Miao Shi, Chao Wang*, Xian-Zhe Li, Ming-Qiang Li, Lu Wang and Neng-Gang Xie*

EEG signal classification based on SVM with improved squirrel search algorithm

<https://doi.org/10.1515/bmt-2020-0038>

Received February 9, 2020; accepted July 30, 2020; published online September 29, 2020

Abstract: Electroencephalography (EEG) is a complex bioelectrical signal. Analysis of which can provide researchers with useful physiological information. In order to recognize and classify EEG signals, a pattern recognition method for optimizing the support vector machine (SVM) by using improved squirrel search algorithm (ISSA) is proposed. The EEG signal is preprocessed, with its time domain features being extracted and directed to the SVM as feature vectors for classification and identification. In this paper, the method of good point set is used to initialize the population position, chaos and reverse learning mechanism are introduced into the algorithm. The performance test of the improved squirrel algorithm (ISSA) is carried out by using the benchmark function. As can be seen from the statistical analysis of the results, the exploration ability and convergence speed of the algorithm are improved. This is then used to optimize SVM parameters. ISSA-SVM model is established and built for classification of EEG signals, compared with other common SVM parameter optimization models. For data sets, the average classification accuracy of this method is 85.9%. This result is an improvement of 2–5% over the comparison method.

Keywords: EEG; parameter optimization; squirrel search algorithm; SVM.

Introduction

Electroencephalography (EEG) signals were first detected by Berger in the human brain in 1929, after which more research has been carried out on brain electrical signals, including brain-computer interface (BCI). A BCI (or brain-machine interface [BMI]) is a device that constructs information channels with external devices by decoding human brain neural activity information [1, 2]. In recent years, faster computer hardware, machine learning, and application of neuroscience have made research in BMI technology become an emerging trend [3]. BCI has been widely applied in disease diagnosis, motion recognition, character spelling, and so on [4, 5]. Soekadar et al. [6] made a study of An EEG/EOG-based hybrid brain-neural computer interaction (BNCI) system. They used the system to help disabled people to control exoskeletons.

BCI system is generally goes through two stages: offline training and online operation. Offline stage refers to the improvement of the model, and the online stage refers to the conversion of subjects' real-time EEG signals into commands to operate external devices [7]. In general, most BCI rely on classification algorithms and signal analysis [8]. Analysis of EEG signals has a certain impact on the subsequent classification effects. EEG signals are usually analyzed from the time domain, the frequency domain, or the time-frequency domain. Gunes et al. [9] combined the time-domain feature extracted from EEG signals with eye movement signals to identify and classify people's sleep state, and finally applied this study result to treat sleep disorder. Because EEG signals are non-stationary signals, researchers have started to propose nonlinear features from EEG signals as feature vectors for classification in recent years. Such as various entropy values of the signal [10]. Sharma et al. [11] used the empirical model decomposition (EMD) method to extract the different entropy of EEG signals as features, classification and recognition of them have obtained very accurate results. Schlögl et al. [12] proposed that mutual information of the feedback might be important for BCI. Researcher can analyze entropy to measure the BCI experiment. In addition to the study of signal analysis methods, many researchers also pay attention to the improvement and innovation of classification algorithms.

***Corresponding authors:** Chao Wang, Department of Mechanical Engineering, Anhui University of Technology, Ma'anshan 243002, Anhui, China, E-mail: cw2013@ahut.edu.cn and Neng-Gang Xie, Department of Management Science and Engineering, Anhui University of Technology, Ma'anshan 243002, Anhui, China, E-mail: xienenggang@aliyun.com

Miao Shi and Lu Wang, Department of Mechanical Engineering, Anhui University of Technology, Ma'anshan, Anhui, China

Xian-Zhe Li, Department of Management Science and Engineering, Anhui University of Technology, Ma'anshan, Anhui, China

Ming-Qiang Li, Department of Electrical and Electronic Engineering, Imperial College, London, UK

The classification algorithms commonly used in BCI systems are divided into linear classifiers and neural networks [13]. Linear method is mainly linear discriminant analysis (LDA) [14], a model of probability density function is generated based on the data of each class, and the new data is classified by this model. LDA is a classification algorithm based on this idea [15]. Hsu et al. [16] proposed an adaptive BCI system model based on adaptive LDA to classify EEG signals. The results show that this method has a good application in adaptive systems. Neural networks [17] and nonlinear SVM [18] are the two main nonlinear methods. Nihal et al. [19] used recurrent neural networks to classify the EEG signals of epileptic patients at onset and normal onset, and to predict the onset time. Another nonlinear method is the support vector machine (SVM), which was proposed by Vapnik et al. [20] in 1995. It is a supervised machine learning algorithm and has broad applications in image recognition, text classification, and BCI [21–26]. In addition, with the availability of big data and the development of graphic processing unit (GPU) in recent years, deep learning has been widely applied to classification problems [27]. Robin et al. [28] proposed to use deep learning with convolutional neural networks (CNN) for solving EEG decoding and visualization. Based on the comparison of different experiments and methods, ConvNets are considered to be an effective method to solve EEG decoding and visualization. Jirayucharoensak et al. [29] used a deep learning network (DLN) to solve the problem of emotion recognition. The method performed better compared with other classification algorithm.

SVM is one of the more mature classification algorithms. It has a good performance in classification problem [30]. Yeo et al. [31] thought it is a good candidate to use SVM to analysis and detection of drowsiness EEG. Compared with other traditional classification algorithms such as k-nearest neighbor (KNN) and multi-layer perceptron (MLP), SVM has lower computational complexity [32, 33]. Although the computational complexity of KNN decreases with the increase of parameter K, its classification accuracy also decreases [34, 32]. In recent years, with the advent of the era of big data, deep learning has become a research hotspot [35]. However, SVM still has certain competitiveness compared with it. The most significant feature of SVM is that it is not a black box method and has a solid mathematical theoretical basis. In addition, SVM has fewer parameters and is easier to adjust than deep learning. But it's worth noting that many researchers in the current study are working to help neural networks get out of the process of adjusting parameters. When the size of data sets is general, the performance of deep learning is not necessarily higher than that of SVM, and the computational efficiency is lower than that of SVM.

Admittedly, the performance advantage of deep learning is obvious when the data scale is very large [36, 37]. Though SVM is a classification algorithm that develops for a long time, there still have insufficiency [38–40]. In SVM, parameters need to be adjusted manually, and different parameters will impact on classification results. Therefore, parameter optimization of SVM has always been the research direction of SVM. SVM has two important parameters, the penalty factor (c) and the kernel function parameters (g). Selection of these two parameters exerts a great impact on the classification effect. An ordinary method is to use the optimization algorithm to optimize the parameters of the SVM which further optimizes the classification effect. Huang et al. [38] proposed a genetic algorithm to optimize SVM parameters. In comparison with the grid optimization method, the classification accuracy of a genetic algorithm is significantly better after using test set tests. Wang et al. [39] used particle swarm optimization (PSO) combined with SVM method to predict the daily minimum temperature and achieved high performance. Xing et al. [40] proposed an emotion-driven retrieval system based on DE-SVM, which has better performance than Back Propagation (BP), Linear Regression (LR), and other methods and proves the feasibility of the method. Wang et al. [41] use ACQIMO to optimize the SVM for pattern recognition of EMG signals. It can be seen that different optimization algorithms will have different SVM parameter optimization results.

Due to different principles and mechanisms of the difference of intelligent optimization algorithms, the result will thereby be a gap in different applications. In practice, selection of optimization algorithms that combine with SVM deserves consideration. Squirrel search algorithm (SSA) [42] is a new optimization algorithm, which has the advantages of fast convergence and high accuracy. SSA is simple and efficient; and the experimental simulation results showed that SSA has better performance than the current popular optimization algorithm [42]. Walaa et al. [43] applied SSA to the Bin packing problem, and proposed an improved method to apply to low-dimensional bin packing problem, which boosted the accuracy and execution time of the solution compared with similar algorithms. Though the SSA is simple and efficient, it still has some shortcomings. In this paper, the SSA is improved, and combined with SVM. Improved squirrel search algorithm (ISSA) is used to optimize the parameters of SVM and increase the classification accuracy of SVM.

The content of this paper is divided into the following four sections. The Section “*Squirrel search algorithm*” introduces the basic SSA. The Section “*Improvement of squirrel search algorithm*” discusses our improvement. And in this section, ISSA and SVM are combined to propose the ISSA-SVM model. The Section “*Simulation*

experiment” is divided into two parts, the first, the ISSA is applied to test functions and compared with common optimization algorithm. Then, after EEG data preprocessing and feature extraction, the ISSA–SVM model was used to identify and classify EEG signals, and compared with similar methods. Finally, classification accuracy and model score were used as measurement indexes. The summary and deficiencies of this research content will be discussed in the Section “Conclusion”.

Squirrel search algorithm

The SSA was proposed by Mohit et al. in 2018. This algorithm simulates the dynamic foraging behavior of the southern squirrel [42]. In the SSA [42], it is assumed that there are only one hickory nut tree, several oak trees, and most normal trees in a forest, with each tree owning a squirrel. The hickory nut tree, oak trees, and normal trees respectively represent the optimal, suboptimal, and general solutions. Squirrels will jump between different trees until they find the best food source, the hickory nut tree, which indicates the best solution. During their food seeking, squirrels alter their positions, which represent update of solutions. Mathematically, an n -dimensional vector represents one squirrel’s position, with each of its values indicating one position variable.

In SSA [42], Eq. (1) is used to generate an initial position with a uniform random distribution. After assessing the fitness of the population location, these locations are ranked in a descending order. According this order, the population is divided into three groups – the best solution, the suboptimal solution and the general solution.

$$FS_i = FS_L + U(0, 1) \times (FS_U - FS_L) \quad (1)$$

In Eq. (1), FS_i refer to the position of the i th squirrel, which is an n -dimensional vector, $U(0, 1)$ is a uniform distribution across $[0, 1]$, FS_U is the upper limit of squirrel search, and FS_L is the lower limit.

Location update formula

In the iterative process, the position updated formula is divided into three types, in the same way as Eqs. (2)–(4). One squirrel on an oak will move in the direction of the hickory nuts tree. Squirrels on ordinary trees are divided into two batches, one for the oak and the other for the hickory nuts tree. In order to improve the global search ability, Mohit added the condition of a hunter appearing [42]. Once this

condition is met, a squirrel will encounter a hunter and will be forced to abandon the current route, and the squirrel will randomly locate to a safe location in the search space.

Case1. Squirrel on the oak tree heading towards the hickory nuts tree.

$$FS_{at}^{t+1} = \begin{cases} FS_{at}^t + d_g \times G_c \times (FS_{ht}^t - FS_{at}^t) & R_1 \geq P_{dp} \\ \text{Random location} & \text{otherwise} \end{cases} \quad (2)$$

Case2. Squirrel on the normal tree heading towards the oak tree.

$$FS_{nt}^{t+1} = \begin{cases} FS_{nt}^t + d_g \times G_c \times (FS_{at}^t - FS_{nt}^t) & R_2 \geq P_{dp} \\ \text{Random location} & \text{otherwise} \end{cases} \quad (3)$$

Case3. Squirrel on the normal tree heading towards the hickory nuts tree.

$$FS_{nt}^{t+1} = \begin{cases} FS_{nt}^t + d_g \times G_c \times (FS_{ht}^t - FS_{nt}^t) & R_3 \geq P_{dp} \\ \text{Random location} & \text{otherwise} \end{cases} \quad (4)$$

In Eqs. (2)–(4), P_{dp} refers to the predator presence probability. R_1 , R_2 , and R_3 are all random numbers within $[0, 1]$. G_c is a gliding constant in a mathematical model, whose value is calculated as 1.9 after rigorous analysis [42]. d_g is the gliding distance of a squirrel that jump between trees. To prevent excessive disturbances from being introduced into the equation, the normal gliding distance is scaled to obtain a perturbation range of d_g across $[0.5, 1.11]$. During each iteration, d_g is a sliding change [42], which greatly improves the local search ability of the algorithm. FS_{ht}^t , FS_{at}^t , FS_{nt}^t are the positions where a flying squirrel reaches the hickory nuts tree, an oak tree and a normal tree respectively, with t being the number of iterations.

Seasonal constant

Since squirrels’ mobility change with the seasons, seasonal monitoring conditions are proposed in the algorithm [42]. These conditions can abandon local optimal solutions in algorithms to a certain extent. First, we have to calculate the seasonal constant according to Eq. (5).

$$S_c^t = \sqrt{\sum_{k=1}^d (FS_{at,k}^t - FS_{ht,k}^t)^2} \quad (t = 1, 2, 3) \quad (5)$$

Next, check the seasonal monitoring condition $S_c^t \geq S_{\min}$. The calculation of $S_{\min}$ is as:

$$S_{\min} = \frac{10E - 6}{(365)^{t/(t_{\max}/2.5)}} \quad (6)$$

When seasonal detection conditions are satisfied, the population of the squirrel will be relocated, and the formula is as follows:

$$FS_{nt}^{\text{new}} = FS_L + Levy(n) \times (FS_U - FS_L) \quad (7)$$

In Eq. (7), $Levy(n)$ can be calculated by the following formula:

$$Levy(x) = 0.01 \times \frac{r_a \times \sigma}{|r_b|^{1/\beta}} \quad (8)$$

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{\beta+1}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{1/\beta} \quad (9)$$

$$\Gamma(x) = (x-1)! \quad (10)$$

where r_a, r_b are two normally distributed random numbers in $[0, 1]$, and β is a constant 1.5 [42].

Improvement of squirrel search algorithm

The SSA is improved from two perspectives. The first one is improving the initial position by using the good point set method. The other is increasing the impact of the optimal position on the population as a whole, rather than just using the global optimal position-guided evolutionary algorithm as the original does.

Good-point set

In most optimization algorithms, the initial population approach is based on a uniform random distribution. It is known that results of each iteration of the optimization algorithm have certain relationship with the previous generation of population. Many researchers use different methods of initial population location to improve the algorithm. The improved algorithm has more accurate calculation results than the original one [44, 45]. And the distribution of the initial population thus impacts on the final result and the number of iterations to some extent [46].

Improving the distribution of initial population can make the algorithm perform better in convergence speed. However, uniformly randomly distributed initial population locations cannot evenly cover the entire solution space. In contrast, good-point sets method can fulfill this requirement.

Assuming that the initial population is n and each population has s dimensions, we can use a good-point set

to initialize the initial population location of the SSA. Select a good point set $P_n(k) = \{x_1^{(n)} \times k, x_2^{(n)} \times k, \dots, x_s^{(n)} \times k, k = 1, 2, \dots, n\}$ to form the initial population space, There are three main methods to generate this set [47].

- (1) Square root sequence method: $r_k = \{\sqrt{p_k}, 1 \leq k \leq s\}$
- (2) Cyclotomic field method: $r_k = \{2\cos(2\pi k/p), 1 \leq k \leq s\}$, p is the smallest prime number meeting $(p-3)/2 \geq s$
- (3) Exponential sequence method: $r_k = \{e^k, 1 \leq k \leq s\}$

Figure 1 respectively shows the random conditions and the set of points generated by the method of dividing the circle. It is clear to see the point set spatial distribution generated by the good-point set is more uniform. In comparison with the good-point set, initial positions resulting from the uniform random distribution are clumping and overlapping. Furthermore, if the popsize n is constant, the distribution of the good-point set is more stable. Hence we conclude that a uniform distribution of initial positions can be generated by applying a good-point set method.

Updated rule based on the global optimal position

- (1) Chaotic perturbation of the optimal point

As a new optimization method, chaos has been verified in many algorithm improvements [48–51]. Combining chaos with PSO, Liu et al. [51] proposed chaotic particle swarm optimization (CPSO) algorithm, which improved the accuracy of calculation results. Wang et al. [50] introduced the idea of chaos into Krill Herd algorithm (KH), and accelerated the convergence rate of KH. According to the result of test function, Chaotic KH algorithm is improved compared with the original algorithm. We also introduce chaotic thought into the SSA, and perform chaotic perturbation on the optimal point after each update. In the original algorithm, the global optimal point does not change itself during each iteration, but only serves as the evolution direction of other points. In the combination one, the global best solution is used to generate a new position by chaotic mapping in each iteration. This new position and the global optimal position are selected by the method of survival of the fittest. The optimal solution gains local search ability after the method is improved, which helps the solution steer clear of the local optimal one. The chaotic map used in chaos optimization is usually logistic mapping, with equation shown below.

$$k = \mu \times k \times (1 - k) \quad (11)$$

where μ is the system constant. When $\mu = 4$, the system is in a chaotic state. For $k \in [0, 1]$, k is generated by Eq. (12) in this

Figure 1: The initial population of two methods in two-dimensional space.

paper, FS_A is the current best position, FS_A^{new} is the new point produced by the chaotic map of the global best point. L_{max} , L_{min} is the upper and lower limits of the search range.

$$k = \frac{FS_A - L_{min}}{L_{max} - L_{min}} \quad (12)$$

$$FS_A^{new} = L_{min} + k \times (L_{max} - L_{min}) \quad (13)$$

(2) Mechanism of elimination

In the SSA, the population is sorted by fitness after each iteration. The last few individuals in the ranking have deviated from the actual goal. The position update of the individuals with lower fitness ranking is slow, which may cause the convergence speed of the algorithm to slow down. In order to reduce the negative influence of poor individuals on the whole population, the elimination mechanism is proposed in the algorithm improvement. Reverse learning is proposed by Tizhooshti [52]. In the process of optimization, the current candidate solution and its reverse individual are considered simultaneously to avoid premature convergence of the algorithm. At present, this method has been applied in Artificial Bee Colony Algorithm (ABC) and PSO, and achieved good results [53, 54].

In the improved squirrel algorithm, the last few individuals in the rank are eliminated unconditionally after each position updates. It also uses the reverse learning based on global best guidance to generate new 10 individuals to fill the gaps, and then step into the next iteration. There is no condition for the replacement in this process. The calculation is shown below:

$$F_{new} = rand \times (F_{old} + FS_A) - F_{old} \quad (14)$$

where F_{new} , F_{old} refer to the newly created individual and the eliminated individual respectively. This method increases the diversity of the population and the impact of

the global optimal position on the new population. This can improve the population diversity and global exploration ability of the algorithm.

Time complexity

The time complexity of ISSA depends on the maximum number of iterations, problem size, and population size. The calculation process is given as Eq. (15)

$$\text{Time} \sim T_{max} \times (2 \times O(N) + N \times O(f)) \quad (15)$$

where T_{max} and N respectively represent the maximum number of iterations and the population size. $2 \times O(N)$ refers to the time complexity generated by population location update and sorting during each iteration, which is associated with the population size. $O(f)$ refers to the time complexity generated by fitness calculation, which is related to the complexity of the problem. It can be observed in the calculation formula that the time complexity of ISSA is mainly depends on the scale of problem when solving complex optimization problems.

Being based on the aforementioned improvements, we construct pseudo code of the improved squirrel algorithm.

ISSA-SVM

SVM is a machine learning algorithm based on statistical learning theory, with minimum structure risk principle. It has strong generalization ability, a solid mathematical theory behind its model and can handle small samples easily. The standard SVM uses linear regression to fit the training set data, assume that the training set is $S = \{(X_i, Y_i) | i = 1, 2, \dots, n\}$, X_i is the input, Y_i is the output, and n is the number of samples. The regression function is:

Improved squirrel search algorithm (ISSA)**Input:** Iterator times: $t_{\max}$; Population: popsize; Dimension: dim;**Output:** best solution

1. Setting parameters, and Initial population location using a good point set approach.
2. Calculate the fitness of the population, and rank the initial population in descending order according to fitness.
3. According to the sorting, the population is divided into three groups, the first one is the best solution, and the second to fourth is considered to be the suboptimal solution, the rest is considered a general solution.
4. Update the squirrel position using Eqs. (2)–(4).
5. Use Eqs. (12) and (13) to generate a new position for the best point chaotic map and make an elite selection.
6. Repeat steps 2, 3.
7. If Meet seasonal conditions.
Update the squirrel position using Eq. (7);
end
8. Eliminate the ranking of dozens of squirrel individuals and use Eq. (14) to generate new individuals.
9. Update the seasonal constant using Eq. (6).
10. Determine whether the number of iterations is met. If the condition is met, end the iteration and output the current best position.
11. End.

The current best position is the final solution

$$f(x) = \omega^T X + b \quad (16)$$

where ω is weight vector, b is bias. ω and b are obtained by solving the optimal problem:

$$\min \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n (\zeta_i + \zeta_i^*) \quad (17)$$

And the above formula needs to satisfy the following conditions:

$$\begin{cases} y_i - \omega^T x_i - b \leq \varepsilon + \zeta_i \\ \omega^T x_i + b - y_i \leq \varepsilon + \zeta_i^* & \zeta_i, \zeta_i^* \geq 0 \end{cases} \quad (18)$$

where C is the penalty factor, ζ_i, ζ_i^* is the relaxation factor, and ε is the insensitivity factor. In order to solve the nonlinear problem, kernel function is introduced into the model. Radial basis kernel function (RBF) is usually used in SVM, RBF is as shown:

$$K = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) = \exp(-g\|x_i - x_j\|^2) \quad (19)$$

The optimal problem is transformed into quadratic programming problem by Lagrange equation, the formula is converted to:

$$\min \frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(x_i, x_j) + \sum_{i=1}^N (\alpha_i - \alpha_i^*) y_i - \sum_{i=1}^N (\alpha_i + \alpha_i^*) \varepsilon \quad (20)$$

$$s.t. \quad \begin{cases} \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i \leq C, 0 \leq \alpha_i^* \leq C \quad i = 1, 2, \dots, N \end{cases} \quad (21)$$

Finally, the SVM regression equation based on kernel function is:

$$f(x) = \sum (\alpha_i - \alpha_i^*) K(x_i, x) + b \quad (22)$$

Penalty factor (c) and kernel function parameters (g) are two important parameters in SVM. The selection of these two parameters has a great influence on classification results. In this paper, the ISSA is combined with SVM, and is used to optimize c and g to improve the classification accuracy. The BCI2003 competition's EEG data set was used to test the model, which is shown in Figure 2.

Simulation experiment

In this section, two simulation experiments are introduced. First, several test functions were used to test the performance of the improved squirrel algorithm. In the second simulation experiment, the ISSA-SVM model was used to classify EEG signals. In the comparison of optimization algorithms, we choose the standard PSO [55], Grasshopper optimisation algorithm (GOA) [56], Ions motion algorithm (IMO) [57] and SSA. In order to ensure the fairness of the comparison, we calculate the results under the same circumstance: The computer is configured as a Windows 7 system, the CPU is Inter (R) Xeon (R), and the memory is 32 GB. The common parameter of the optimization function is the maximum number of evaluations of $1000 \times \text{dim}$, where the population size is 50. Each optimization function runs 30 times independently.

Figure 2: ISSA-SVM training and prediction process.

Performance test of improved squirrel search algorithm

Four single-peaks, four multi-peak and four complex multi-peak test functions in fixed dimensions [42, 56, 58] are used to evaluate the algorithm performance. These functions are shown in Table 1. We used four indicators – the best value, the worst value, the average value and the variance in results statistics, with these results being demonstrated in Tables 2–4.

From Tables 2–3 and Figures 3 and 4, it demonstrates that the improved algorithm has, in general, lower values of those four indicators in the multi-peak test function and single-peak test function. From the point of view of convergence speed and accuracy, it is superior to the other four algorithms.

In the multi-peak complex test function of fixed dimensions, the performance of the improved algorithm does not perform as well as it does in the single-peak test function and multi-peak test function. In TF10, all the algorithms

Table 1: Testing function.

Expression	d	Range	$F_{\min}$
$TF1(x) = \sum_{j=1}^n x_j^2$	10	$[-100, 100]$	0
$TF2(x) = \sum_{j=1}^n x_j + \prod_{j=1}^n x_j $	10	$[-10, 10]$	0
$TF3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	10	$[-100, 100]$	0
$TF4(x) = \sum_{j=1}^n (x_j + 0.5)^2$	10	$[-100, 100]$	0
$TF5(x) = \sum_{j=1}^n (x_j^2 - 10 \cos(2\pi x_j)) + 10$	10	$[-5.12, 5.12]$	0
$TF6(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{j=1}^n (x_j^2)}\right) - \exp\left(\frac{1}{d} \sum_{j=1}^n \cos(2\pi x_j)\right) + 20 + e$	10	$[-32, 32]$	0
$TF7(x) = \frac{1}{4000} \sum_{j=1}^n x_j^2 - \prod_{j=1}^n \cos\left(\frac{x_j}{\sqrt{j}}\right) + 1$	10	$[-600, 600]$	0
$TF8(x) = \sum_{j=1}^n -x_j \sin(\sqrt{ x_j })$	10	$[-500, 500]$	$-418.9829 \times d$
$TF9(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} - 6\right)^2 + 10 \left(1 - \frac{1}{8\pi}\right) \cos x_1 + 10$	2	$[-500, 500]$	0.398
$TF10(x) = -\sum_{j=1}^n \sin(x_j) \sin(jx_j^2/\pi)^{20}$	2	$[0, \pi]$	-1.8013
$TF11(x) = -\sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	$[-300, 300]$	0.0003
$TF12(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 100]$	-10.1532

Table 2: Single peak test function results.

Function		GOA	PSO	IMO	SSA	This paper
TF1	Best	5.0724e – 05	3.3130e + 00	6.3887e – 05	4.1101e – 08	<i>5.9537e – 10</i>
	Worst	7.4609e – 03	2.0971e + 01	1.1544e – 03	3.5682e – 06	<i>1.3537e – 07</i>
	Mean	2.0919e – 03	8.7996e + 00	4.5853e – 04	6.8653e – 07	<i>1.7209e – 08</i>
	SD	1.6905e – 03	4.9530e + 00	2.9544e – 04	7.9163e – 06	<i>3.1186e – 08</i>
TF2	Best	4.2001e – 04	3.4922e – 01	9.5966e – 03	5.6975e – 04	<i>8.9418e – 05</i>
	Worst	2.0000e + 01	1.5693e + 00	2.6210e – 02	1.0251e + 01	<i>4.2769e – 02</i>
	Mean	6.6982e – 01	7.8215e – 01	1.7848e – 02	1.3806e + 00	<i>5.6139e – 03</i>
	SD	3.6510e + 00	3.2601e – 01	<i>3.4250e – 03</i>	2.4102e + 00	<i>8.5332e – 03</i>
TF3	Best	1.0341e – 02	1.3183e + 01	3.1679e – 02	1.8195e – 03	<i>3.4348e – 06</i>
	Worst	1.2791e + 00	1.3751e + 02	2.0811e + 00	1.5462e + 00	<i>4.4185e – 04</i>
	Mean	2.6101e – 01	5.3119e + 01	5.1377e – 01	1.1138e – 01	<i>7.6426e – 05</i>
	SD	2.9814e – 01	3.0215e + 01	4.6803e – 01	2.9688e – 01	<i>9.9035e – 05</i>
TF4	Best	4.9451e – 06	2.1389e + 00	4.4356e – 03	2.5764e – 08	<i>4.9923e – 10</i>
	Worst	1.1604e – 02	2.0164e + 01	1.1070e – 01	4.4456e – 06	<i>1.8814e – 07</i>
	Mean	2.2532e – 03	9.2157e + 00	3.5847e – 02	9.0420e – 07	<i>3.6023e – 08</i>
	SD	2.1782e – 03	5.3202e + 00	2.4869e – 02	1.1588e – 06	<i>4.2290e – 08</i>

GOA, Grasshopper optimisation algorithm; PSO, Particle swarm optimization; IMO, Ions motion algorithm; SSA, Squirrel search algorithm. The italics indicated that the value is optimal in the same row.

Table 3: Multi-peak test function results.

Function		GOA	PSO	IMO	SSA	This paper
TF5	Best	8.9546e + 00	8.2987e + 00	1.4637e – 02	1.4924e + 01	<i>5.3834e – 10</i>
	Worst	7.3697e + 01	3.8418e + 01	<i>1.9573e + 01</i>	7.0860e + 01	2.4874e + 01
	Mean	3.4001e + 01	1.7042e + 01	3.5305e + 00	3.4167e + 01	<i>2.8908e + 00</i>
	SD	1.5920e + 01	5.9308e + 00	<i>4.2369e + 00</i>	1.5958e + 01	7.4823e + 00
TF6	Best	4.7938e – 04	1.1455e + 00	8.2213e – 03	2.3616e – 04	<i>4.0877e – 06</i>
	Worst	2.5799e + 00	3.5059e + 00	2.2005e – 02	1.8086e + 01	<i>1.3476e – 04</i>
	Mean	5.6858e – 01	2.7488e + 00	1.3378e – 02	3.3349e + 00	<i>4.6615e – 05</i>
	SD	8.7983e – 01	5.6824e – 01	3.4030e – 03	3.0003e + 00	<i>3.1622e – 05</i>
TF7	Best	9.8499e – 02	8.3340e – 01	8.2347e – 03	4.4367e – 02	<i>1.3692e – 08</i>
	Worst	9.0752e – 01	1.2788e + 00	<i>3.8258e – 01</i>	1.2652e + 00	8.0679e – 01
	Mean	2.8196e – 01	1.0646e + 00	<i>1.5308e – 01</i>	4.3256e – 01	1.6201e – 01
	SD	1.7873e – 01	1.1029e – 01	<i>8.6403e – 02</i>	3.4445e – 01	1.6405e – 01
TF8	Best	7.5012e + 02	<i>4.6437e + 02</i>	1.9696e + 03	6.9280e + 02	5.5272e + 02
	Worst	2.0150e + 03	<i>1.6579e + 03</i>	2.6980e + 03	2.0269e + 03	2.2441e + 03
	Mean	1.3699e + 03	1.1560e + 03	2.4525e + 03	1.4234e + 03	<i>1.0969e + 03</i>
	SD	3.3383e + 02	3.4024e + 02	<i>1.7052e + 02</i>	3.2757e + 02	4.0998e + 02

GOA, Grasshopper optimisation algorithm; PSO, Particle swarm optimization; IMO, Ions motion algorithm; SSA, Squirrel search algorithm. The italics indicated that the value is optimal in the same row.

converge to the optimal value, but as can be seen from Figure 5B, the proposed algorithm has the high convergence speed. In TF11, though the improvement proposed is better than the original algorithm, the performance of PSO is the best in this test function. In general, the improved algorithm has certain competitiveness in complex test functions.

Classification and recognition of EEG signals

In this section, the improved algorithm is applied to the parameter optimization of the SVM, and the BCI

competition 2003 data set III is used to verify its effect of the classification. Results calculated from the experiment are compared with other optimization algorithms to demonstrate the relative effectiveness of this algorithm. In fact, the BCI competition 2003 data set III is a commonly used data set. Bashar et al. [59] decomposed the signal by means of multivariate empirical mode decomposition (MEMD). They selected components that differed significantly from category to category and treated them with the short-time Fourier transform (STFT). The features extracted by this signal processing method had good separability. Lee et al. [60] converted

Table 4: Results of complex test functions in fixed dimensions.

Function		GOA	PSO	IMO	SSA	This paper
TF9	Best	$3.9788e-01$	$4.4070e-01$	$3.9788e-01$	$3.9788e-01$	$3.9788e-01$
	Worst	$4.1685e-01$	$4.0823e+00$	$4.0292e-01$	$1.6568e+01$	$3.9788e-01$
	Mean	$3.9960e-01$	$8.4448e-01$	$3.9820e-01$	$9.3692e-01$	$3.9788e-01$
	SD	$3.7311e-03$	$6.9618e-01$	$9.3018e-04$	$2.9524e+00$	$4.3943e-15$
TF10	Best	$-1.8013e+00$	$-1.8013e+00$	$-1.8013e+00$	$-1.8013e+00$	$-1.8013e+00$
	Worst	$-1.0000e+00$	$-1.8012e+00$	$-1.0000e+00$	$-1.8013e+00$	$-1.8013e+00$
	Mean	$-1.6357e+00$	$-1.8013e+00$	$-1.5653e+00$	$-1.8013e+00$	$-1.8013e+00$
	SD	$3.0964e-01$	$3.3692e-05$	$3.1722e-01$	$8.3388e-16$	$1.1269e-15$
TF11	Best	$1.0211e-03$	$1.0314e-03$	$1.0240e-03$	$1.0222e-03$	$1.0296e-03$
	Worst	$2.5782e-02$	$4.9366e-03$	$2.9617e-02$	$4.3901e-02$	$1.8129e-02$
	Mean	$3.3330e-03$	$1.5854e-03$	$3.0577e-03$	$6.1175e-03$	$4.0529e-03$
	SD	$5.0156e-03$	$8.0453e-04$	$5.3281e-03$	$1.1275e-02$	$3.6472e-03$
TF12	Best	$-1.0153e+01$	$-9.6054e-01$	$-7.8365e-02$	$-1.0153e+01$	$-1.0153e+01$
	Worst	$-8.8245e-01$	$-3.5065e-01$	$-1.9853e-03$	$-5.0546e+00$	$-1.0153e+01$
	Mean	$-4.9699e+00$	$-6.4185e-01$	$-8.3491e-03$	$-5.2251e+00$	$-1.0153e+01$
	SD	$2.7930e+00$	$2.2453e-01$	$1.4152e-02$	$9.3076e-01$	$4.8243e-07$

GOA, Grasshopper optimisation algorithm; PSO, Particle swarm optimization; IMO, Ions motion algorithm; SSA, Squirrel search algorithm. The *italics* indicated that the value is optimal in the same row.

Figure 3: Single peak test function iteration curve.

EEG signals into images using continuous wavelet transform (CWT). These images are classified and identified as inputs to the CNN. The idea of this method is very interesting and worthy of reference. These papers pre-processed the signals by using the current mainstream signal processing methods, which improved the classification accuracy of the signals.

The dataset and preprocessing

(1) Description of dataset

The data set records a 25-year-old healthy female subject who sat casually on a chair and controlled a feedback bar by imagining the left and right hand movements.

Figure 4: Multi-peak test function iteration curve.

Figure 5: Complex test functions in fixed dimensions.

The hints for the left and right hands are random. Single experiment lasts for 9 s, with the subject remaining motionless within the first 2 s. Sound of the prompt sounds at the second second, notifying the subject to prepare for the action. At the third second, a prompt appears on a screen which requires the subject to imagine a left hand

movement or a right hand movement. Three bipolar EEG channels are measured over C3, C4, and Cz, and are sampled at 128 Hz. The experimental process is shown in Figure 6. Seven experiments have been performed in total, with 40 samples each for a length of 9 s [61, 62]. Data sets are derived from <http://bbci.de/competition/ii/>.

Figure 6: Experimental process.

(2) Preprocessing of the dataset

When a certain area of the cerebral cortex is activated, the electrophysiological phenomenon of signal energy degradation in some frequency bands is called event-related-desynchronization (ERD). When a certain area of the brain is in a quiet state, the energy of certain frequency bands rises, which is known as event-related synchronization (ERS) [63]. Once people imagine the movement of the hand, the μ rhythm and β wave of the C3/C4 channel in the sensory motion zone of the human brain have obvious ERD/ERS phenomenon [64]. The μ rhythm and β wave mainly concentrated within 8–30 Hz, hence we have to filter the data first. We use an elliptical filter. The passband frequency of the elliptical filter matches these waves' energy band, with its passband ripples being less than 1 db and signal attenuation in the range of 5 Hz on both sides of the passband being 40 db.

After filtering the signal data, we need to segment the data [65]. The experimental time was in total 9 s, but the ERD/ERS phenomenon did not persist throughout the experiment. If we calculate the energy characteristics of complete length data, the characteristics of time segments unrelated to events will inevitably affect the classification results. Therefore, we use a time window to divide data into different periods, and then calculate the energy difference of the same time window when the subject imagines hands movements. The formula are Eqs. (23) and (24). In Eq. (23), k represents the k th time window and the value of 128 is the signal sampling frequency.

$$E_{c3} = \sum_{i=1+128(k-1)}^{128k} |x_i|^2 \quad (23)$$

$$E = |E_{c3} - E_{c4}| \quad (24)$$

Since the motion starts after the third second, we calculate the energy difference between the C3 and C4 channels after this point. As shown in Figure 7, the energy difference of the channel starts to rise after the third second, peaking at the seventh second, and then begins to fall. This means the ERD/ERS phenomenon appears after the third second and begins to decline after the seventh second respectively. This leads to treating period 4–7 s as the feature time period.

(3) Feature extraction and classification

Time domain features have always been effective indicators of EEG signal classification [66]. After the data is preprocessed by time domain analysis, two time domain features are extracted as eigenvalues, which are integral EEG (IEEG) and root mean square (RMS) values with formulas shown below:

$$\text{IEEG} = \frac{1}{N} \sum_{i=1}^N |X_i| \quad (25)$$

$$\text{RMS} = \sqrt{\frac{\sum_{i=1}^N X_i^2}{N-1}} \quad (26)$$

The EEG signals are collected from the three channels of C3, C4, and Cz. After feature extraction, each single sample is integrated into a six-dimensional vector as a feature vector. The improved squirrel algorithm is used to optimize the SVM parameters and improve the classification accuracy. In this paper, SVM uses a RBF. In order to eliminate the exceptions of the results, we repeat the program 30 times to compare the final classification accuracy in different optimization algorithms.

Analysis of classification results of EEG signals

EEG data set can be divided into training set and test set, the training set accounts for 70% and the other accounts for

Figure 7: The energy different of the hand-imagined motion characteristics in different.

Figure 8: Iteration process.

30% of the total set. The training set is used to optimize the parameters, and the optimal one is to construct the SVM model. The training set and the test set are classified, with the corresponding result being drawn into a confusion matrix, as shown in Tables 5–9. The confusion matrix obtained in 30 operations is normalized and averaged.

As shown in Figure 8A, in the process of optimization, other algorithms converge quickly, but the improved squirrel algorithm constantly rejects the local optimal solution and converges to the best value. To verify its stability, we counted the results of 30 runs into box plot as shown in Figure 8B. We can see that the results of the method proposed in this paper are the most stable.

The optimal parameters found in the train set are put into the classifier model which is utilised to classify and identify the test set. All 30 results are counted into one confusion matrix, during which the confusion matrix is normalized and averaged whenever the a new result is counted. To evaluate the quality of several models, five evaluation indexes are introduced: accuracy, precision, sensitivity (recall), specificity, and F1-Score. F1-Score is a common indicator that distinguishes the quality of the classifier. It is the weighted average of precision and sensitivity (recall). The closer the value of F1-Score is to 1 with the better the performance of the classifier (see Table 10).

It is clear to observe from the confusion matrix of train set and test set that IMO-SVM and GOA-SVM models have higher classification accuracy on train set, but not on the test set, thereby demonstrating limitations of the generalization of the two classifications. This limitation can

Table 5: ISSA-SVM confusion matrix.

(a) Train set			(b) Test set		
	True left	True right		True left	True right
Predict left	0.489	0.066	Predict left	0.397	0.074
Predict right	0.023	0.422	Predict right	0.067	0.462

Table 6: SSA-SVM confusion matrix.

(a) Train set			(b) Test set		
	True left	True right		True left	True right
Predict left	0.45	0.087	Predict left	0.421	0.122
Predict right	0.062	0.401	Predict right	0.044	0.414

Table 7: PSO-SVM confusion matrix.

(a) Train set			(b) Test set		
	True left	True right		True left	True right
Predict left	0.44	0.095	Predict left	0.429	0.126
Predict right	0.071	0.393	Predict right	0.036	0.409

Table 8: IMO-SVM confusion matrix.

(a) Train set			(b) Test set		
	True left	True right		True left	True right
Predict left	0.512	0.02	Predict left	0.4	0.281
Predict right	0.0	0.489	Predict right	0.064	0.255

Table 9: GOA-SVM confusion matrix.

(a) Train set			(b) Test set		
	True left	True right		True left	True right
Predict left	0.496	0.028	Predict left	0.377	0.138
Predict right	0.016	0.46	Predict right	0.088	0.398

Table 10: Test set evaluation index.

	GOA-SVM	PSO-SVM	IMO-SVM	SSA-SVM	This paper
Accuracy	77.5%	83.8%	65.5%	83.4%	85.9%
Precision	77.3%	77.3%	60.9%	77.8%	84.7%
Sensitivity	81.1%	92.3%	86.1%	90.6%	85.6%
Specificity	74.3%	76.4%	47.6%	77.3%	86.2%
F1 score	77.0%	84.1%	71.4%	83.7%	85.1%

GOA-SVM, Grasshopper optimisation algorithm-support vector machine; PSO-SVM, Particle swarm optimization-support vector machine; IMO-SVM, Ions motion algorithm-support vector machine; SSA-SVM, Squirrel search algorithm-support vector machine.

further be seen from Figure 9. Except for these two models, the results calculated by the other three methods in the train set and the test set are relatively close. The ISSA-SVM proposed in this paper, however, has comparative advantages in both the training set and the test set. From the F1-score, the method in this paper scores 85.1%, which is the highest score compared with the other four methods.

For this data set, Brodu et al. [67] obtained classification accuracy of 82.1% for this data set in the study of power extraction technology. In [68], they used multifractal cumulations and predictive complexity to analyze the EEG and obtained an accuracy of 80.7% on the dataset. The average results of this paper show 4–5% improvement over other articles that use the same data set. It's worth noting that in [60], Lee et al. converted the CWT to convert EEG into 93×32 feature map, and then combined with the advantages of CNN in image processing to obtain a high classification accuracy (92.9%). The method in [60] is enlightening, but the time complexity of the model is higher. From the point of view of characteristic engineering, this paper extracts time domain feature of EEG signals, whose time complexity is $O(1)$. The time complexity of CWT is usually $O(s \log s)$, where s represents the signal length. Analyze from the classification model, the time complexity of CNN is shown in Eq. (27).

$$\text{Time} \sim O\left(\sum_l^D M_l^2 \times K_l^2 \times C_{l-1} \times C_l\right) \quad (27)$$

where D and l represent the total number of convolutional layers and the convolutional layer index respectively. M and K represent the length of the side of feature map and convolution kernel. Eq. (27) only includes the computational complexity of convolutional layer, excluding pooling layer and full connection layer [69]. In addition, Eq. (27) represents the time complexity of a single sample in CNN, and the time complexity of the training stage is three times that of the testing stage [69]. As shown in Figure 9, the classification accuracy of ISSA-SVM in training set and test set reached the highest of 100 and 89.5% respectively. The time complexity of ISSA-SVM is mainly determined by the complexity of SVM, which is $O(P \times \dim)$. P is the number of samples and $\dim$ is the number of features [70]. When the sample size is small, the time complexity of the classification model proposed in this paper is lower. To summarize, we have demonstrated that the proposed method is feasible in EEG signal pattern recognition.

Conclusion

In this paper, the SSA is improved. The distribution of initial population was improved by using the method of good-point set. The idea of chaos and the mechanism of reverse learning are introduced into the SSA, which improves the exploration ability and convergence speed of the algorithm. ISSA was tested on 12 commonly used benchmark functions. As can be seen from the tables and figures of result statistics, ISSA has better performance than some commonly used optimization algorithms. We use the ISSA to optimize two important parameters of SVM and construct an ISSA-SVM model to identify and classify EEG signals. Finally, the classification accuracy of this method is 85.9% and F1-score is 85.1%. By comparing the results of combining SVM with other optimization algorithms, the method has better performance on F1-score and accuracy. Compared with the results of other references using the same data set, the classification accuracy obtained in this paper is also very competitive.

Figure 9: Statistical results of classification results for test sets and training sets.

In spite of this, the SSAISSA is not tested on some complex test functions. In addition, ISSA-SVM is mainly used for the binary classification of EEG signals. We hope to use this model for multi-classification recognition of signals in the future. The research shows that the signal pretreatment process has obvious influence on the classification results. Because this paper mainly focuses on the performance of the classified model, the method of signal processing is weak. We will pay more attention to feature extraction and feature selection in the following research. At the same time, the model proposed to this paper is combined to further to improve the accuracy of classification.

Acknowledgment: This work was supported by the Science and Technology Major Project of Anhui Province, China (Grant No. 17030 901037), Scientific Research Foundation of Education Department of Anhui Province, China (Grant No. KJ2019ZD 09 and KJ2019A0091), the Ministry of Education, Humanities and Social Sciences research projects (Grant NO. 19YJAZH098). The authors would like to thank all the anonymous referees for their valuable comments and suggestions to further improve the quality of this work.

Research funding: This work was supported by the Science and Technology Major Project of Anhui Province, China (Grant No. 17030901037), Scientific Research Foundation of Education Department of Anhui Province, China (Grant No. KJ2019ZD09 and KJ2019A0091), the Ministry of Education, Humanities and Social Sciences research projects (Grant No. 19YJAZH098).

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

Competing interests: No conflict of interest exists in the submission of this manuscript, and manuscript is approved by all authors for publication. We stated that the work described has not been published previously, and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

References

- Gao S, Wang Y, Gao X, Hong B. Visual and auditory brain–computer interfaces. *IEEE Trans Biomed Eng* 2014;61:1436–47.
- Lebedev MA, Nicolelis MA. Brain–machine interfaces: past, present and future. *Trends Neurosci* 2006;29:536–46.
- McFarland DJ, Wolpaw JR. EEG-based brain–computer interfaces. *Curr Opin Biomed Eng* 2017;4:194–200.
- Aya R, Mihaly B, Piotr S, Felix G, Abdul S, Ivan V. Brain–computer interface spellers: a review. *Brain Sci* 2018;8:57.
- Abdulkader SN, Atia A, Mostafa MSM. Brain computer interfacing: applications and challenges. *Egypt Inf J* 2015;16: 213–30.
- Soekadar SR, Witkowski M, Vitiello N, Birbaumer N. An EEG/ EOG-based hybrid brain–neural computer interaction (BNCI) system to control an exoskeleton for the paralyzed hand. *Biomed Eng–Biomed Tech* 2015;60:199–205.
- Lotte F, Bougrain L, Cichocki A, Clerc M, Congedo M, Rakotomamonjy A, et al. A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *J Neural Eng* 2018;15: 031005. 1-031005.28.
- Lotte F, Congedo M, Lécuyer A, Lamarche F, Arnaldi B. A review of classification algorithms for EEG-based brain–computer interfaces. *J Neural Eng* 2017;4:R1.
- Güneş S, Polat K, Dursun M, Yosunkaya S. Examining the relevance with sleep stages of time domain features of EEG, EOG, and chin EMG signals. In: 2009 14th national biomedical engineering meeting. Turkey: Izmir; 2009:1–4 pp.
- Acharya UR, Hagiwara Y, Deshpande SN, Suren S, Koh JEW, Oh SL, et al. Characterization of focal EEG signals: a review. *Future Generat Comput Syst* 2019;91:290–9.
- Sharma R, Pachori RB, Acharya UR. Application of entropy measures on intrinsic mode functions for the automated identification of focal electroencephalogram signals. *Entropy* 2015;17:669–91.
- Schlögl A, Neuper C, Pfurtscheller G. Estimating the mutual information of an EEG-based brain–computer interface. *Paralyzed hand. Biomed Eng–Biomed Tech* 2002;47:3–8.
- Xiao D, Mu Z, Hu J. A linear discrimination method used in motor imagery EEG classification. In: 2009 fifth international conference on natural computation. TianJian: IEEE; 2009, vol 2: 94–8 pp.
- Hauk O, Davis MH, Ford M, Pulvermüller F, Marslen-Wilson WD. The time course of visual word recognition as revealed by linear regression analysis of ERP data. *Neuroimage* 2006;30: 1383–400.
- Garrett D, Peterson DA, Anderson CW, Thaut MH. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *IEEE Trans Neural Syst Rehabil Eng* 2003;11: 141–4.
- Hsu WY. EEG-based motor imagery classification using enhanced active segment selection and adaptive classifier. *Comput Biol Med* 2011;41:633–39.
- Titterton D, McCheng B. Neural networks: a review from a statistical perspective. *Stat Sci* 1994;9:2–30.
- Übeyli ED. Analysis of EEG signals by combining eigenvector methods and multiclass support vector machines. *Comput Biol Med* 2008;38:14–22.
- Güler NF, Übeyli ED, Güler I. Recurrent neural networks employing lyapunov exponents for EEG signals classification. *Expert Syst Appl* 2005;29:506–14.
- Cortes C, Vapnik V. Support-vector networks. *Mach Learn* 1995; 20:273–97.
- Niu XX, Suen CY. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recogn* 2012;45:1318–25.
- Manevitz LM, Yousef M. One-class SVMs for document classification. *J Mach Learn Res* 2001;2:139–54.
- Tarabalka Y, Fauvel M, Chanussot J, Benediktsson JA. SVM-and MRF-based method for accurate classification of hyperspectral images. *Geosci Rem Sens Lett IEEE* 2010;7:736–40.

24. Antunes M, Silva C, Ribeiro B, Correia M. A hybrid AIS–SVM ensemble approach for text classification. In: International conference on adaptive and natural computing algorithms. Berlin, Heidelberg: Springer; 2011:342–52 pp. https://doi.org/10.1007/978-3-642-20267-4_36.
25. Garcia GN, Ebrahimi T, Vesin JM. Support vector EEG classification in the Fourier and time-frequency correlation domains. In: First international IEEE EMBS conference on neural engineering, 2003. Conference Proceedings. Capri Island, Italy, Italy: IEEE; 2003:591–94 pp.
26. Bajaj V, Pachori RB. Classification of seizure and non-seizure EEG signals using empirical mode decomposition. *IEEE Trans Inf Technol Biomed* 2012;16:1135–42.
27. Craik A, He Y, Contreras-Vidal JL. Deep learning for electroencephalogram (EEG) classification tasks: a review. *J Neural Eng* 2019;16: 031001.
28. Schirrmester RT, Springenberg JT, Fiederer LDJ, Glasstetter M, Eggensperger K, Tangermann M, et al. Deep learning with convolutional neural networks for EEG decoding and visualization. *Hum Brain Mapp* 2017;38:5391–420.
29. Jirayucharoensak S, Pan-Ngum S, Israsena P. EEG-based emotion recognition using deep learning network with principal component based covariate shift adaptation. *Sci World J* 2014; 2014. <https://doi.org/10.1155/2014/627892>.
30. Yan W, Shao H. Application of support vector machine nonlinear classifier to fault diagnoses. In: Proceedings of the 4th world congress on intelligent control and automation (Cat. No. 02EX527), vol. 4 IEEE; 2002:2697–700 pp.
31. Yeo MV, Li X, Shen K, Wilder-Smith EP. Can SVM be used for automatic EEG detection of drowsiness during car driving?. *Saf Sci* 2009;47:115–24.
32. Palaniappan R, Sundaraj K, Sundaraj S. A comparative study of the SVM and K-NN machine learning algorithms for the diagnosis of respiratory pathologies using pulmonary acoustic signals. *BMC Bioinf* 2014;15:223.
33. Osowski S, Siwek K, Markiewicz T. MLP and SVM networks – a comparative study. In: Proceedings of the 6th nordic signal processing symposium, 2004. NORSIG 2004. Espoo, Finland, Finland: IEEE; 2004:37–40 pp.
34. Beyer KS, Goldstein J, Ramakrishnan R, Shaft U. When is “nearest neighbor” meaningful? In: International conference on database theory. Berlin, Heidelberg: Springer; 1999.
35. Schmidhuber J. Deep learning in neural networks: an overview. *Neural Network* 2015;61:85–117.
36. Liu P, Choo KKR, Wang L, Huang F. SVM or deep learning? A comparative study on remote sensing image classification. *Soft Comput* 2017;21:7053–65.
37. Liu T, Abd-Elrahman A, Morton J, Wilhelm VL. Comparing fully convolutional networks, random forest, support vector machine, and patch-based deep convolutional neural networks for object-based wetland mapping using images from small unmanned aircraft system. *GIScience Remote Sens* 2018;55: 243–64.
38. Huang CL, Wang CJ. A GA-based feature selection and parameters optimization for support vector machines. *Expert Syst Appl* 2006; 31:231–40.
39. Wang G, Qiu Y, Li H. Temperature forecast based on SVM optimized by PSO algorithm In: 2010 international conference on intelligent computing and cognitive informatics. Kuala Lumpur, Malaysia: IEEE; 2010:259–62 pp.
40. Xing B, Zhang K, Sun S, Zhang L, Gao Z, Wang J, et al. Emotion-driven Chinese folk music–image retrieval based on DE–SVM. *Neurocomputing* 2015;148:619–27.
41. Wang BZ, Wang C, Wang L, Xie NG, Wei W. Recognition of semg hand actions based on cloud adaptive quantum chaos ions motion algorithm optimized SVM. *J Mech Med Biol* 2019;19: 1950047.
42. Jain M, Singh V, Rani A. A novel nature-inspired algorithm for optimization: squirrel search algorithm. *Swarm Evol Comput* 2019;44:148–75.
43. El-Ashmawi WH, Elminaam DSA. A modified squirrel search algorithm based on improved best fit heuristic and operator strategy for bin packing problem. *Appl Soft Comput* 2019;82: 105565.
44. Gao W, Liu S, Huang L. A global best artificial bee colony algorithm for global optimization. *J Comput Appl Math* 2012;236: 2741–53.
45. Ali M, Siarry P, Pant M. An efficient differential evolution based algorithm for solving multi-objective optimization problems. *Eur J Oper Res* 2012;217:404–16.
46. Maaranen H, Miettinen K, Mäkelä MM. Quasi-random initial population for genetic algorithms. *Comput Math Appl* 2004;47: 1885–95.
47. Li Y-P, Ni Z-W, Jin F-F, Li J-M, Li F-G. Research on clustering method of improved glowworm algorithm based on good-point set. *Math Probl Eng* 2018. <https://doi.org/10.1155/2018/8724084>.
48. Sayed GI, Khoriba G, Haggag MH. A novel chaotic salp swarm algorithm for global optimization and feature selection. *Appl Intell* 2018;48:3462–81.
49. Ibrahim RA, Elaziz MA, Lu S. Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Syst Appl* 2018;108:1–27.
50. Wang GG, Guo L, Gandomi AH, Hao GS, Wang H. Chaotic krill herd algorithm. *Inf Sci* 2014;274:17–34.
51. Liu B, Wang L, Jin YH, Tang F, Huang DX. Improved particle swarm optimization combined with chaos. *Chaos, Solit Fractals* 2005; 25:1261–71.
52. Tizhoosh HR. Opposition-based learning: a new scheme for machine intelligence. In: International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC’06). Vienna, Austria: IEEE; 2005, vol 1:695–701 pp.
53. Xiang WL, Meng XL, Li YZ, He RC, An MQ. An improved artificial bee colony algorithm based on the gravity model. *Inf Sci* 2018; 429:49–71.
54. Dong H-b, Zhang H, Han S, Li X-h, Wang X-w. Reverse-learning particle swarm optimization algorithm based on niching technology. In: 2018 IEEE/ACIS 17th international conference on computer and information science (ICIS). Singapore: IEEE; 2018:405–10 pp.
55. Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN’95-international conference on neural networks. Perth, WA, Australia: IEEE; 1995, vol 4:1942–48 pp.
56. Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: theory and application. *Adv Eng Software* 2017;105: 30–47.
57. Javidy B, Hatamlou A, Mirjalili S. Ions motion algorithm for solving optimization problems. *Appl Soft Comput* 2015;32:72–9.

58. Mirjalili S, Lewis A. The whale optimization algorithm. *Adv Eng Software* 2016;95:51–67.
59. Bashar SK, Bhuiyan MIH. Classification of motor imagery movements using multivariate empirical mode decomposition and short time fourier transform based hybrid method. *Eng Sci Technol Int J* 2016;19:1457–64.
60. Lee HK, Choi YS. Application of continuous wavelet transform and convolutional neural network in decoding motor imagery brain–computer interface. *Entropy* 2019;21:1199.
61. Blankertz B, Muller KR, Curio G, Vaughan TM, Schalk G, Wolpaw JR, et al. The BCI competition 2003: progress and perspectives in detection and discrimination of EEG single trials. *IEEE Trans Biomed Eng* 2004;51:1044.
62. Lemm S, Schafer C, Curio G. BCI competition 2003-data set III: probabilistic modeling of sensorimotor/spl mu/rhythms for classification of imaginary hand movements. *IEEE Trans Biomed Eng* 2004;51:1077–80.
63. Pfurtscheller G, Neuper C, Flotzinger D, Pregenzer M. EEG-based discrimination between imagination of right and left hand movement. *Electroencephalogr Clin Neurophysiol* 1997;103: 642–51.
64. Zhao W, Wu R, Wang S, Qi H, Qian Y, Wang S. Behavioral and neurophysiological abnormalities during cued continuous performance tasks in patients with mild traumatic brain injury. *Brain Behav* 2018;8:e00966.
65. Hohnsbein J, Falkenstein M, Hoormann J, Blanke L. Effects of crossmodal divided attention on late ERP components. I. Simple and choice reaction tasks. *Electroencephalogr Clin Neurophysiol* 1991;78:438–46.
66. Vidaurre C, Krämer N, Blankertz B, Schlögl A. Time domain parameters as a feature for EEG-based brain–computer interfaces. *Neural Network* 2009;22:1313–19.
67. Brodu N, Lotte F, Lécuyer A. Comparative study of band-power extraction techniques for motor imagery classification. In: 2011 IEEE symposium on computational intelligence, cognitive algorithms, mind, and brain (CCMB). Paris, France: IEEE; 2011:1–6 pp.
68. Brodu N, Lotte F, Lécuyer A. Exploring two novel features for EEG–based brain–computer interfaces: multifractal cumulants and predictive complexity. *Neurocomputing* 2012;79:87–94.
69. He KM, Sun J. Convolutional neural networks at constrained time cost. In: IEEE conference on computer vision and pattern recognition (CVPR). Boston, MA: IEEE; 2015. p. 5353–60. JUN 07-12, 2015.
70. Chang CC, Lin CJ. LIBSVM: a library for support vector machines. *ACM Trans Intell Syst Technol* 2011;2:3. SI.