Original Article
ISSN (Online): 2582-7472

ShodhKosh: Journal of Visual and Performing Arts
June 2024 5(6), 945–965

CHALLENGES IN EVOLUTIONARY ALGORITHM TO FIND OPTIMAL PARAMETERS OF SVM: A REVIEW

Ashish Kumar Namdeo[1], Dr. Dileep Kumar Singh[2]

[1]School of Engineering and Technology, Jagran Lakecity University, Bhopal (M.P.), India
[2]School of Engineering and Technology, Jagran Lakecity University, Bhopal (M.P.) India

Corresponding Author
Ashish Kumar Namdeo,
ashish1namdeo@gmail.com

DOI
10.29121/shodhkosh.v5.i6.2024.1985

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ABSTRACT

In rapidly changing classification and predation environment, optimization techniques in determining the hyper parameter of Support Vector Machine has become crucial for the accuracy of result. It's an important tool for improving output quality of classification and prediction which includes modeling parameters relationship and resolution of optimal hyper parameter. However, determination of regularization (C) and gamma (γ), through mathematical models have undergone substantial development and expansion. In this paper, optimization techniques are categorized under several criteria. We also included the benchmarks for measuring the performance of classifier after parameter tuning, and found there is still a scope of improvement.

Keywords: Particle Swarm Optimization; Genetic Algorithm, Firefly Algorithm, Fruitfly Optimization Algorithm, Cuckoo Search, Ski-Driver Algorithm, Support Vector Machines Introduction

1. INTRODUCTION

Now a day's datasets have enormous elements that needed to be classified and predicted, as efficiently and fast as possible. For this purpose, we use Support vector machines as It can handle a large number of features.

The performance of an SVM model mainly depends on the value of hyper parameters.

So, to determine this we are finding a decent blend of Evolutionary algorithm which can ensure Lower computation cost with good efficiency of the SVM model by parameter optimization.

Support vector machines (SVMs), is a novel machine learning algorithm used for classification, and have been applied to other fields which gives higher accuracy than customary learning machines [1], [2], [3]. The idea of SVMs starts from finding an ideal isolating hyperplane keeping in mind the end goal to isolate the biggest conceivable division of preparing set of a similar class on the same side, while augmenting the separation from either class to the isolating hyperplane. As indicated by Vapnik [1], this hyperplane limits the danger of misclassifying of data.

SVM HAS TWO PHASES: LEARNING AND CLASSIFICATION.

While the classification phase of SVM depends on the model evolved during SVM learning stage and has no different parameters than test data, the learning phase of SVM has numerous parameters (called hyperparameters) [8], e.g. SVM piece parameters or regularization parameter (C), which may strongly affect the SVM classifier conduct for hard classification problem. Such hyperparameters, if not well picked, may reduce classification accuracy of SVM. To pick the proper parameters for a particular problem is an imperative research issue in the information mining range [25]. Picking the best values for these parameters is critical issue, since that requires either a thorough seek over the hyperparameter space, or utilization of streamlined strategies that investigate a limited subset of the conceivable values.

As a rule, experts select these parameters observationally by attempting a limited number of values, keeping those that give the slightest test error. However, for a vast number of parameters, this approach is not possible.

Distinctive researchers consider a few methodologies for choice of ideal estimations of the SVM hyperparameters: grid search [26], Nelder–Mead algorithm [27]. Genetic Algorithm [13],Levenberg-Marquardt algorithm [4], gradient descent algorithm [17], Newton strategy [30], simulated annealing [28], pattern search algorithm [27], cross validation[29], heuristic parameter optimization[16], selection based on data characteristic[18], by applying measurable strategies to the matrix framework in an unsupervised fashion [19] utilizing the Online Gaussian Process model of the determining error in parameter space [11], utilizing Advancement Strategy to decide the piece from a parameterized part space and to control the regularization [10], utilizing a subordinate free numerical streamlining agent [9], utilizing Principal Component Analysis (PCA) to discover the kernel parameter in view of the distribution of information in the feature space [20], utilizing an exploratory outline based producing set scan for a proficient investigation of the error surface and local optimization [21], utilizing a nonlinear programming algorithm for figuring kernel and related parameters of the SVM [22], in view of earlier data of the information distribution and Bayesian deduction [25], in light of the Receiver Operating Characteristic (ROC) bends measured on an appropriate approval set [23], in view of the calculation of the slope of penalty function as for the kernel parameters [24], utilizing the Nystrom guess to the SVM kernel [12], utilizing an agent based procedure for minimization of span/edge and traverse headed for leave one out error and Genetic-Quasi-Newton algorithm for the advancement [14], utilizing fractal measurements [15].

Optimization means supplying a set of numerical parameter values which will provide the best fit of an equation. For simple function this can be done by differentiating parameter with reference to parameters this equation depends upon. In more complicated cases, however, it's not possible to differentiate the equations. This can be done by two ways:

i) Direct search methods, in which only values of the function to be minimized (or maximized) are used.

ii) Gradient methods, which also use derivatives of the function.

For Support Vector Machine (SVM), optimizations may be accomplished for obtaining local and global optima. Support vector machine, one of the prevalent machine learning algorithms initially conceived by Cortes and Vapnik [4], have been widely applied to pattern Classification problems [5] [6] and nonlinear regressions [7][6].

A Support Vector Machine is a classifier specified by a dividing hyperplane. That is to say, known labeled training data, the algorithm builds an optimal hyperplane as shown in fig. 1. There are multiple hyperplanes and we have to choose best among them, which is farthest from either side data point because if a line is too close to a data point it won't be able to segregate the data correctly.

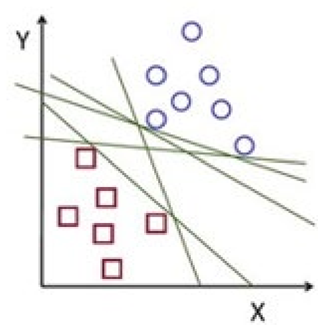

Fig.1. Separating hyperplanes

Thus, the working of the SVM algorithm discovers the hyperplane that gives the largest minimum distance to the training data sets, twice this distance known as margin shown in fig.2. Therefore, the optimal separating hyperplane maximizes the margin of the data sample.

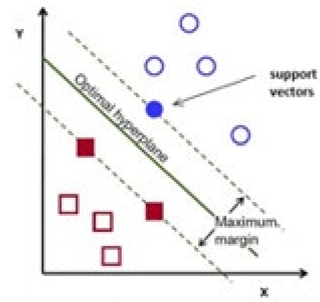

Fig. 2. Support Vector Machine Model

Let's introduce the notation used to define formally a hyperplane:

$$f(x) = \beta_0 + \beta^T x\,. \qquad (1)$$

Where βis known as the weight vector and β_0 as the bias. The optimal hyperplane (Fig.2) defined in different ways by scaling of βand β_(0.)Out of all the possible representations of the hyperplane, the following is chosen as

$$|\beta_0 + \beta^T x| = 1. \qquad (2)$$

Where,x = sample data set nearest to the hyperplane.

The training data sets that are nearest to the hyperplane are called support vectors as shown in figure 2.The distance between a point x and a hyperplane (β+β_o) is given as.

$$\text{distance} = \frac{|\beta_0 + \beta^T x|}{||\beta||}. \qquad (3)$$

For the hyperplane, the numerator is equal to one and the distance to the support vectors is

$$\text{distance}_{\text{support vector}} = \frac{|\beta_0 + \beta^T x|}{||\beta||} = \frac{1}{||\beta||}. \qquad (4)$$

The margin here designated asM, is twice the distance to the closest examples:

$$M = \frac{2}{||\beta||} \qquad (5)$$

Lastly, the problem of maximizingM (Fig.2) is analogous to the problem of minimizing a function under some constraints. The constraints model the requirement for the hyperplane to classify correctly all the training examples x_i. Formally,

$$\min_{\beta,\beta_0} L(\beta) = \qquad (6)$$

$$\frac{1}{2}||\beta||^2 \text{ subject to } y_i(\beta^T x_i + \beta_0)\,.$$

Where y_i represents every individual labels of the training examples. This (6) comes under Lagrangian optimization which will be solved by using Lagrange multipliers to get the weight vector β and the bias β_0of the optimal hyperplane.

CLASSIFIER PERFORMANCE

When we build a model we must foresee accuracy of a model as the number of correct prediction with respect to all other predicts. Once our model is build we need to decide whether it is good enough to make prediction. Classification accuracy is the total instances of correct prediction divided by total number of prediction made for binary classifier we use confusion matrix (fig.3) which is clean and unambiguous way to represent prediction of classifier.

	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Fig. 3. Confusion Matrix.

For binary classification we have a table of 2 rows and 2 columns over the top is observed class labels whereas down side shows predicted class labels True Negative, means that our prediction is correct that class is negative. False Negative, we had incorrectly predicted that class as negative similarly False Positive, is when we incorrectly predict that the class as positive and True Positive is when we correctly predict our class as positive but there is a limitation this method that is we can only calculate the accuracy of classifier when you have dataset divided into different categories but while evaluating continuous measure where we don't have clear choice to make the cutoff it describes the range of possible behavior between true positive and false positive rates we use Receiver Operating Characteristic (ROC) [32][33][34]curve or AUC .If you know the value of positives and negatives you can construct a confusion matrix form any point lies on ROC curve[35][36][37].

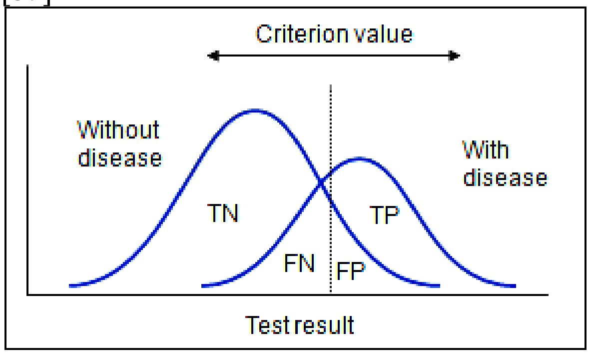

Fig. 4. Overlapping datasets

Since we are comparing two models (Fig.4.) it's often more convenient to have a single metric rather than single thus we compute two metrics from the confusion matrix and combine them to one. True positive rate (TPR) also known as sensitivity, hit rate and recall, which is define as follows:

$$\mathrm{TP} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \qquad (7)$$

The proportion of positive data consider as positive with respect to all positive data points , higher its valve lesser the point we miss. False positive rate (FPR) also known as dropout which is defined as

$$\mathrm{FP} = \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}}. \qquad (8)$$

The proportion of negative data point which are mistakenly considered as positive with respect to all negative data points in other words the higher FPR data the more negative data points will be misclassified. In order to combine TPR and FPR into single metrics we have to compute former metrics with variable threshold then plot them on a single graph with FPR value on X-axis and TPR value on Y-axis. The curve obtained(Fig.5) is known as ROC curve [38][39] and the metric we consider is AUC of this curve.AUC is the area under the ROC curve, which is one of the finest means for analyzing classifiers in two-class problems. In this view the method considered in [31] was realized to measure the AUC.

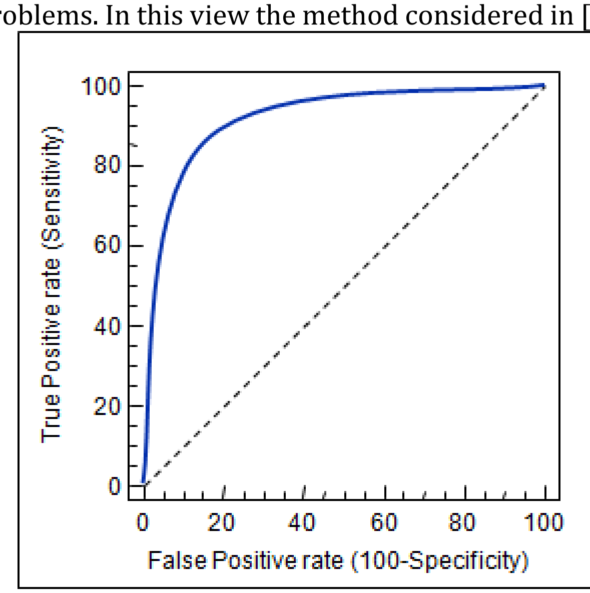

Fig. 5. ROC curve derived from two overlapping distribution

Evolutionary algorithms (EA), are meta-heuristics imitating the long term optimization technique of the biological development for determining numerical optimization problems. The problem solutions are general 'individuals' ina

group. Each solution is estimated by a fitness function. The fitness cost expresses closeness to optimal solution. Moreover, the fitness function has major influence on performance [60][61][62].
In combination with SVM, EA can be made for solving problems like, for instance, construction of a suitable kernel function, optimization of the SVM's parameters and for judgment of the most judicious input data. Combinations of EA and SVM for solving classification tasks can be made out example in [63][64][65][66][67].
Yet, there exist some limitations in the EA, such as low convergence precision, quickly captured in a local optimum value at the next evolution step. This paper gives an improvement suggestion by looking at the knowledge of the global worst, average, and best solutions into the exploration approach to enhance the exploitation.

2. OPTIMIZATION TECHNIQUES FOR SVM

A variety of optimization techniques have been applied to the SVM parameter optimization problem including Grid Search technique as described earlier as well as metaheuristic techniques for instance Particle Swarm Optimization, Genetic Algorithm, FruitFly Optimization Algorithm (FOA)and Firefly Algorithm. In this section we will focus on optimization algorithm and its parameter that search for optimum value of regularization (C) and gamma (γ).

A. PARTICLE SWARM OPTIMIZATION(PSO)

This algorithm was first developed by Dr.Eberhart and Dr. Kenley in 1995 [68]and inspired by social behavior bird flocking or fish schooling PSO has wider variety of application such as function Optimization, Artificial Neural Network and Fuzzy system.
It is influenced by the social nature of bird flight and fish class.PSO has been referred to many design problems owing to its unique searching Process, naive thought, computational performance and successful implementation.
It uses a "population" of particles that pass through the problem hyperspace with indicated sum of speeds. At each repetition, the speeds of the single particles are stochastically adjusted corresponding to the past best position for the particle itself and the locality best position. Both the particle best and the locality best are collected corresponding to a custom defined fitness function. Movement of each particle naturally grows to an optimal or near-optimal result.PSO is not generally influenced by the volume and nonlinearity of the distributed problem, and can concentrate to the optimal result in many problems where most judicious methods are inadequate to concentrate. Each particle (group segment) in the swarm correspond to a quick fix in a large-dimensional area with four vectors, its present position, highest position found so far, the best position found so far by its proximity and its speed and sets its position in the exploration area situated on the best position achieved by itself (pbest)and its neighbor (gbest) during the search method. Steps to perform search for getting optimal solution:

ALGORITHM 1: PSO ALGORITHM

Input: Number of particles search space (Δhd).
Output: best position achieved (Δcf).

1: Each particle (position) Δhd ≠ ø.
2: For each particle Δhd ε set(fitness value, velocity)
3: For optimum solution
Particle search through given problem space
if particles are not at optimum location
then update particle position(Δcf)
4: Find common best (Δcf).
Particles share local information presentbest or pbest(di)
5: Move particle closer to target.
if present value (pbest) which is assigned to the individual particle is better then optimized result will be assigned as Global Best particle which is represented by gbest. (Δgf). check desired results(Δc)analysis each particle position
6:Exit .

Note that each particle structure will be represented by possible solution in search space; velocity value will dictate how much change in the data is appreciable pbest indicate best particle location has ever came to target.

B.GENETIC ALGORITHM(GA)

Genetic Algorithm is developed by Prof. John Holland in 1970 [69]. Genetic Algorithm is basically metaheuristic search algorithm which is based on the Darwinian Theory, whose concept is "survival of the fittest", where successive generation is better than the previous generation. Genetic Algorithm supports the multiobjective optimization Technique and search problem.

GAs exists among the most outstanding group of technique under EA's which are carried away by the evolutionary theories of genetic choice. They moreover support the ideals of Charles Darwin Philosophy of survival of the fittest. Though, because of its superior practice in optimization, GA has been referred to as a function optimizer. Method begins by loading a group of solution (chromosome).It consists of description of the problem generally in the bit vector class. Later for each chromosome calculate the fitness applying a suitable fitness function good enough for the problem. Established on this, the most suitable chromosomes are taken into the matching pool, where they are subject to cross over and mutation thus providing different set of solutions (offspring). There are three types of operators in Genetic Algorithm Selection, Mutation and Crossover.

Natural selection: It is a process in which evolutionary changes in the organism takes place.
Mutation: Mutation operator is a unary operator which operates one chromosome at a time.
Crossover: Crossover operator is a binary operator; it can use two chromosomes at a time.
In GA-SVM there are steps to perform feature selection and parameters optimization which are given as follows:

ALGORITHM 2: GENETIC ALGORITHM

Input: Population must be initialized
Output: Optimized values of parameters ΔC,γ.

1: phenoType ← genoType (Convert)
2:Feature subset determination(Feature ← create feature values)
3: fitnessEvaluation(testingData)
4: while convergence criteria reached () do
5: selection
6: mutation
7: crossover
8:exit.

C.FRUITFLY OPTIMIZATION ALGORITHM(FOA)

Fruit fly is a recently developed evolutionary optimization algorithm by W-T Pan in 2012[40] which is used in determining Optima of the numerical functions. Fruit fly is based on foraging behavior of fruit flies. Fruit fly algorithm involves few parameters and its algorithmic structure is relatively simple as compared to other evolutionary optimization algorithms, due to these advantages it is employed in a variety of applications such as Optimization problems like prediction and classification. SVM classification model build by radial basis kernel function has got two parameters to be optimized, that is regularization parameter C and γ. In this paper, the FOA technique is used to find the parameters of SVM. The process of optimizing the SVM parameters is described as follows:

ALGORITHM 3: FRUIT FLY OPTIMIZATION ALGORITHM (FOA)

Input: Initial location of the each fruit fly that decides the range of SVM parameter vector array (C,γ) (Δhd).
Output: Parameter C and γ (Δcf)

1: Set iteration variable: t=0 and perform the training process from 2~3. Δhd ≠ ø.
2: For each every fruit fly Δhd
    Calculate fitness value(di)
        check fitness value
            analysis each fruitfly fitness value
                update the best fruit fly and global best fruit

if stopping condition is reached
then transforms Go to 3
else go to 2.
3:Exit .

D.FIREFLY ALGORITHM(FA)

The Firefly Algorithm is a metaheuristic algorithm which is inspired by the natural process such as flashing behavior of fireflies. Firefly Algorithm is developed by Xin-She Yang in 2008 [70].It is designed to solve continuous domain problem. Yang has proved that the firefly is having Exploration and Exploitation feature which is superior to the other nature inspired algorithms, when we solve the multiobjective nonlinear function. Applications of the Firefly Algorithm are fault detection, feature selection; clustering and scheduling. The basic goal for a firefly's glow is to make progress as a sign system to appeal to different fireflies. The technique establishes a community-centered iterative technique with various factors (known as fire flies) concurrently having considered optimization problem. Agents make contact with each other via bioluminescent intensity which allows them to seek cost function capacity more adequately than in conventional distributed random exploration.

Swarm optimization technique is established on the theory that solution of an optimization problem can be picked up as representative (firefly) which radiates proportionally to its quality in a thought of problem environments. Consequently each brighter firefly attracts its participants (indifferently of their sex), that comes to the exploration area further closely.

ALGORITHM 4: FIREFLY ALGORITHM (FA)

Input: light absorption coefficient, initial brightness of every firefly(Δhd).
Output: An update dataset common format (Δcf).

1: Initialize number of flies, maximum iteration,Δhd ≠ ø.
2: For each firefly Δhd ε set(candidate solution)
selecting solution from solution (di)
check lesser brighter firefly will move towards brighter firefly. (Δc)
Update candidate solution
3: For each firefly
if best solution contains the nth solution
then it will show random walk to reach optimal solution (Δcf)
if new solution is better than best
then solution it is updated
else the new solution is discarded.
4: Repeat step2-3 until t reaches it's maximum value
5:Exit .

E. CUCKOO SEARCH(CS)

The CS technique is a nature-inspired metaheuristic optimization technique which was proposed by Yand and Deb in 2009 [76]. The reproduction technique of cuckoos is the center idea behind the CS technique. The CS strategy has been produced in view of three idealized assumptions: (i) each cuckoo lays one egg at any given moment and stores it at an random picked nest, (ii) the best nests with the highest quality eggs are conveyed to the next generation , and (iii) the quantity of host nests for saving eggs are settled. Eggs laid by a cuckoo are found by the host bird with a pre-set portion probability, $P_a \in [0,1]$. If there should arise an occurrence of finding outsider eggs, the host bird may just through away them or abandon the nest and fabricate a totally new one In terms of optimization implementation, eggs in nests demonstrate to the solutions. The idea is to replace not-so-good solutions in the nests with new and potentially better solutions. Based on the three idealized assumption, algorithm steps demonstrates the pseudo code for implementation of the CS technique. The strategy applies two exploration techniques. A few solutions are created in the area of the present best solution (a Levy walk). This speeds up the local search. In the meantime, a major division of new solution is produced by a long shot field randomization and whose areas are far away from the present best solution area. This is done to ensure the strategy is not trapped in local search or local optimum. Algorithm shows the pseudo code for CS strategy including Levy flights. Note that CS strategy is when all is said in done population based, elitist, and single goal. A Levy flight is considered while creating new arrangements for X_i^(t+1) the i^thcuckoo.

.

$$X_i^{t+1} = X_i^t + \alpha \oplus \text{Levy} \qquad 9$$

In the above equation, α is the step size which relies upon the scales of the problem of interest.

Where, α=O(L/100), fulfills the search necessities for most optimization problems. L represents the difference between the greatest and least substantial estimation of the problem of interest. The item ⊕ implies entry-wise multiplication.

ALGORITHM 5: CUCKOO SEARCH (CS)

Input: Number of nests, N , the probability parameters, P_a

Output: Generate C,γ

1: Objective function f(x), x= 〚(x_1……x_d)〛 ^T

2: Generate initial population of n host x_i(i=1, 2,…n)

3: While (termination criteria)

Get a cuckoo randomly by levy flights.

Calculates its fitness f_i

Select a nest among n (say ,j) randomly

4: If (f_i>f_j)

Replace solution j by the new solution

end

A fraction (P_a)of worse nests are abandoned and new ones are built.

Keep the best solution.

Rank the best solution and find the current best

end while

Post process results and visualization

5: End

The Lévy flight gives an arbitrary walk where its progression is drawn from a Lévy distribution. There are a some approaches to create this arbitrary steps [20]. The Mantegna's calculation is a standout amongst the most productive algorithm for generating symmetric (positive or negative) Lévy distribution steps. In this technique, the progression length in (9) is evaluated as:

$$S = \frac{u}{|v|^{1/\beta}} \qquad 10$$

Where the value of u and v derived from normal distribution.

u ~N(0,σ_u), v~N(0,σ_v)

where, σ_v=1 and

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin\left({}^{\pi\beta}/_2\right)}{\Gamma\,{}^{(1+\beta)}/_2\ \beta 2^{(\beta-1)/2}} \right\}^{1/\beta} \qquad 11$$

where, Γ(2) is a gamma function

$$\Gamma(2)=\int_0^\infty t^{z-1}\, e^{-t}\, dt \qquad 12$$

Two normal distributions are utilized by Mantegna's algorithm to produce a third irregular variable which has a similar behavior of a Levy distribution. In the CS technique proposed by Yang and Deb [75], the entry-wise multiplication of the random number and distance between the current or present and best solution is used as a transition probability to move from the present location to the next location. As per this, (9) can be changed as,

$$x_i^{t+1} = x_i^t + \alpha\, s(x_i^t - x_i^{best})r \qquad 13$$

Where, x_i^best is the current best solution and r is a random number which is drawn from a normal distribution with zero mean and unit variance. The step length s is also evaluated applying (10). Remaining discussion about CS technique and its details can be found in [75] [76].

F. SOCIAL SKI DRIVER ALGORITHM (SSD)

The Ski Driver Algorithm is inspired by the path followed by the Ski Driver while going down the hill, proposed by Alaa and Thomas in 2019 [107].

The design parameter of Ski Driver algorithm consists of position of the agents as well as the velocity which is used to position based on the Global of the mean's of the previous three best position. The position is updated. As per this, (14) ,

$$x_i^{t+1} = x_i^t + V_i^t \qquad 14$$

Here, v_i^t is the velocity given as follows.

$$v_i^{t+1} = \begin{cases} c\sin(r_1)\,(P_i^t - X_i^t) + c\sin(r_1)\,(M_i^t - X_i^t)\ if\ r_2 \le 0.5 \\ c\cos(r_1)\,(P_i^t - X_i^t) + c\cos(r_1)\,(M_i^t - X_i^t)\ if\ r_2 > 0.5 \end{cases}$$

In (15) Equation M_i is the mean of top best three solutions [108] obtained, and r is a random number which is drawn from a uniform distribution within the range [0, 1]. P_i is best solution obtained by calculating the fitness function, the value of this is then compared to the current best solution, just like in PSO [68].

ALGORITHM 6: SOCIAL SKI DRIVER ALGORITHM (SSD) ALGORITHM

Input: Agent's position X_iand velocityV_i.
Output: best position achieved.

1: Assign Each Agent's (position) with minimum fitness value.
2: For each particle set (fitness value, velocity) until stop criteria are met, compute the steps below.
3: Calculate Fitness value
4: Sort agent with respect to fitness value.
Calculate the last best solution known and global mean.
Find new solution by updating agent's location (14),
Also update its velocity (15).
5: Move agent closer to target.
6. End loop and Return best position.
7: Exit.

But the trajectory of path in SDD is not same as in case of PSO, in this case it follows sine and cosine path as this gives a better approach in determining the optimal path which can lead to a better optimal solution.

3. ANALYSIS OF OPTIMIZATION ALGORITHMS

An optimization method can be considered from various perspectives. In this field, we will consider it as three evolutionary operators as well as its application to optimize C,γ.

A EXPLORATION AND EXPLOITATION

Nature-inspired optimization methods can further be considered from the actions they seek the search space. In principle, all methods should have two key factors: exploitation and exploration, which are again cited to as intensification and diversification [77] [78].

Exploitation purposes any knowledge obtained from the issue of significance so as to facilitate to develop new solutions that are better than existing solutions. However, this process is generally local, and knowledge is too local.

Therefore, it is for local search. The influence of exploitation is that it usually contributes to excessively large convergence rates, but its harm is that it can get caught in a local optimum because the ultimate solution point largely depends on the initiating point.

On the other hand, exploration makes it feasible to examine the search space more quickly, and it can generate solutions with sufficient diversity and distant from the current solutions.
Therefore, the search is normally on a global scale. The influence of exploration is that it is less likely to get stuck in a local mode, and the overall optimality can be more accessible.
However, its disadvantages are slow convergence and waste of most computational efforts because many different solutions can be far from global optimality.
Therefore, a definite balance is needed so that a method can obtain satisfactory performance. Too much exploitation and too little exploration means the system may converge more rapidly, but the possibility of finding the proper global optimality may be small. On the other hand, very little exploitation and too much investigation can contribute to the search path wonder around with excessively low convergence.
The optimal balance should mean the proper measure of exploration and exploitation, which may contribute to the optimal performance of an algorithm. Therefore, balance is crucially important [77].
However, how to carry out such balance is still an open problem. In fact, no method can claim to have produced such balance in the current literature. In principle, the balance itself is a hyper optimization problem, because it is the optimization of an optimization method.
In addition, such balance may depend on many aspects such as the driving mechanism of an algorithm, its setting of parameters, tuning and supervision of these parameters and indeed the problem to be considered.
Furthermore, such balance may not always exist, and it may vary from problem to problem. This is consistent with the so-called "no-free-lunch (NFL)" theorems [79].
The NFL theorems indicated that for any two methods A and B, if A performs better than B for some problems, there must be some problems on which that B will do better than A. That is, if worked out over all possible problems, the average performance of both methods is approximately identical.
In alternative terms, there is no universally better method that can be productive for all problems. Though theoretically convincing, NFL theorem may have specified impact in process because we do not require figuring out all problems and we do not require the average performance either
One of the major objectives of problem-solving in practice is to seek to discover the optimal solution in a limited, sufficient timescale. For a given type of problem, some algorithms can certainly perform better than others. For example, for convex problems, methods that operate problem specific convexity instruction will perform better than black-box type algorithms. However, some studies indicate that free lunches can occur for some types of problems, specifically for co-evolutionary approaches [81].
These unresolved problems can drive further investigation in this field. Some variations of meta heuristics could address such issues like Chaotic fruit fly optimization algorithm (CFOA) [80], improved fruit fly optimization algorithm via Levy flight [72], Accelerated Particle Swarm Optimization (APSO) [71].

CUCKOO SEARCH

Cuckoo search (CS) was formed in 2009 by Xin-She Yang and Suash Deb [82].
CS is based on the brood parasitism of some cuckoo species. In addition, this method is reinforced by the so-called L´evy flights [83], rather than by basic isotropic random walks.
Later studies show that CS is potentially much more efficient than PSO and genetic algorithms [84][85][86][87][88][89][90]. Infact, CS and its variations have employed in practically every area of engineering design and applications [91][92][93][94] [95][96].
CS acquires two particular improvements over alternative methods such as GA, and these improvements are: efficient random walks and balanced mixing. Since L´evy flights are generally much more efficient than any alternative random-walk-based randomization techniques, CS can be valuable in global search.
In fact, recen investigations indicate that CS can have secured global convergence [97]. In extension, the similarity between eggs can yield better new solutions, which is approximately fitness-proportional generation with a suitable blending ability.

In alternative terms, CS has varying mutation realized by L´evy flights, and the fitness-corresponding generation of current solutions based on relationship provides an indirect form of crossover.
In extension, selection is driven out by employing p a where the appropriate solutions are developed onto the next generation, while not so favorable solutions are taken over by new solutions.

Furthermore, simulations further indicate that CS can have auto-diving capability in the anticipate that new solutions can frequently dive into the locality where the potential global optimality is located.
It is worth marking out that both CS and FA can pick up the essential aspects of SA and APSO, but there are some notable distinction between FA and CS. One significant distinction between FA and CS is that FA needs the distance-based, landscape-modulated attraction.
As local attraction is greater than long-distance attraction, FA can subdivide the whole population into such populations, while CS does not. In supplement, FA uses random walks and ranking, while CS uses L´evy flights and random permutation, which will proceeds in distinct behavior. These distinctions make both FA and CS exclusive.

FIREFLY ALGORITHM

One unique component of FA is that attraction is employed, and this is the initial of its kind in any SI-based algorithms. Since local attraction is greater than long-distance attraction, the population in FA can naturally subdivide into numerous subgroups, and each group can possibly swarm arounda local system. Among all the local system, there is consistently an overall best solution which is the normal optimality of the issue. Thus, FA can handle with multimodal issues uniformly and effectively. The movement of a firefly i is drawn to another more brighter firefly j is measured by

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha \epsilon_i^t, \qquad 16$$

where the middle is because to the attractiveness, and β_0is the attractiveness at r = 0. The last term is randomization where α is randomization parameter, and ϵ_i^t is a vector of random numbers picked from a Gaussian distribution at moment t. Other investigations still adopt the randomization in terms of ϱt i that can readily be expanded to alternative distributions such as L´evy flights [98,99]. A thorough analysis of firefly algorithm and its variations has been driven out by Fister et al. [100]. From the above equation, we can understand that mutation is employed for both local and global search. When ϵ_i^tis taken from a Gaussian distribution and L´evy flights, it produces mutation on a wider scale. On the other hand, if α is taken to be an exceedingly small value, then mutation can be exceedingly small, and therefore restricted to a local subspace. Interestingly, there is no clear selection in the formula as global best is not applied in FA. However, during the update step in the loops in FA, ranking as well as selection is applied. One novel quality of FA is that attraction is applied, and this is the first of its type in any SI-based algorithms. Since local attraction is greater than long-distance attraction, the population in FA can automatically subdivide into various subgroups, and each group can potentially swarm around a local system. Among all the local system, there is invariably a global best solution which is the appropriate optimality of the problem. Thus, FA can handle with multimodal problems easily and efficient.
From above (14). , we can understand that FA disintegrates into a variation of simulated annealing (SA) when β_0 = 0 . Further, when x_j^t is changed by global best, FA likewise turns into the accelerated PSO. Therefore, APSO and SA are particular instances of the firefly algorithm, and therefore FA can have the gains of these algorithms. It is no surprise that FA can be resourceful and capable, and perform better than other algorithms such as GA and PSO.As a result, the firefly algorithm and its variations have been employed in a diverse area of applications [102], comprising hard problems and multiobjective problems [103].

PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) was established by Kennedy and Eberhart in 1995 [68], based on the swarming behaviour such as fish and bird teaching in nature. In principle, the position and velocity of a particle, x_iand v_i , correspondingly, can be adjusted as follows:

$$v_i^{t+1} = v_i^t + \alpha\epsilon_1\left[g^* - x_i^t\right] + \beta\epsilon_2\left[x_i^* - x_i^t\right], \qquad 17$$

$$x_i^{t+1} = x_i^t + v_i^t, \qquad 18$$

where ϵ_(1)and ϵ_2 are two arbitrary vectors, and each item can pick up the costs between 0 and 1. The parameters α and β are the training parameters or acceleration constants, which can often be selected as, α ≈ β ≈ 2.
By analyzing the above equations, we can understand that the recent position is brought about by pattern-search-type variation, while selection is implicitly determined by employing the present global best solution g^* found so far, and likewise through the individual best x_i^*. However, the performance of individual best is not quite obvious, even though the current global best seems particularly significant for selection, as this is determined in the accelerated particle swarm optimization [68][101].

Therefore, PSO consists of mainly mutation and selection. There is no crossover in PSO, which means that PSO can have a significant flexibility in particles with a large degree of exploration. However, the use of g^* seems strongly selective, which may be like a double-edge sword. Its advantage is that it serves to boost up the convergence by drawing towards the present best g^*, while at the same moment it may get to premature convergence even though this may not be the true optimal solution of the problem of interest.

GENETIC ALGORITHM

Genetic algorithm (GA) is the most extensively used and researched evolutionary optimization method in the scientific world. It is a guided stochastic search approach inspired from the principles of natural fittest selection and population genetics. In general terms, it is based on the parent and offspring iterations and their transformations through times. GA generates candidate solutions from the location of all feasible solutions and measures their performance as per the considered objective function. It has been demonstrated that GA performs strongly together in constrained and unconstrained search problems where the quantity of stable solutions is exceedingly small compared to the volume of the search space. GA converges towards more competitive solutions by applying elitism, crossover, and mutation operations. GA first makes a population (usually randomly) of possible solutions (also called chromosomes) for the optimization problem. This population is later evaluated employing the objective function of the interest. Then GA uses its three operators to make the current population for the later generation. The best performing chromosome(s) is copied to the next generation unchanged. This refining is called elitism and produces the best solution(s) is not absorbed as the optimization proceeds.

Crossover operator is employed for combing good parents and generating offspring. This operator is handled with the hope of retaining the essence of good chromosomes. In its simplest form, i.e., single point, a random point (crossover point) is randomly chosen. Then the operator swaps portions of pair chromosomes at the crossover point. Alternative crossover techniques are multi-points and uniform. Regardless of the type of applied crossover operator, its generated offspring only have information carried by the present population new operator is expected to introduce and create new knowledge (solutions) to the population. Mutation operator produces new offspring by randomly modifying the contents of genes at one or more positions of a selected chromosome.

GA is stochastic and gradient-independent, so it can be conveniently implemented for minimization or maximization of discontinuous and non differentiable objective functions. Theoretical literature of GA is very rich and various applications of GA for real world optimization problems have been accounted for in the last two decades. Detailed discussion about GA and its operators can be established in basic reading sources such as [104].

FRUITFLY OPTIMIZATION ALGORITHM

The main principle of FOA's components is that it is easy to understand, contains a simple searching methodology, and is simple to implement. Due to its great execution and magnificent properties, FOA has been generally utilized in some real-world classification tasks and in the SVM optimization field. Although FOA can achieve significant outcomes in terms of search proficiency and running time in different fields, FOA's seeking execution depends only on its natural product fly swarm area, which can easily lead the process of the FOA to fall into the trap of local optima. Consequently, to address this problem and enhance FOA's searching capacity, we present the disorderly PSO in combination with the proposed change methodologies to be utilized as a part of FOA. The transformation system is proposed to produce two diverse osphresis forging techniques for at the same time searching for the local optimum and the global optimum. One of these methodologies is the worldwide looking stage, which replaces the random method that fruit flies employ to find food sources.

Here analogy is carried on distinct characteristics of various meta-heuristic techniques like parameters whose values should be initialized before commencing the implementation, convergence i.e. how the algorithm converge in local maxima, Exploration & Exploitation Component means Growth i.e. the exploitation of search space and Exploitation means moving the unexplored segment of the search space. Some techniques as listed by Table 1. used by optimization algorithm to optimize SVM parameters and compared with rest of the techniques that describes their Initial Parameters, Dataset used, Initial problem domain, Exploitation, Exploration mechanism, convergence and as well as SVM kernel used.

SOCIAL SKI DRIVER ALGORITHM

The nature of the Ski-Driver Algorithm was inspired by the path of Ski Driver, while moving down the hill to find the optimal solution by exploration.

Who's Optimization consist of an Objective Function where we will find an optimal value of C and Gamma parameter of SVM, where are already aware about the solution.
Thus we have our Search Space is bounded by [107] C [0.01, 1000], and the range of [107] Gamma is bounded to [0.01, 50]. By extending the search space we could slower our convergence rate.
Next comes Optimization part,here we do not use accuracy as a means to calculate fitness of optimized solution because this will leads to erroneous results, As it does not differentiate between correlated labels that belongs to different label. Thus we will use sensitivity [107] as we are dealing small number of sample of minority class. Our aim is to find C and Gamma value, which will provide us maximum sensitivity.

As soon as our termination condition is met then this algorithm will end else it will continue to iterate throw next generation of population.

Table I. Summerization of optimization techniques

Techniques	Genetic Algorithm (GA) Cheng-Lung Huang,et al.2006 [43]	Particle Swarm Optimization (PSO) Huang, et al.2008 [41]	Fruit Fly Optimization Algorithm (FOA) Shen, Liming, et al. 2015[44]	Firefly Algorithm(FA) Chao, et al.2015[45]	Cuckoo Search(CS) Minlan Jiang, et al. 2016[74]	Social Ski Driver Algorithm (SSD)Alaa Tharwat , et al. 2019[107]
Initial Parameters	Crossover rate, mutation rate, population size.	Population size, velocity of each particle.	population size, initial swarm location, random flight distance range.	Population of fireflies, maximum generation, new solution, best firefly.	Population size, inertia weight, generations, and the range of hyper parameters.	Ski Driver agent's, position and velocity.
Dataset used	UCI repository data sets used as benchmarks to compare the performance with different optimization technique.	UCI repository data set.	Biomedical datasets from the UCI machine learning data repository were used.	Binary class dataset from UCI repository database [46].	UCI machine learning repository	KEEL[106] , machine learning dataset repository
Initial problem domain	Optimization problems in which the constraints and objective functions are non-linear and/or discontinuous GA can solve such problems.	Continuous problem.	Continuous problem.	Continuous problem .	non-convex and noisy function	Continuous problem.
Exploitation Mechanism	Performed by selection.	Updating the position of particle towards the global best position.	Global worst, mean, and best solutions according to distance from origin and smell concentration.	Firefly movement according to attractiveness.	average accuracy from kfold cross validation is exploited as a fitness function,	Agents position are updated towards the Global point which comprises mean of three best solution.
Exploration mechanism	Exploration is done by Crossover (two-point crossover) and mutationoperator.	Updating the velocity of particle.	Random flight direction and distance for food by individual fruitfly.	Random motion of fireflies.	explores the search space using Levy flights	Updating the Velocity of the agents.
SVM kernel used	Radial Basis Function (RBF).	Radial Basis Function (RBF).	Radial Basis Function (RBF).	Radial Basis Function (RBF).	Radial Basis Function (RBF).	Radial Basis Function (RBF).
Convergence	Rapid	Rapid but less than GA	Rapid	Rapid but less than GA	Better as compared to GA	Better as compared to PSO

In paper [41] authors discussed about Grid Search and GA technique. Grid Search does not perform feature subset selection but GA performs feature subset selection and optimizes C, γsimultaneously thus it is better than grid search, PSO is one of evolutionary computing technique to find global optima which uses social behavior of birds regarded as particles in solution space its computationally efficient relatively and easy to implement and it is used in variety of application in case of SVM it is used to optimize C, γas well as in feature subset set selection so the proposed PSO-SVM will simultaneously optimize all three parameter author proposes discrete PSO for feature subset selection and continuous PSO for C, γoptimization. In order to measure the performance [42] of algorithm hit cover and ratio indexes

had been used which were defined as number of hit on correct false, number of features correctly classified and ratio of correct feature respectively.
In GA paper[43] author discuss about the drawback of Grid Search technique that it doesn't perform feature selection so, he proposes GA-SVM to perform feature selection along with C,γoptimization and its performance is measured in terms of accuracy which is measured as the product of specificity and sensitivity of dataset of binary class in case of multiple class it is measured by overall hit rate.
The FOA[44] discusses previously used techniques like grid and gradient descent technique to find out C,γ and points out that both suffers from local optimum problem and also discusses about evolutionary algorithm like GA and PSO which is considered to have a better potential of finding optimum solution. Author finds out recently developed evolutionary algorithm, fruit fly algorithm is computationally feasible and simple to find global optimum of continuous function. Hence he uses it to optimize C,γ whose value is explored by using smell concentration which is inversely proportional to distance between fruit fly and its food. In order to evaluate the performance of FOA-SVM author has used classification characteristic curve in which we find area under the operating characteristic curve, specificity and sensitivity.
In [45] it is observed that the Grid Search and PSO are vulnerable to stuck in local optimum solutions and also the Lagrangian multiplier α is not taken into account in grid search algorithm while optimizing C, γparameter as well as Bee Colony Optimization is used only to optimize C, γ as given in this paper, so author proposed FA-SVM to optimize C, γ and ∝ simultaneously by Firefly algorithm which discusses thebehavior of firefly particularly their bioluminescent property is highlighted in order to explore the solution search space for the optimum value.
In [74] it is a technique of dynamic measurement error prediction based on CS-optimized SVM parameters(C,γ) is presented. The CS algorithm is applied to select the proper SVM parameters to effectively avoid the "overfitting" or "underfitting" occurrence of SVM, thus improving the prediction efficiency. The simulation experiments illustrate that the projected model performs well in all tested cases. The results of the CS-SVM model were compared with those of the PSO-SVM and GS-SVM in terms of MAPE and RMSE, with the results that the CS-SVM has higher accuracy and a better effect than other algorithm. The proposed method may present a new modeling method for dealing with dynamic measurement error and has definite value for application in error correction
In order to evaluate the performance of algorithm OAA (One Against All) strategy is employed in which binary SVM for each class is used to separate member that belongs to that class from rest of classes. By careful observation we see that recently developed FOA excels rest optimization techniques stated, either the remaining evolutionary technique tries to optimize C,γ and feature selection or C,γ and α (lagrangian multiplier) simultaneously to optimize SVM whereasFOA is used to optimize C,γ only yet had shown better result as compared to PSO and GA,where as in case of firefly the attractiveness function seems to be relatively complex than smell function of FOA for exploration purpose so its computation will be more in relative to FOA.

4. ISSUES AND CHALLENGES

After going through research papers we observe that Support Vector Machine parameter optimization algorithms are premature or fallen into the local optimum. However, due to the complexity of identified models, these algorithms are disadvantageous in locating the global optimal solutions unless the good initial values of parameters are selected. In addition, the priori knowledge about the scopes of parameters is needed to obtain a faster convergence speed. Some of the challenges and issues are listed below.
An idea of evolutionary algorithm works only when social behavior have some similar characteristics so there is a need to understand the algorithm and its process that's about to model and its analogy as well as the following consideration must be in mind before suggesting any evolutionary computation model up to what extent the model is satisfying social analogy?, is the analogy with its independent solution could be supported independently and empirically?
Does the model justify current development towards evolutionary algorithm the choice of appropriate operator to produce desired results?, since the descriptive model developed are the part of social science behavior rather than computer science it must be showed that heuristics of such kind exists on both the side of analogy that they not only increase our understanding but also gives us the direction of further development although this heuristics couldn't be quantified, qualitative performance is still an area for further research.
What's the reason behind using evolutionary algorithm for modeling a descriptive process? is the application was inspired by theoretical or empirical suitability?
Hybridizing evolutionary algorithm with other algorithm to improve quality of the solution. Metahueristic algorithm can be improved by combining it with other local or global search algorithm Advantage of combining is that one algorithm

can overcome the weakness of another algorithm Rizk M Rizk Allah Hybridization of Fruit Fly Optimization Algorithm and Firefly Algorithm for Solving Nonlinear Programming Problems [48] in which he had combined fruit fly with fire fly so as to solve premature convergence problem.

Improving solution by changing exploitation operator. Fei Ye, et al. An improved chaotic fruit fly optimization based on a mutation strategy for simultaneous feature selection and parameter optimization for SVM and its applications[49] in which the result had shown superiority of chaotic Firefly over other algorithm in which he uses chaotic movement to find the better solution and parameters Optimization of SVM.

Based on Improved FOA and its Application in Fault Diagnosis[50] by Qiantu Zhang , et al. which introduces levy fly search strategy to exploit the search space while searching for food long jump via levy flight helps in performing extensive search where as in paper Research on parameters Optimization of SVM and an Improved Fruit Fly Optimization Algorithm by Qiantu Zhang , et al. [50]by dynamically dividing solution in to subgroup accordingly in which global search is performed under the influence of best individual and a local search is made via levy flight near the best solution and after that we will combine local and global best solution by exchanging information which leads to overall optimum solution by those recombination experimental results have shown that IFOA performs well as compared to FOA ,PSO, GA.

In Short-term load forecasting using Support Vector Machine optimized by the improved fruit fly algorithm and the similar day technique Ai-hua Jiang ,Ni-xiao Liang[51] had suggested a way to improve the Firefly Algorithm because the value of distance is selected from a large random value thus the smell concentration value is appearing on small scale which had resulted in premature convergence so he had introduced an optimal parameter which helps in escaping from local minima .Mixed Fruit Fly Optimization Algorithm Based on Lozi's Chaotic Mapping HuixiaLuo[52] which uses Lozi's mapping to perform global search for finding optimal parameter.

Selection of classification technique while converting binary class classifier [53] into multiclass classifier [54][55][56][57] for instance one one-vs-all (OVA) or all-vs.-all (AVA). AVA seems to be much more efficient since it requires O(N^2) instead of O(N) but each classifier is much smaller.

5. PERFORMANCE EVALUATION OF SVM

In this section we analyze classification accuracy according to data set used on a number of medical dataset Wisconsin breast cancer (Wisconsin), Pima Indians diabetes (Pima), Parkinson Thyroid. Each data set has different training and test split to reliably calculate performance. We can divide data set according to training data in to different sizes of large and small data set. In most dataset number of feature and instances are large. To reduce the search space of parameter set we will use radial basis function (RBF).In addition to this the mod of solving Binary classifier(one against-one and one-against-all).For each problem of binary classification we will consider same values of regularization(C) and Gamma(γ).

The performance of classifier is measured in terms of classification accuracy, area under operating characteristic curve specificity and sensitivity in paper Liming Shena, el. Evolving support vector machines using FruitFly Optimization Algorithm for medical data classification FOA-SVM has achieved highest performance among those five technique [58] PSO-SVM Grid-SVM GA-SVM BFO-SVM FOA-SVM by studying the standard deviation among its four technique which comes out to relatively less. their classification performance is also measured in terms of Tpair test in which it's used to measure the pair of each of element from a sample of Wisconsin dataset where the pvalue less than 0.05[58] have got statistical significance in which FOA-SVM performs well, in order to know the computational efficiency in terms of CPU given

same number of crossfold validation and population size time FOA-SVM is not the best but performs well as compared with PSO-SVM GA-SVM BFO-SVM.

Huang, et al [41] proposes a PSO-SVM technique to optimize the performance of SVM classifier. 10-fold cross validation is used to validate and evaluate the provided solutions. The results obtained Optimization of SVM Multiclass by particle swarm (PSO-SVM) gives a better classification in respect of accuracy and paper Huang,et al. [43]highlights that GA out performs Grid search algorithm and it's not only used to optimize C and γ parameters but also used for feature subset selection processwe also see that Shen, Liming, et al. 44] introduces FOA-SVM as newly developed technique used for Parkinson's disease which was compared against PSO-SVM and Grid-SVM result shows that FOA-SVM performs well relatively to those algorithms at last paper Chao, et al. [45]on Firefly-SVM training algorithm shown better result than the other two techniques as given in table2. Alaa, et al [107] introduced a SSD-SVM based algorithm, in which we

discussed, the reasons which makes SSD algorithm better in comparison to PSO and BA algorithm[109] because it can escape from local maxima or minima as we are following the path which is lead by the mean of three best possible solutions. On other hand in PSO and BA algorithm we follow path relative to either Global maxima or minima. And the exploration path which is followed by the PSO and BA algorithm is pretty straight forward in comparison to SDD. As in SDD we use sine and cosine function to explore search space. Hence, SDD-SVM is better than PSO-SVM and BA-SVM.
The "No Free Lunch" (Wolpert and Macready) [59] theorem for optimization, all optimization algorithms that search for optimum solution performance are exactly same when averaged all cost function so any optimization algorithm if modified to perform well for one class of problem had equally paid performance over other class but this doesn't prevent us from developing new better algorithms for specific nature of problem.

6. DISCUSSIONS

Many optimization methods are made on swarm judgment, and use population-centered strategies. Most will need some order of three key evolutionary operators: crossover, mutation and selection; usually, all methods use mutation and selection, while crossover may manifest itself in some indirect manner in some methods. Crossover is precisely in exploitation and can frequently give good convergence in a subspace. If this subspace is where the global optima lies, then crossover with elitism can nearly prove to produce global optimality. However, if the subspace of crossover is not in the region where the global optima lies, there is a possibility for premature convergence.
The widespread utilization of mutation and selection can often enable a stochastic method to have a strong capability of exploration. As the exploitation is comparably small, the convergence movement is normally small, contrasted with that of established methods. As a consequence, most metaheuristic algorithms can normally work out adequately for nonlinear problems, having relatively tough optimization. However, the quantity of function evaluations can be exceedingly costly.
The use of crossover and mutation in exploration is very slight, even though selection as an exploitation process can be direct and even powerful. However, it is yet not clear how the sequence of crossover, mutation and selection can immediately link to the balance of exploration and exploitation.
Infact, this is yet an open problem. For instance, in genetic algorithms, the probability of crossover can be as large as 0.95, while the mutation can be generally small in the order of 0.01 to 0.05. In comparison with other algorithms, the exploration seems small, but genetic algorithms have been demonstrated to be very much useful. On the other part, modification-associated L´evy flights in cuckoo search can have significant exploration ability, and yet cuckoo search can converge very quickly. At the same moment, it is not clear what percentage of the examination is in exploration in the standard firefly algorithm, and even though it has been illustrated that firefly algorithm is very productive in handling with multimodal, highly nonlinear problems.
Indeed in the conventional particle swarm optimization, it is not evident what proportion of the search iterations is in the exploration. The purpose of the present global best can be beneficial and disadvantageous as well. The present global best may serve to boost up the convergence, but it may similarly contribute to the false optimality if the present global best is picked up from a biased collection of individuals taken from a subspace where a local optimum (not the global optimum) is established. All these indicate that it is yet obscure how to achieve the optimal equilibrium of exploration and exploitation by tuning the incorporation of evolutionary operators.
As matter of fact, an accurate balance cannot be accomplished by placing together all evolutionary operators in a proper sense without tuning parameters. From direct observation, we are aware of that the setting or values of any algorithm-dependent parameters can impact on the behavior of an algorithm considerably. In order to produce decent performance, we need to identify the appropriate values for parameters. In other terms, parameters essential to be refined so that the algorithm can perform to the most desirable result. Parameter tuning is still an active area of research [12]. An interesting development is a combined framework for self-tuning optimization methods [60].

Despite the weightage of the above issues, little improvement has been carried out. On the contrary, there is some alteration in research efforts aside from important problems. Nature has grown into millions of different species with a diverse range of properties, but this does not mean that researchers should develop millions of other algorithms, such as the grass algorithm, leave algorithm, beatles algorithm, sky algorithm, universe algorithm, or even hooligan algorithm. Attention should put on dealing with important issues. Though, this seems not mean new algorithms should not be developed at all. The research community should promote truly novel and efficient algorithms in terms of better evolutionary operators and a better balance of exploration and exploitation.

7. CONCLUSION

In this paper, we have analyzed the performance of some evolutionary optimization techniques for SVM parameter tuning. For such analysis, we depict the techniques used by the different algorithm to optimize independent parameters. For benchmark we see that UCI repository datasets had been used, in which all the evolutionary optimization techniques overcame the default values and previous technique in respect to performance. The review showed that the evolutionary optimization techniques are able to produce good results for most of the datasets. The evolutionary optimization techniques can still be less computationally expensive than the commonly accepted techniques, to get more favorable results. The review showed the potential of the evolutionary optimization techniques for the SVM parameters tuning.

As future work we also intend to improve the classification estimation error for the SVMs to obtain a smaller test error. In order to improve the results of the evolutionary optimization, since they stuck to local optima. This paper suggests that new evolutionary algorithm should be made to solve this problem by finding the best solution. This modification may enhance the performance and reduce computational costs.

CONFLICT OF INTERESTS

None

ACKNOWLEDGMENTS

None

REFERENCES

V. N. Vapnik, Statistical learning theory, Wiley, NewYork, 1998.

B. Schölkopf, A.J. Smola, Learning with kernels:Support vector machines, regularization, optimization,and beyond, Cambridge, Mass: MIT Press, London,2002.

O. L. Mangasarian and D.R. Musicant, ``Successive overrelaxation for support vector machines," IEEETrans. on Neural Networks, vol. 10, no. 5, pp

Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." sMachine learning 20.3 (1995): 273-297.

Drucker, Harris, Donghui Wu, and Vladimir N. Vapnik. "Support vector machines for spam categorization." IEEE Transactions on Neural networks 10.5 (1999): 1048-1054.

G.O.Young, “Syntheticstructureofindustrialplastics,”in Plastics, 2nd ed., vol. 3, J. Peters, Ed. New York: McGraw-Hill,1964,pp.15–64.

Mukkamala, Srinivas, Andrew H. Sung, and Ajith Abraham. "Intrusion detection using an ensemble of intelligent paradigms." Journal of network and computer applications 28.2 (2005): 167-182.

Duan K, Keerthi SS, Poo AN (2003) Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing 51:41–59

Eitrich T, Lang B (2006) Efficient optimization of Support Vector Machine learning parameters for unbalanced data sets. J ComputAppl Math 196(2):425–436

Friedrichs F, Igel C (2004) Evolutionary tuning of multiple SVM parameters. In: Trends in neurocomputing: 12th European symp on artificial neural networks 2004, vol 64, pp 107–117

Frohlich H, Zell A (2005) Efficient parameter selection for Support Vector Machines in classification and regression via model-based global optimization. In: Proc of IEEE int joint conf on neural networks (IJCNN ’05), 31 July–4 Aug 2005, vol 3, pp 1431–1436

Gold C, Sollich P (2005) Fast Bayesian Support Vector Machine parameter tuning with the nystrom method. In: Proc. of the IEEE int joint conf on neural networks (IJCNN ’05), July 31–August 4, 2005, Montréal, Québec, Canada, vol 5, pp 2820–2825

Imbault F, Lebart K (2004) A stochastic optimization approach for parameter tuning of support vector machines. In: Proc of the 17th intconf on pattern recognition (ICPR 2004), 23–26 August 2004, Cambridge, UK, vol 4, pp 597–600

Kulkarni A, Jayaraman VK, Kulkarni BD (2004) Support vector classification with parameter tuning assisted by agent-based technique. ComputChemEng 28(3):311–318

Kurasova O, Dzemyda G, Vainoras A (2007) Parameter system for human physiological data representation and analysis. In: Proc of 3rd Iberian conf on pattern recognition and image analysis, IbPRIA 2007, Girona, Spain, June 6–8, 2007. LNCS, vol 4477, pp 209–216

Mattera D, Haykin S (1999) Support vector machines for dynamic reconstruction of a chaotic system. In: Schölkopf B, Burges CJC, Smola AJ (eds) Advances in kernel methods: support vector learning, pp 209–241

Chapelle O, Vapnik V, Bousquet O, Mukherjee S (2002) Choosing multiple parameters for support vector machines. Mach Learn 46(1–3):131–159

Smola AJ, Murata N, Schölkopf B, Miller KR (1998) Asymptotically optimal choice of ε-loss for support vector machines. In: Proc of 8th int conference on artificial neural networks, Berlin, Germany, pp 105–110

Cassabaum ML, Waagen DE, Rodriguez JJ, Schmitt HA (2004) Unsupervised optimization of Support Vector Machine parameters. In: Kadar I (ed) Automatic target recognition XIV. Proc of SPIE, vol 5426(1), SPIE Defense & Security Symposium, Orlando, FL, April 13–15, 2004, pp 316–325

Debnath R, Takahashi H (2004) An efficient method for tuning kernel parameter of the support vector machine. In: Proc of the IEEE intsymp on communications and information technology (ISCIT 2004), Sapporo, Japan, October 2004, vol 2, pp 1023–1028

Lim H (2004). Support vector parameter selection using experimental design based generating set search (SVEG) with application to predictive software data modeling. PhD thesis, Syracuse University

Schittkowski K (2005) Optimal parameter selection in Support Vector Machines. J IndManagOptim 1(4):465–476

Ancona N, Cicirelli G, Stella E, Distante A (2002) Object detection in images: Run-time complexity and parameter selection of Support Vector Machines. In: Proc of the 16th intconf on pattern recognition (ICPR’02), 11–15 August 2002, Quebec, Canada, vol 2, pp 426–429

Quan Y, Yang J (2003) An improved parameter tuning method for support vector machines. In: Proc of 9th intconf on rough sets, fuzzy sets, data mining, and granular computing (RSFDGrC 2003), Chongqing, China, May 26–29, 2003, pp 607–610

Ali S, Smith KA (2003) Automatic parameter selection for polynomial kernel. In: Proc of the IEEE intconf on information reuse and integration (IRI 2003), October 27–29, 2003, Las Vegas, NV, USA, pp 243–249

Zhuang L, Dai H (2006) Parameter optimization of kernel-based one-class classifier on imbalance learning. J Comput 1(7):32–40

Christmann A, Luebke K, Rüping S, Marin-Galianos M (2005) Determination of hyperparameters for kernel-based classification and regression. Technical report 38/05, SFB475, University of Dortmund, Germany

Boardman M, Trappenberg T (2006) A heuristic for free parameter optimization with Support Vector Machines. In: Proc of IEEE int joint conf on neural networks (IJCNN 2006), July 16–21, 2006, Vancouver, Canada, pp 1337–1344

Cherkassky V, Mulier F (1998) Learning from data: concepts, theory, and methods. Wiley, New York

Ayat NE, Cheriet M, Suen CY (2002) Empirical error based optimization of SVM kernels: Application to digit image recognition. In: Proc of the 8th int workshop on frontiers in handwriting recognition (IWFHR’02), August 6–8, 2002, p 292

Fawcett, Tom. "ROC graphs: Notes and practical considerations for researchers." Machine learning 31.1 (2004): 1-38.

Hanley, James A., and Barbara J. McNeil. "The meaning and use of the area under a receiver operating characteristic (ROC) curve." Radiology 143.1 (1982): 29-36.

Hajian-Tilaki, Karimollah. "Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation." Caspian journal of internal medicine 4.2 (2013): 627.

Lisowsky, Petro. "Seeking shelter: Empirically modeling tax shelters using financial statement information." The Accounting Review 85.5 (2010): 1693-1720.

Fawcett, Tom. "An introduction to ROC analysis." Pattern recognition letters 27.8 (2006): 861-874.

Fürnkranz, Johannes, and Peter A. Flach. "Roc ‘n’rule learning—towards a better understanding of covering algorithms." Machine Learning 58.1 (2005): 39-77.

Hand, David J., and Robert J. Till. "A simple generalisation of the area under the ROC curve for multiple class classification problems." Machine learning 45.2 (2001): 171-186.

Zou, Kelly H. "Receiver operating characteristic (ROC) literature research." On-line bibliography available from:< http://splweb. bwh. harvard. edu 8000 (2002).

Fawcett, Tom. "An introduction to ROC analysis." Pattern recognition letters 27.8 (2006): 861-874.

W.-T. Pan, “A new fruit fly optimization algorithm: Taking the financial distress model as an example”, Knowledge-Based Systems, vol. 26, (2012), pp. 69-74.

Huang, Cheng-Lung, and Jian-Fan Dun. "A distributed PSO–SVM hybrid system with feature selection and parameter optimization." Applied Soft Computing 8.4 (2008): 1381-1391.

Provost, Foster J., and Tom Fawcett. "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions." KDD. Vol. 97. 1997.

Huang, Cheng-Lung, and Chieh-Jen Wang. "A GA-based feature selection and parameters optimizationfor support vector machines." Expert Systems with applications 31.2 (2006): 231-240.

Shen, Liming, et al. "Fruit fly optimization algorithm based SVM classifier for efficient detection of Parkinson's disease." International Conference in Swarm Intelligence. Springer International Publishing, 2015.

Chao, Chih-Feng, and Ming-HuwiHorng. "Theconstruction of support vector machine classifier using the firefly algorithm." Computational intelligence and neuroscience 2015 (2015).

A. Asunicion , D. Newman, UCI Machine Learning Repository2013,http://www.ics.uci.edu/~mlearn/MLRepository.html.

Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: a library for support vector machines." ACM Transactions on Intelligent Systems and Technology (TIST)2.3 (2011): 27.

Allah, Rizk M. Rizk. "Hybridization of Fruit Fly Optimization Algorithm and Firefly Algorithm for Solving Nonlinear Programming Problems." International Journal of Swarm Intelligence and Evolutionary Computation 5.2: 1000134.

Ye, Fei, Xin Yuan Lou, and Lin Fu Sun. "An improved chaotic fruit fly optimization based on a mutation strategy for simultaneous feature selection and parameter optimization for SVM and its applications." PloS one 12.4 (2017): e0173516.

Zhang, Qiantu, et al. "Parameters Optimization of SVM Based on Improved FOA and Its Application in Fault Diagnosis." JSW 10.11 (2015): 1301-1309

Jiang, Ai-hua, and Ni-xiao Liang. "Short-term load forecasting using support vector machine optimized by the improved fruit fly algorithm and the similar day method." Electricity Distribution (CICED), 2014 China International Conference on. IEEE, 2014.

Luo, Huixia, et al. "Mixed Fruit Fly Optimization Algorithm Based on Lozi's Chaotic Mapping." P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2014 Ninth International Conference on. IEEE, 2014.

Egan, James P. "Signal detection theory and {ROC} analysis." (1975).

Weston, Jason, and Chris Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, May, 1998.

Bredensteiner, Erin J., and Kristin P. Bennett. "Multicategory classification by support vector machines." Computational Optimization. Springer US, 1999. 53-79.

Crammer, Koby, and Yoram Singer. "On the algorithmic implementation of multiclass kernel-based vector machines." Journal of machine learning research 2.Dec (2001): 265-292.

Lee, Yoonkyung, Yi Lin, and Grace Wahba. "Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data." Journal of the American Statistical Association 99.465 (2004): 67-81.

Shen, Liming, et al. "Evolving support vector machines using fruit fly optimization for medical data classification." Knowledge-Based Systems 96 (2016): 61-75.

D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1997),pp.67–82.

D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley, 1989

J. H. Holland, Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence. Cambridge, MA: MIT Press, 1994.

Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 2nd ed. Berlin: Springer, 1994.

H.-N. Nguyen, S.-Y. Ohn, and W.-J. Choi, "Combined kernel function for support vector machine and learning method based on evolutionary algorithm," in Neural Information Processing: 11th International Conference - Proc. of ICONIP 2004, Calcutta, India, Nov 22-25, 2004. Heidelberg: Springer, 2004, pp.1273–1278, dOI 10.1007/b103766.

S.-Y. Ohn, H.-N. Nguyen, D. S. Kim, and J. S. Park, "Determining optimal decision model for support vector machine by genetic algorithm," in Computational and Information Science: First International Symposium - Proc. of CIS 2004, Shanghai, China,Dec 16-18, 2004. Heidelberg: Springer, 2004, pp. 592–597, dOI 10.1007/b104566.

S.-Y. Ohn, H.-N. Nguyen, and S.-D. Chi, "Evolutionary parameter estimation algorithm for combined kernel function in support vector machine," in Content Computing: Advanced Workshopon Content Computing – Proc. of AWCC 2004, ZhenJiang, JiangSu, China, Nov 15-17, 2004. Heidelberg: Springer, 2004, pp. 592–597, dOI 10.1007/b103383.

F. Friedrichs and C. Igel, "Evolutionary tuning of multiple svm parameters," Neurocomputing (in press), 2005.

D. Eads, D. Hill, S. Davis, S. Perkins, J. Ma, R. Porter, and J. Theiler, “Genetic algorithms and support vector machines for time series classification,” in Proc. of Fifth Conf. on the Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation. Symposium on Optical Science and Technology of the 2002 SPIE Annual Meeting, Seattle, WA, July 2002., 2002, pp. 74–85, lA-UR-02-6212.

Kennedy, J.; Eberhart, R.,“Particle Swarm Optimization”,Proceedings of IEEE International Conference on NeuralNetworks. IV. pp. 1942–1948, 1995.

J.H. Holland,"Genetic algorithms and the optimal allocationof trials, SIAM J. Comput. 2 (2), pp. 88–105, 1973.

Yang, X. S.,"Nature-Inspired Metaheuristic Algorithms",Frome: Luniver Press, 2008.

Yang, Xin-She, Suash Deb, and Simon Fong. "Accelerated particle swarm optimization and support vector machine for business optimization and applications." Networked digital technologies (2011): 53-66.

Zhang, Qiantu, et al. "Parameters Optimization of SVM Based on Improved FOA and Its Application in Fault Diagnosis." JSW 10.11 (2015): 1301-1309.

Mitić, Marko, et al. "Chaotic fruit fly optimization algorithm." Knowledge-Based Systems 89 (2015): 446-458.

Jiang, Minlan, et al. "A cuckoo search-support vector machine model for predicting dynamic measurement errors of sensors." IEEE Access 4 (2016): 5030-5037.

Yang X. Nature-inspired metaheuristic algorithms. Luniver Press; 2008.

Yang X-S, Deb S. Cuckoo search via levy flights. In: World congress on nature and biologically inspired computing; 2009. p. 210–4.

Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: overview and conceptural comparision, ACM Comput. Surv., 35(2), 268–308.

Yang X. S., (2010). Engineering Optimisation: An Introduction with Metaheuristic Applications, John Wiley and Sons.

Wolpert D. H. andMacreadyW. G. No free lunch theorems for optimization, IEEE Transaction on Evolutionary Computation 1997; 1(1): 67–82

Mitić, Marko, et al. "Chaotic fruit fly optimization algorithm." Knowledge-Based Systems 89 (2015): 446-458.

Wolpert D. H. and Macready W. G. Coevolutonary free lunches, IEEE Trans. EvolutionaryComputation 2005; 9(6):,721–735.

Yang, X. S. and Deb, S., (2009). Cuckoo search via L´evy flights, Proceeings of World Congresson Nature & Biologically Inspired Computing (NaBIC 2009), IEEE Publications, USA, pp.210–214.

Pavlyukevich I. (2007). L´evy flights, non-local search and simulated annealing, J. Computational Physics, 226, 1830–1844.

Yang X. S. and Deb S., (2010). Engineering optimization by cuckoo search, Int. J. Math.Modelling Num. Opt., 1 (4), 330–343 (2010).

Yang, X.S. and Deb, S. (2013). Multiobjective cuckoo search for design optimization, Computers and Operations Research, 40(6), 1616–1624 (2013).

Walton, S., Hassan, O., Morgan, K., and Brown, M.R. (2011). Modified cuckoo search: a newgradient free optimization algorithm. Chaos, Solitons & Fractals, 44(9), 710-718.

Chandrasekaran, K., and Simon, S. P., (2012). Multi-objective scheduling problem: hybrid appraoch using fuzzy assisted cuckoo search algorithm, Swarm and Evolutionary Computation,5(1), pp. 1-16.

Dhivya, M., Sundarambal, M., Anand, L. N., (2011). Energy efficient computation of data fusion in wireless sensor networks using cuckoo based particle approach (CBPA), Int. J. of Communications, Network and System Sciences, 4, No. 4, pp. 249-255.

Dhivya, M. and Sundarambal, M., (2011). Cuckoo search for data gathering in wireless sensor networks, Int. J. Mobile Communications, 9, pp.642-656.

Durgun, I, Yildiz, A.R. (2012). Structural design optimization of vehicle components using cuckoo search algorithm, Materials Testing, 3, 185-188.

Layeb, A., (2011). A novel quantum-inspired cuckoo search for Knapsack problems, Int. J.Bio-inspired Computation, 3, No. 5, pp. 297-305.

Moravej, Z., Akhlaghi, A., (2013). A novel approach based on cuckoo search for DG allocation in distribution network, Electrical Power and Energy Systems, 44, pp. 672-679.

Noghrehabadi, A., Ghalambaz, M., and Vosough, A., (2011). A hybrid power series – Cuckoo search optimization algorithm to electrostatic deflection of micro fixed-fixed actuators, Int. J. Multidisciplinary Science and Engineering, 2, No. 4, pp. 22-26.

Srivastava, P. R., Chis, M., Deb, S., and Yang, X. S., (2012). An efficient optimization algorithm for structural software testing, Int. J. Artificial Intelligence, 9 (S12), pp. 68-77.

Valian, E., Mohanna, S., and Tavakoli, S., (2011). Improved cuckoo search algorithm for feedforward neural network training, Int. J. Articial Intelligence and Applications, 2, No. 3, 36-43(2011).

Yildiz, A. R., (2012). Cuckoo search algorithm for the selection of optimal machine parametersin milling operations, Int. J. Adv. Manuf. Technol., (2012). doi:10.1007/s00170-012-4013-7.

Fan Wang, Xing-Shi He, Yan Wang, and Song-Ming Yang, (2012). Markov model and convergence analysis based on cuckoo search algorithm. Jisuanji Gongcheng/ Computer Engineering, 38 (11), 181–185.

Yang X. S., (2009). Firefly algorithms for multimodal optimization, in: Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, Vol. 5792, 169–178.

Yang X.-S., (2010). Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-inspired Computation, 2(2), 78–84.

Fister I., Fister Jr I., Yang X. S., Brest J., A comprehensive review of firefly algorithms, Swarm and Evolutionary Computation, (2013). http://dx.doi.org/10.1016/j.swevo.2013.06.001.

Yang X. S., (2008). Nature-Inspired Metaheuristic Algorithms, First Edition, Luniver Press, UK.

Sina K. Azad, Saeid K. Azad, Optimum Design of Structures Using an Improved FireflyAlgorithm, International Journal of Optimisation in Civil Engineering, 1(2), 327-340(2011).

[G. K. Jati and S. Suyanto, Evolutionary discrete firefly algorithm for travelling salesman problem, ICAIS2011, Lecture Notes in Artificial Intelligence (LNAI 6943), pp.393-403 (2011).

Holland JH. Adaptation in natural and artificial systems. Michigan Press; 1975.

Ye, Fei, Xin Yuan Lou, and Lin Fu Sun. "An improved chaotic fruit fly optimization based on a mutation strategy for simultaneous feature selection and parameter optimization for SVM and its applications." PloS one 12.4 (2017): e0173516.

Alcalá-Fdez, Jesús, et al. "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework." Journal of Multiple-Valued Logic & Soft Computing 17 (2011).

Tharwat, Alaa, and Thomas Gabel. "Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm." Neural Computing and Applications (2019): 1-14

Mirjalili S, Mirjalili SM, Lewis A "Grey wolf optimizer" Adv Eng Softw(2014) 69:46–61

Tharwat A, Hassanien AE, Elnaghi "A ba-based algorithm for parameter optimization of support vector machine."BE Pattern Recognit Lett(2017) 93:13–22