithm to enhance the probability of obtaining the global optimal solution. Literature [22], combined with catfish effect and cloud model in particle swarm optimization, increased particle diversity and improved accuracy and finally is applied to the circuit breaker, effectively improving the design efficiency. Literature [23] uses the catfish effect to improve the bat algorithm to attract the population to jump out of the current local optimal solution, so as to maintain population diversity and get better global search ability and high convergence speed.

In the CS algorithm, if the "limit" times do not update the nest position, then we can see that the algorithm has been trapped in the local optimal solution, which leads to the algorithm failure which cannot get the globally optimal solution, and only at this local solution stagnation. According to literature [16]'s idea of chaotic catfish bee, this paper proposes a chaotic catfish nest, in order to jump out of the local optimal solution which is trapped in the local optimal solution and finally converge to the global optimal solution. The concrete steps are as follows.

Step 1 (sort and mark). The corresponding fitness values of the current n bird's nest were arranged in ascending order, marking the last 10% bird nest as the worst bird nest position.

Step 2 (the nest of a chaotic catfish is produced). Set the initial position vector of the nest of the chaotic catfish $V_m^d(0) = (v_m^1(0), v_m^2(0), \dots, v_m^d(0))$, $m = 1, 2, \dots, M$ in accordance with the following formula:

$$v_m^j(0) = \begin{cases} v_{m,j}^{max}, & r > 0.5 \\ v_{m,j}^{min}, & r \leq 0.5 \end{cases} \quad (5)$$

Among them, $v_{m,j}^{max}$ and $v_{m,j}^{min}$ are the maximum and minimum value of the j dimension of the m catfish wasp, respectively. r is a random number in the [0, 1] interval, and 0.5 is the boundary point to make the distribution of v_m^j approximate at the two polar values.

Step 3 (update the nest of the chaotic catfish). The position of the chaotic catfish nest is updated with formula (6), in which T is a chaotic iterative threshold.

$$V_m^d(t) = V + (V_{max} - V_{min}) \times V_m^d(t), \quad V_m^d \in [V_{max}, V_{min}] \quad (6)$$

$$V_m^d(t+1) = \mu \times V_m^d(t) \times (1 - V_m^d(t)),$$

$$t = 0, 1, \dots, T$$

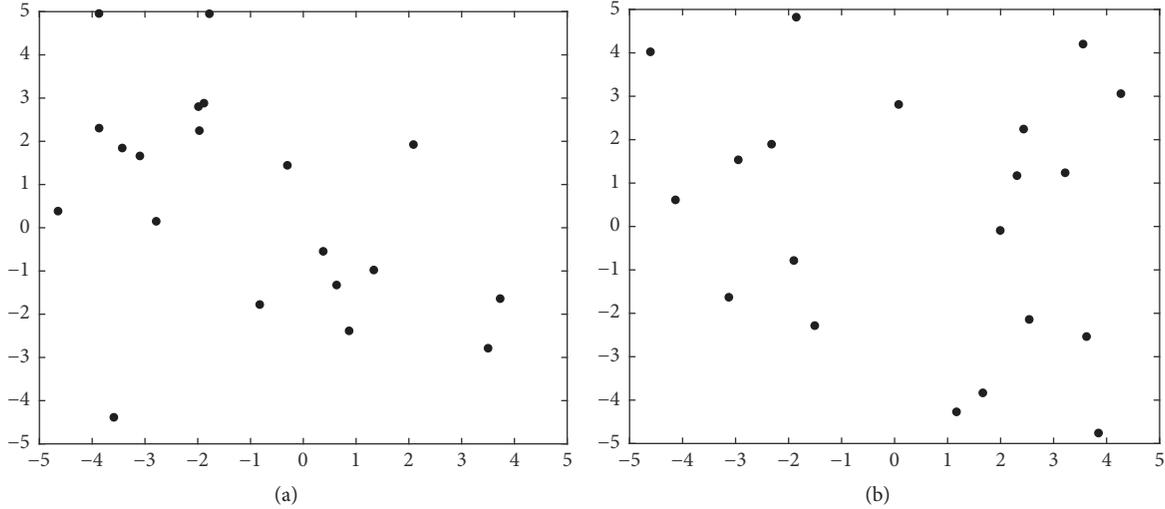


FIGURE 1: Distribution of nest: (a) distribution of random nest; (b) distribution of chaotic nest.

Step 4 (the nest of a chaotic catfish is introduced). The nest which is marked as the worst bird nest position is abandoned, and the nest of the same amount of chaotic catfish bird's nest is placed in the original nest of the worst bird nest.

Step 5 (update the location of the bird's nest). The fitness value of the chaotic catfish nests was calculated by using formula (4), and the new nest location and corresponding fitness value were updated to enter the next iteration.

Finally, the steps of cuckoo search algorithm based on chaotic catfish effect (CCCS) can be shown in Algorithm 1.

Among them, $Count$ is a column vector of $N \times 1$. If the optimal solution is updated at i_{th} iteration, then $Count(i) = 1$, otherwise $Count(i) = 0$. c is the number of successive 0 in the vector $Count$, if $c \geq limit$, the algorithm can be found to be in the local optimal, and the new nest of catfish is added to replace some inferior nests in order to help the algorithm jump out of the local optimal.

2.3. Simulation Test and Analysis

2.3.1. Benchmark Functions. In order to verify the effectiveness and generalization of the improved algorithm, we compare the improved algorithm with CS algorithm [7], particle swarm optimization Algorithm(PSO) [24], and chaotic gravitation search algorithm (CGSA) that the first method to incorporate chaos into gravitation search algorithm (GSA) in literature [25] on 8 benchmark functions. The 8 benchmark functions and their parameter settings are shown in Table 1.

2.3.2. Parameter Settings. In our experiment, the dimension's (Dim) size for each function was 2, and the corresponding maximum number of iteration was 1000. We carried out 30 independent runs for CS, CCCS, PSO, and CGSA.

The parameter settings of CS, CCCS, PSO, and CGSA were assigned as follows: size of population/nest $size = 20$, maximum iteration $Iteration_{max} = 1000$, $P_{\alpha} = 0.25$, the

largest renewal number of bird's nest $limit = 5$, and the largest iteration number of chaos $T = 300$.

2.3.3. Experimental Results and Discussion. The evolution curves of the best fitness value of the standard CS, CCCS, PSO, and CGSA when the maximum iteration number is 1000 is shown in Figure 2, for each kind of algorithm, the program runs 30 times independently.

The experiment selected Schaffer, Schwefel, Sphere, and Sum of different power four unimodal functions and Ackley, Rastrigrin, Branin, and Griewank four multimodal functions. Firstly, we can see that when testing on unimodal functions from Figure 2, the CCCS algorithm can find the optimal solution for most benchmark functions in the number of iterations that are far less than 1000 times. For example, when using the Sphere function, the optimal solution is obtained only in the 750th iteration. Secondly, the CCCS algorithm can get the best solution for $f_2, f_3, f_4, f_5, f_6, f_7, f_8$. For example, the optimal solution is obtained in the 957st iteration when using the Griewank function, however, CS, PSO, and CGSA cannot find the optimal solution within 1000 iterations. Thirdly, when multimodal functions are selected, because of the complexity of multimodal functions and the existence of multiple local minima, the CCCS algorithm still has a strong ability to find the global optimal solution, and the convergence speed of the CCCS algorithm is better than the other three algorithms. For instance, the CCCS algorithm can find the optimal solution for all multimodal functions, such that the optimal solution is obtained in 824th iteration when using the Ackley function. Finally, comparing CCCS algorithm with CS algorithm, the CCCS algorithm always converges quickly and finds the global optimal solution in the same iteration.

Evaluating algorithm performance by statistics of the maximum value, minimum value, average value, and variance, the numerical test results are shown in Table 2.

It can be seen from the numerical result of Table 2 that the CCCS algorithm has a superior search performance to the

```

(01) Input: Dimension of the nest: d, Total number of the nest: n, Maximum number of
iterations: time, The probability of being discovered by the host:  $p_a$ , The maximum number
of updates of the bird's nest: limit, Maximum chaotic iterations: T
(02) Output: The best nest  $X_i$ 
(03) Begin
(04) Initial chaotic nest:  $n$  host nests  $X_i, i = 1, 2, \dots, n$ 
(05) Calculated fitness:  $F_i (i = 1, 2, \dots, n)$ ;
(06) While (present iterations  $\leq$  time)
(07)   Generate a new solution  $X_i$  by Levy flight;
(08)   If ( $c \geq \text{limit}$ )
(09)      $F_i$  from small to large in accordance with  $X_i$ ;
(10)     Delete the solution  $X_i$  of the end 10%
(11)     Generate  $m$  catfish new nest Placed in the tail of  $X_i, m = n \times 10\%$ ;
(12)     Output the catfish new solution  $X_i$ ;
(13)   End
(14)   Select candidate solution  $X_j$ ;
(15)   If ( $F_i > F_j$ )
(16)     The new solution is used instead of the candidate solution;
(17)     Count(present iterations)=1;
(18)   Else
(19)     Count(present iterations)=0;
(20)   End
(21)   Discarding the worst solution according to the probability  $P_a$ ;
(22)   A new solution is used to replace the discarded solution with a preference random walk;
(23)   Keep the best solution;
(24) End
(25) End

```

ALGORITHM 1: CCCS.

other three algorithms under all the performance index and different functions.

3. Improved CS Algorithm Optimized Semisupervised SVM Model

3.1. Semisupervised SVM Basic Model. Semisupervised SVM can improve the prediction accuracy of an algorithm based on SVM with both labeled and unlabeled samples. It can be considered as a perfect realization based on low-density hypothesis and clustering hypothesis [26]. Recently, it has become a hotspot in the field of machine learning. The main principles are as follows.

Given a labeled dataset L

$$(x_1, y_1), \dots, (x_l, y_l), \quad x_i \in R^m, \quad y_i \in \{-1, +1\} \quad (7)$$

and unlabeled dataset U

$$x_1^*, \dots, x_k^* \quad (8)$$

find an effective method to predict the label of unlabeled samples and get its semilabeled samples Y^* :

$$y_1^*, \dots, y_k^* \quad (9)$$

Getting a hybrid set containing labeled and semilabeled samples

$$(x_1, y_1), \dots, (x_l, y_l), (x_1^*, y_1^*), \dots, (x_k^*, y_k^*) \quad (10)$$

can be separated at the maximum interval, and the larger classification interval means that the classifier has better generalization ability.

When the optimal hyperplane

$$\omega \cdot x + b = 0 \quad (11)$$

separates the mixed data set, the classification results can maximize the interval, where ω is a vector normal to the hyperplane, and b is a constant such that $b/\|\omega\|$ is the distance of the hyperplane from the origin. The above problem can be formalized as the following optimization problem:

$$\min \left\{ \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^l \xi_i + C^* \sum_{j=1}^k \xi_j^* \right\}$$

$$\forall_{i=1}^l : y_i^* (\omega \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, 2, \dots, l \quad (12)$$

$$\forall_{j=1}^k : y_j^* (\omega \cdot x_j^* + b) \geq 1 - \xi_j^*, \quad \xi_j^* \geq 0, \quad j = 1, 2, \dots, k$$

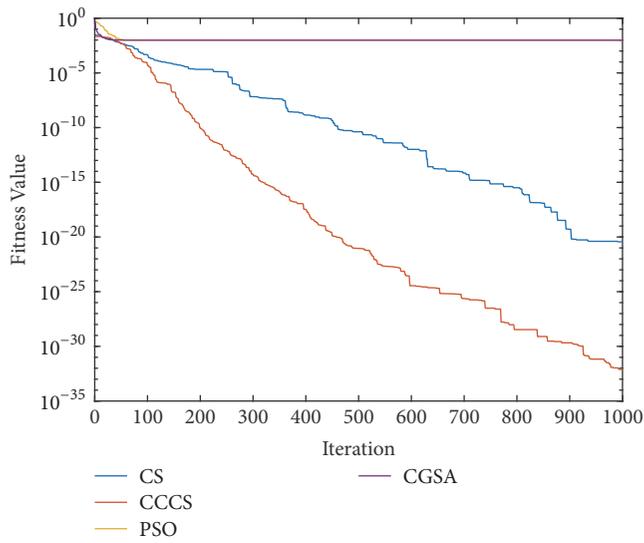
C and C^* are the regularization parameters associated with the labeled and semilabeled samples, respectively; ξ_i and ξ_j^* are the corresponding slack variables. The kernel function is radial basis kernel function (RBF), $K(x, x_i) = \exp(-\gamma \|x - x_i\|^2)$, and γ is kernel parameters.

TABLE 1: Formula and parameter settings of the eight benchmark functions.

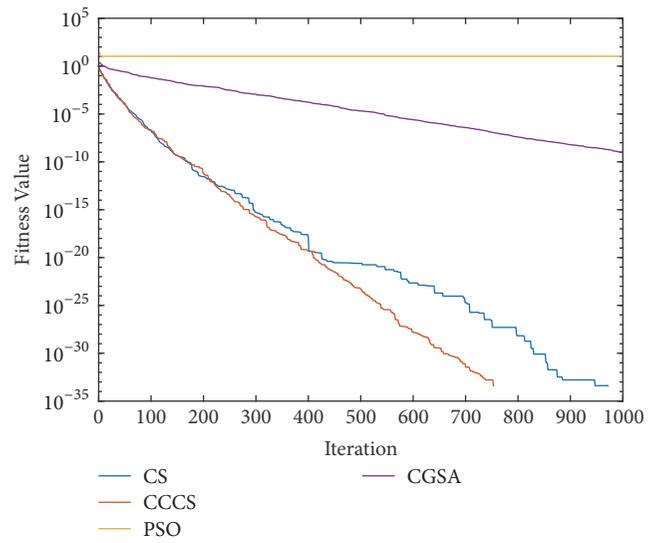
Function	Name	Formula	Searching Space	Opt.	Trait
f_1	Schaffer	$f_1(x, y) = 0.5 + \frac{\sin^2(x^2 - y^2) - 0.5}{(1 + 0.001(x^2 - y^2))^2}$	$[-10, 10]$	0	Unimodal
f_2	Schwefel 2.22	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]$	0	Unimodal
f_3	Sphere	$f_3(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0	Unimodal
f_4	Sum of different power	$f_4(x) = \sum_{i=1}^N x_i ^{i+1}$	$[-3, 3]$	0	Unimodal
f_5	Ackley	$f_5(x) = -20e^{-0.2\sqrt{(1/n)\sum_{i=1}^n x_i^2}} - e^{(1/n)\sum_{i=1}^n \cos 2\pi x_i}$	$[-32, 32]$	0	Multimodal
f_6	Rastrigrin	$f_6(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0	Multimodal
f_7	Branin	$f_7(x) = \left(1 - 2y + \frac{1}{20} \sin 4\pi y - x\right)^2 + \left(y - \frac{1}{2} \sin 2\pi x\right)^2$	$[-10, 10]$	0	Multimodal
f_8	Griewank	$f_8(x) = \sum_{i=1}^N \frac{x_i^2}{4000} - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0	Multimodal

TABLE 2: Comparison of numerical testing results.

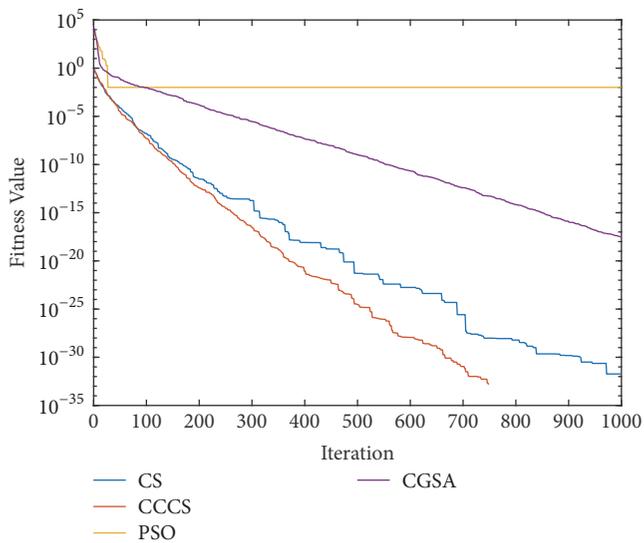
Function	Algorithm	Maximum	Minimum	Average	Variance
f_1	CS	0.0296	3.6427e-21	8.4440e-04	1.3215e-05
	CCCS	0.0218	7.8064e-33	8.1488e-04	1.2582e-05
	PSO	0.4786	0.0100	0.0158	0.0017
	CGSA	0.5362	0.0098	0.0108	3.0341e-04
f_2	CS	0.6626	0	0.0029	0.0010
	CCCS	0.5690	0	0.0027	7.3280e-04
	PSO	26.6960	10.6037	10.6359	0.5174
	CGSA	2.2731	9.4231e-10	0.0431	0.0358
f_3	CS	0.7740	1.8489e-32	0.0031	0.0013
	CCCS	0.6371	0	0.0031	0.0011
	PSO	5.3003e+03	0.0101	17.4638	5.8025e+04
	CGSA	1.3660e+04	3.0835e-18	30.6816	2.8436e+05
f_4	CS	0.6047	0	0.0034	0.0012
	CCCS	0.5616	0	0.0022	6.2698e-04
	PSO	3.3220e+05	0.0100	951.0100	2.7004e+08
	CGSA	14.8439	9	9.0251	0.1110
f_5	CS	1.1919	2.0256e-31	0.0044	0.0028
	CCCS	0.9859	0	0.0036	0.0019
	PSO	20.2544	3.7769	4.9084	6.4985
	CGSA	20.7509	3.4519e-09	0.2963	1.6965
f_6	CS	1.1351	0	0.0048	0.0031
	CCCS	1.0239	0	0.0038	0.0022
	PSO	5.4272e+03	1.5607	24.8489	7.5573e+04
	CGSA	44.0831	3.6602	4.2440	7.2125
f_7	CS	0.5721	1.1381e-30	0.0032	0.0011
	CCCS	0.5493	0	0.0027	7.9073e-04
	PSO	4.9874e+04	0.0298	143.0958	5.5309e+06
	CGSA	268.3960	60.6185	60.9033	45.2938
f_8	CS	0.1298	4.4998e-30	0.0021	1.5972e-04
	CCCS	0.0984	0	0.0014	7.1452e-05
	PSO	2.2087	0.0050	0.0174	0.0128
	CGSA	129.8682	6.4933	7.9713	90.9936



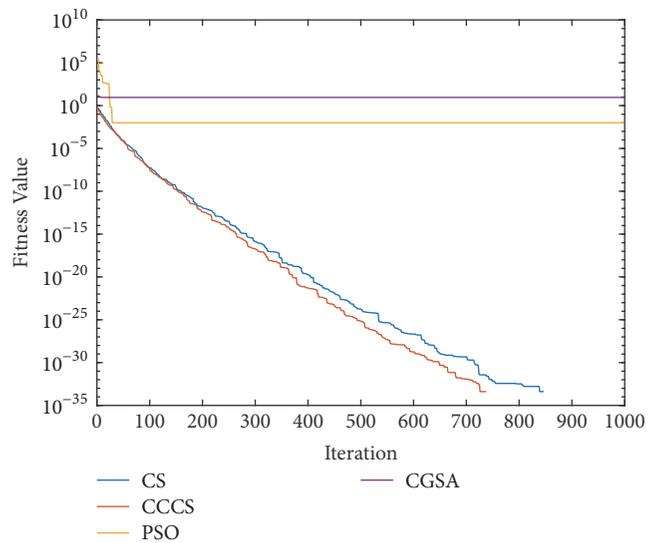
(a) Function f_1



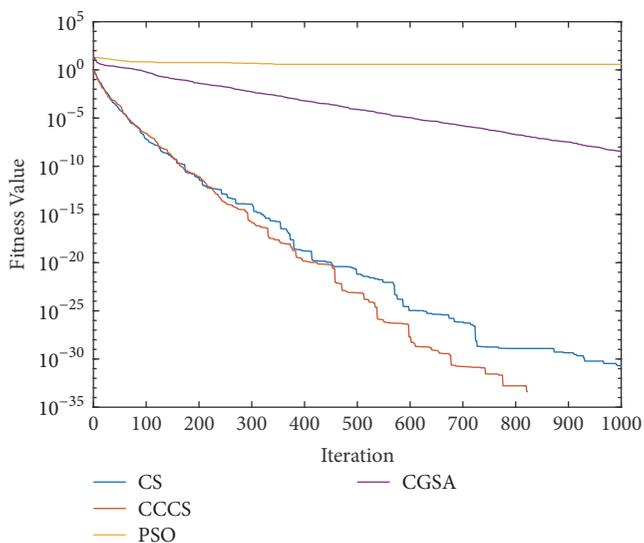
(b) Function f_2



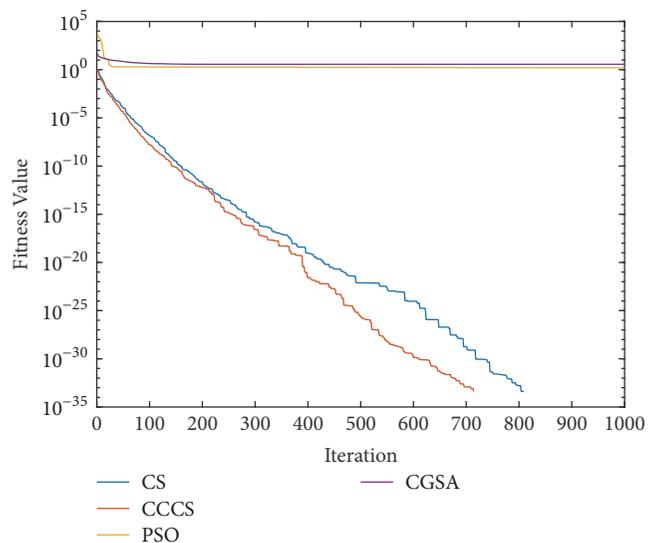
(c) Function f_3



(d) Function f_4



(e) Function f_5



(f) Function f_6

FIGURE 2: Continued.

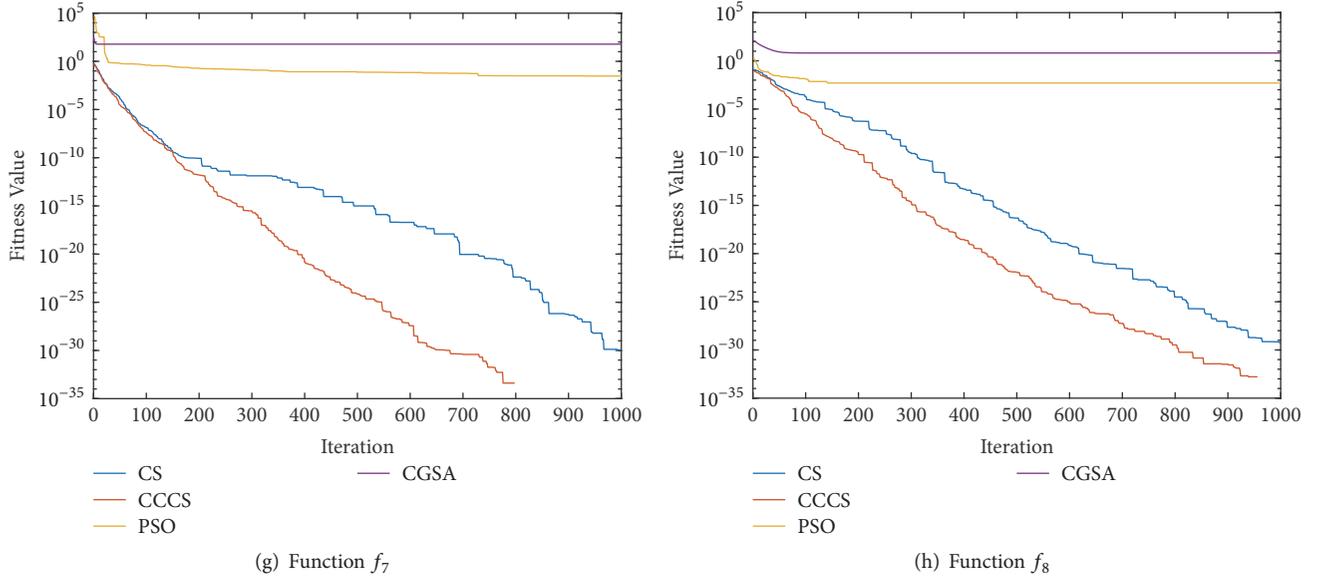


FIGURE 2: Fitness value curves of CS, CCCS, PSO, and CGSA on eight benchmark functions.

The S3VM is retrained with the new training set, and this procedure is continued iteratively until a stopping criterion is reached (e.g., maximum number of iterations). At each iteration, new semilabeled samples are added to the training set constructed in the previous iteration, while the semilabeled samples that change their label are removed from the new semilabeled set and moved back to the unlabeled set.

3.2. Semisupervised SVM Model Based on an Improved Cuckoo Search Algorithm. Improved cuckoo search algorithm optimized semisupervised SVM model (CCCS-S3VM) is based on improved cuckoo search algorithm optimized SVM (CCCS-SVM), by adding the semisupervised learning, which introduced unlabeled samples into the training process and then being trained with labeled samples together, finally getting the CCCS-S3VM model. The main idea is to use SVM's regularization parameter C and RBF's kernel parameters γ as the optimization target of CCCS algorithm, in order to get a set of combination parameters that make SVM get the best classification accuracy. The specific algorithm steps are shown as in Algorithm 2.

3.3. Oil Layer Recognition Application

3.3.1. Design of Oil Layer Recognition System. Block diagram of the oil layer recognition system based on CCCS-S3VM is shown in Figure 3.

The oil layer recognition mainly has the following steps.

Step 1 (data selection and preprocessing). The selection of the data set is complete and comprehensive and should be closely related to layer evaluation. The dataset should be divided into two parts of training and testing samples, which is conducted normalization processing to avoid calculation saturation phenomenon.

Step 2 (attribute discretization and generalization). In order to achieve the reduction of the sample information attribute, first extract sample information, conduct decision attribute generalization, and use curve inflection point approach to achieve continuous attribute discretization.

Step 3 (attribute reduction of data information). Oil logging data includes sound, electricity, nuclear, and other logging information. There are more than 10 kinds of logging property in Logging series, but some properties are not important, attribute reduction must be carried out to eliminate concentrated redundant attributes in data. We use the attribute reduction based on consistent covering rough set [27].

Step 4 (CCCS-S3VM modeling). In CCCS-S3VM model, input the sample information after attribute reduction, use an S3VM algorithm to train, and label the unlabeled samples. Use the CCCS algorithm to speed up the solution speed and accuracy and finally get the CCCS-S3VM classification model.

Step 5 (recognition output). The layer of the entire well section is recognized by the trained CCCS-S3VM model and output the results.

In order to verify the application effect of the S3VM layer recognition model based on CCCS, we select three logging data for training and testing.

3.3.2. Practical Application. Hardware environment: MacOS10.13.3/ Matlab 2016a/2.8GHz/Intel Core i7/16GB.

In Section 2.3, the effectiveness of the improved CCCS algorithm is simulated and analyzed on benchmark functions. In order to verify the application effect of the semisupervised model optimized by the improved algorithm, we select three logging data for training and testing, record as

```

(01) Input:
(02) Labeled dataset L
(03) Unlabeled dataset U
(04) Regularization parameter of unlabeled dataset  $C^*$ 
(05) Output:
(06) The label of unlabeled samples  $Y^*$ ;
(07) Final model CCCS-S3VM;
(08) Begin
(09) The regularization parameter  $C$  and the kernel parameter  $\gamma$  of the labeled samples
that obtain by CCCS-SVM algorithm;
(10) Training the initial model  $SVM_0$  with labeled dataset L;
(11) Predict the label  $Y^*$  of unlabeled dataset U using  $SVM_0$ :
 $y_1^*, \dots, y_k^*$ 
(12) Initialization:  $C^* \ll C$ ;
(13) While  $C^* < C$  do
(14) Solve formula (12) based on L, U,  $Y^*$ ,  $C^*$ ,  $C$ , obtain  $(\omega, b)$  and  $\xi$ ;
(15) // There is a pair of unlabeled samples  $x_i$  and  $x_j$ , Its label  $y_i$  and  $y_j$  is reverse,
and the corresponding relaxation variables are satisfied:  $\xi_i + \xi_j^* > 2$ , it means that  $y_i$  and
 $y_j$  likely to be a wrong label. The labels of the two are exchanged and the SVM
problem is solved again. The approximate solution of the minimization of the
objective function can be obtained after each round of iteration.
(16) While  $\exists i, j \mid (y_i y_j < 0) \wedge (\varepsilon_i > 0) \wedge (\varepsilon_j > 0) \wedge (\varepsilon_i + \varepsilon_j > 2)$  do
(17)  $y_i = -y_i; y_j = -y_j$  // label exchange;
(18) Solve formula (12) based on L, U,  $Y^*$ ,  $C^*$ ,  $C$ , obtain  $(\omega, b)$  and  $\xi$ ;
(19) End while
(20)  $C^* = \min(2C^*, C)$ 
(21) End while
(22) End

```

ALGORITHM 2: CCCS-S3VM.

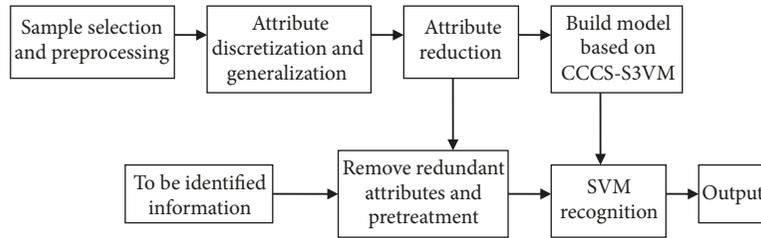


FIGURE 3: Block diagram of the oil layer recognition system based on CCCS-S3VM.

w_1, w_2, w_3 , respectively. In addition, its reductions results and distribution of those logging data are as in Tables 3 and 4, respectively.

The CCCS-S3VM model trained on the training data set is used to identify the oil layer of the test sample. Table 5 shows the normalized range of four sample attributes in the whole well after the reduction. These attributes are normalized as shown in Figure 4, where the horizontal axis represents the depth, and the vertical axis represents a normalized value.

In CCCS-S3VM model, the sample information is input by attribute reduction, we use CCCS algorithm to find the optimal C and γ and obtain trained S3VM forecasting model. There are three test datasets from three wells in different good section, which are used in oil layer recognition by trained prediction model; we establish SVM model, S3VM model, and QPSO-S3VM model, respectively; their recognition

results are compared with CCCS-S3VM model. In addition, supervised SVM model trained with the full available training set, but the training set of semisupervised models consists of the labeled set and the unlabeled set. The labeled set consists of 10% of the training set, and the unlabeled set consists of the remaining data of the training set.

In order to measure the performance of the recognition model, we define the following performance index:

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2} \quad (13)$$

$$MAE = \frac{1}{m} \sum_{i=1}^m |h(x^{(i)}) - y^{(i)}| \quad (14)$$

TABLE 3: Reduction results of logging data.

Well	Attributes
Actual Results (w_1)	{ AC, CNL, DEN, GR, RT, RI, RXO, SP, R2M, R025, BZSP, RA2, C1, C2, CALI, RINC, PORT, VCL, VMA1, VMA6, RHOG, SW,VO, WO, PORE, VXO, VW, AC1 }
Reduction Results (w_1)	{AC, GR, RI, RXO, SP }
Actual Results (w_2)	{ GR, DT, SP, WQ, LLD, LLS, DEN, NPHI, PE, U, TH, K, CALI }
Reduction Results (w_2)	{ GR, DT, SP, LLD, LLS, DEN, K }
Actual Results (w_3)	{ AC, CALI, GR, NG, RA2, RA4, RI, RM, RT, RXO, SP }
Reduction Results (w_3)	{ AC, NG, RI, SP }

TABLE 4: Distribution of logging data.

Well	Training set			Test set		
	Depth(m)	Oil layers	Dry layers	Depth(m)	Oil layers	Dry layers
w_1	1200~1280	48	112	1230~1310	96	545
w_2	3300~3450	63	137	3100~3200	172	828
w_3	1220~1290	53	228	1180~1320	105	895
Decision attribute	$D = \{d\}, d = \{d_i = i, i = 0, 1\}$, where 0, 1 represents the dry layer and oil layer respectively.					

TABLE 5: The normalized range of attributes in the well.

Attribute	Minimum value			Maximum value		
	w_1	w_2	w_3	w_1	w_2	w_3
AC	50	-	0	150	-	600
GR	5	4	-	200	20	-
RI	0	-	-5	100	-	100
RXO	0	-	-	350	-	-
SP	-35	-50	-50	-5	30	30
DT	-	150	-	-	500	-
LLD	-	0	-	-	25000	-
LLS	-	0	-	-	3310	-
DEN	-	0	-	-	5	-
K	-	0	-	-	5	-
NG	-	-	0	-	-	50

Here, $h(x^{(i)})$ and $y^{(i)}$ are the recognition output value and the desired output value, respectively.

RMSE measured the deviations between the recognition output value and the desired output value, which can well reflect the precision of the measurement. MAE is the mean of the absolute value of the deviation of all individual and arithmetic mean values. The smaller the RMSE and MAE, the better the performance of the algorithm model. Therefore, RMSE is used as the standard to evaluate the accuracy of each algorithm model. MAE is used as the actual situation of the prediction and prediction error. The performance index data of each algorithm model are shown in Table 6.

Table 6 clearly shows that the recognition efficiency of semisupervised algorithms (S3VM, PSO-S3VM, CCCS-S3VM) is significantly close to the supervised algorithm(SVM), and the proposed model (CCCS-S3VM) in this paper has higher accuracy than SVM algorithm in wells w_2 and w_3 . Secondly, both the PSO and the improved CS algorithm can

improve the performance of semisupervised SVM, but the improvement of its accuracy means longer running time and large computational complexity. However, the CS algorithm based on chaotic catfish effect can make the semisupervised SVM algorithm improve the classification accuracy and keep running speed fast at the same time. In summary, CCCS-S3VM constituted catfish effect and chaotic theory is better than the traditional SVM and S3VM model in oil layer recognition. The classification is shown in Figure 5, among them, (a), (c), and (e) represent the actual oil layer distribution; (b), (d), and (f) represent CCCS-S3VM oil layer distribution.

According to a comparison of recognition results in Figure 5, firstly, we know both recognition accuracies in three wells go up to 90%. It can effectively recognize the oil layer distribution accurately. Secondly, the CCCS-S3VM model can predict the distribution of a large number of reservoir samples only using a small number of labeled samples; it

TABLE 6: Comparison on performance index.

Well	Model	RMSE	MAE	Accuracy(%)	Running time(s)
w_1	SVM	0.1894	0.0359	96.41	0.01
	S3VM	0.3824	0.1463	85.37	0.17
	PSO-S3VM	0.3673	0.1349	86.51	5.32
	CCCS-S3VM	0.2371	0.0562	94.38	1.78
w_2	SVM	0.3317	0.1100	89.00	0.02
	S3VM	0.3536	0.1250	87.50	0.42
	PSO-S3VM	0.3151	0.0993	90.07	10.21
	CCCS-S3VM	0.3043	0.0926	90.74	2.99
w_3	SVM	0.3240	0.1050	89.50	0.02
	S3VM	0.3521	0.1240	87.60	0.57
	PSO-S3VM	0.3351	0.1123	88.77	11.72
	CCCS-S3VM	0.2581	0.0666	93.34	3.89

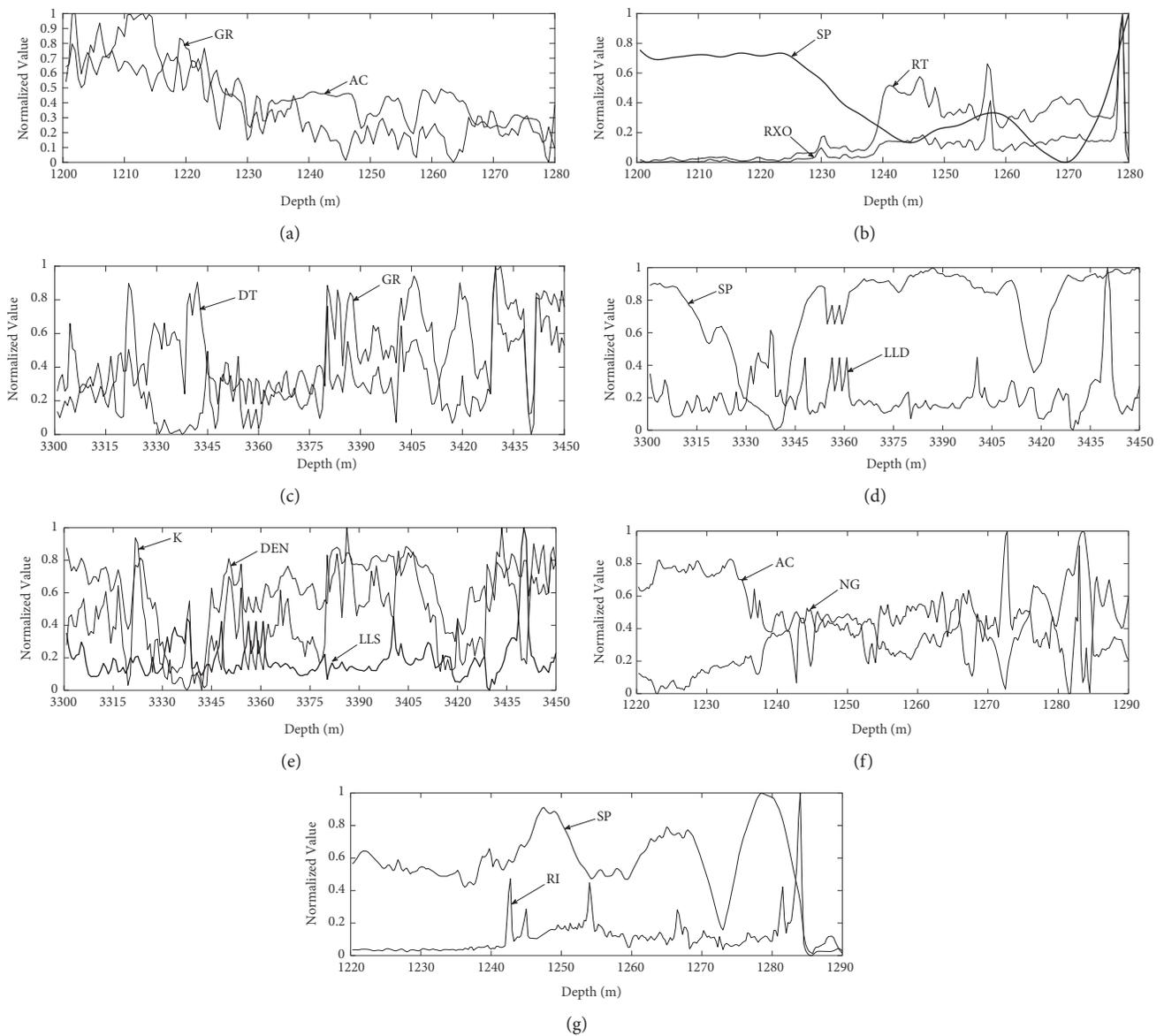


FIGURE 4: The normalized curves of attributes: (a) and (b) attribute normalization of w_1 ; (c), (d), and (e) attribute normalization of w_2 ; (f) and (g) attribute normalization of w_3 .

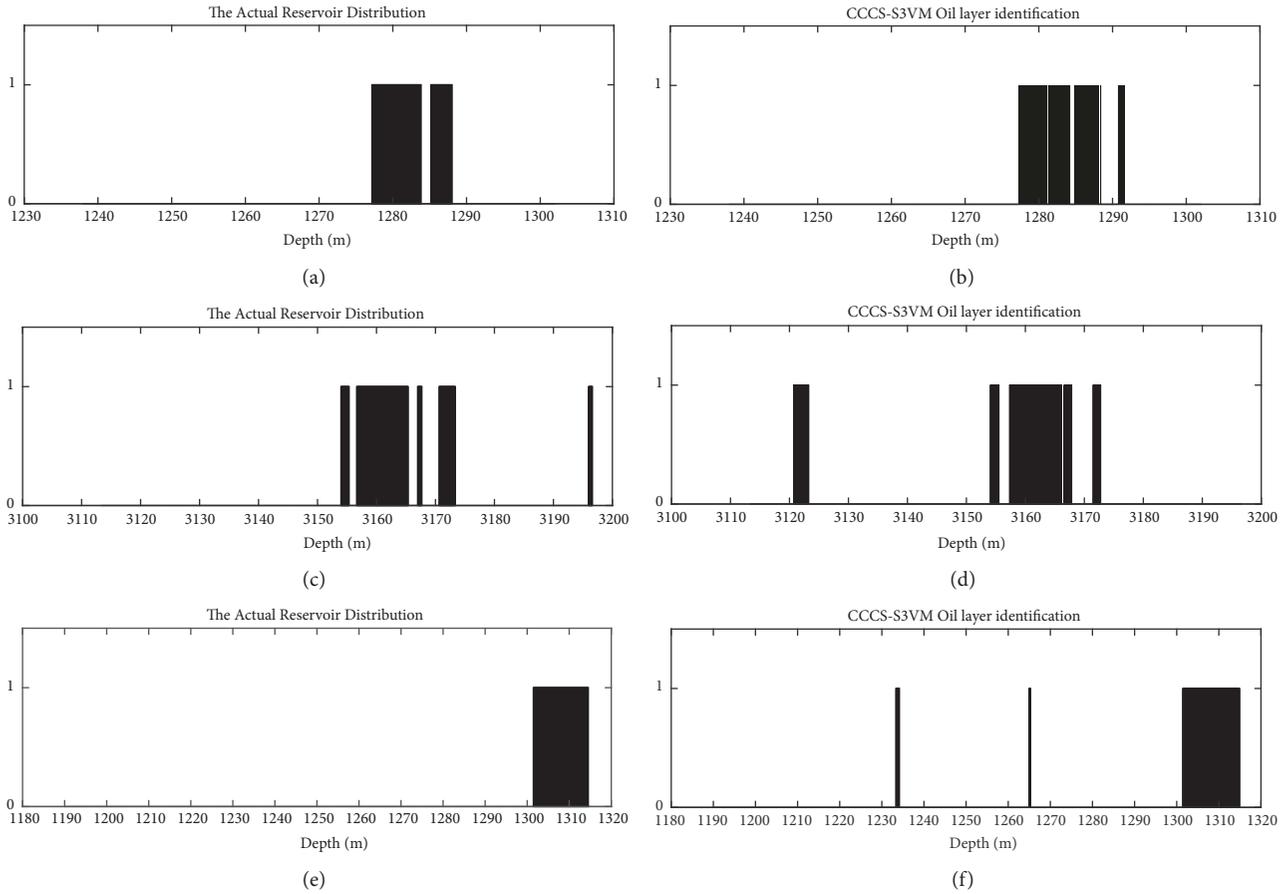


FIGURE 5: Classification of CCCS-S3VM: (a) the actual oil layer distribution of w_1 ; (b) CCCS-S3VM oil layer distribution of w_1 ; (c) the actual oil layer distribution of w_2 ; (d) CCCS-S3VM oil layer distribution of w_2 ; (e) the actual oil layer distribution of w_3 ; (f) CCCS-S3VM oil layer distribution of w_3 .

greatly reduces the need of labeled samples and has a good application prospect.

4. Conclusion

In view of the possibility that the cuckoo algorithm is locally optimal, the chaotic catfish effect is introduced into the cuckoo algorithm, which enhances the ability of the algorithm to search for the global optimal solution and validates its effectiveness on the benchmark functions. Then, the improved algorithm is used to optimize the semisupervised SVM algorithm. Finally, it is applied to the oil layer recognition to solve the problem that the labeled data are difficult to get when logging. This method has good classification effect. It improves the recognition rate and reduces data extraction cost when using a small number of labeled data, and runs faster, which is obviously better than other classical algorithms. It has good prospects.

Data Availability

All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by Tianjin Natural Science Foundation (no. 18JCYBJC16500) and Hebei Province Natural Science Foundation (no. E2016202341).

References

- [1] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd Annual meeting of the Association for Computational Linguistics*, pp. 189–196, Morristown, NJ, USA, 1995.
- [2] B. Zu, K. Xia, Y. Pan, and W. Niu, "A novel graph constructor for semisupervised discriminant analysis: Combined low-rank and κ -nearest neighbor graph," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 9290230, 11 pages, 2017.
- [3] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, NY, USA, 1998.

- [4] Q. Wu, S. Y. Liu, and L. Y. Zhang, "Particle swarm optimization for semi-supervised support vector machine," *Journal of Information Science & Engineering*, vol. 26, no. 5, pp. 1695–1706, 2010.
- [5] W.-P. Luo, H.-Q. Li, and N. Shi, "Semi-supervised least squares support vector machine algorithm: application to offshore oil reservoir," *Applied Geophysics*, vol. 13, no. 2, pp. 406–415, 2016.
- [6] D. Zhang, L. Jiao, X. Bai, S. Wang, and B. Hou, "A robust semi-supervised SVM via ensemble learning," *Applied Soft Computing*, vol. 65, pp. 632–643, 2018.
- [7] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC '09)*, pp. 210–214, IEEE Publication, Coimbatore, India, 2009.
- [8] R. B. Payne, M. D. Sorenson, and K. Klitz, *The Cuckoos*, Oxford University Press, 2005.
- [9] X. S. Yang and S. Deb, "Engineering optimization by cuckoo search," *International Journal of Mathematical Modelling & Numerical Optimization*, vol. 1, no. 4, pp. 330–343, 2012.
- [10] P. R. Srivastava, R. Khandelwal, S. Khandelwal, S. Kumar, and S. S. Ranganatha, "Automated test data generation using cuckoo search and tabu search (CSTS) algorithm," *Journal of Intelligent Systems*, vol. 21, no. 2, pp. 195–224, 2012.
- [11] X.-S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Computers & Operations Research*, vol. 40, no. 6, pp. 1616–1624, 2013.
- [12] Y. Zhang, L. Wang, and Q. Wu, "Modified Adaptive Cuckoo Search (MACS) algorithm and formal description for global optimisation," *International Journal of Computer Applications in Technology*, vol. 44, no. 2, pp. 73–79, 2012.
- [13] L. J. Wang, Y. L. Yin, and Y. W. Zhong, "Cuckoo search algorithm with dimension by dimension improvement," *Journal of Software*, vol. 24, no. 11, pp. 2687–2698, 2013.
- [14] W.-H. Yang, J.-R. Liu, and Y. Zhang, "A new local-enhanced cuckoo search algorithm," *International Journal of Computational Science and Mathematics*, vol. 8, no. 2, pp. 175–182, 2017.
- [15] S. Wang, S. Y. Song, Y. Yu, Z. Xu, H. Yachi, and S. C. Gao, "Multiple chaotic cuckoo search algorithm," in *Proceedings of the 8th International Conference on Swarm Intelligence (ICSI '17)*, vol. 10385, pp. 531–542, Springer International Publishing, Fukuoka, Japan, 2017.
- [16] S.-S. Wang, J.-J. Yang, and S. Chai, "Artificial bee colony algorithm with chaotic catfish effect and its application," *Acta Electronica Sinica*, vol. 42, no. 9, pp. 1731–1737, 2014.
- [17] C. T. Brown, L. S. Liebovitch, and R. Glendon, "Lévy flights in dove Juhoansi foraging patterns," *Human Ecology*, vol. 35, no. 1, pp. 129–138, 2007.
- [18] A. M. Reynolds and M. A. Frye, "Free-flight odor tracking in *Drosophila* is consistent with an optimal intermittent scale-free search," *PLoS ONE*, vol. 2, no. 4, p. e354, 2007.
- [19] I. Pavlyukevich, "Cooling down Lévy flights," *Journal of Physics A: Mathematical and Theoretical*, vol. 40, no. 41, pp. 12299–12313, 2007.
- [20] P. Barthelemy, J. Bertolotti, and D. S. Wiersma, "A Lévy flight for light," *Nature*, vol. 453, no. 22, pp. 495–498, 2008.
- [21] L. Y. Chuang, S. W. Tsai, and C. H. Yang, "Fuzzy adaptive catfish particle swarm optimization," *Artificial Intelligence Research*, vol. 1, no. 2, pp. 149–170, 2012.
- [22] W. Z. Ju, K. W. Xia, and S. D. Dai, "Improved cloud particle swarm optimization algorithm and its application on circuit breaker optimization," *Application Research of Computers*, vol. 35, no. 7, 2018.
- [23] L. Yi, X. C. Diao, J. J. Cao, and B. Zhang, "Local Enhanced Catfish Bat Algorithm," in *Proceedings of the 2016 International Conference on Robots & Intelligent System (ICRIS '16)*, pp. 238–245, Institute of Electrical and Electronics Engineers Inc, ZhangJiaJie, China, August 2016.
- [24] K. James and E. Russell, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp. 1942–1948, IEEE Publication, Perth, Australia, 1995.
- [25] S. Gao, C. Vairappan, Y. Wang, Q. Cao, and Z. Tang, "Gravitational search algorithm combined with chaos for unconstrained numerical optimization," *Applied Mathematics and Computation*, vol. 231, pp. 48–62, 2014.
- [26] N. V. Chawla and G. Karakoulas, "Learning from labeled and unlabeled data: An empirical study across techniques and domains," *Journal of Artificial Intelligence Research*, vol. 23, pp. 331–366, 2005.
- [27] J. Bai, K. Xia, Y. Lin, and P. Wu, "Attribute Reduction Based on Consistent Covering Rough Set and Its Application," *Complexity*, vol. 2017, Article ID 8986917, 9 pages, 2017.

