

## Research Article

# Multistrategy Improved Sparrow Search Algorithm Optimized Deep Neural Network for Esophageal Cancer

Yanfeng Wang <sup>1</sup>, Qing Liu <sup>1</sup>, Junwei Sun <sup>1</sup>, Lidong Wang,<sup>2</sup> Xin Song,<sup>2</sup> and Xueke Zhao<sup>2</sup>

<sup>1</sup>School of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou, China

<sup>2</sup>State Key Laboratory of Esophageal Cancer Prevention & Treatment and Henan Key Laboratory for Esophageal Cancer Research of the First Affiliated Hospital, Zhengzhou University, Zhengzhou 450052, China

Correspondence should be addressed to Junwei Sun; [junweisun@yeah.net](mailto:junweisun@yeah.net)

Received 23 June 2022; Revised 5 September 2022; Accepted 9 September 2022; Published 27 September 2022

Academic Editor: Muhammad Fazal Ijaz

Copyright © 2022 Yanfeng Wang 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.

Deep neural network is a complex pattern recognition network system. It is widely favored by scholars for its strong nonlinear fitting ability. However, training deep neural network models on small datasets typically realizes worse performance than shallow neural network. In this study, a strategy to improve the sparrow search algorithm based on the iterative map, iterative perturbation, and Gaussian mutation is developed. This optimized strategy improved the sparrow search algorithm validated by fourteen benchmark functions, and the algorithm has the best search accuracy and the fastest convergence speed. An algorithm based on the iterative map, iterative perturbation, and Gaussian mutation improved sparrow search algorithm is designed to optimize deep neural networks. The modified sparrow algorithm is exploited to search for the optimal connection weights of deep neural network. This algorithm is implemented for the esophageal cancer dataset along with the other six algorithms. The proposed model is able to achieve 0.92 under all the eight scoring criteria, which is better than the performance of the other six algorithms. Therefore, an optimized deep neural network based on an improved sparrow search algorithm with iterative map, iterative perturbation, and Gaussian mutation is an effective approach to predict the survival rate of esophageal cancer.

## 1. Introduction

Esophageal cancer is a malignant tumor with high incidence, high mortality, and high recurrence rate [1]. The way of cancer patient treatment is mainly based on physicians' experience, which inevitably leads to physicians' treatment errors [2]. How to effectively forecast the survival time of esophageal cancer patients to decrease the misdiagnosis rate of physicians is a current research hotspot [3–6]. In recent years, the quick advancement of artificial intelligence has enabled the construction of intelligent systems, a simple and easy task [7]. Artificial intelligence has been able to simulate human intelligence more accurately to learn and make predictive actions on medical datasets [8, 9]. In particular, deep learning excels at complex machine learning tasks by building multilayer neural networks [10, 11]. Deep learning has made remarkable progress and excellent performance in

areas such as medical image process, biological image process, and target inspection completion [12–15]. Since deep learning is adept at handling complex nonlinear problems, it can perform comparably or even better than professional physicians in the fields of disease identification and disease prediction [16, 17]. Therefore, survival prediction models based on deep learning will hopefully provide a scientific basis for clinical medical decisions in esophageal cancer.

Neural network [18] is the basis of deep learning, a mathematical simplification of single-layer perception of human nerve cells. Deep neural network (DNN) has been continuously trained by researchers regarding the exploration of the human nervous system with multiple implicit layers [19, 20]. DNN has been used to identify different ECG abnormalities, and its identification results are better than hospitalists with heart disease and emergency medical

medicine [21]. DNN model is effective in identifying tumors or hyperplastic polyps, and the identification time of the model is shorter than endoscopy [22]. 3D-DNN has achieved progress in automated lung cancer diagnosis by computed tomography. The model can identify all suspicious lung pathologies and evaluate the grades of lung malignancies [23]. DNN has exhibited state-of-the-art performance in areas such as medical image recognition [24], cancer diagnosis, pathology examination [25], stock price prediction [26], and daily plant transpiration estimation [27]. However, the DNN model trained on small datasets typically exhibits poorer properties than traditional machine learning approaches, such as shallow neural networks and support vector machines.

Gradient descent is a common optimization approach for neural network learning, which cleverly utilizes gradients to find function minima [28]. Nevertheless, when gradient descent is used as an optimization method for DNN, it is usually difficult to get rid of the local minimum value and slow convergence speed. The metaheuristic algorithm can effectively eliminate the above problem by obtaining the optimal solution through global search [29]. Metaheuristic algorithm is extensively implemented in function optimization, fuzzy logic system, and image treatment [30–32]. The DNN model optimized by the Gray Wolf Optimization (GWO) algorithm has been applied to extract and classify features from the CAD image dataset of diabetic retinopathy [33]. The model has advantages in terms of accuracy, precision, recall, sensitivity, and specificity. Gravitational search algorithm (GSA) plays an essential role in enhancing the prediction accuracy of DNN models [34]. The model can more precisely differentiate between benign and malignant nodules in CT scan lung images. Whale Optimization Algorithm (WOA) of DNN has achieved significant results [35]. Dimensionality reduction by principal component analysis (PCA) and firefly algorithm is performed on the diabetic retinopathy dataset [36]. The simplified dataset is sent to the DNN model for classification. The DNN model is shown to outperform other machine learning algorithms in accuracy, precision, recall, sensitivity, and specificity. Optimization algorithm for feature dimensionality reduction is a common neural network optimization method. However, the impact of dimensionality reduction on data results is significant, which may result in DNN performing even worse than traditional methods on small sample datasets [37]. This makes the application of DNN to small sample datasets a challenge. In this study, a method based on a metaheuristic algorithm for reducing the mean square deviation difference of DNN models and changing the interlayer connection weights makes the accuracy of models for small datasets improved.

In this study, the chaotic map is introduced into the sparrow search algorithm (SSA), which allows the sparrow population to be increased in sample diversification and improved in even distribution. The ability to escape local optimum and increase convergence speed is obtained by the SSA. Gaussian mutation and chaotic perturbations are also introduced into the SSA, which makes it possible to adjust the aggregated sparrow individuals. The local search ability

of SSA in the focal search region is enhanced by Gaussian mutation and chaotic perturbation. The iterative map is employed in the SSA to verify its optimal performance by fourteen benchmark functions. Iterative map, iterative perturbation, and Gaussian mutation optimized SSA (IIGSSA) are proposed. IIGSSA is adopted to find the connection weights of DNN. To evaluate the algorithm performance, IIGSSA-DNN is applied to the esophageal cancer dataset along with eight predictive classification algorithms, which are DNN, PSO-DNN, GSA-DNN, GWO-DNN, WOA-DNN, SSA-DNN, IIGSSA-K Nearest Neighbor (KNN), and IIGSSA-Support Vector Machine (SVM). They also use 26 metrics as input and survival time as output. Due to the small samples used in this paper, five-fold cross-validation is selected to evaluate the authenticity of the model performance. Ten scoring criteria are used as judging criteria, which are accuracy (Acc), false positive rate (FPR), recall rate (REC), true positive rate (TPR), precision (PRE), true negative rate (TNR), area under the curve (AUC), F1-measure (F1-M), and pooled mean (G-M) [38]. The IIGSSA-DNN model has an FPR value of 0.1, a  $P$ -value of 0.01, and all other scoring criteria have a value of 0.92. The model is evidenced to possess superior predictive accuracy and statistical value. Therefore, IIGSSA-DNN is an algorithm to accurately predict the survival time for esophageal cancer patients. IIGSSA-DNN is expected as a novel approach for the future clinical treatment of esophageal cancer.

A mixed metaheuristic algorithm is formulated to improve the prediction algorithm of the DNN network structure. The iterative map, iterative perturbation, and Gaussian mutation optimized SSA (IIGSSA) is developed. The optimum network structure of DNN can be defined by IIGSSA. IIGSSA-DNN is expected to be an enabling instrument for the clinical diagnosis and treatment of esophageal cancer. This is also an innovation of IIGSSA-DNN applied to nonimage datasets of esophageal cancer. The primary contributions of this study can be summarized as follows:

- (1) The iterative map, iterative perturbation, and Gaussian mutation optimized SSA with the fastest convergence rate and best accuracy are validated in fourteen benchmark functions.
- (2) IIGSSA is used as an optimization strategy to improve the model accuracy by optimizing the optimal connection weights for the DNN model.
- (3) IIGSSA-DNN is presented to the esophageal cancer dataset, and IIGSSA can predict the survival time of esophageal cancer patients more precisely than other algorithms.

The remainder of this study is structured as follows. Section 2 introduces the SSA and improvement ideas and identifies the iterative map, iterative perturbation, and Gaussian variation as the best optimization strategy on fourteen benchmark functions. Section 3 describes the principles and improvement ideas of DNN and applies IIGSSA-DNN to the esophageal cancer dataset. The conclusions are described in Section 4.

## 2. Development and Validation of an Improved SSA Based on Iterative Map, Iterative Perturbation, and Gaussian Mutation

**2.1. SSA.** SSA [39, 40] is a swarm intelligence optimization algorithm that simulates sparrow foraging behavior and antipredation behavior. The sparrow population achieves foraging behavior through three task divisions, which are discovery, follow, and alert. Discoverers are leaders in the population due to their high fitness. This is associated with their capacity to seek and provide the location and direction of food resources. Epigones follow and forage around the discoverers for greater fitness. Epigones in a population supervise the behavior of other individuals, and they compete with high-intake peers for food resources to improve their predation rates. When the whole population is threatened by predators or perceives danger, the sparrows immediately counter-hunt. The sparrows in the outer circle of the population are vulnerable to predators, and they need to constantly relocate to the center of the population. The sparrow in the center of the population adjusts its position to keep its distance from others as short as possible. SSA simulates the foraging process of sparrows to obtain the solution for the optimization problem.

Suppose a population of  $N$  sparrows searches in  $D$  dimensional search space; then, the position of the  $i_{th}$  sparrow in  $D$  dimensional search space is  $X_i = [x_{i1}, x_{i2}, \dots, x_{id}, \dots, x_{iD}]$ ,  $i = 1, 2, \dots, N$ .  $x_{id}$  stands for the position of the  $i_{th}$  sparrow in  $D$  dimension. SSA takes 10%~20% of the finders in the sparrow population as constraint conditions, and the position update equation is as follows:

$$x_{id}^{t+1} = \begin{cases} x_{id}^t \cdot \exp\left(\frac{-i}{\alpha T}\right), & R_2 < S_T, \\ x_{id}^t + QL, & R_2 \geq S_T, \end{cases} \quad (1)$$

where  $t$  is the current iteration number.  $T$  is the maximum number of iterations.  $\alpha$  is the uniform random number between (0, 1].  $Q$  is a random number that follows the standard normal distribution.  $L$  is a  $1 \times D$  dimensional matrix composed of element 1.  $R_2 \in [0, 1]$  is the warning value.  $S_T \in [0.5, 1]$  is the safety value. If  $R_2 < S_T$ , predators or other hazards are not near the population, and the current search environment is safe. Discoverers conduct extensive searches to guide the population to obtain higher fitness. If  $R_2 \geq S_T$ , epigones are keen to spot predators and quickly release danger signals for reminding the population to act immediately against predation. The population adjusts its search strategy and quickly moves towards the safe area.

The update equation for follower position is as follows:

$$x_{id}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{xw_d^t - x_{id}^t}{i^2}\right), & i > \frac{n}{2} \\ xb_d^{t+1} + |x_{id}^t - xb_d^{t+1}|A^+ \cdot L, & i \leq \frac{n}{2} \end{cases} \quad (2)$$

where  $A^+$  is  $1 \times D$  vector allocation with the value 1 or -1, and  $A^+ = A^T (AA^T)^{-1}$ .  $xw_d^t$  is the worst position of sparrows in the  $D$  dimension in the  $t$  iteration.  $xb_d^{t+1}$  is the optimal position of sparrows in the  $D$  dimension at the  $t + 1$  iteration. If  $i > n/2$ , the  $i_{th}$  epigone is starving without food. To obtain higher fitness, the epigone shifts to the area for food. If  $i \leq n/2$ , the  $i_{th}$  epigone has a better fitness at the current best location, and the epigone randomly seeks a location near the best location for foraging. 10%~20% of the population is responsible for reconnaissance, and their positions are updated as follows:

$$x_{id}^{t+1} = \begin{cases} xb_d^{t+1} + \beta(x_{id}^t - xb_d^{t+1}), & f_i \neq f_g, \\ x_{id}^t + K \left( \frac{x_{id}^t - xw_d^t}{|f_i - f_w| + e} \right), & f_i = f_g, \end{cases} \quad (3)$$

where  $\beta$  is a step size control parameter with the mean value of 0 and variance of 1, which satisfies normal distribution random number.  $K$  is a random step size control parameter in the interval  $[-1, 1]$ , referring to the orientation of the sparrow's flight.  $e$  is a minimum constant to prevent the numerator being zero.  $f_i$  is the fitness value of the  $i_{th}$  sparrow.  $f_g$  is the optimal fitness value of the current sparrow population.  $f_w$  is the worst fitness value of the current sparrow species. If  $f_i \neq f_g$ , sparrows on the edge of the population are vulnerable to predator attack, and they need to position themselves to avoid the attack. If  $f_i = f_g$ , the sparrow is in the middle of the population. To avoid being attacked by predators, sparrows in this area adjust their search strategy by approaching other sparrows in time after realizing the threat of predators.

**2.2. Chaotic Map, Chaotic Perturbation, and Gaussian Mutation Strategies.** Chaotic [41] is a complex nonlinear motion type ubiquitous in nature. This nonlinear phenomenon usually occurs under certain conditions, which makes the ordered trajectory deviates from the original path suddenly into a disorderly form. The chaotic map is favored by scholars because of its randomness, ergodicity, and regularity. This is associated with its capacity to sustain a rich population variety. The chaotic map optimization meta-heuristic algorithm enables the algorithm to escape local optimum while gaining higher global search capability. Nine common chaotic maps are listed in this study to optimize SSA.

(1) Tent map [42]:

$$Z_{k+1} = \begin{cases} Z_k/\beta, & Z_k \in (0, \beta], \\ (1 - Z_k)/(1 - \beta), & Z_k \in (\beta, 1], \end{cases} \quad (4)$$

where  $\beta$  indicates the mapping parameter for the tent. If  $\beta = 0.7$ , the obtained sequence distribution is more uniform.

(2) Chebyshev map [43]:

$$Z_{k+1} = \cos(\phi \cos^{-1} Z_k), \quad (5)$$

where  $\phi$  is the order of the Chebyshev map. When  $\phi$  is bigger than 2, the Lyapunov exponent is non-negative and the system is chaotic.

(3) Circle map [44]:

$$Z_{k+1} = \text{mod}\left(Z_k + 0.2 - \frac{0.5}{2\pi} \sin(2\pi Z_k), 1\right), \quad (6)$$

where  $a = 0.5b = 2.2$ .

(4) Iterative map [45]:

$$Z_{k+1} = \sin\left(\frac{a\pi}{Z_k}\right), \quad (7)$$

where  $a \in (0, 1)$  is the iterative map control parameter.

(5) Sine map [46]:

$$Z_{k+1} = \frac{4}{a} \sin(\pi Z_k), \quad (8)$$

where the sine map is a single-peak map, whose range is  $[-1, 1]$ , and the sine map parameter is  $a \in (0, 4]$ .

(6) Singer map [47]:

$$Z_{k+1} = \mu(7.86Z_k - 23.31Z_k^2 + 28.75Z_k^3 - 13.302875Z_k^4), \quad (9)$$

where  $\mu$  is the singer parameter whose value range is  $(0.9, 1.08)$ . When  $Z_k \in [0, 1]$ , the singer map is distributed in the range  $[0, 1]$ , which can make a more uniform distribution of  $Z_k$ .

(7) Sinusoidal map [48]:

$$Z_{k+1} = aZ_k^2 \sin(\pi Z_k), \quad (10)$$

where  $a = 2.3$ ,  $Z_0 = 0.7$ .

(8) Logistic map [49]:

$$Z_{k+1} = \mu Z_k (1 - Z_k), \quad (11)$$

where  $\mu$  is a logistic parameter whose value range is  $[0, 4]$ . When  $Z_k \in [0, 1]$ , the logistic map operates in chaos. The nearer the  $\mu$  value gets to 4, the more uniformly  $Z_k$  is dispersed between 0 and 1.

(9) Cubic map [50]:

$$Z_{k+1} = \rho Z_k (1 - Z_k^2), \quad (12)$$

where  $\rho \in (0, 1)$  is the Cubic map control parameter. Cubic map is one of the most common chaotic maps.

Chaotic perturbation is the introduction of a random perturbation quantity obeying chaotic distribution based on the original solution. A chaotic variant is derived from the chaotic map, and the chaotic variant is brought into the solution space by (13).

$$X_n^d = m_{\min} + C(m_{\max} - m_{\min}), \quad (13)$$

where  $m_{\min}$  is the minimum value of the variable  $X_n^d$  in  $d_{th}$  dimension.  $m_{\max}$  is the maximum value of the variable  $X_n^d$  in  $d_{th}$  dimension.  $X_n^d$  is the amount of chaotic perturbation generated by the solution in the  $m$  dimension.  $C$  is chaotic variable. The chaotic perturbation equation is

$$X_n' = \frac{(X' + X_n)}{2}, \quad (14)$$

where  $X'$  is the individual requiring chaotic perturbation.  $X_n$  is the amount of chaotic perturbation generated.  $X_n'$  is the individual after chaotic perturbation.

Gaussian mutation [51] is an optimized strategy modified from a genetic algorithm mutation operation. Gaussian mutation operation generates random numbers obeying normal distribution to generate new positions by acting on the original position vector. It implements neighborhood search instruction in a small range to distribute most of the variational operators in the original position. The advantages of high optimization accuracy and the difficulty of falling into local optimization are gained by the optimization algorithm. A small fraction of mutation operators away from the current position makes the potential region search more advantageous and the population diversity richer. Therefore, Gaussian mutation is exploited to modify the algorithm, which will result in a much faster search speed and a much faster convergence trend. The Gaussian probability density equation is as follows:

$$M(x) = x(1 + G(0, 1)), \quad (15)$$

where  $x$  is the initial parameter value and  $G$  is the Gaussian normally displaced stochastic number with an expectation value of 0 and a standard deviation of 1.

**2.3. An Improved SSA.** SSA is an algorithm with simple structure, easy implementation, few control parameters, and strong local searchability. It obtains the initial position of the sparrow based on the random initialization method. Although this approach ensures the randomness of the initial positions, the optimal values of the initial positions of some individuals are too different from the actual optimal values, which reduces the convergence speed and the accuracy of the solution. The blind production of initial positions is prone to the phenomenon of overlapping aggregation of initial solutions. It will lead to a low probability of solution space coverage and a low rate of change of population individuals. The pseudorandom number generator is an ideal information source with excellent statistics and stochastic properties. The chaotic map has high randomness and easy implementation, and it can randomly generate chaotic numbers between 0 and 1. Therefore, chaotic maps are ideal for pseudorandom number generators. The introduction of chaotic map in SSA can effectively improve the initialization population blindness problem of the algorithm. The introduction of the chaotic map can effectively increase the global search capability of SSA. Gaussian mutation optimization strategy can strengthen the local search capability of the population and improve search accuracy. To protect against the solution stagnation phenomenon caused by the

emergence of local optimum, the chaotic perturbation strategy is introduced into SSA. Some local optimal individuals are endowed by chaotic perturbations with a “new dynamism” capable of stepping outside the local optimum. The optimization strategy in SSA directly affects the convergence precision, search capability, and velocity. Strategy selection is crucial to the performance of SSA. In this study, a chaotic Gaussian sparrow search algorithm (CGSSA) based on a multistrategy fusion mechanism is developed by introducing a chaotic map, Gaussian mutation, and chaotic perturbation strategies. The detailed steps of the CGSSA execution are described below.

*Step 1.* Initialize the population size  $N$ , the number of discoverers  $P_a$ , the number of scouting warning sparrows  $S_a$ , the dimensionality of the objective function  $D$ , the upper bound  $ub$ , lower bound  $lb$  of the initial value, and the maximum number of iterations  $T$ .

*Step 2.* Initialize the sparrow population by chaotic sequences to generate  $N * D$  dimensional vectors  $Z$ . Each component of  $Z$  is brought into a defined range of values by equations (4)–(12).

*Step 3.* Calculate the fitness  $f_i$  of each sparrow, select the current optimal fitness  $f_b$  and its correspondent position  $X_b$  of each sparrow.

*Step 4.* Select the top  $P_a$  sparrows with the best fitness as discoverers and the rest as followers. Update the positions of discoverer and follower according to (1) and (2).

*Step 5.* Randomly select  $S_a$  sparrows from the sparrow population as reconnaissance alerts. Update their positions according to (3).

*Step 6.* Recompute the fitness value of individual sparrows and the average fitness value of the sparrow population after each iteration.

*Step 7.* If  $f_i < f_a$ , perform a Gaussian mutation operation on the aggregated sparrow population according to (15). Compare the postmutation individuals with the pre-mutation individuals. Determine whether to accept the position of the postmutation sparrow individual. If  $f_i \geq f_a$ , perform a chaotic perturbation operation on the dispersed population of sparrows by (4)–(12). Compare post-disturbance individuals with pre-disturbance individuals. Determine whether to accept the location of the post-perturbation individual sparrows.

*Step 8.* Get the current state of the sparrow population. Update the optimal position  $X_b$  and its fitness  $f_b$  by the whole sparrow population.

*Step 9.* If the algorithm runs to the maximum number of iterations, end the loop and output the search results. Otherwise, return to Step 4.

Chaotic map strategy is used by CGSSA to initialize the population to improve the population diversity. Both Gaussian mutation and chaotic perturbation strategies are introduced into the SSA, targeting to solve the sparrow divergence and aggregation problems. The local search ability of SSA in the focal search region is enhanced by Gaussian mutation and chaotic perturbation. To find the optimal chaotic map and chaotic perturbation strategies, this study tests the performance of nine chaotic map and chaotic perturbation strategies combined with the Gaussian mutation strategy to optimize SSA, respectively. The benchmark functions are chosen for the test functions, respectively.

*2.4. Benchmark Functions Test.* To select the chaotic map with the highest adaptation to the SSA, fourteen benchmark functions [52–54] are selected in this study. The original SSA and the nine improved algorithms are validated. The fourteen benchmark functions are given in Table 1. The solution space diagram of the fourteen benchmark functions is illustrated in Figure 1. The parameters of the 10 algorithms are set as follows.

The number of populations is 30. The maximum number of iterations is 500. The dimension of the objective function and the range of initial values are kept consistent with Table 1. The number of discoverers and followers is set to 20% of the sparrow population size. To avoid contingency in the search outcomes, each benchmark function is examined 20 times individually. The optimal value, mean, and standard deviation of the run results are assessed to determine the robustness of each algorithm. Test outcomes of fourteen benchmark functions are listed in Table 2. The bolded data in the table indicate the best value of each function. A comparison of convergence curves of 10 algorithms on benchmark functions is illustrated in Figure 2.

From Table 2 and Figure 2, no matter which chaotic strategy is chosen to improve SSA, its convergence accuracy is better than the traditional SSA. CGSSA is proved to be effective and feasible. Seven functions obtained theoretical optimal values in ten algorithm tests, which are  $F_1$ ,  $F_2$ ,  $F_3$ ,  $F_7$ ,  $F_9$ ,  $F_{11}$ , and  $F_{14}$ . Ten algorithms fall into the same local optimum when testing functions  $F_8$  and  $F_{10}$ . In addition to this, the Chebyshev map achieves a minimum optimum on function  $F_6$ . The iterative map obtains a minimum optimum on functions  $F_4$  and  $F_5$ . The sine map reaches a minimum optimum on two functions, which are  $F_{12}$  and  $F_{13}$ . The iterative map and sine map strategies have the highest number of minimum optimums, which means that these two strategies perform best in terms of optimal solutions.

The mean and standard deviation are a pair of statistical indicators describing the overall characteristics of the data. The mean reaction dataset trend is the standard deviation reaction dataset trend. The overall characteristics of the data can be obtained more comprehensively and accurately by means and standard deviations. From the mean and standard deviation analysis, ten algorithms acquire the smallest mean and standard deviation when testing function  $F_{10}$ . Except for the sinusoidal map, the remaining nine algorithms obtain the smallest standard deviation on function

TABLE 1: Fourteen benchmark functions.

Function name	Function	Dimension	Search space	Optimal value
Sphere	$F_1(x) = \sum_{i=1}^d x_i^2$	30	[-100, 100]	0
Schwefel 2.21	$F_2(x) = \max\{ x_i , 1 \leq i \leq d\}$	30	[-100, 100]	0
Schwefel 2.22	$F_3(x) = \sum_{i=1}^d  x_i  + \prod_{i=1}^d  x_i $	30	[-10, 10]	0
Rosenbrock	$F_4(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30, 30]	0
Step	$F_5(x) = \sum_{i=1}^d ([x_i + 0.5])^2$	30	[-100, 100]	0
Quartic	$F_6(x) = \sum_{i=1}^d ix_i^4 + \text{random}[0, 1]$	30	[-1.28, 1.28]	0
Alpine	$F_7(x) = \sum_{i=1}^d  x_i \sin(x_i) + 0.1x_i $	30	[-10, 10]	0
Rastrigin	$F_8(x) = \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i) + 10)$	30	[-5.12, 5.12]	0
Sum squares	$F_9(x) = \sum_{i=1}^d ix_i^2$	30	[-10, 10]	0
Ackley	$F_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}) - \exp(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)}) + 20 + e$	30	[-100, 100]	0
Maytas	$F_{11}(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	2	[-10, 10]	0
Levi	$F_{12}(x) = \sin^2(3\pi x_1) + (x_1 - 1)(1 + \sin^2(3\pi x_2)) + (x_2 - 1)(1 + \sin^2(2\pi x_2))$	2	[-10, 10]	0
Booth	$F_{13}(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	2	[-10, 10]	0
Three-Hump	$F_{14}(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	2	[-10, 10]	0

$F_{11}$ . Seven algorithms obtained the smallest standard deviation on  $F_1$ , which are SSA, Chebyshev map, circle map, iterative map, sine map, singer map, and logistic map. Five algorithms obtained the smallest standard deviation on function  $F_{11}$ . Seven algorithms obtained the smallest standard deviation on  $F_1$ , which are SSA, Chebyshev map, circle map, iterative map, sine map, singer map, and logistic map. Five algorithms obtained the smallest standard deviation on  $F_9$ , which are Chebyshev map, circle map, iterative map, logistic map, and cubic map. Six algorithms obtained the smallest standard deviation on  $F_{14}$ , which are SSA, Chebyshev map, iterative map, sine map, logistic map, and cubic map. In addition to this, the Chebyshev map reaches a smallest standard deviation on  $F_8$ . Sine map acquires the smallest standard deviation on two functions, which are  $F_7$  and  $F_{11}$ . Iterative map acquires the smallest standard deviation on seven functions, which are  $F_2, F_3, F_4, F_5, F_6, F_{12}$ , and  $F_{13}$ . Eight algorithms fall into the same mean when testing functions  $F_8$ , which are Chebyshev map, circle map, iterative map, sine map, singer map, sinusoidal map, logistic map, and cubic map. In addition to this, the Chebyshev map reaches a smallest mean on  $F_{11}$ . Circle map acquires the smallest mean on  $F_1$ . Sine map acquires the smallest mean on  $F_7$ . Iterative map acquires the smallest standard deviation on nine functions, which are  $F_2, F_3, F_4, F_5, F_6, F_9, F_{12}, F_{13}$ , and  $F_{14}$ . The minimum means and minimum standard deviations of iterative map strategy are the highest. Iterative map strategy is best played in the standard deviation of the average value. Therefore, iterative map, iterative perturbation, and Gaussian mutation optimized SSA (IIGSSA) are identified for subsequent studies.

### 3. Development and Evaluation of IGSSA-DNN Model for Esophageal Cancer Dataset

**3.1. Deep Neural Network.** DNN is a method for learning the neural structure of the brain to mimic the processing of

information [55, 56]. The DNN structure is comprised by multiple perceptrons, also known as multilayer feedforward neural network. DNN has strong learning capability, self-learning capability, and self-adaptive capability. DNN is a complex pattern recognition network system capable of simulating more complex models or representing more abstract relationships of things. DNN is appropriate for big data analysis, which is reflected in disease prediction, medical image recognition, cancer diagnosis, etc.

From the structural observation, DNN can be divided into an input layer, a multilayer hidden layers, and an output layer, as illustrated in Figure 3.  $I_1, I_2, \dots, I_h$  stands the input layer, which is intended to accept messages from exterior devices or systems. The  $j$  layer hidden layer effectively assumes the task of the computational engine in the entire network.  $O_1, O_2, \dots, O_z$  stands for the output layer, which enables to take decisions on the inputs.  $h$  indicates the amount of the input layer's neurons.  $z$  means the amount of the output layer's neurons. The output of each vector in the network layer is expressed by (16).

$$Y_{m,n} = f(W_{m,n}^T X_{m-1} + BH_{m-1}), \quad (16)$$

where  $Y_{m,n}$  is the output value of the  $n$  biased neurons at layer  $m$ , and  $W_{m,n}^T$  is the weight value of the  $n$  biased neurons at layer  $m$ .  $X_{m-1}$  is the output value of all neurons of  $m-1$  layer,  $B$  is the biased neuron of  $m-1$  layer, and  $f(\bullet)$  is the activation function.

To enhance the representativeness and versatility of the model, activation functions are introduced into DNN to perform nonlinear transformations on the inputs. In this study, the Sigmoid function as the activation function of the DNN is elected to prevent data scattering during transmission [57].  $x$  is set as the input. The Sigmoid function is expressed in (17).

$$\text{Sigmoid}(x) = \frac{1}{1 + e^x}. \quad (17)$$

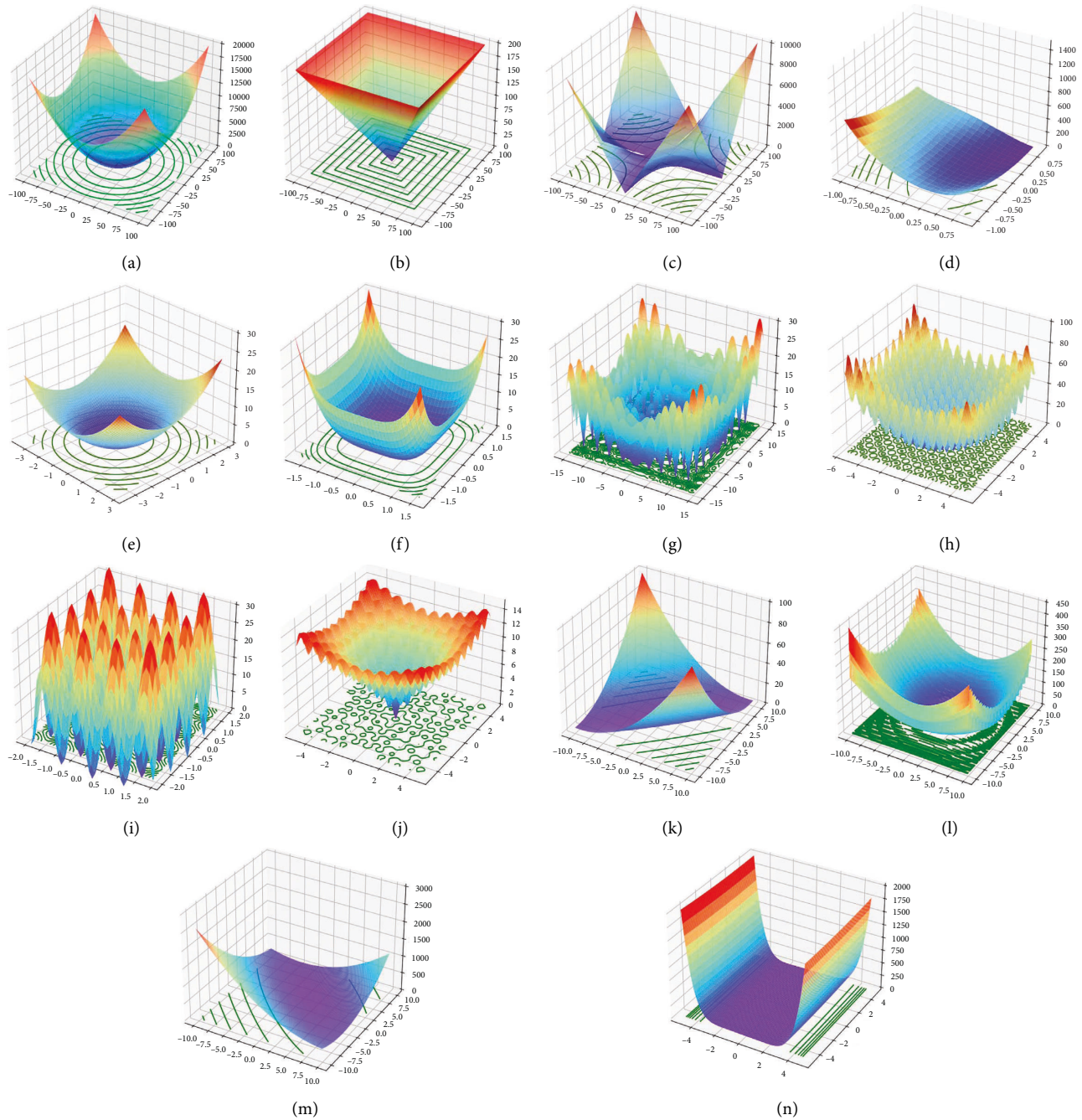


FIGURE 1: Solution space diagram for nineteen benchmark functions. (a) Sphere function. (b) Schwefel 2.21 function. (c) Schwefel 2.22 function. (d) Rosenbrock function. (e) Step function. (f) Quartic function. (g) Alpine function. (h) Rastrigin function. (i) Sum squares function. (j) Ackley function. (k) Maytas function. (l) Levi function. (m) Booth function. (n) Three-Hump function.

The training procedure of DNN follows the five steps:

- (1) Define the network structure including the features of the input and output layers, the number of nodes in the network layer, and the implicit layer.
- (2) Randomly generate initialized weights and biases.
- (3) Forward propagation to obtain the predicted values.
- (4) Calculate the loss function and prediction error for forwarding propagation.

- (5) Reverse propagation to update the weights and biases.

Until the loss function is minimized and the training data is not trained by overfitting, the optimal model is obtained. Therefore, weights play a crucial role in DNN training. The loss function is a fundamental criterion to judge the performance of a neural network. The loss function is sensitive to small changes in weights and biases. The susceptibility of the loss function renders it the best way to

TABLE 2: Examine outcomes for fourteen benchmark functions.

Function	Algorithms	Optimal value	Mean value	Standard deviation
$F_1$	SSA	0	8.037E-114	3.594E-113
	Tent	0	5.311E-202	0
	Chebyshev	0	7.017E-245	0
	Circle	0	0	0
	Iterative	0	4.101E-183	0
	Sine	0	1.298E-198	0
	Singer	0	7.994E-212	0
	Sinusoidal	0	1.202E-160	5.376E-160
	Logistic	0	4.07E-257	0
	Cubic	0	3.662E-163	2.223E-162
$F_2$	SSA	0	3.856E-45	1.724E-44
	Tent	0	8.548E-155	3.823E-154
	Chebyshev	0	4.031E-130	1.803E-129
	Circle	0	1.481E-127	6.621E-127
	Iterative	0	<b>1.235E-156</b>	<b>4.303E-156</b>
	Sine	0	1.985E-136	8.877E-136
	Singer	0	9.621E-140	5.524E-140
	Sinusoidal	0	1.4722E-122	2.2693E-128
	Logistic	0	2.154E-133	1.312E-132
	Cubic	0	2.934E-133	2.0752E-79
$F_3$	SSA	0	1.673E-27	7.483E-27
	Tent	0	2.714E-122	1.214E-121
	Chebyshev	0	9.729E-96	4.351E-95
	Circle	0	1.564E-74	6.994E-74
	Iterative	0	<b>5.909E-129</b>	<b>2.643E-129</b>
	Sine	0	2.99E-109	1.337E-108
	Singer	0	7.224E-105	3.231E-104
	Sinusoidal	0	1.316E-65	5.887E-65
	Logistic	0	1.106E-122	4.946E-122
	Cubic	0	1.165E-117	5.209E-117

TABLE 2: Continued.

Function	Algorithms	Optimal value	Mean value	Standard deviation
$F_4$	SSA	1.223E-05	1.142E-03	2.778E-03
	Tent	4.321E-08	5.348E-06	8.617E-06
	Chebyshev	5.946E-07	7.212E-05	8.992E-05
	Circle	1.033E-05	3.929E-04	5.017E-04
	Iterative	<b>2.255E-08</b>	<b>4.452E-06</b>	<b>6.736E-06</b>
	Sine	2.714E-07	1.764E-05	1.828E-05
	Singer	1.128E-06	7.727E-05	8.092E-05
	Sinusoidal	2.735E-04	1.373E-05	2.526E-04
	Logistic	1.344E-07	1.124E-05	1.206E-05
	Cubic	3.213E-08	1.333E-05	1.568E-05
$F_5$	SSA	3486E-04	6.432E-03	5177E-03
	Tent	4.018E-07	9.14E-05	9.994E-05
	Chebyshev	1.981E-08	1.796E-05	1.97E-05
	Circle	4.387E-08	8.52E-04	9.895E-05
	Iterative	<b>3.0466E-08</b>	<b>1.774E-05</b>	<b>1.711E-05</b>
	Sine	1.496E-06	9.726E-04	9.667E-05
	Singer	1.618E-06	7.313E-04	8.238E-05
	Sinusoidal	3.918E-06	1.317E-04	1.371E-04
	Logistic	9.34E-06	7.424E-04	6.168E-04
	Cubic	7.205E-06	1.032E-04	1.236E-04
$F_6$	SSA	3.1E-03	3.399E-02	2.059E-02
	Tent	2.912E-04	6.554E-03	4.036E-03
	Chebyshev	<b>7.183E-05</b>	4.367E-03	4.201E-03
	Circle	8.295E-04	8.557E-03	6.841E-03
	Iterative	3.63E-04	<b>4.008E-03</b>	<b>2.842E-03</b>
	Sine	1.218E-03	6.723E-03	4.694E-03
	Singer	5.581E-05	7.878E-03	5.441E-03
	Sinusoidal	5.971E-04	8.378E-03	7.758E-03
	Logistic	2.541E-04	7.907E-03	6.586E-03
	Cubic	4.965E-04	7.344E-03	4.835E-03
$F_7$	SSA	0	5.837E-60	2.611E-59
	Tent	0	3.32E-119	1.485E-118
	Chebyshev	0	8.955E-90	4.005E-89
	Circle	0	1.03E-140	4.608E-140
	Iterative	0	9.454E-138	4.228E-137
	Sine	0	<b>2.375E-212</b>	<b>0</b>
	Singer	0	1.417E-102	6.335E-102
	Sinusoidal	0	3.156E-99	1.412E-98
	Logistic	0	5.296E-85	2.368E-84
	Cubic	0	3.802E-100	1.701E-99
$F_8$	SSA	<b>1.008E-01</b>	2.237E-01	1.688E-01
	Tent	<b>1.008E-01</b>	1.011E-01	3.295E-02
	Chebyshev	<b>1.008E-01</b>	<b>1.009E-01</b>	<b>5.37E-05</b>
	Circle	<b>1.008E-01</b>	<b>1.009E-01</b>	1.034E-04
	Iterative	<b>1.008E-01</b>	<b>1.009E-01</b>	1.323E-04
	Sine	<b>1.008E-01</b>	<b>1.009E-01</b>	1.323E-04
	Singer	<b>1.008E-01</b>	<b>1.009E-01</b>	1.502E-04
	Sinusoidal	<b>1.008E-01</b>	<b>1.009E-01</b>	1.408E-04
	Logistic	<b>1.008E-01</b>	<b>1.009E-01</b>	1.608E-04
	Cubic	<b>1.008E-01</b>	<b>1.009E-01</b>	1.407E-04



TABLE 2: Continued.

Function	Algorithms	Optimal value	Mean value	Standard deviation
$F_9$	SSA	0	6.364E-105	3.486E-104
	Tent	0	5.818E-162	3.19E-161
	Chebyshev	0	2.456E-195	0
	Circle	0	1.067E-198	0
	Iterative	0	<b>2.373E-244</b>	0
	Sine	0	3.023E-112	1.656E-111
	Singer	0	2.521E-141	1.381E-140
	Sinusoidal	0	5.721E-140	3.133E-139
	Logistic	0	3.719E-203	0
	Cubic	0	3.758E-173	0
$F_{10}$	SSA	<b>4.441E-16</b>	<b>4.441E-16</b>	0
	Tent	<b>4.441E-16</b>	<b>4.441E-16</b>	0
	Chebyshev	<b>4.441E-16</b>	<b>4.441E-16</b>	0
	Circle	<b>4.441E-16</b>	<b>4.441E-16</b>	0
	Iterative	<b>4.441E-16</b>	<b>4.441E-16</b>	0
	Sine	<b>4.441E-16</b>	<b>4.441E-16</b>	0
	Singer	<b>4.441E-16</b>	<b>4.441E-16</b>	0
	Sinusoidal	<b>4.441E-16</b>	<b>4.441E-16</b>	0
$F_{11}$	SSA	0	4.594E-182	0
	Tent	0	1.434E-245	0
	Chebyshev	0	0	0
	Circle	0	5.773E-238	0
	Iterative	0	6.493E-226	0
	Sine	0	1.862E-243	0
	Singer	0	4.438E-315	0
	Sinusoidal	0	2.401E-145	1.074E-144
	Logistic	0	2.014E-282	0
	Cubic	0	1.378E-190	0
$F_{12}$	SSA	2.324E-06	2.009E-05	9.768E-05
	Tent	1.093E-07	1.083E-05	1.845E-05
	Chebyshev	4.92E-07	5.578E-06	6.295E-06
	Circle	9.458E-08	6.071E-06	6.68E-06
	Iterative	1.19E-08	<b>3.015E-06</b>	<b>3.344E-06</b>
	Sine	<b>5.444E-09</b>	5.002E-06	4.757E-06
	Singer	2.386E-08	5.343E-06	5.608E-06
	Sinusoidal	4.846E-08	6.556E-06	5.957E-06
Logistic	1.144E-07	6.205E-06	5.926E-06	
Cubic	2.044E-07	6.126E-06	8.436E-06	

TABLE 2: Continued.

Function	Algorithms	Optimal value	Mean value	Standard deviation
$F_{13}$	SSA	2.647E-06	2.611E-05	2.979E-05
	Tent	9.904E-07	7.571E-04	8.334E-04
	Chebyshev	8.105E-07	1.441E-05	1.482E-05
	Circle	5.826E-07	1.133E-05	1.619E-05
	Iterative	5.674E-07	<b>1.009E-05</b>	<b>8.207E-06</b>
	Sine	<b>1.501E-08</b>	1.272E-05	1.293E-05
	Singer	1.144E-07	1.387E-05	1.481E-05
	Sinusoidal	2.314E-08	1.504E-05	2.021E-05
	Logistic	3.14E-08	1.363E-05	1.683E-05
	Cubic	2.296E-07	1.36E-05	1.427E-05
$F_{14}$	SSA	0	4.752E-133	2.125E-132
	Tent	0	5.872E-268	0
	Chebyshev	0	7.727E-268	0
	Circle	0	2.862E-152	1.28E-151
	Iterative	0	0	0
	Sine	0	2.875E-311	0
	Singer	0	2.985E-122	1.335E-121
	Sinusoidal	0	1.003E-159	4.487E-159
	Logistic	0	9.108E-224	0
	Cubic	0	1.866E-260	0

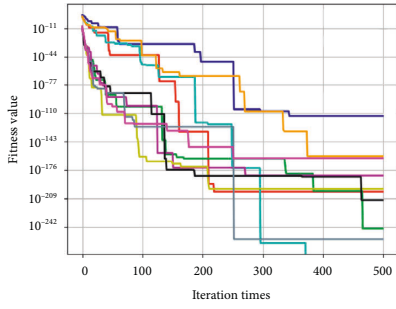
find the best weight parameter for DNN [58]. The interlayer weights of the DNN model are optimized by adjusting the error rate, and the adaptability of the model to the current dataset is adjusted, which has a substantial gain on the accuracy of the DNN model output. Mean square error, cross-entropy error, and root mean square error are common loss functions. In this study, the mean square error (MSE) is employed to calculate the error rate, as shown in

$$MSE = \frac{1}{m} \sum_{k=1}^m (O_k - A_k)^2, \quad (18)$$

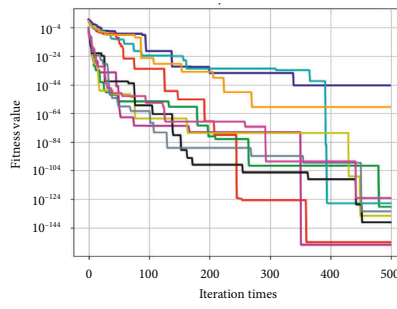
where  $m$  is the sample number in the training dataset.  $o_k$  is the model output produced by the  $k^{th}$  input.  $A_k$  is the  $K^{th}$  actual output. The DNN model is trained by data inputs, and the weight values are obtained to predict the output results.

#### 4. An Improved DNN Based on IGSSA

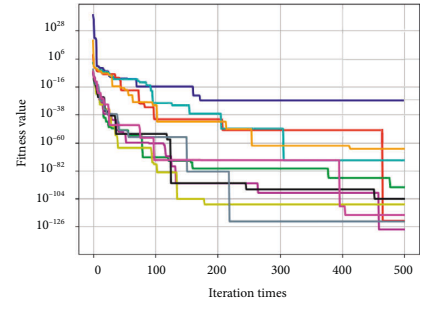
The optimal weights of a DNN have a direct correlation with the prediction results of the model. In this study, the accuracy of IGSSA has been verified in Section II, and thus a method to improve DNN based on IIGSSA is proposed. DNN mixed with the proposed IIGSSA is applied to determine the optimal weights of DNN, which will help to enhance the efficiency of the model. The optimum solution of all sparrows in IIGSSA algorithm is exploited to renew the



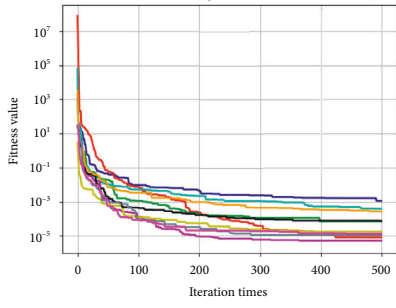
(a)



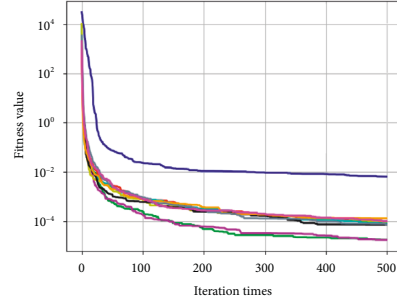
(b)



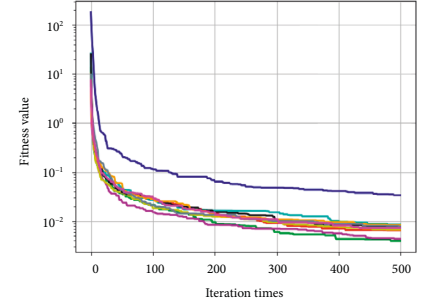
(c)



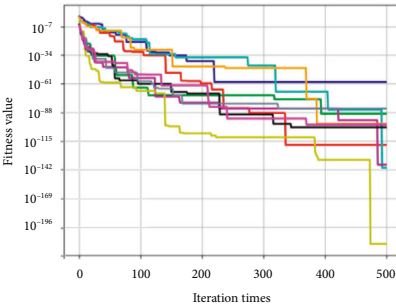
(d)



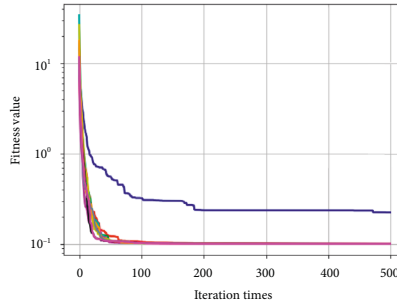
(e)



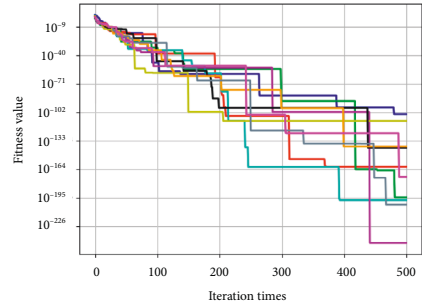
(f)



(g)



(h)



(i)

FIGURE 2: Continued.

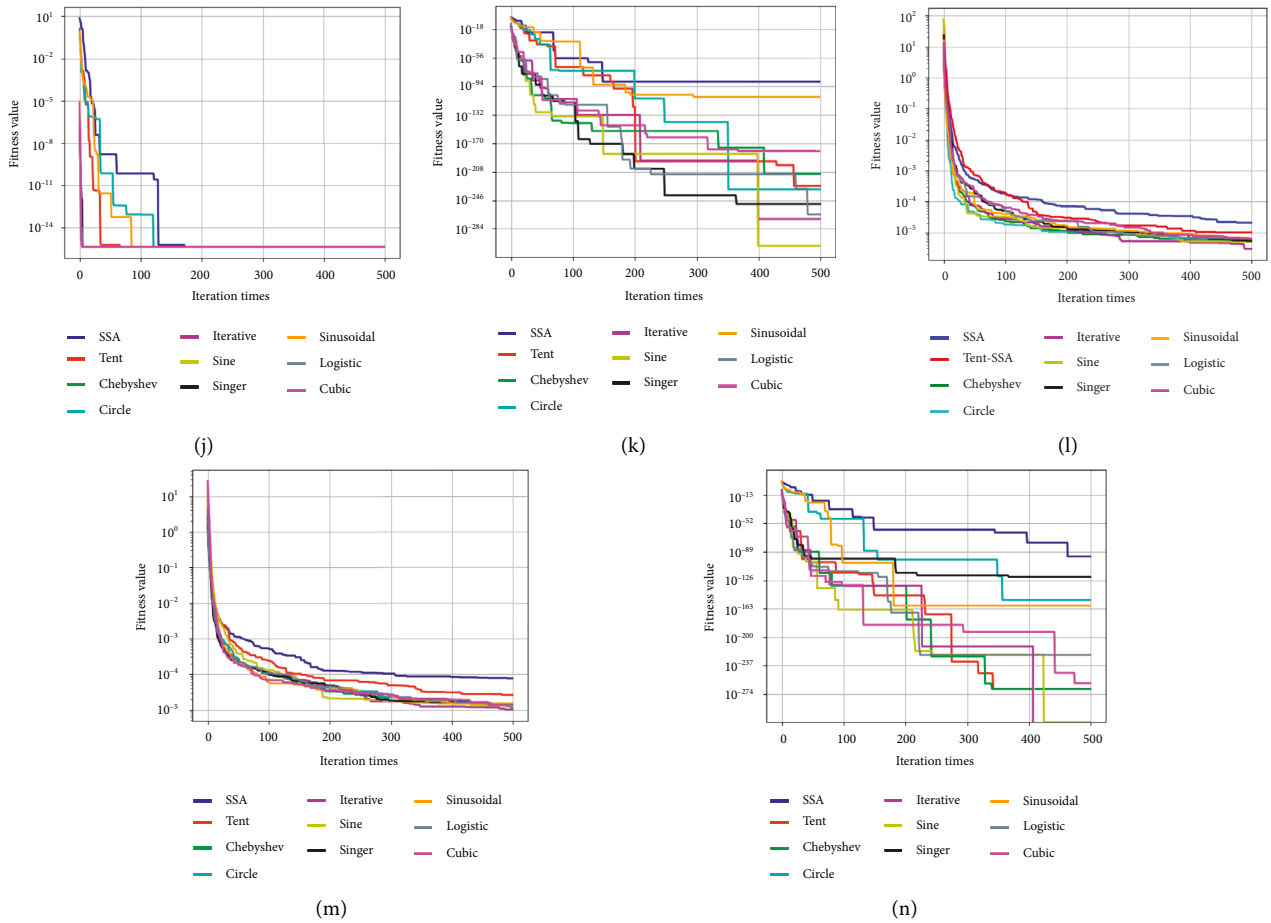


FIGURE 2: Comparison of convergence curves of 10 algorithms on benchmark functions. (a) Sphere function. (b) Schwefel 2.21 function. (c) Schwefel 2.22 function. (d) Rosenbrock function. (e) Step function. (f) Quartic function. (g) Alpine function. (h) Rastrigin function. (i) Sum squares function. (j) Ackley function. (k) Maytas function. (l) Levi function. (m) Booth function. (n) Three-Hump function.

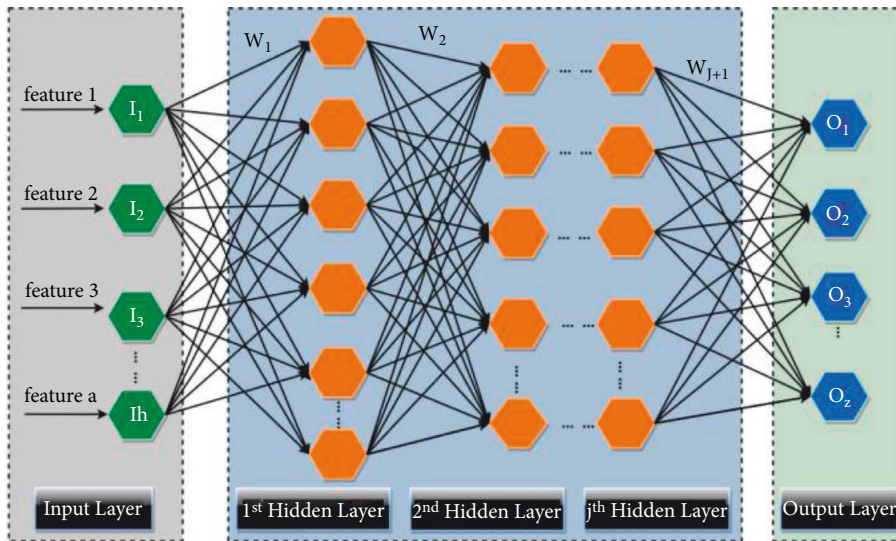


FIGURE 3: DNN architecture.

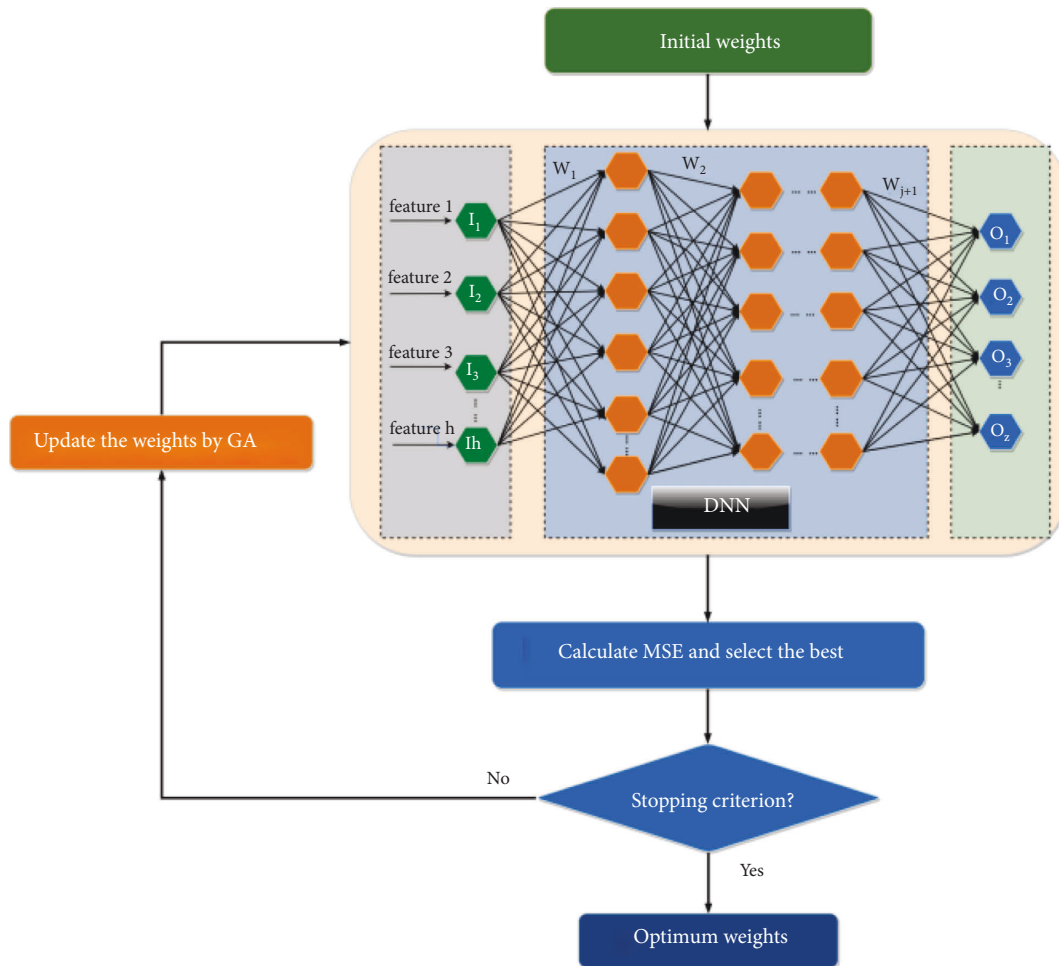


FIGURE 4: Training phase of the proposed IGSSA for DNN.

location, and the optimum solution can be reached by the highest number of iterations. The initial weight values are stochastically produced depending on the specified range. The amount of weights of the DNN stands for the number of dimensions of IIGSSA. MSE is chosen as the fitness function of IIGSSA. The training phase of the proposed IGSSA for DNN is displayed in Figure 4. The AUC values are used to test the performance of the evaluated DNN.

**4.1. Dataset Analysis.** The clinical dataset and biological sample dataset of esophageal cancer patients are selected for this study. The dataset is obtained from the State Key Laboratory of Esophageal Cancer Control in Henan Province. A total of 398 patients in the available dataset have complete information, and they are all diagnosed with esophageal cancer in 2007. Continuous metrics in the dataset are shown in Table 3. Discrete metrics in the dataset are indicated in Table 4.

**4.2. Performance Evaluation.** In this study, a modeling method of optimizing DNN through IIGSSA is proposed. Seventeen blood indicators are provided in the existing esophageal cancer dataset, namely WBC, LY, MONO,

NEUT, EOS, BASO, RBC, HB, PLT, TP, ALB, GLB, PT, APTT, TT, and FIB. Seven items of tumor information are supplied, namely tumor length, tumor width, tumor thickness, degree of differentiation, tumor location, transfer situation, and TNM stages. The two physical characteristics are age and gender. Use these 26 features as the input dataset, and the IIGSSA-DNN algorithm is used to build a prognostic model. The overall flow chart of IIGSSA-DNN is illustrated in Figure 5. The robustness and accuracy of IIGSSA-DNN are validated by this study. The IIGSSA-DNN is evaluated against existing scoring criteria to measure the strengths and weaknesses of the proposed algorithm. The most commonly available rubrics [59] are Acc, FPR, REC, TPR, PRE, TNR, AUC, F1-M, and G-M. To avoid the limitations and specificity of fixed division datasets, five-fold cross-validation is employed for objective evaluation of the model.

The survival time of esophageal cancer patients is predicted by the proposed IIGSSA-DNN. Five optimization algorithms have been achieved well in the field of neural networks [60–63], which are particle swarm algorithm (PSO), GWO, GSA, WOA, and SSA. Five optimization algorithms are also used to optimize the DNN and tested on the esophageal cancer dataset in comparison with IIGSSA-

TABLE 3: Continuous metrics in the dataset.

Variable	Mean	Median (range)	Variance
Tumor length	4.112	4 (1–11)	3.208
Tumor width	2.649	2.5 (0.3–9)	1.148
Tumor thickness	1.1776	1 (0.1–8)	0.471
WBC	6.5366	6.2 (2.5–13.6)	3.6958
LY	1.7622	1.8 (0–4)	0.3652
MONO	0.3899	0.4 (0–1.4)	0.06661
NEUT	4.0011	3.7 (0–9.8)	2.8097
EOS	0.1238	0.1 (0–0.9)	0.0198
BASO	0.04163	0 (0–5)	0.005549
RBC	4.43	4.48 (2.73–5.75)	0.2289
HB	137.4347	138 (64–169)	223.7577
PLT	236.8518	231 (100–448)	52.606
TP	71.0377	71 (50–92)	54.4092
ALB	42.0201	42 (26–59)	25.1281
GLB	29.1533	29 (16–45)	28.8656
PT	10.2271	10.2 (7–16.6)	2.4610
APTT	35.9095	35.1 (15.4–62.2)	52.9934
TT	15.3420	15.5 (10.9–21.3)	2.9607
FIB	387.3433	378.3960 (167.613–774.433)	985.7021
Age	60	60 (38–82)	70.099
Survival time	4	3 (0–11)	12.873

The unit of tumor length, tumor width, tumor thickness is centimeter. The unit of WBC, LY, MONO, NEUT, EOS, BASO, RBC, and PLT is 10<sup>9</sup>/L. The unit of HB, TP, ALB, and GLB is g/L. The unit of PT, APTT, and TT is second(s). The unit of FIB is mg/dL. The unit of survival time is year.

TABLE 4: Discrete metrics in the dataset.

Project	Category	Number of population	Percentage of population (%)
Gender	Male	247	62
	Female	151	38
Degree of differentiation	Poorly differentiated	158	40
	Moderately differentiated	217	54
	Highly differentiated	23	6
Tumor site	Lower thoracic	78	20
	Mid thoracic	267	67
	Upper thoracic	53	13
Transfer situation	Negative	200	50
	Positive	198	50
TNM stages	I	39	10
	II	172	43
	III	166	42
	IV	21	5
Survival status	Live	101	25
	Dead	297	75

DNN. To further verify the classification performance of IIGSSA-DNN, SVM and KNN are used to compare with IIGSSA-DNN. KNN and SVM are two machine learning algorithms widely used in the diagnosis of breast cancer [64]. Python is the operating platform of algorithms mentioned in this paper. The deep learning framework PyTorch is employed to implement DNN partial training. 26 features are set as the input dataset, survival time, and survival state as the output dataset, and a DNN model with 10 layers and 128 nodes is constructed. The Sigmoid function is adopted as the activation function of the DNN. The learning rate of the model is fixed to 0.001 [65]. The learning rate is adaptively and randomly adjusted by Adam algorithm. The first-order moment estimates and second-

order moment estimates of the gradient are calculated by Adam algorithm [66] to adjust the learning rates of the different parameters. In each iteration, the learning rate is limited to a rough range, which makes the parameters more stable. IIGSSA is employed to enable minimizing the MSE to find the optimal weight value. The population size is set to 30, and the maximum number of iterations is set to 1000. The results of the predictive model assessment are shown in Table 5. The five ROC curves of nine models are displayed in Figure 6. Five ROC curves of DNN are expressed in Figure 6(a). Five ROC curves of PSO-DNN are indicated in Figure 6(b). Five ROC curves of GSA-DNN are illustrated in Figure 6(c). Five ROC curves of GWO-DNN are depicted in Figure 6(d). Five ROC curves of WOA-DNN are

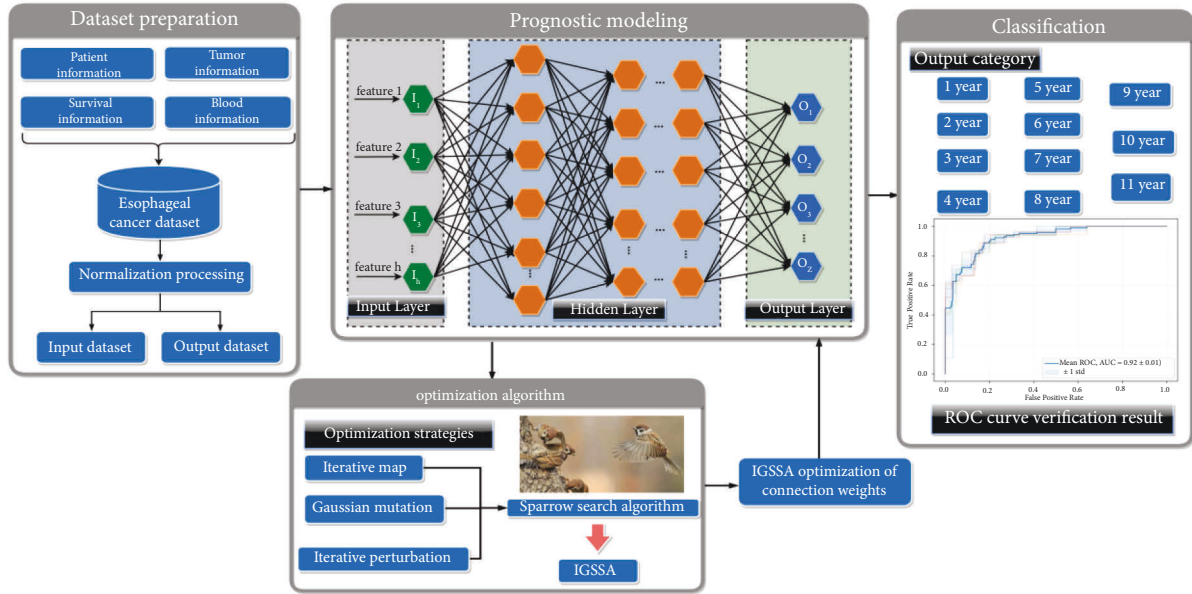


FIGURE 5: The overall flow chart of IGSSA-DNN.

TABLE 5: Results of the predictive model evaluation.

Algorithms	Acc	FPR	REC	PRE	TNR	F1-M	AUC	P
DNN	0.76	0.16	0.76	0.76	0.76	0.76	0.76	0.03
PSO-DNN	0.79	0.133	0.79	0.79	0.79	0.79	0.79	0.04
GSA-DNN	0.83	0.133	0.83	0.83	0.83	0.83	0.83	0.04
GWO-DNN	0.81	0.117	0.81	0.81	0.81	0.81	0.81	0.01
WOA-DNN	0.86	0.140	0.86	0.86	0.86	0.86	0.86	0.01
SSA-DNN	0.89	0.147	0.89	0.89	0.89	0.89	0.89	0.04
IIGSSA-KNN	0.75	0.21	0.75	0.75	0.75	0.75	0.75	0.04
IIGSSA-SVM	0.82	0.18	0.82	0.82	0.82	0.82	0.82	0.04
IIGSSA-DNN	0.92	0.100	0.92	0.92	0.92	0.92	0.92	0.01

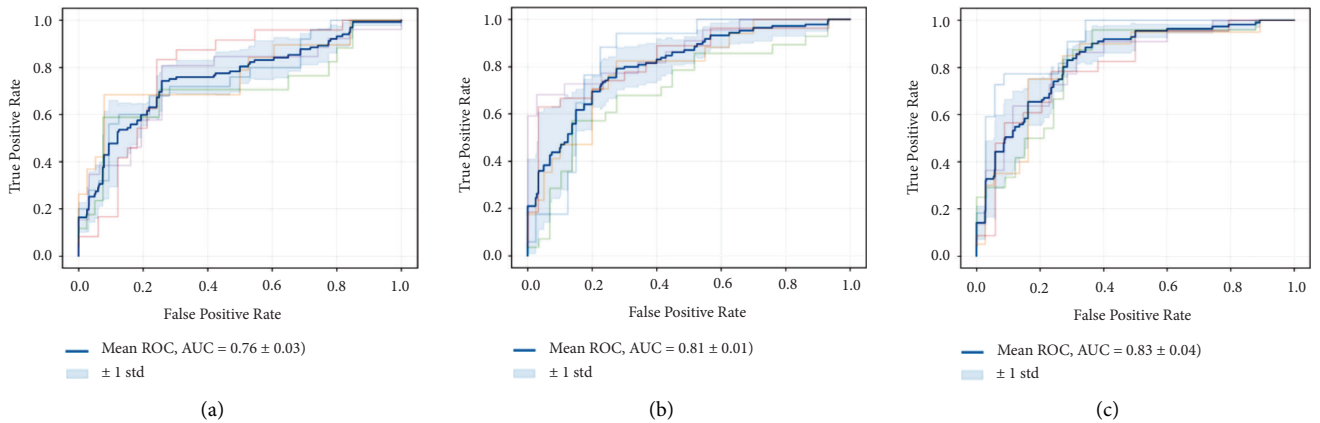


FIGURE 6: Continued.

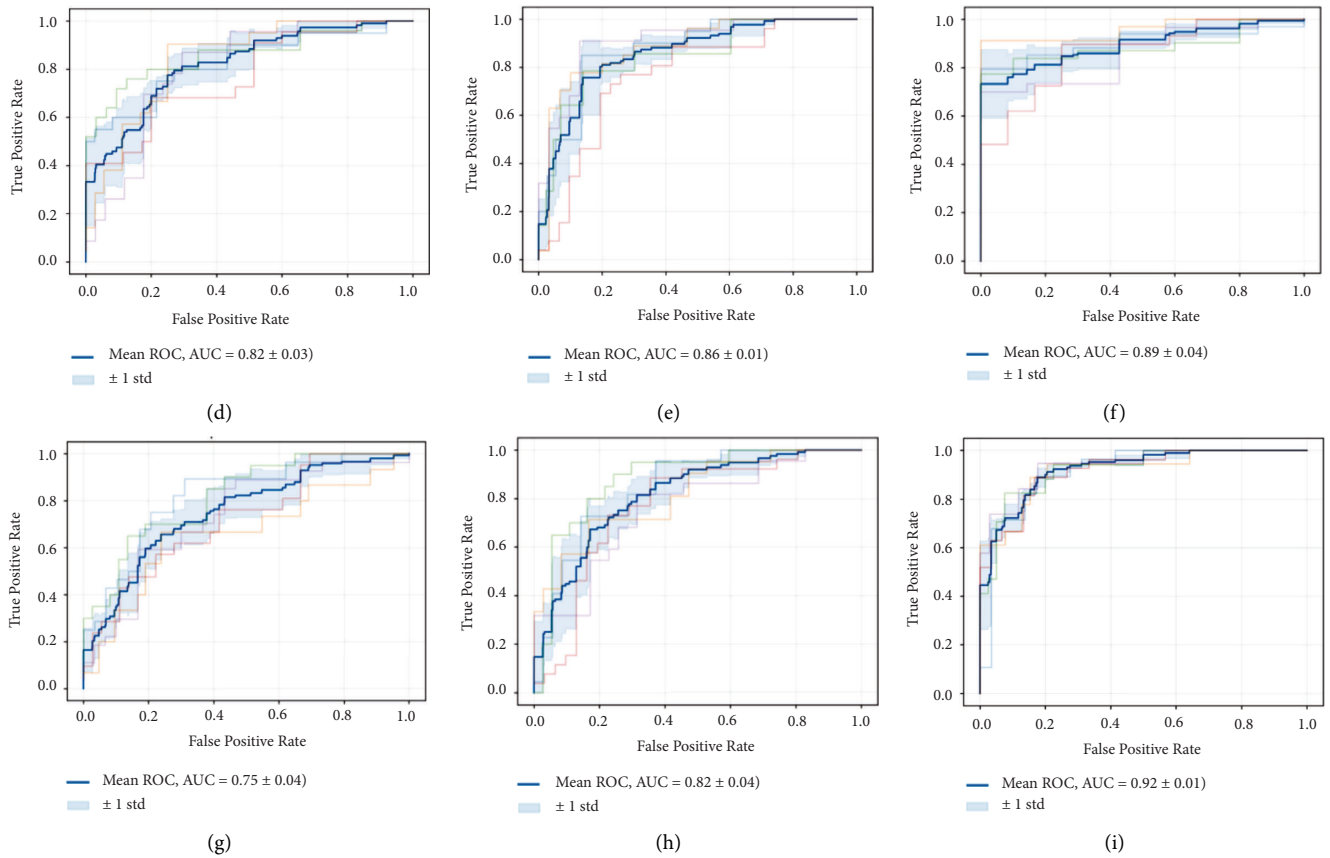


FIGURE 6: Five ROC curves of nine models. (a) Five ROC curves of DNN. (b) Five ROC curves of PSO-DNN. (c) Five ROC curves of GSA-DNN. (d) Five ROC curves of GWO-DNN. (e) Five ROC curves of WOA-DNN. (f) Five ROC curves of SSA-DNN. (g) Five ROC curves of IGSSA-KNN. (h) Five ROC curves of IGSSA-SVM. (i) Five ROC curves of IGSSA-DNN.

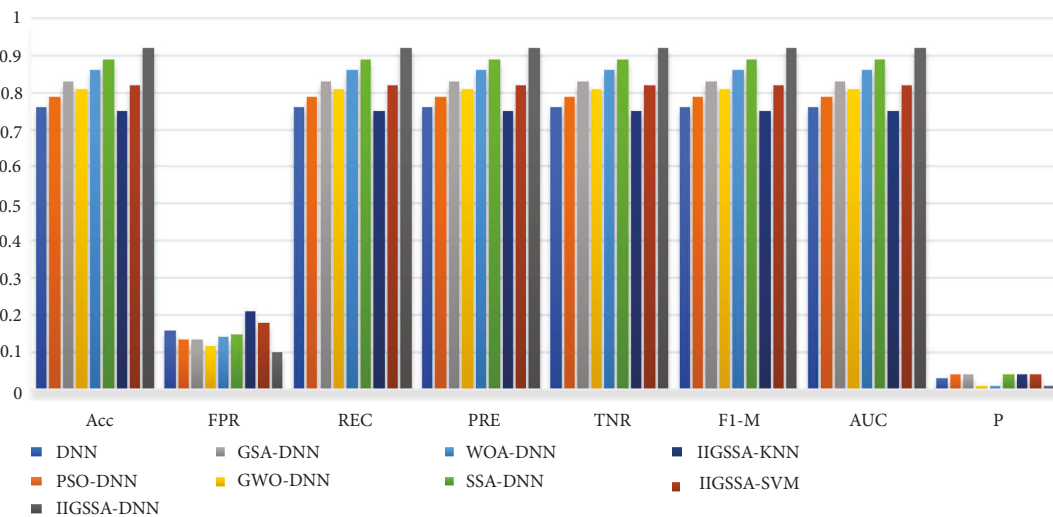


FIGURE 7: Test accuracy of nine models.

demonstrated in Figure 6(e). Five ROC curves of SSA-DNN are seen in Figure 6(f). Five ROC curves of IIGSSA-KNN are expressed in Figure 6(g). Five ROC curves of IIGSSA-SVM are indicated in Figure 6(h). Five ROC curves of IIGSSA-DNN are represented in Figure 6(i). Test accuracy of nine models is shown in Figure 7.

From Table 5, the performance of IIGSSA-DNN for training esophageal cancer dataset is proved to be better than other optimization algorithms. The accuracy of the proposed IIGSSA-DNN is 0.92. The accuracy of IIGSSA-DNN is outperformed by other optimization algorithms of hybrid DNN. The patient’s survival time is well predicted by IIGSSA-

DNN. REC is the probability of correctly classifying positive samples. PRE is the proportion of samples with positive classification results versus those which are actually positive. REC and PRE are a good way to determine the number of accurate samples for classification. The IIGSSA-DNN's REC and PRE are 0.92 outperformed by other algorithms. The IIGSSA-DNN is proven to be better at identifying positive samples. FPR is the rubric for judging percentage of correct positive samples. TPR is the accuracy rate for evaluating the correct classification of positive samples. FPR is a scale for judging the percentage of correct positive samples. PRE is an assessment of the accuracy of classifying positive samples correctly. The value of IIGSSA-DNN is 0.92 on all these rubrics, and the model is confirmed to have a strong performance in distinguishing between negative and positive samples. F1-M is a synthesis assessment index to assess the quality of the model. The F1-M value of IIGSSA-DNN is 0.92 higher than the other indexes. The model is confirmed to have higher model quality than the other models listed in this study. The AUC and  $P$ -values are represented in Figure 6. In Figure 6, the average ROC curve for each algorithm model is plotted as the blue curve. The AUC of IIGSSA-DNN has approached 1 and the  $P$  value is less than 0.05. The model has great statistical significance and better classification performance. Consequently, IIGSSA-DNN is exhibited to be a reliable classification model with much higher performance than the other models listed in this study.

## 5. Conclusion

DNN is a complex pattern recognition network system. The network structure and optimal weights have a drastic impact on the classification results of DNN. How to value the network structure and connection weights is a daunting task. In this study, a new chaotic map and Gaussian mutation are proposed to improve the optimal search strategy of SSA. The iterative chaotic map is validated by fourteen benchmark functions, and the iterative chaotic map improves SSA to a better rate than other chaotic maps. IIGSSA is identified as the best strategy to find the optimal fully connected weights of DNN. IIGSSA-DNN is contrasted with seven algorithms on the esophageal cancer dataset, and the seven algorithms are DNN, PSO-DNN, GSA-DNN, GWO-DNN, SSA-DNN, IIGSSA-KNN, and IIGSSA-SVM. The IIGSSA-DNN is proven to possess optimal performance in predicting the survival time of esophageal cancer patients. The accuracy of IIGSSA-DNN is substantially improved by the self-learning capability of DNN and the efficient search capability of IIGSSA. Therefore, IIGSSA-DNN gains more than traditional machine learning algorithms in dealing with complex issues. It is more accurate to solve the classification and recognition problems. According to this model, doctors can more keenly monitor the progress of each patient, thereby establishing a more benign diagnosis and treatment system. The parameters of the IIGSSA-DNN model rely on empirical selection, which limits the accuracy of prediction. The dataset sample samples used in this institute are relatively small, and the future period is verified in a large number of datasets and clinical trials. Taking the IIGSSA-DNN model

as an opportunity, a secure smart device application is expected to be developed by researchers in the future. The application can bind the preliminary and retraining information of each patient, enabling doctors to customize personalized diagnosis and treatment schemes for each patient.

## Data Availability

The datasets presented in this article are not readily available because the data used in the study are private and confidential data. The datasets' access can be obtained from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the Joint Funds of the National Natural Science Foundation of China, under Grant U1804262, in part by the Foundation of Young Key Teachers from University of Henan Province, under Grant 2018GGJS092, in part by the Youth Talent Lifting Project of Henan Province, under Grant 2018HYTP016, in part by the Henan Province University Science and Technology Innovation Talent Support Plan, under Grant 20HASTIT027, in part by the Zhongyuan Thousand Talents Program, under Grant 204200510003, and in part by the Open Fund of State Key Laboratory of Esophageal Cancer Prevention and Treatment, under Grants K2020-0010 and K2020-0011.

## References

- [1] H. Qiu, S. Cao, and R. Xu, "Cancer incidence, mortality, and burden in China: a time-trend analysis and comparison with the United States and United Kingdom based on the global epidemiological data released in 2020," *Cancer Communications*, vol. 41, no. 10, pp. 1037–1048, 2021.
- [2] R. J. Kelly, J. A. Ajani, J. Kuzdzal et al., "Adjuvant nivolumab in resected esophageal or gastroesophageal junction cancer," *New England Journal of Medicine*, vol. 384, no. 13, pp. 1191–1203, 2021.
- [3] C.-Y. Wang, J.-T. Tsai, C.-H. Fang, T.-F. Lee, and J.-H. Chou, "Predicting survival of individual patients with esophageal cancer by adaptive neuro-fuzzy inference system approach," *Applied Soft Computing*, vol. 35, pp. 583–590, 2015.
- [4] E. Smyth, V. Gambardella, A. Cervantes, and T. Fleitas, "Checkpoint inhibitors for gastroesophageal cancers: dissecting heterogeneity to better understand their role in first-line and adjuvant therapy," *Annals of Oncology*, vol. 32, no. 5, pp. 590–599, 2021.
- [5] S. Urakawa, T. Makino, M. Yamasaki et al., "Lymph node response to neoadjuvant chemotherapy as an independent prognostic factor in metastatic esophageal cancer," *Annals of Surgery*, vol. 273, no. 6, pp. 1141–1149, 2021.
- [6] M. Watanabe, R. Otake, R. Kozuki et al., "Recent progress in multidisciplinary treatment for patients with esophageal cancer," *Surgery Today*, vol. 50, no. 1, pp. 12–20, 2020.



- [7] T. Shan, F. Tay, and L. Gu, "Application of artificial intelligence in dentistry," *Journal of Dental Research*, vol. 100, no. 3, pp. 232–244, 2021.
- [8] A. Vulli, P. N. Srinivasu, M. S. K. Sashank, J. Shafi, J. Choi, and M. F. Ijaz, "Fine-tuned densenet-169 for breast cancer metastasis prediction using fastai and 1-cycle policy," *Sensors*, vol. 22, no. 8, p. 2988, 2022.
- [9] M. F. Ijaz, M. Attique, and Y. Son, "Data-driven cervical cancer prediction model with outlier detection and over-sampling methods," *Sensors*, vol. 20, no. 10, p. 2809, 2020.
- [10] X. Zhou, Y. Li, and W. Liang, "Cnn-rnn based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, 2021.
- [11] Y. Horie, T. Yoshio, K. Aoyama et al., "Diagnostic outcomes of esophageal cancer by artificial intelligence using convolutional neural networks," *Gastrointestinal Endoscopy*, vol. 89, no. 1, pp. 25–32, 2019.
- [12] C. Liu, Y. Huang, F. Huang, and J. Yu, "Multifeature Deep Cascaded Learning for Ppg Biometric Recognition," *Scientific Programming*, p. 2022, 2022.
- [13] S. Jin, X. Zeng, F. Xia, W. Huang, and X. Liu, "Application of deep learning methods in biological networks," *Briefings in Bioinformatics*, vol. 22, no. 2, p. 1902, 2021.
- [14] S. Zhao, Y. He, J. Qin, and Z. Wang, "A semi-supervised deep learning method for cervical cell classification," *Analytical Cellular Pathology*, vol. 2022, pp. 1–12, 2022.
- [15] X. Li and W. Zhang, "Deep learning-based partial domain adaptation method on intelligent machinery fault diagnostics," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4351–4361, 2021.
- [16] P. Kedia, R. Katarya, and R. Katarya, "Covnet-19: a deep learning model for the detection and analysis of covid-19 patients," *Applied Soft Computing*, vol. 104, Article ID 107184, 2021.
- [17] X. Zheng, Z. Yao, Y. Huang et al., "Deep learning radiomics can predict axillary lymph node status in early-stage breast cancer," *Nature Communications*, vol. 11, no. 1, pp. 1236–1239, 2020.
- [18] J. Dong, Y. Chen, B. Yao, X. Zhang, and N. Zeng, "A Neural Network Boosting Regression Model Based on Xgboost," *Applied Soft Computing*, Article ID 109067, 2022.
- [19] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.
- [20] Y. Shiu, K. J. Palmer, M. A. Roch et al., "Deep neural networks for automated detection of marine mammal species," *Scientific Reports*, vol. 10, no. 1, pp. 607–612, 2020.
- [21] A. H. Ribeiro, M. H. Ribeiro, G. M. M. Paixão et al., "Automatic diagnosis of the 12-lead ecg using a deep neural network," *Nature Communications*, vol. 11, no. 1, pp. 1760–1769, 2020.
- [22] L. Sk, S. N. Mohanty, K. Shankar, N. Arunkumar, and G. Ramirez, "Optimal deep learning model for classification of lung cancer on ct images," *Future Generation Computer Systems*, vol. 92, pp. 374–382, 2019.
- [23] M. Hammad, A. M. Iliyasa, A. Subasi, E. S. L. Ho, and A. A. El-Latif, "A multitier deep learning model for arrhythmia detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.
- [24] P.-J. Chen, M.-C. Lin, M.-J. Lai, J.-C. Lin, H. H.-S. Lu, and V. S. Tseng, "Accurate classification of diminutive colorectal polyps using computer-aided analysis," *Gastroenterology*, vol. 154, no. 3, pp. 568–575, 2018.
- [25] F. Liao, M. Liang, Z. Li, X. Hu, and S. Song, "Evaluate the malignancy of pulmonary nodules using the 3-d deep leaky noisy-or network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3484–3495, 2019.
- [26] P. Yu and X. Yan, "Stock price prediction based on deep neural networks," *Neural Computing & Applications*, vol. 32, no. 6, pp. 1609–1628, 2020.
- [27] J. Fan, J. Zheng, L. Wu, and F. Zhang, "Estimation of daily maize transpiration using support vector machines, extreme gradient boosting, artificial and deep neural networks models," *Agricultural Water Management*, vol. 245, Article ID 106547, 2021.
- [28] W. Bai, Q. Zhou, T. Li, and H. Li, "Adaptive reinforcement learning neural network control for uncertain nonlinear system with input saturation," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3433–3443, 2020.
- [29] W. Waseem, M. Sulaiman, S. Islam et al., "A study of changes in temperature profile of porous fin model using cuckoo search algorithm," *Alexandria Engineering Journal*, vol. 59, no. 1, pp. 11–24, 2020.
- [30] G. Yavuz, B. Durmuş, and D. Aydın, "Artificial bee colony algorithm with distant savants for constrained optimization," *Applied Soft Computing*, vol. 116, Article ID 108343, 2022.
- [31] R.-E. Precup, R.-C. David, R.-C. Roman, E. M. Petriu, and A.-I. Szedlak-Stinean, "Slime mould algorithm-based tuning of cost-effective fuzzy controllers for servo systems," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 1042–1052, 2021.
- [32] M. Canayaz, "Mh-covidnet: diagnosis of covid-19 using deep neural networks and meta-heuristic-based feature selection on x-ray images," *Biomedical Signal Processing and Control*, vol. 64, Article ID 102257, 2021.
- [33] T. R. Gadekallu, N. Khare, S. Bhattacharya, S. Singh, P. K. R. Maddikunta, and G. Srivastava, "Deep neural networks to predict diabetic retinopathy," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2020.
- [34] T. Mahalingam and M. Subramoniam, "Optimal object detection and tracking in occluded video using dnn and gravitational search algorithm," *Soft Computing*, vol. 24, no. 24, pp. 18301–18320, 2020.
- [35] H. Guo, J. Zhou, M. Koopialipoor, D. Jahed Armaghani, and M. M. Tahir, "Deep neural network and whale optimization algorithm to assess flyrock induced by blasting," *Engineering with Computers*, vol. 37, no. 1, pp. 173–186, 2021.
- [36] T. R. Gadekallu, N. Khare, S. Bhattacharya et al., "Early detection of diabetic retinopathy using pca-firefly based deep learning model," *Electronics Now*, vol. 9, no. 2, p. 274, 2020.
- [37] S. Feng, H. Zhou, and H. Dong, "Using deep neural network with small dataset to predict material defects," *Materials & Design*, vol. 162, pp. 300–310, 2019.
- [38] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of skin disease using deep learning neural networks with mobilenet v2 and lstm," *Sensors*, vol. 21, no. 8, p. 2852, 2021.
- [39] C. Zhang and S. Ding, "A stochastic configuration network based on chaotic sparrow search algorithm," *Knowledge-Based Systems*, vol. 220, p. 106924, 2021.
- [40] G. Liu, C. Shu, Z. Liang, B. Peng, and L. Cheng, "A modified sparrow search algorithm with application in 3d route planning for uav," *Sensors*, vol. 21, no. 4, p. 1224, 2021.
- [41] B. Khokhar, S. Dahiya, and K. S. Parmar, "Load frequency control of a microgrid employing a 2d sine logistic map based

- chaotic sine cosine algorithm,” *Applied Soft Computing*, vol. 109, Article ID 107564, 2021.
- [42] G. Kaur and S. Arora, “Chaotic whale optimization algorithm,” *Journal of Computational Design and Engineering*, vol. 5, no. 3, pp. 275–284, 2018.
- [43] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, “Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 824–839, 2018.
- [44] M. E. Fernandez, G. A. Ten Hoor, S. Van Lieshout et al., “Implementation mapping: using intervention mapping to develop implementation strategies,” *Frontiers in Public Health*, vol. 7, p. 158, 2019.
- [45] Q. Xu, K. Sun, C. Cao, and C. Zhu, “A fast image encryption algorithm based on compressive sensing and hyperchaotic map,” *Optics and Lasers in Engineering*, vol. 121, pp. 203–214, 2019.
- [46] H. Bao, Z. Hua, N. Wang, L. Zhu, M. Chen, and B. Bao, “Initials-boosted coexisting chaos in a 2-d sine map and its hardware implementation,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1132–1140, 2021.
- [47] M. Tubishat, S. Ja’afar, M. Alswaiti et al., “Dynamic salp swarm algorithm for feature selection,” *Expert Systems with Applications*, vol. 164, Article ID 113873, 2021.
- [48] D. Yousri, D. Allam, M. Eteiba, and P. N. Suganthan, “Static and dynamic photovoltaic models’ parameters identification using chaotic heterogeneous comprehensive learning particle swarm optimizer variants,” *Energy Conversion and Management*, vol. 182, pp. 546–563, 2019.
- [49] A. V. Tutueva, E. G. Nepomuceno, A. I. Karimov, V. S. Andreev, and D. N. Butusov, “Adaptive chaotic maps and their application to pseudo-random numbers generation,” *Chaos, Solitons & Fractals*, vol. 133, Article ID 109615, 2020.
- [50] A. Bodaghi, “Equalities and inequalities for several variables mappings,” *Journal of Inequalities and Applications*, vol. 2022, no. 1, pp. 6–19, 2022.
- [51] X. Zhang, Y. Xu, C. Yu et al., “Gaussian mutational chaotic fruit fly-built optimization and feature selection,” *Expert Systems with Applications*, vol. 141, Article ID 112976, 2020.
- [52] L. Rodríguez, O. Castillo, M. García, and J. Soria, “A new randomness approach based on sine waves to improve performance in metaheuristic algorithms,” *Soft Computing*, vol. 24, no. 16, pp. 11989–12011, 2020.
- [53] F. Zitouni, S. Harous, and R. Maamri, “A novel quantum firefly algorithm for global optimization,” *Arabian Journal for Science and Engineering*, vol. 46, no. 9, pp. 8741–8759, 2021.
- [54] M. Jamil, X.-S. Yang, and H.-J. Zepernick, “Test Functions for Global Optimization: A Comprehensive Survey,” *Swarm Intelligence and Bio-Inspired Computation*, pp. 193–222, 2013.
- [55] N. Wu, J. Phang, J. Park et al., “Deep neural networks improve radiologists’ performance in breast cancer screening,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 4, pp. 1184–1194, 2020.
- [56] T. C. Hollon, B. Pandian, A. R. Adapa et al., “Near real-time intraoperative brain tumor diagnosis using stimulated Raman histology and deep neural networks,” *Nature Medicine*, vol. 26, no. 1, pp. 52–58, 2020.
- [57] F. Giannakas, C. Troussas, I. Voyiatzis, and C. Sgouropoulou, “A deep learning classification framework for early prediction of team-based academic performance,” *Applied Soft Computing*, vol. 106, Article ID 107355, 2021.
- [58] Q. Chen, W. Hao, and J. He, “A weight initialization based on the linear product structure for neural networks,” *Applied Mathematics and Computation*, vol. 415, Article ID 126722, 2022.
- [59] I. Ahmed, G. Jeon, and F. Piccialli, “A deep-learning-based smart healthcare system for patient’s discomfort detection at the edge of internet of things,” *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10318–10326, 2021.
- [60] A. Darwish, D. Ezzat, and A. E. Hassanien, “An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis,” *Swarm and Evolutionary Computation*, vol. 52, Article ID 100616, 2020.
- [61] H. Moayedi, H. Nguyen, and L. Kok Foong, “Nonlinear evolutionary swarm intelligence of grasshopper optimization algorithm and gray wolf optimization for weight adjustment of neural network,” *Engineering with Computers*, vol. 37, no. 2, pp. 1265–1275, 2021.
- [62] D. Ezzat, A. E. Hassanien, and H. A. Ella, “An optimized deep learning architecture for the diagnosis of covid-19 disease based on gravitational search optimization,” *Applied Soft Computing*, vol. 98, Article ID 106742, 2021.
- [63] P. Stephan, T. Stephan, R. Kannan, and A. Abraham, “A hybrid artificial bee colony with whale optimization algorithm for improved breast cancer diagnosis,” *Neural Computing & Applications*, vol. 33, no. 20, pp. 13667–13691, 2021.
- [64] P. Sudharshan, C. Petitjean, F. Spanhol, L. E. Oliveira, L. Heutte, and P. Honeine, “Multiple instance learning for histopathological breast cancer image classification,” *Expert Systems with Applications*, vol. 117, pp. 103–111, 2019.
- [65] P. N. Srinivasu, G. JayaLakshmi, R. H. Jhaveri, and S. P. Praveen, “Ambient assistive living for monitoring the physical activity of diabetic adults through body area networks,” *Mobile Information Systems*, vol. 2022, pp. 1–18, 2022.
- [66] A. Barakat and P. Bianchi, “Convergence and dynamical behavior of the Adam algorithm for nonconvex stochastic optimization,” *SIAM Journal on Optimization*, vol. 31, no. 1, pp. 244–274, 2021.