International Journal of Communication Networks and Information Security
2024, 16(4)
ISSN: 2073-607X, 2076-0930
https://https://ijcnis.org/ **Research Article**

Enhanced SDD Algorithm Optimization Technique For Finding Hyper Parameter Of SVM

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

[1,2]JLU School of Engineering & Technology, Bhopal, India

Email: [1]ashish1namdeo@gmail.com, [2]dileep.singh@jlu.edu.in

ARTICLE INFO

Received: 28 Apr 2024
Accepted: 04 Sep 2024

ABSTRACT

Background: It is crucial to pay attention to the classify data. The classification of data via Support Vector Machine (SVM) approach has severe restrictions. Corresponding to this, the intriguing improvements could not be accomplished without a suitable Support Vector Machine (SVM) classifier improvement and it is of high significance to build a machine learning model which can accurately classify the data. In this paper, an enhanced framework is proposed mainly used for classifying the data by introducing a hyperplane.
Objective: The most important aspect of this whole framework is to create an enhanced version of recently developed evolutionary algorithm known as Social Ski Driver (SSD) optimization. As far as we know, enhanced version of SDD optimization algorithm have not yet used in SVM hyperparameter optimization to classify data.
Methods: We, improvise Social Ski Driver (SSD) exploitation ability, with Levy flight. To verify this, the proposed method is then applied to balanced, imbalance and multiclass datasets with higher dimensionality from the UCI repository then empiercally compared with Grid Search, PSO and SSD-SVM.
Results: The result achieved shows that ESSD-SVM is capable of finding, optimal solution and better performance classification as compared with other approaches
Conclusion: The proposed ESSD-SVM model's effectiveness is demonstrated by its accuracy that indicates that it optimizes classification performance for hybrid models, which takes less time.

Keywords: Particle swarm optimization, Ski-driver algorithm, Metaheuristic Algorithm, Support vector machines, Grid Search Evolutionary optimization.

INTRODUCTION

Support Vector Machines are primarily used in area of classification and regression problems [1] [2]. They solve problem by creating a boundary among them. It uses kernel trick for complex data transformation from higher dimensional space based on input and output parameters [2].

Hence, we should carefully choose kernel parameter values as classifier performance is majorly dictated by its value. Searching of all the possible values of parameter, could take polynomial time [2] problem, considering all the possible subset of values. The search can also be considered at random. But we often might end up with a solution with is not optimal. Thus, to obtain optimal solution heuristic or evolutionary optimization comes handy.

Support vector machine algorithm takes training set $\{X_i\}$ as an input and output $\{Y_i\}$ vector whose value is {-1, +1} known as class labels. Now, what this algorithm does it tries to search for a hyperplane which is defined mathematically as (W.X) + B = 0 Here W is perpendicular to hyper plane, B is called as Bias.

This hyperplane is obtained by finding minima of $\frac{1}{2}|W|^2$ under the constraint

$Y_i(WX_i + B) \geq 1$; where i is a natural number

After solving that minimization problem to find hyperplane we get a quadratic equation.

$Q = \sum\alpha_i - \frac{1}{2}\sum\alpha_i\alpha_j Y_i Y_i X_i X_j$

Where i and j are natural number from 1 to n

Q is maximized under the condition $\sum\alpha_i Y_i = 0$ and $C \geq \alpha_i \geq 0$

C is called as penalty whose value is a balance between training error and the margin width.

And width is calculated as $W = \sum\alpha_i Y_i X_i$

Hence on solving dual optimization we get following decision boundary

$\text{sign}(\sum\alpha_i Y_i (X_i X) + B)$

This is applicable for linear classification but for non-linear classification above equation is updated as follows

$\text{sign}(\sum\alpha_i Y_i K(X_i X) + B)$

Where $K(X_i X)$ is transformation function

We will use Radial Basis Function (RBF) $e^{\frac{|x_i - x|^2}{\sigma^2}}$ kernel. for our classification. Where σ is the length of the width of the hyperplane separating different classes of data.

Evolutionary algorithm optimization strategy is derived from the biological paradigm which aims in finding best possible results [3] It is simple yet effective. Evolutionary computation has been used in various areas such as scientific and industrial research [4]. Over the recent years many Evolutionary Algorithms had been developed such as particle swarm optimisation (PSO) [5] which mimics food searching behaviour of swarms of birds. Ant colony optimization (ACO) [6][39] which is inspired by ants behaviour to find food by following the pheromone lay down by them for directing each other.

However, these optimization techniques have some cons such as PSO algorithm could get stuck at local optima or ACO performance might get affected due to slow convergence rate.

In this paper we have proposed a variation to Ski Driver Algorithm [37], which has shown considerably a better result than other optimization algorithm as mentioned in literature survey. To improve the Ski Driver's optimization performance, we enhanced its solution exploration capability by using levy Flight [7] mechanism, so that it will not fall into local optima. And these levy flight jumps not only improve solution exploration capability, but also improve its convergence rate as shown in empirical analysis.

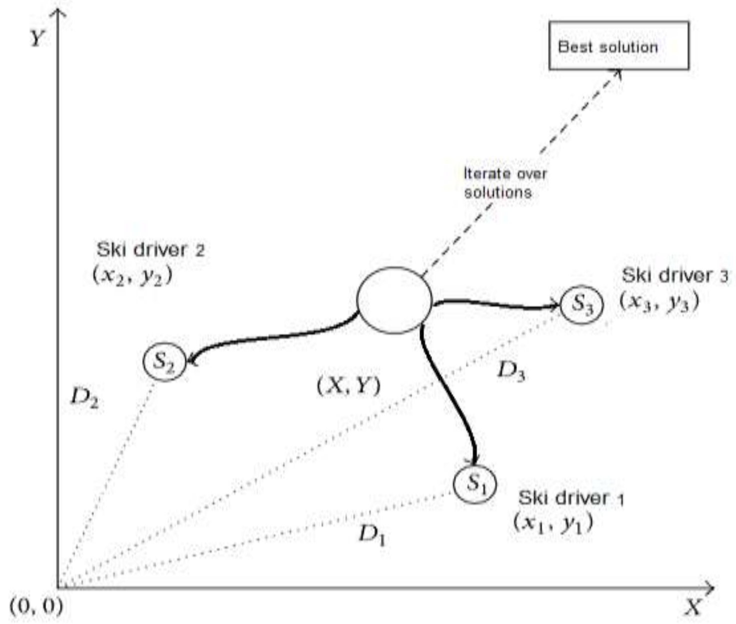

Fig. 1 Ski drivers agents position to find best solution.

Fig. 2 the figure depicts the movement of three Ski drivers agents in solution space. S1, S2 and S3 are ski drivers agents moving towards the average of the best three solutions. S1, S2 and S3 travel in a nonlinear direction to find best solution.

Levy flights follow the paths of intervenient free exploration, and its effectiveness [8][9][10][11][12] has been proven in finding optimal pattern of various problem, we can set a threshold to follow some levy flight exploration at this rate along with Ski driver's solution exploration pattern, thus creating a hybrid approach.

The main highlights of this paper's contribution are.

1. Enhanced Ski driver's algorithm has been used for first time in SVM.
2. Enhancing Exploration capability of Ski Driver's Algorithm by using Levy Flight Algorithm.
3. Using threshold to control exploration pattern by using two functions, for instance like SSD function and Levy flight function for developing Enhance SDD.
4. Validating proposed model again different data types for instance balanced imbalance and multiclass datasets.

The remainder of the document is arranged accordingly: Section II gives us the literature review regarding Evolutionary Algorithm with regards to finding optimal solution. In Section III we describe working of proposed ESDD Algorithm. While its result and analysis has been given has been mentioned at Section IV. Section V discusses about the future direction of this research and its applicability, along with its conclusion.

LITERATURE SURVEY

Selection of kernel parameters is an optimization problem, with the objective in mind to enhance classifier Performance. Many articles have been published in the literature whose main goal is to find optimal solution. Like [13] discusses about the challenges for finding the optimum value of C and gamma which also suggested to use evolve method to find a better solution.

As described here, evolutionary algorithms are prevalent in this domain because they are agile and have the capability to escape local optima.

Since the inception of Genetic Algorithm (GA) [14], is popularly being used in finding optimal solution. With respect to SVM it has been used to fine tune the parameters of SVM.

Then, we had developed hybrid version Genetic Algorithm as an advance Genetic-Quasi-Newton [15], where the agent based which is used to minimize the edge to traverse, by leaving one out error.
Particle Swarm Algorithm (PSO) [16] is a powerful technique, which has a wide variety of application, such as an optimizer function, in Artificial intelligence and Fuzzy system. In regard to optimize C, gamma parameters we observe in, lagrangian multiplier has not been considering in PSO. Thus, introducing need to consider need to be taken that in to account while optimizing hyperparameter.
This algorithm was then used by [17], as he observed that this algorithm might stuck in local optima, and he also considers lagrangian multiplier, thus introducing, Firefly Algorithm in order to simultaneously optimize them and uses bioluminescent property to explore solution. That finds optimal value. While In [18], they consider Dynamic error measurement forecasting technique in terms of MAPE and RSME, to find C and gamma values.
A recent study [10] shows us that in some, evolutionary algorithm population can shift 90 degree after an examination of exploration space, which then follows a pattern of levy flight, to explore the optimal solution, whose effectiveness has already been seen in various areas [8][9][10][11][12].
This paper [35] discusses about the Challenges in evolutionary algorithms in context of machine learning for instance running time, convergence, and parameters values. Along sides its various application for example to find the optimum value of machine learning algorithm parameters and feature selection.
As there are a lot of evolutionary algorithms, it poses the question about its effectiveness.
According to No Free lunch theory [19], no single solution is possible for all the optimizations; with the emergence of new algorithm there exists a possibility of achieving a better mechanism that will provide trade off rate between exploration and exploitation. But these are domain sensitive, as NFL states that we should focus on problem which is currently at hand along with its assumptions. And if we have a multi-objective function then we need to simultaneously address those entire objectives which are rarely possible. Hence, we use heuristic approach in finding optimal solution.
This inspires us to enhance and develop hybrid meta-heuristic algorithm based on Levy flight.
SSD itself has been motivated by PSO [17] gray wolf optimization [20] and sine cosine algorithm [21], to find optimum solution.
Following is the contribution of this paper.

1. To create a solution for the problem this would be optimum with faster convergence rate.
2. To address the viability the proposed methodology with variety number of data set with already established benchmark.
3. To find a balance rate between the exploration and exploitation rate with the help of levy flight.

SSD has been used recently in many areas like this paper [22] discusses about drugs toxicity levels analysis via rough set to deduce number of features SSD is also used in signature authentication [23] in conjunction with deep convolutional neural network, basically its used to optimize weighting function that has been build on the foundation of classification algorithm, with high precision and sensitivity. Another example is power restoration system with Voltage Source Converter based High Voltage Direct Current (VSC-HVDC) in which [24] where SSD along with Deep Neural Network is used with VSC-HVDC to restore power faster and gracefully in comparison to conventional PI conte Proportional and Integral, controller.

A. Current Work

1)Social Ski Driver Algorithm.

Social Ski Driver Algorithm is inspired by the nature's evolutionary phenomena. As Ski Driver exploration pattern mimics this process while descending hill.

Following parameters use to mathematically describe the process.

Agents Position ($x_i^{R^n}$): Location of agent helps us in calculating the objective function value in N dimensional space

Last best position (P_i): This Defines an agent's cost function, which is used to compare the best position of the other agent. And its location is then preserved as the best location in the region.

Mean global best (M_i): It represents the Global best, obtained by taking mean of top three Solution in terms of fitness achieved so far.

Symbols	Meaning
$\mathbf{V_i}$	Agents Velocity at X_i position
$\mathbf{X_\alpha, X_\beta}$ **and** $\mathbf{X_\gamma}$	Best 3 solutions
$\mathbf{r_1, r_2}$	Random number between [0, 1]
$\mathbf{P_i}$	Agent's local best position
$\mathbf{M_i}$	Global best position
$\mathbf{c}$	constant value trade-off between exploitation and exploration rate
$\mathbf{t}$	current iteration
$\boldsymbol{\alpha}$	Value of α reduce c

$M_i = \frac{X_\alpha + X_\beta + X_\gamma}{3}$	(1)
$X_i^{t+1} = X_i^t + V_i^t$	(2)
$V_i^{t+1} = \begin{cases} \text{when } r_2 > \frac{1}{2} \text{ then } c.\cos r_1 (P_i^t - X_i^t) + c.\cos r_1 (M_i^t - X_i^t) \\ r_2 \leq \frac{1}{2} \text{ then } c.\sin r_1 (P_i^t - X_i^t) + c.\sin r_1 (M_i^t - X_i^t) \end{cases}$	(3)
$c^{t+1} = \alpha c^t$	(4)
$\alpha \in (0,1)$	(5)

Sine and cosine function in Equation 3 gives a better exploration capability, along with guided search.

2). Levy Flight

Levy Flight [25][26][27][28][29][30][31] is defined as a random distribution of number over non-Gaussian search space, from levy stable distribution. In that we have two components, random direction and step length, the step length comes from probability distribution function over levy distributionWhile random direction is governed by levy stable distribution which follows power law frequency, defined as in equation 6.

Symbols	Meaning
u,**v**	derived from normal distribution
B	Index between (0, 2)
S	Step length

$L(s) \sim \lvert s \rvert^{-1-\beta}$	(6)
$S = \dfrac{u}{\lvert v \rvert^{1/\beta}}$	(7)
$u \sim N(0, \sigma_u^2), \qquad v \sim N(0, \sigma_v^2),$	(8)
Where $\sigma_u = \left\{ \dfrac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left[\frac{1+\beta}{2}\right]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}$	(9)
Hence $\text{stepsize} = 0.01 \times S$	(10)

In which 0.01 is multiplier factor, as S/100 is the normal length, else we will aggressively do levy flight search, dropping off some solution outside domain.

THE PROPOSED NEW ENHANCED SKI DRIVER ALGORITHM.

Initial stage involves pre-processing of data, where we normalize feature in the range 0 to 1. This not only improves classification performance but also help us to realize the important feature. Then we use SMOTE Algorithm to create balance among various classes of data. The list of parameters involving optimization decides the dimension of search space.

Agents positions are initialized randomly and updated as per the equation 2, if random probability has value more than 0.5, else will be updated by our designed position update equation.

Symbol	Meanings
$\mathbf{X_i^t}$	Position of ith particle at t iteration
$\otimes$	Element wise multiplication
$\mathbf{V_i^t}$	Agents Velocity at X_i position
$\mathbf{Levy\_walk}(\mathbf{X_i^t})$	Stepsize

By using levy flight we can able to make a leap, to explore other optimal solution, after reaching a particular threshold. This will facilitate global exploration within search space.

$X_i^{t+1} = V_i^t + \omega * \text{Levy}_{\text{walk}}(X_i^t)$	(11)
Where $\omega = 0.1 + 0.8 \times \left(1 - \dfrac{\text{current}_{\text{iteration}}}{\text{total}_{\text{iteration}}}\right),$	(12)
And $\text{Levy\_walk}(X_i^t) = X_i^t + \text{step} \otimes \text{random}\left(\text{size}(X_i^t)\right)$	(13)
Where $\text{step} = \text{stepsize} * X_i^t$	(14)

stepsize Value is derived from Equation (13).

While number of agents that are going to participate, is predetermined by user

Steps	Description
Normalization	data is trasformed so that all features are scaled within same range, for further processing by SVM.
Data Partition	Data is divided into equal size subsets such that they do not overlap
Training dataset	Data set used to train SVM

Validation dataset	Dataset used to tune hyperparameters
Testing Dataset	Dataset used to evaluate the model
Fitness Evaluation	Evaluates the proximity of solution to desired optimal solution

.

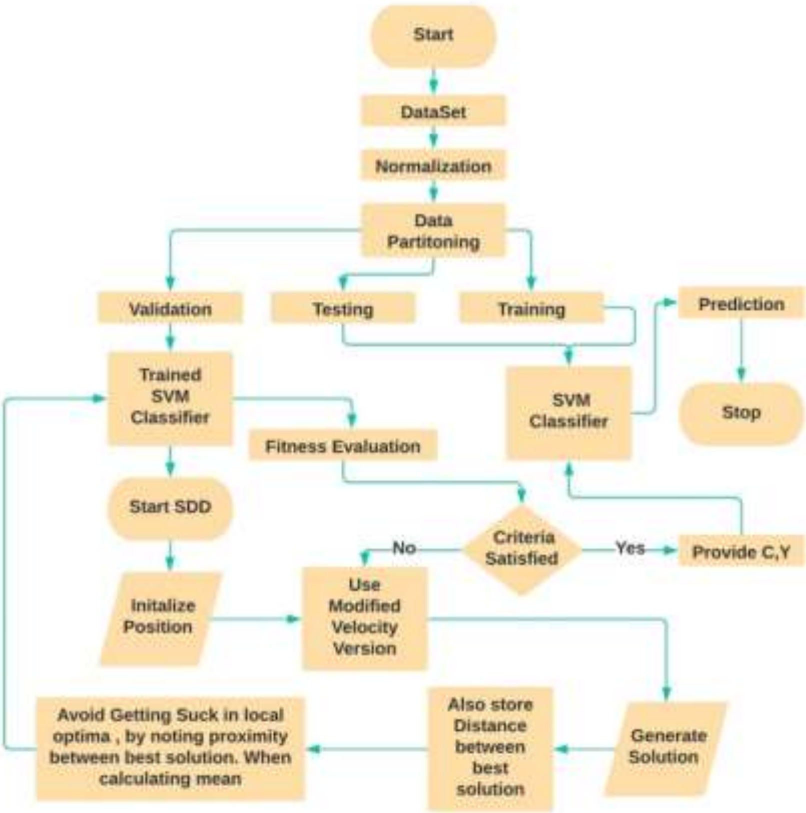

Fig.2. Flow Chat of Enhanced SDD based Optimization method for finding Hyperparameter of SVM

The above figure 2 depicts the optimization of SVM parameters by proposed method. The optimization process is divided in two phases. In first phase we will find the value of C, gamma by calculating and updating fitness value, with the help of proposed model. Thus, we will be able to get better fitness value by using levy flight mechanism to escape local minima. While in second phase we will validate the result obtained in first phase. These phases are same as that of SDD algorithm with only modification applied as per the above stated definition, of optimization SDD algorithm. The pseudo code of ESDD is shown below

Algorithm 1

ESDD for parameter optimization.

1: Initialize the agents with random positions, Max_Iter, Pop_Size
2: Generate the velocity for agents
3: Evaluate fitness function value
4: iteration=1
5: while (iteration < Max_Iter) do
6:$\omega = 0.1 + 0.8 \times \left(1 - \frac{\text{current}_{\text{iteration}}}{\text{total}_{\text{iteration}}}\right)$
7: for j=1 to Pop_Size do
8: if rand ( ) < 0.5 then

```
Update agent's velocity and position by Equation (2) and (3) respectively
    else
    Update agent's velocity and position by Equation (2) and (11) respectively
    end if
    Position values are brought to the boundary value when its values are moved out of the boundary
    Compute the fitness function for new agent X_j
    if fit(i) < fitbest then
    select k best agents through Classifier's Evaluation and store its value.
    end if
  end for
Record the best solution
iteration=iteration+1
end while
Output the best solution
```

The ESSD algorithm is effective in preserving equilibrium between exploration and exploitation and avoids premature convergence and local optima. Such advantages are an additional add-on to find global optima for SVM.

A. Data Pre-processing Stage

In this stage we first change all the nominal features to numerical values followed by normalizing those numerical values in the range [0, 1] so that they can follow normal distribution curve having mean $\mu=0$ and standard deviation $\sigma=1$.

Now, the data set is divided into three segment that is testing, training and validation. During validation phase algorithm searches the best value of C and gamma based on fitness value, then uses it to train model then test it.

B. Fitness Computation

Since, our model will be trained against imbalance class data. This could lead us to erroneous inference as accuracy may not be able to differentiate between correlated labels which belong to other class. Thus, the objective of our model will be to maximize sensitivity.

$$\mathrm{fitness} = -S = \frac{TP}{TP + FN} \qquad (15)$$

The optimum value of C and gamma will be obtained at position where the sensitivity is maximum.

I. Experiment result and observation

In this part, we will assess our proposed algorithm and contrast it with the state-of-the-art SDD SVM. algorithm along with PSO-SVM [17] and BA-SVM [32]. The algorithm is implemented in python3 [33],

Matplotlibtools.py [34] library was used to create graph to analyze the result.

A. Dataset Used

In our work we have used 8 benchmark dataset from KEEL[1] to access the performance of ESSD SVM. The variance in dataset can be seen as the data which is having imbalance ratio more than 9 while other less than that, here imbalance ratio means magnitude of majority class instances present every minority group. As mentioned in table 1.

We will use K folds cross validation method to evaluate our model, by dividing dataset into K sub groups. Out of those group one group will be used in used in validation and one in testing phase while the rest group, will be used to train our model. This process will be repeated K times, then in the end we compute the result by taking average of every iteration along with standard deviation.

The criteria used to access the suitability of our proposed model are sensitivity, specificity and Area Under the curve. That determines the proportion of true positives or true negatives that are correctly classified by our classifier in terms of sensitivity and specificity respectively.

TABLE 1 Datasets Property

Dataset				No. of Instances	Attributes No.	Imbalance Ratio			
Ecoli2	Ecoli3			336	7	5.46	8.6		
Glass0	Glass6	Glass4	Glass2	214	9	2.06	6.38	15.46	11.59
WineQuality_white-9_vs_4				168	11	32.6			
Poker Hand				244	10	29.5			

B. Parameters optimization

In this part, we will equate the efficiency of the proposed method to those of other approaches and then analyze the result obtained. To test the statistical significance of the result, Wilcoxon Signed rank test is done, which is a non-parametric test for pair wise data, based on ranking and considers the sign and magnitude of differences between absolute values. In which null hypothesis is no difference between data pair. We reject null hypothesis when p value is less than 0.005 means the difference statistically significant. Now let's first we investigate Grid search [38] method, as we can observe the table 2 the sensitivity of proposed model is better in comparison to Grid search. Followed by Wilcoxon Signed rank test expect Glass6 data set, rest having statistical significance value less than 0.005. Apart from this we also see that Grid search computation time is higher as compared proposed method. This is because in case of Grid Search the computation increases exponentially when number of searching parameter increase or searching range is increased.

When we compare to PSO-SVM and BA-SVM with ESDD-SVM, we see that proposed method performs better in terms of specificity, sensitivity and AUC but also observe that statistically significant value is more than 0.005 for Glass6 and Glass4 in case PSO while in case of BA significant value is more than 0.005 for Ecoli2, Glass0 and Glass4.

Now we compare SDD with ESDD, out of 8 dataset 3 dataset has similar specificity, sensitivity and AUC measures, while the rest have significantly higher value of specificity and AUC with statistical significance value less than 0.005. This shows that ESDD outperforms the stated methods.

Table 2 Comparison between The Essd-Svm And Grid Search Svms Algorithms In Terms Of Sensitivity, Specificity, And Auc

Dataset	IR	Grid Search SVMs			ESSD-SVM			p value for Wilcoxon testing		
		Sen.	Spc.	AUC	Sen.	Spc.	AUC	Sen.	Spc.	AUC
D1	5.46	0.90 ± 0.08	0.93 ± 0.03	0.92 ± 0.04	0.98 ± 0.05	0.95 ± 0.02	0.97 ± 0.06	<0.005	<0.005	<0.005
D2	8.6	0.88 ± 0.08	0.91 ± 0.03	0.89 ± 0.05	0.93 ± 0.06	0.96 ± 0.04	0.94 ± 0.06	<0.005	<0.005	<0.005
D3	2.06	0.85 ± 0.10	0.90 ± 0.02	0.88 ± 0.05	0.96 ± 0.04	0.91 ± 0.01	0.93 ± 0.02	<0.005	<0.005	<0.005
D4	6.38	0.76 ±	0.81	0.80	0.79	0.85	0.83	0.0054	<0.005	<0.005

		0.05	± 0.04	± 0.04	± 0.14	± 0.08	± 0.1			
D5	15.46	0.89 ± 0.13	0.97 ± 0.02	0.95 ± 0.06	0.94 ± 0.02	1 ± 0	0.96 ± 0.02	<0.005	<0.005	<0.005
D6	11.59	0.96 ± 0.02	0.97 ± 0.02	0.96 ± 0.02	0.98 ± 0.01	0.99 ± 0.06	0.99 ± 0.04	<0.005	<0.005	<0.005
D7	32.6	0.93 ± 0.04	0.97 ± 0.02	0.95 ± 0.03	0.97 ± 0.02	1 ± 0	0.98 ± 0.06	<0.005	<0.005	<0.005
D8	29.5	0.84 ± 0.05	0.88 ± 0.06	0.85 ± 0.04	0.87 ± 0.07	0.98 ± 0.04	0.95 ± 0.06	<0.005	<0.005	<0.005

In this table 2 we not only see that ESSD have been outperformed in terms of AUC but also had shown significance improvement in terms of sensitivity as well for all D1 to D8 dataset which would help us in finding local optima with better prediction.

TABLE 3 Comparison between the ESSD-SVM and PSO-SVM algorithms in terms of sensitivity, specificity, and AUC

Dataset	PSO-SVM			ESSD-SVM			p value for Wilcoxon testing		
	Sen.	Spc.	AUC	Sen.	Spc.	AUC	Sen.	Spc.	AUC
D1	0.97 ± 0.03	0.93 ± 0.02	0.95 ± 0.03	0.98 ± 0.05	0.95 ± 0.02	0.97 ± 0.06	0.0064	<0.005	0.0051
D2	0.91 ± 0.04	0.88 ± 0.06	0.89 ± 0.05	0.93 ± 0.06	0.96 ± 0.04	0.94 ± 0.06	<0.005	<0.005	<0.005
D3	0.90 ± 0.03	0.88 ± 0.02	0.90 ± 0.03	0.96 ± 0.04	0.91 ± 0.01	0.93 ± 0.02	<0.005	<0.005	<0.005
D4	0.73 ± 0.08	0.79 ± 0.06	0.78 ± 0.06	0.79 ± 0.14	0.85 ± 0.08	0.83 ± 0.1	<0.005	0.006	<0.005
D5	0.94 ± 0.03	1 ± 0.00	0.97 ± 0.01	0.94 ± 0.02	1 ± 0	0.96 ± 0.02	0.0062	0.0059	0.0058
D6	0.96 ± 0.02	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.99 ± 0.06	0.99 ± 0.04	<0.005	<0.005	<0.005
D7	0.95 ± 0.03	1 ± 0.00	0.97 ± 0.02	0.97 ± 0.02	1 ± 0	0.98 ± 0.06	<0.005	<0.005	<0.005
D8	0.84 ± 0.06	0.86 ± 0.04	0.85 ± 0.04	0.87 ± 0.07	0.98 ± 0.04	0.95 ± 0.06	<0.005	<0.005	<0.005

Form the above table 3, it's clear that we get so see a lot of improvement in AUC for ESSD as compared to PSO with is under statistically significant as proven by Wilcoxon test.

TABLE 4 Comparison Between The ESSD-SVM And BA-SVM Algorithms In Terms Of Sensitivity, Specificity, And AUC

Dataset	BA-SVM			ESSD-SVM			p value for Wilcoxon testing		
	Sen.	Spc.	AUC	Sen.	Spc.	AUC	Sen.	Spc.	AUC
D1	0.98 ±	0.94 ±	0.96 ±	0.98 ±	0.95 ±	0.97 ±	0.0063	0.0058	0.0064

	0.02	0.03	0.03	0.05	0.02	0.06			
D2	0.91 ± 0.03	0.89 ± 0.05	0.9 ± 0.04	0.93 ± 0.06	0.96 ± 0.04	0.94 ± 0.06	<0.005	<0.005	<0.005
D3	0.92 ± 0.02	0.9 ± 0.03	0.91 ± 0.03	0.96 ± 0.04	0.91 ± 0.01	0.93 ± 0.02	0.006	<0.005	<0.005
D4	0.72 ± 0.05	0.78 ± 0.07	0.76 ± 0.06	0.79 ± 0.14	0.85 ± 0.08	0.83 ± 0.1	<0.005	<0.005	<0.005
D5	0.94 ± 0.04	1 ± 0	0.96 ± 0.03	0.94 ± 0.02	1 ± 0	0.96 ± 0.02	0.0061	0.006	0.0059
D6	0.97 ± 0.04	0.98 ± 0.02	0.98 ± 0.03	0.98 ± 0.01	0.99 ± 0.06	0.99 ± 0.04	<0.005	<0.005	<0.005
D7	0.94 ± 0.02	0.99 ± 0.01	0.97 ± 0.04	0.97 ± 0.02	1 ± 0	0.98 ± 0.06	<0.005	<0.005	<0.005
D8	0.85 ± 0.05	0.88 ± 0.04	0.86 ± 0.05	0.87 ± 0.07	0.98 ± 0.04	0.95 ± 0.06	<0.005	<0.005	<0.005

In table 4, we see that there is some gain in overall AUC with respect to all data set but for some data like D1 and D3 we observe there is not much improvement in terms of sensitivity for ESSD

TABLE 5 Comparison between the SSD-SVM and ESSD-SVM algorithms in terms of sensitivity, specificity, and AUC

Dataset	SSD-SVM			ESSD-SVM			p value for Wilcoxon testing		
	Sen.	Spc.	AUC	Sen.	Spc.	AUC	Sen.	Spc.	AUC
D1	0.97 ± 0.05	0.94 ± 0.02	0.95 ± 0.04	0.98 ± 0.05	0.95 ± 0.02	0.97 ± 0.06	0.0063	0.0058	0.0064
D2	0.92 ± 0.08	0.94 ± 0.04	0.91 ± 0.06	0.93 ± 0.06	0.96 ± 0.04	0.94 ± 0.06	<0.005	<0.005	<0.005
D3	0.92 ± 0.04	0.91 ± 0.02	0.92 ± 0.03	0.96 ± 0.04	0.91 ± 0.01	0.93 ± 0.02	0.006	<0.005	<0.005
D4	0.78 ± 0.12	0.82 ± 0.08	0.81 ± 0.1	0.79 ± 0.14	0.85 ± 0.08	0.83 ± 0.1	<0.005	<0.005	<0.005
D5	0.94 ± 0.02	1 ± 0	0.96 ± 0.02	0.94 ± 0.02	1 ± 0	0.96 ± 0.02	0.0061	0.006	0.0059
D6	0.98 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.98 ± 0.01	0.99 ± 0.06	0.99 ± 0.04	<0.005	<0.005	<0.005
D7	0.97 ± 0.03	1 ± 0	0.98 ± 0.01	0.97 ± 0.02	1 ± 0	0.98 ± 0.06	<0.005	<0.005	<0.005
D8	0.86 ± 0.07	0.92 ± 0.04	0.87 ± 0.06	0.87 ± 0.07	0.98 ± 0.04	0.95 ± 0.06	<0.005	<0.005	<0.005

As we observe table 5, we notice that ESSD have gained some performance in terms of AUC as well as specificity and sensitivity by some nominal points.

The explanation behind this is as follows.

1. The agents move towards mean of top three best positions, even if there is a possibility of getting stuck at local minima it will take a leap towards other search space and check whether if it's still better just before convergence and update location accordingly. While in case of PSO and BA they move towards global best or previous best location without considering other option, hence there exists a possibility of trapping in local optima.

2. The agents in BA and PSO follows straight path, whereas in case of SSD agents have flexibility to change the direction of exploration as it uses Sin and Cosine function to update velocity. In case of ESSD apart from following Sine and Cosine path it also follows a shot path followed by sudden shift of 90° at regular intervals.
3. The parameters in PSO algorithm needed to be determined in advance, but in BA and SSD parameters are updated iteratively, thus able to escape local solution better.

It's worth mentioning that for imbalanced data it performs better for higher IR ratio, which shows resilience towards imbalance dataset. It also maintains the balance between sensitivity and specificity. Also, the value of hyperparameters changes for every dataset as it depends on the dimensionality and transformation applied which changes the distance between data pairs.

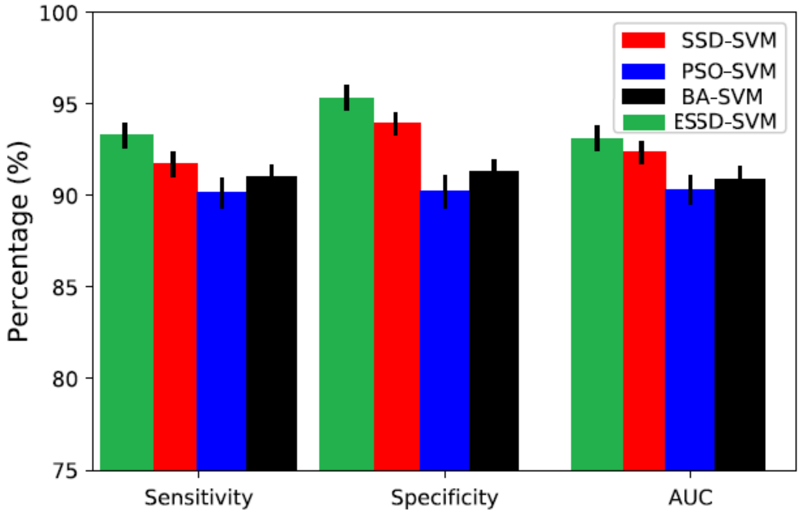

Fig.3. Comparison graph of optimization algorithms

Now to check its performance under different dataset, we have used the following data set where as above figure 3 represent comparison in terms of sensitivity, specificity and AUC.
1. Balanced dataset: We are using Iris and Liver data dataset from UCI repository and have been compared to paper [36] result to seek its effectiveness in case of balanced data
2. High dimensional dataset: In this we have Sonar dataset from UCI repository with more features as compared to other dataset to test its suitability among higher dimensional data. Table 6 shows its potential to perform better in this scenario as well. Result obtained compared with result in [17].
3. Multiclass dataset: For this Iono data set obtained from UCI repository which has of three classes. To classify these classes, we had slightly modified SMOTE part, which is used in binary classifier to balance data, by generating minority class data. In multiclass problem we identify majority class data one with highest sample, then we generate minority class data to over sample data. The result then compared with result in paper [36]

TABLE 6 ESSD Test on different dataset.

Datasets	Classes	Dim No.	Sen.	Spc.	AUC	Previous results
Iris	(50/50)	4	1 ± 0	1 ± 0	1 ± 0	Acc. (100%)
Liver	(145/200)	6	81.07 ± 0.9	78.18 ± 2.24	76.13 ± 1.84	Acc. (78.7%)
Sonar	(97/111)	60	88.47 ± 1.05	89.54 ± 1.18	88.23 ± 0.97	Acc. (88.3%)
Iono	(59/71/48)	13	93.12 ± 2.14	94.22 ± 0.82	93.73 ± 1.23	Acc.

						(97.94%)

From Table 6 result we observe that our algorithm performs better as compared to previous result in terms of AUC for those four different data sets. On running proposed algorithm, across multiple datasets gives satisfactory performance. But this does not guarantee always because several factor like distance between samples, ratio of majority and minority classes and dimensionality of data, play a key role in determining the performance of classifier

CONCLUSION AND FUTURE DIRECTION

In our research we had built a new metaheuristic algorithm based on SSD algorithm. To enhance the exploration ability, we use Levy flight mechanism, for updating path. The hyperparameter selection problem is used which is then formatted as multi-objective optimization with fitness value to achieve better performance in terms of sensitivity. We have applied this algorithm on 11 different datasets obtained from KEEL and UCI repository and compared the result with other state of art algorithms, which shows that proposed method works better when incorporated with Levy flight mechanism. Thus, we conclude that ESSD works effectively to find optimal value of hyperparameters in comparison to other methods. As the working of ESSD method depends on levy flight which may require some optimal parameter setting, this can be considered as its limitation. Also, as per No free lunch [19] theory this method is not guaranteed to provide best result for all datasets. This method could be also applied with another classifier as well for instance random forest search. And it can be hybridized with another metaheuristic algorithm or any other optimization algorithm as well.

Author Contributions: Ashish Namdeo is responsible for: Conceptualization, Methodology, Writing - Original Draft, Writing - Review & Editing, , Software, Investigation, Data Curation, Writing - Original Draft, Investigation, Data Curation Dr. Dileep Singh is responsible for: Supervision, reviewing research work.
All authors have read and agreed to the published version of the manuscript.

Funding: This research received no specific grant from any funding agency.
Conflicts of Interest: The authors declare no conflict of interest.
Data Availability: The dataset for this are obtained from KEEL and UCI repository and source code cannot be shared openly publically because since paper is not published yet and others might use that code which could result in loss of copyright for my work since other may publish same work.
ORCID:
First Author: https://orcid.org/0000-0002-8723-0284

REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks", Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.
[2] L. Wang, Support vector machines. Berlin: Springer, 2005.
[3] X. Yang, "Swarm intelligence based algorithms: a critical analysis", Evolutionary Intelligence, vol. 7, no. 1, pp. 17-28, 2013.
[4] M. Lozano, D. Molina and F. Herrera, "Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems", Soft Computing, vol. 15, no. 11, pp. 2085-2087, 2010.

[5] Z. Liu, C. Mao, J. Luo, Y. Zhang and C. Philip Chen, "A three-domain fuzzy wavelet network filter using fuzzy PSO for robotic assisted minimally invasive surgery", Knowledge-Based Systems, vol. 66, pp. 13-27, 2014.
[6] M. Reed, A. Yiannakou and R. Evering, "An ant colony algorithm for the multi-compartment vehicle routing problem", Applied Soft Computing, vol. 15, pp. 169-176, 2014.
[7] P. Barthelemy, J. Bertolotti and D. Wiersma, "A Lévy flight for light", Nature, vol. 453, no. 7194, pp. 495-498, 2008.
[8] X.-S. Yang and S. Deb, "Eagle Strategy Using Lévy Walk and Firefly Algorithms for Stochastic Optimization," Nature Inspired Cooperative Strategies for Optimization, vol. 284, pp. 101-111, 2010.
[9] I. Pavlyukevich, "Lévy flights, non-local search and simulated annealing", Journal of Computational Physics, vol. 226, no. 2, pp. 1830-1844, 2007.
[10] A. Reynolds and M. Frye, "Free-Flight Odor Tracking in Drosophila Is Consistent with an Optimal Intermittent Scale-Free Search", PLoS ONE, vol. 2, no. 4, p. e354, 2007.
[11] M. F. Shlesinger, G. M. Zaslavsky and U. Frisch, Lévy Flights and Related Topics in Physics, vol. 450, Nice: springer nature, 1994.
[12] M. F. Shlesinger, "Mathematical physics: Search research", Nature, vol.443, no. 7109, pp. 281–282, 2006.
[13] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends", Neurocomputing, vol. 408, pp. 189-215, 2020.
[14] F. Imbault and K. Lebart, "A stochastic optimization approach for parameter tuning of support vector machines", In Proc. 17th International Conf. on Pattern Recognition", Aug 2004. Vol. 4, pp. 597-600.
[15] A. Kulkarni, V. K. Jayaraman and B. D. Kulkarni, "Support vector classification with parameter tuning assisted by agent-based technique," Computers & chemical engineering, vol. 28, pp. 311-318, 2004.
[16] J. Kennedy and R. Eberhart, "A discrete binary version of the particle swarm algorithm," in IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, 1997.
[17] S.-W. Lin, K.-C. Ying, S.-C. Chen and Z.-J. Lee, Eds."Particle swarm optimization for parameter determination and feature selection of support vector machines," Expert Systems with Applications, pp. 1817-1824, November 2008.
[18] C. Chao and M. Horng, "The Construction of Support Vector Machine Classifier Using the Firefly Algorithm", Computational Intelligence and Neuroscience, vol. 2015, pp. 1-8, 2015.
[19] David H. Wolpert and William G. Macready, "No Free Lunch Theorems for Optimization" IEEE Transactions on Evolutionary Computation Vol.1 No.1 pp.67-83,1997", Journal of Japan Society for Fuzzy Theory and Systems, vol. 9, no. 5, pp. 696-697, 1997.
[20] S. Mirjalili, S. Mirjalili and A. Lewis, "Grey Wolf Optimizer", Advances in Engineering Software, vol. 69, pp. 46-61, 2014.
[21] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems", Knowledge-Based Systems, vol. 96, pp. 120-133, 2016.
[22] A. Tharwat, A. Darwish and A. Hassanien, "Rough sets and social ski-driver optimization for drug toxicity analysis", Computer Methods and Programs in Biomedicine, vol. 197, p. 105-702, 2020.
[23] N. nalla, "Social Ski Driver-Jaya Optimization-Enabled Deep ConvolutionNeural Network for Signature Verification," Turkish Journal of Computer and Mathematics Education, vol. 12, pp. 62-77, 2021.

[24] J. Shaik and G. V, "Deep Neural Network and Social Ski-Driver Optimization Algorithm for Power System Restoration with VSC - HVDC Technology", Journal of Computational Mechanics, Power System and Control, vol. 3, no. 1, pp. 1-9, 2020.
[25] H. Haklı and H. Uğuz, "A novel particle swarm optimization algorithm with Levy flight", Applied Soft Computing, vol. 23, pp. 333-345, 2014.
[26] A. Chechkin, R. Metzler, J. Klafter and V. Y. Gonchar, "Introduction to the Theory of Lévy Flights," in Anomalous Transport: Foundations and Applications, 2008, pp. 129 - 162.
[27] X.-S. Yang and S. Deb, "Cuckoo Search via Levey Flights," in Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress, 2009.
[28] M. Shehab and A. Khader, "Modified Cuckoo Search Algorithm using a New Selection Scheme for Unconstrained Optimization Problems", Current Medical Imaging Formerly Current Medical Imaging Reviews, vol. 16, no. 4, pp. 307-315, 2020.
[29] N. Bacanin, "An object-oriented software implementation of a novel cuckoo search algorithm," in In Proc. of the 5th European Conference on European Computing Conference (ECC'11), 2011.
[30] X.-S. Yang, "Firefly Algorithm, Levy Flights and Global Optimization," Research and Development in Intelligent Systems XXVI, pp. 209-218, 2010.
[31] [31] J. Xie, Y. Zhou and H. Chen, "A Novel Bat Algorithm Based on Differential Operator and Lévy Flights Trajectory", Computational Intelligence and Neuroscience, vol. 2013, pp. 1-13, 2013.
[32] A. Tharwat, A. Hassanien and B. Elnaghi, "A BA-based algorithm for parameter optimization of Support Vector Machine", Pattern Recognition Letters, vol. 93, pp. 13-22, 2017.
[33] R. Guido Van and D. F. L., The Python Language Reference Manual, Network Theory Ltd., 2011.
[34] A. Price-Whelan et al., "The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package", The Astronomical Journal, vol. 156, no. 3, p. 123, 2018.
[35] H. Al-Sahaf et al., "A survey on evolutionary machine learning", Journal of the Royal Society of New Zealand, vol. 49, no. 2, pp. 205-228, 2019.
[36] Y. Zhang and P. Zhang, "Machine training and parameter settings with social emotional optimization algorithm for support vector machine", Pattern Recognition Letters, vol. 54, pp. 36-42, 2015.
[37] A. Tharwat and T. Gabel, "Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm", Neural Computing and Applications, vol. 32, no. 11, pp. 6925-6938, 2019.
[38] E. Ndiaye, T. Le, O. Fercoq, J. Salmon and I. Takeuchi, "Safe Grid Search with Optimal Complexity," International Conference on Machine Learning, pp. 4771-4780, 2019.
[39] M. Lopez-Ibanez and T. Stutzle, “The automatic design of multiobjective ant colony optimization algorithms,” IEEE Transactions on Evolutionary Computation, vol. 16, no. 6, pp. 861–875, 2012.

Ashish Kumar Namdeo is a Ph.d. Research Scholar in the Department of Computer Science and Engineering at Jagran lakecity University, Bhopal India. He pursued his Bachelor’s in Computer Science and Engineering from UIT-RGPV, Bhopal and a M. Tech in Computer Technology & Application from NITTTR, Bhopal.
Dileep Kumar Singh holds a position as a Head, JLU-SOET at Jagran Lakecity University Bhopal. Has 16 years of experience in research and academics at various colleges. He pursued his bachelor’s from Government Engineering College Bilaspur Also did post-

graduation and PhD in CSE from MANIT Bhopal. His field of research includes, machine learning, security, web applications. Published over 15 articles in SCIE and Scopus indexed journals.