Computing and Informatics, Vol. 42, 2023, 805–833, doi: 10.31577/cai_2023_4_805

VIGILANT SALP SWARM ALGORITHM FOR FEATURE SELECTION

N. B. Arunekumar*

Department of Artificial Intelligence and Data Science
Koneru Lakshmaiah Education Foundation, Vaddeswaram-522302, AP, India
e-mail: arunekumarbala@gmail.com

K. Suresh Joseph

Department of Computer Science, Pondicherry University Puducherry, India
e-mail: ksjoseph.csc@gmail.com

J. Viswanath

Department of Artificial Intelligence and Data Science
Madanapalle Institute of Technology and Science, AP, India
e-mail: viswaj20@gmail.com

A. Anbarasi

Department of Computing Technologies
SRM Institute of Science and Technology, Kattankulathoor, TN, India
e-mail: anbarasi.a@gmail.com

N. Padmapriya

Department of Statistics, Sri Sarada College for Women (Autonomous)
Salem, TN, India
e-mail: theen.91@gmail.com

* Corresponding author

Abstract. Feature selection (FS) averts the consideration of unwanted features which may tend the classification algorithm to classify wrongly. Choosing an optimal feature subset from the given set of features is challenging due to the complex associations present within the features. In non-convex conditions, the gradient-based algorithms suffer due to local optima or saddle points with respect to initial conditions where swarm intelligence algorithms pose a higher chance to converge over the global optima. The Salp Swarm Algorithm (SSA) proposed by Mirjalili et al. is based on the chaining behaviour of sea salps but the algorithm lacks diversity in the exploration stage. Rectifying the exploratory behaviour and testing the algorithm against the FS problem is the motivation behind this work. Three variants of the algorithm are proposed, of which the Vigilant Salp Swarm Algorithm (VSSA) inherits the vigilant mechanism in Grey Wolf Optimizer (GWO), the second variant and the third variant replace a simple crossover operator and shuffle crossover operator instead of the follower's position update mechanism used in the VSSA to form Vanilla Crossover VSSA (VCVSSA) and Shuffle Crossover VSSA (SCVSSA).

Keywords: Feature selection, optimization, k-nearest neighbors, salp swarm algorithm

Mathematics Subject Classification 2010: 68T01

1 INTRODUCTION

Feature selection is a challenging problem where two contradicting objectives of selecting the minimal number of features and achieving maximum accuracy on classification have to be attained. The feature selection discards the un-impacting or misleading features in training the algorithm for the classification. Using unwanted features for classification may deteriorate the algorithm's performance as fitting an extra dimension with respect to any learning algorithm takes considerable time due to the curse of dimensionality. The predominantly used feature selection (FS) models are of three types: filter, embedded and wrapper. The filter feature selection models are independent of the learning algorithm and rank the features based on any relationship amid the features. The ranking models are computationally low in cost. However, after ranking, choosing the n number of best features would be sub-optimal as ranking algorithms would investigate only the necessity of a single feature at a time. This phenomenon biases the feature selection only towards some specific data relationship alone. But, in a real scenario, the features may have complex dependencies. For example, a feature when being alone may not be essential but, when combined with any other features it would become a vital indicator for classification. As the complexity of the dependency between the features increases, the filter models would fail to mine out the significant feature subset. In the case of the wrapper model, the association with the classification algorithm provides feedback to the feature selection algorithm [1]. It aids the FS algorithm in procuring the

minimal and better optimal subset of features. The bio-inspired algorithms [2] can swiftly investigate the problem space and find the optimal solution in contrast to the gradient-based algorithms [3], which use only the slope of the current position. The convergence speed of the swarm intelligence algorithms is higher than the gradient following algorithms. The gradient-based algorithms may get stuck in local minima or on saddle points, paving the way to choose unwanted features and thereby deteriorating the learning algorithm's performance. The above phenomenon can be seen in the comparison table. The balanced exploration and exploitation capabilities of the swarm algorithms bestow the capability of mining the best optimal solution. A hybrid algorithm fabricated by inheriting the existing swarm algorithms' best traits will be much more efficient than their parent algorithms. The Salp Swarm Algorithm proposed by [4] uses two different mechanisms for updating a salp. One is used for updating the leader regarding the food position and the second is for updating the followers as a chain. Using a single solution for guiding may stagnate the algorithm in local optima as it lacks diversity. Introducing the influence of other eminent solutions would enhance the exploratory behaviour in the initial search and make the particles more vigilant. The GWO [5] algorithm replicates the vigilant hunting strategy of the wolves where a group of wolves surround the prey and attack them. Both the Salp Swarm Algorithm and the GWO are being used on several applications as they uncover promising solutions.

1.1 Goals

The prime goal of the proposed paper is to adopt an enhanced position update mechanism for the Salp Swarm Algorithm. The following objectives will be scrutinized to ensure enhanced performance of the proposed algorithm.

- A hybrid position update model that increases the efficiency of the guiding mechanism.
- An algorithm capable of finding a minimal and the optimal subset of features best suited for classifying the objects with high accuracy compared to the SSA.
- An algorithm that could outperform the other primarily used feature selection algorithms.
- An algorithm that aids in classifying data of different dimensions.

1.2 Organization

The paper is organized as follows: Section 1 comprises the introduction to feature selection, introduction to swarm intelligence and defines the goal. Section 2 enumerates the related work on bio-inspired and feature selection algorithms. Section 3 provides the needed preliminaries and the proposed variants. Section 4 depicts the experimentation setup. Section 5 discusses the results and analysis. Section 6 concludes the paper with the findings accomplished.

2 RELATED WORK

Feature selection is an NP-hard problem that tries to avert the curse of dimensionality. Choosing a subset of m features out of n would result in 2^n different combinations of feature subsets. Each feature in a feature vector is represented as either 1 or 0 to denote whether the respective feature is selected or not. Various algorithms have been proposed starting from the Genetic Algorithm [6], Simulated Annealing [7], Ant Colony Optimization [8] and PSO [9]. Other state-of-the-art swarm algorithms are Cuckoo Search [10], Bat Algorithm that mimics the echolocation behaviour of bats [11], Firefly Algorithm [12], Biogeography-Based Optimizer [13] and Whale Optimization Algorithm [14].

All the data observed need not necessarily be used for the classification and the data may possess several complexities such as dependency between features and irrelevant features. These feature selection algorithms pick out the essential features among the complete set and facilitate the execution of classification algorithms to provide high accuracy and low running time. The F-score [15], PCA [16], and correlation-based feature (CBF) selection [17] are some of the filter model feature selection algorithms. The filter models have no interaction with the learning algorithm and are purely dependent on the features' characteristics. The wrapper models on the other extreme works on the feedback from the learning algorithm. The elegant behaviour of the bio-inspired algorithm has attracted researchers to adopt these algorithms for wrapper feature selection algorithms. For feature selection, algorithms like hybrid genetic algorithm [18], hybrid PSO [19] with micro genetic algorithm and Gaussian mutation were used. Unlike the problems with continuous space, the binary algorithm takes values of either 0 or 1. Binary variants of the algorithms like bGWO [20], BPSO [21] and Binary Ant Lion Optimizer [22] were introduced specifically for the feature selection problems. To convert the continuous algorithms into their binary equivalent without altering any of their characteristics, the transfer functions [23] were introduced. There are totally 8 different functions that can be broadly divided into two families of S-shaped and V-shaped functions. Both the S-shaped and the V-shaped family of transfer functions map the input from the continuous range into values amid the range [0,1] which is later converted to binary value with conditions similar to Equation (11). Binary algorithms like the Binary Salp Swarm [24] and the Binary Dragonfly Algorithm [25] also used transfer functions.

3 PRELIMINARIES AND PROPOSED ALGORITHM

3.1 Brief on SSA

The SSA algorithm proposed by Ali Mirjalili et al. [4] is a population-based algorithm inspired by the swarming behaviour of the transparent jelly-like fish. This fish moves analogous to the motion exhibited by jet propulsion where the water is inhaled and exerted from its body to move forward. Along with the motion, the salps feed from

Type	Sl. No.	Dataset Name	Instances	Features		Sl. No.	Dataset Name	Instances	Features (n.o.f)
LOW DIM f < 100	1	Wine	178	13	L D	13	Movement_libras	360	90
	2	Hepatitis	155	19	HIGH DIM f ≥ 100	14	Spambase	4 601	57
	3	Vehicle	94	18		15	Arrhythmia	452	279
	4	Zoo	101	16		16	Clean1	1 593	265
	5	Heart disease	270	13		17	Hill valley	1 212	100
	6	Wisconson	682	10		18	Leukemia	72	7 070
	7	ionosphere	351	34		19	Colon	62	2 000
	8	Lung-cancer	32	56		20	Arcene	200	10 000
	9	Dermatology	366	34		21	Lymphoma	96	4 026
	10	Sonar	208	60		22	Smk_can_187	187	19 993
	11	BreastEW	569	29		23	Tox_171	171	5 748
	12	Soybean-small	47	35		24	Coil20	1 440	1 024

Table 1. Dataset description

the inhaled water by filtering out the plankton. The locomotive behaviour of the salp swarm is modelled mathematically to solve the optimization problems. They bind together as salp chains and exhibit swarming behaviour. These salps move as long chains and are attached to each other.

3.1.1 Salp Swarm Algorithm

The salps chain can be divided into two parts: the leader and the followers. The first salp is termed the leader, and the rest form the chain members or followers, as shown in Figure 2.

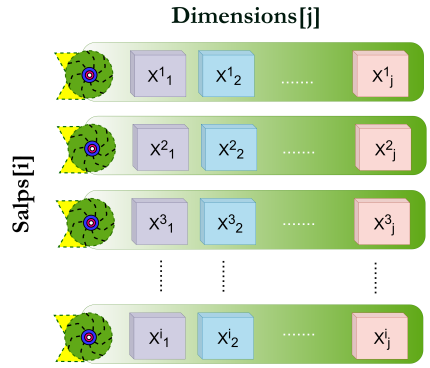


Figure 1. Representation of salp swarm in problem space

The complete population data is stored as the matrix comprising the number of individuals and the number of dimensions. All the agents and their respective features are combined to form the matrix x_j^i as in Figure 1, where i denotes the salp or agent number and j denotes the dimension. The position update of the salp members is done separately in two stages. The leader's position is updated using the Equation (2) which relies on the target food position.

$$c_1 = 2 * e^{-\left(\frac{4l}{L}\right)^2}, \tag{1}$$

$$x_j^1 = \begin{cases} T_j + c_1 * ((ub_j - lb_j) * c_2 + lb_j), & c_3 \geq 0.5, \\ T_j - c_1 * ((ub_j - lb_j) * c_2 + lb_j), & c_3 \geq 0.5, \end{cases} \tag{2}$$

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right), \quad i \geq 2. \tag{3}$$

The variable x_j^1 is j^{th} dimension of the first salp. Variable T denotes the target or food, and ub and lb are the upper and lower bound respectively. The parameters c_2, and c_3 are random numbers amid $[0, 1]$ and the parameter c_1 is calculated as

Algorithm 1: SSA pseudo code

```
Initialize the population within the bound;
while current iteration ≤ total no. of iterations do
    Compute fitness of each salp;
    T = salp possessing best fitness (Target or Food);
    Update parameter c1 using Equation (1);
    for each salp particle xi do
        if i == 1 then
            Update leader salp solution using Equation (2);
        else
            Update followers on-chain using Equation (3);
        end
    end
end
Return T
```

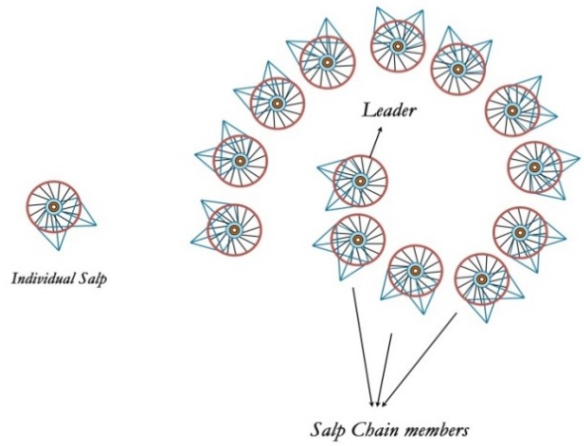


Figure 2. Salp swarm

given in Equation (1), which changes in accordance to the iteration count. The parameter c_1 is a crucial parameter that balances the algorithm between exploration and exploitation. The parameter l refers to the current iteration and L depicts the total number of iterations. The remaining salps other than the leader are being modified using Equation (3). The Pseudo code for the complete working model of the SSA is given in Algorithm 1. Several variants of the Salp Swarm Algorithm have been proposed so far. Among these variants specialized for feature selection are the bSSA [26] and iSSA [27]. In bSSA three major variants were proposed the S-shaped transfer function variants, the V-shaped transfer function variants and the

simple crossover variant. The transfer functions are one of the ways by which the values at any range are transformed into a range amid $[0, 1]$. The algorithm was run over 22 different datasets with 30 independent runs each. The iSSA algorithm used the inertia weight ω parameter from PSO and the target food. The algorithm was run over 23 different datasets with 20 independent runs each. The multi-salp chain [28] algorithm split the salps into sub-chains, updated the parameter with different strategies for each sub-chain and was run over 20 datasets with 30 iterations on each.

3.2 Proposed Algorithm

The salp chain updates the food in two phases where updating the leader position is highly critical. Based on the leader's position, the chain particles will be updated. In such cases, if the leader gets into local optima there is a huge chance for the followers to avoid promising search areas. To avert this situation the vigilant mechanism found in the GWO is introduced. Three algorithms are proposed: the first is VGSSA algorithm, the second is VCVSSA and the third is SCVGSSA.

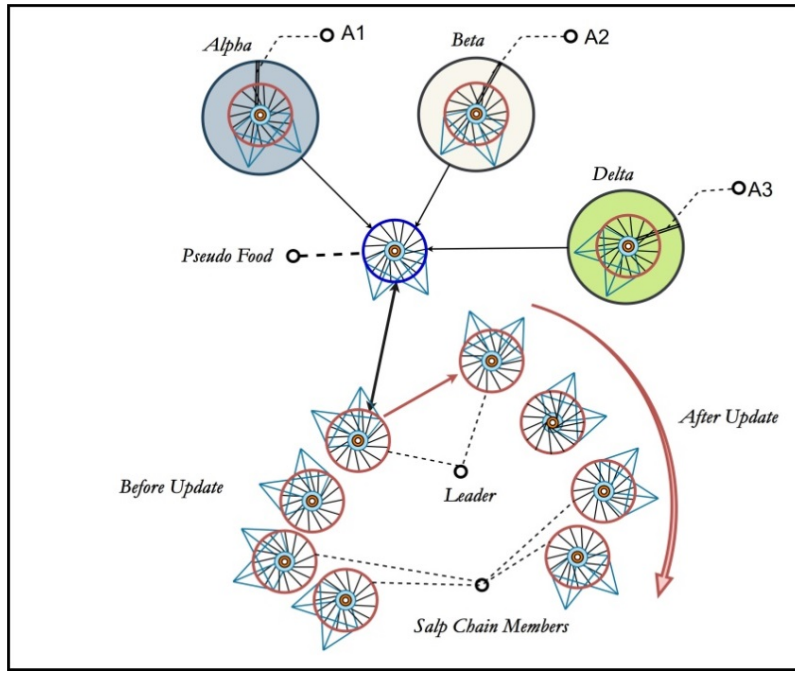


Figure 3. Vigilant SSA

3.2.1 Vigilant Salp Swarm Algorithm

Instead of relying on only one best solution as the target food, three best solutions namely Alpha, Beta and Delta are adopted. The Pseudo Food given in Equation (4) is the average of the three solutions. The Pseudo Food (PF_j) for each dimension j is replaced instead of Target Food T_j used in the SSA. The parameters r_1, r_2, r_3 are random numbers in between the range $[0, 1]$. The overall working mechanism of the VSSA is depicted in Figure 3. After the accomplishment of maximum iterations, the best solution (α) is returned as given in the pseudo-code of VSSA in Algorithm 2.

Algorithm 2: VSSA algorithm pseudo code

```
Initialize population with respect to the bounds;
while Max iterations ≥ current iteration do
    Derive fitness for each salp;
    Update α, β, δ food sources;
    α = 1st best solution;
    β = 2nd best solution;
    δ = 3rd best solution;
    PF = Compute Pseudo Food with Equation (4);
    Update c1 w.r.t. Equation (1);
    for each salp particle xi do
        if xi is leader then
            Use Leader position update as in Equation (8);
        else
            Use followers position update as in Equation (3);
Return α;
```

$$PF_j = ((A_1 * \alpha_j) + (A_2 * \beta_j) + (A_3 * \delta_j))/3, \tag{4}$$

$$A_1 = 2 * r_1, \tag{5}$$

$$A_2 = 2 * r_2, \tag{6}$$

$$A_3 = 2 * r_3, \tag{7}$$

$$x_j^1 = \begin{cases} PF_j + c_1 * ((ub_j - lb_j) * c_2 + lb_j), & c_3 \geq 0.5, \\ PF_j - c_1 * ((ub_j - lb_j) * c_2 + lb_j), & c_3 \geq 0.5. \end{cases} \tag{8}$$

3.2.2 Vanilla Crossover Vigilant Salp Swarm Algorithm (VCVSSA)

As a binary problem, feature selection either rejects or accepts a feature. Using the proposed VSSA algorithm an enhancement in the exploration of the agents can

be achieved. However, the salp chain followers still update their positions using Equation (3) which intuitively relocates the solution amid itself and its predecessor. This phenomenon can be more efficiently modelled by using a crossover operator ψ as in Equation (9). The crossover operator is predominantly used in the inheritance phase of the genetic algorithm [6, 29, 30]. The crossover operator can extract the exact features from both of its parents. The vanilla (simple) single-point crossover as given in Figure 4 inherits half of its characteristics from parent A and the rest half from parent B which is equivalent to Equation (3).

$$x_i^{t+1} = \psi(x_i, x_{i-1}). \tag{9}$$

In Equation (9), the child feature set x_i^{t+1} is the salp i at time $t+1$ which is derived from its parents, x_i the salp itself at time t and the salp's predecessor x_{i-1}.

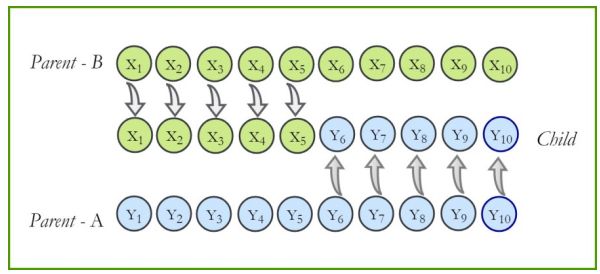


Figure 4. Simple single point crossover (vanilla crossover)

3.2.3 Shuffle Crossover Vigilant Salp Swarm Algorithm (SCVSSA)

A simple single-point crossover would be a better choice as it averts having a complete, continuous domain calculation and also depicts the behaviour of Equation (3). But, the single-point crossover has a substantial drawback of always inheriting either the left or right half of the parent as a whole.

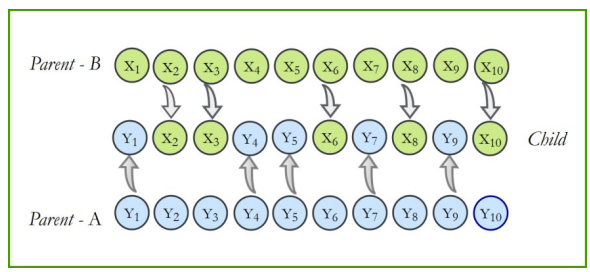


Figure 5. Shuffle crossover

The single-point crossover again introduces the paucity of inheriting multiple combinations of features. To overcome all these difficulties the proposed SCVSSA uses shuffled crossover operator φ which carries out crossover as given in Equation (10) instead of the position update by Equation (3).

$$x_j^i = \varphi(x_j^i, x_j^{i-1}). \tag{10}$$

In shuffle crossover, both the parents are shuffled with the same indices and then a single point crossover is done with the final reverse shuffling of the children to roll them back into their original indices again. The single-point crossover after the shuffling overcomes its earlier difficulties. The shuffled crossover also inputs the parents and outputs the children as done by the simple crossover. The overall mechanism of the shuffle crossover can be observed in Figure 5 and the combined flowchart for both the crossover mechanisms is given in Figure 6.

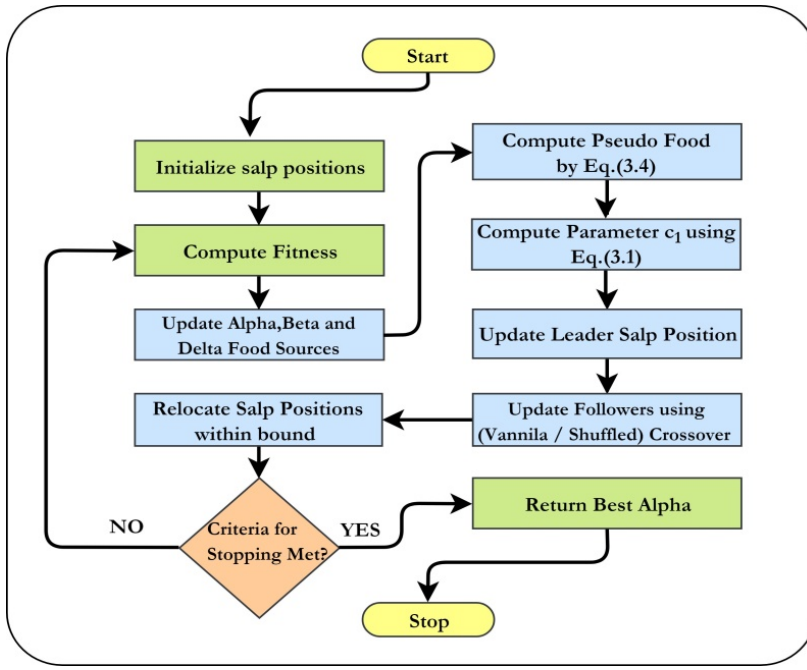


Figure 6. Flowchart for vanilla (VCVSSA)/shuffled (SCVSSA)

4 EXPERIMENTATION

4.1 Feature Representation

As discussed in Section 2, feature selection is a discrete problem. The feature sets can be modelled as binary solutions represented as an n-dimensional vector as in Figure 7 which uses 0 to reject and 1 to accept the respective feature.

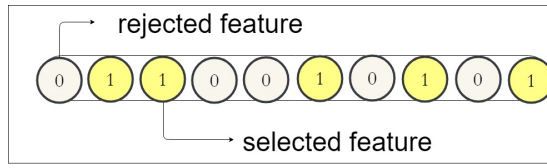


Figure 7. Feature vector representation

The algorithm is initialized and later updated over continuous domain values. To directly convert j^{th} dimension of the i^{th} continuous agent x_j^i into its respective binary agent (bx_j^i) Equation (11) is used. The converted bx_j^i is utilized to calculate the fitness of the corresponding continuous solution.

$$bx_j^i = \begin{cases} 1, & \text{if } x_j^i \geq 0.5, \\ 0, & \text{if } x_j^i < 0.5. \end{cases} \tag{11}$$

4.2 Classification Algorithm and Parameter Setup

The KNN classification used in this experiment uses certain distance measures to classify the data. A generic KNN model uses Euclidean distance as given in Equation (12) for the classification.

$$D(X_1, X_2) = \sqrt{(X_1 - X_2)^2}. \tag{12}$$

Various parameters used for the other algorithms are enumerated in the table, for the sake of fair comparison all the algorithms were implemented in the same language and compared with the same sample on each iteration.

4.3 Fitness Function

The feature selection's fitness function comprises two objectives contradicting each other. The first objective is selecting the feature that yields high accuracy for the classification algorithm and the second is selecting the least number of features. Aggregating both these objectives the fitness function which is utilized in most of the literature is being adopted as given in Equation (13). The error rate γ_R of the solution is given as (1 − accuracy). R is the number of features selected in the

solution and N is the total number of features. The hyperparameters (ρ, σ) decide the weights for the error rate and the features selected. The hyperparameter σ is given by $\sigma = (1 - \rho)$ and the other hyperparameter ρ is given as 0.99 because the reduction of error rate has to be weighted higher than the number of features. These hyperparameter values are adopted from the literature [22, 31].

$$\text{Fit} = \rho * \gamma_R(D) + \sigma * \frac{|R|}{|N|}. \tag{13}$$

4.4 Datasets

The proposed variants are tested over 24 datasets of various modalities and characteristics to prove the durability of the algorithm in versatile conditions. All the datasets have been downloaded from the standard UCI machine learning dataset repository [32] and ASU feature selection repository [33]. Very few algorithms in the literature have tested the feature selection with such enormously high-dimension datasets. Failing to test on such large dimension datasets will fail to portray the exact capabilities of the exploration and the exploitation of the algorithm. In this work, datasets of both large and small dimensions have been tested to generalize the algorithm's capability under various conditions. The datasets with feature sizes greater than 100 are termed large-dimension datasets. The number of instances in the datasets also varies in accordance with the change in dimensions. Some of the datasets possess missing values too. The above-given traits provide a challenging task of testing the exploration and exploitation ability of the algorithm such that the algorithms have to elect the optimal feature subset.

4.5 Experiment Setup

Including the raw VSSA, two other variants incorporating crossover have been proposed. All the proposed three variants are compared with the baseline Salp Swarm [4], its recently proposed hybrids bSSA [24], iSSA [27] and predominantly used feature selection algorithms bGWO [34], GOA [35], ALO [36] and PSO [37]. The general parameter setting derived from the literature [24] is used for the experimentation as enumerated in Table 2.

The feature selection algorithms are compared over three standard metrics: fitness, accuracy and number of selected features. Each algorithm is run 30 times over a dataset and its arithmetic mean is counted for the final comparison. The dataset is split into 80-20 ratio where 80 % is utilized for training the classifier and 20 % is utilized for testing it. For each of the 30 iterations performed, a unique sample of training and testing data was subjected to all the algorithms. By providing the same sample to all the algorithms for each round, the bias of the classifier with respect to the sample could be completely averted and all the algorithms would have an equal opportunity to showcase their performance.

Sl. No.	Parameters	Value
1	Population	7
2	Neighbor count in KNN	5
3	Independent run	30
4	Fitness parameters	$\rho = 0.99$, $\sigma = 0.01$
5	GOA	$c = [\text{Min} = 0, \text{Max} = 1]$, cMin = 0.0004, cMax = 1
6	PSO	$w = 1$; $c1 = 1.5$; $c2 = 2.0$;
7	(SSA/VSSA/SCVSSA), bGWO	$c1, a = [\text{Max} = 2, \text{Min} = 0]$

Table 2. Parameter configuration

5 RESULTS AND DISCUSSION

5.1 Assessment of Results

The fitness, accuracy and the number of features selected are tabulated in Tables 3, 4 and 5. The fitness value must be minimal as the error rate and no features are considered. Each cell in the table corresponds to either the mean (avg) or the standard deviation (std) for 30 independent runs on each dataset of the respective algorithms. The proposed variants are compared with the baseline, modified SSA and the other existing algorithms. On accuracy alone, the KNN without feature selection is compared.

5.1.1 Comparison over SCVSSA

Among the three proposed variants, the SCVSSA algorithm has outperformed every other algorithm over most datasets. It has been ranked 1 on both fitness and accuracy against all the other algorithms including the other proposed variants which is evident from Table 3 and Table 4. From those tables, it is also evident that the SCVSSA has outperformed all the other algorithms on 87.5 % of the datasets in terms of fitness and accuracy. Considering individually, SCVSSA has outperformed VCVSSA, bSSA [24] and PSO [37] over 95.8 % of the datasets and the VSSA, iSSA [27], bGWO [34], GOA [35], ALO [36] and baseline SSA [4] over 100 % of the datasets on accuracy and fitness. In addition, the naïve KNN classifier was also subjected to experimentation and its accuracy is compared in Table 4. It clearly shows the need for the feature selection that has increased the accuracy by 16 %. From Table 3 inference can be made that, on all datasets, the algorithm performs better. In terms of the number of features chosen, the SCVSSA is ranked second. The parameters in the fitness Equation (13) facilitate the primary goal to acquire good accuracy and the secondary goal to elect the minimal number of features.

Therefore, the SCVSSA has balanced well in accordance with the fitness equation and has tried to converge with the global minima without getting stuck or stagnating. For example, On the Soybean-small dataset, all the algorithms have achieved the

Sl.	Algorithm	SCVSSA		VCVSSA		VSSA		iSSA		bSSA		bGWO		GOA		ALO		PSO		SSA	
	Dataset	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
1	Wine	0.0184	0.0172	**0.0171**	0.0173	0.0216	0.0181	0.0238	0.0209	0.0202	0.0184	0.0419	0.0295	0.0905	0.0373	0.0359	0.0270	0.0301	0.0250	0.0244	0.0188
2	Hepatitis	**0.0769**	0.0441	0.0790	0.0530	0.0782	0.0407	0.0878	0.0530	0.0930	0.0512	0.1138	0.0593	0.1609	0.0609	0.1111	0.0561	0.1249	0.0674	0.0795	0.0456
3	Vehicle	**0.2235**	0.0816	0.2389	0.0748	0.2427	0.0753	0.2446	0.0817	0.2439	0.0730	0.3005	0.0867	0.3506	0.1018	0.2886	0.0805	0.2609	0.0825	0.2378	0.0893
4	Zoo	0.0171	0.0226	0.0173	0.0254	0.0155	0.0251	0.0151	0.0217	**0.0148**	0.0207	0.0307	0.0371	0.0612	0.0415	0.0310	0.0332	0.0280	0.0305	0.0143	0.0241
5	Heart disease	**0.1233**	0.0325	0.1244	0.0335	0.1260	0.0284	0.1323	0.0303	0.1347	0.0293	0.1818	0.0576	0.2224	0.0544	0.1550	0.0444	0.1693	0.0594	0.1284	0.0333
6	Wisconson	**0.0161**	0.0093	0.0179	0.0097	0.0174	0.0103	0.0162	0.0094	0.0178	0.0096	0.0207	0.0098	0.0305	0.0135	0.0215	0.0108	0.0220	0.0105	0.0182	0.0103
7	Ionosphere	**0.0581**	0.0214	0.0629	0.0267	0.0659	0.0265	0.0691	0.0234	0.1020	0.0361	0.1030	0.0394	0.1327	0.0382	0.0771	0.0319	0.0871	0.0378	0.0856	0.0353
8	Lung-cancer	**0.0581**	0.0881	0.0726	0.0962	0.0823	0.0958	0.0812	0.0800	0.1322	0.1133	0.1474	0.1173	0.3238	0.1330	0.1150	0.1014	0.0975	0.1008	0.0837	0.0961
9	Dermatology	**0.0064**	0.0055	0.0086	0.0074	0.0085	0.0080	0.0115	0.0083	0.0096	0.0075	0.0132	0.0092	0.0406	0.0180	0.0174	0.0114	0.0121	0.0097	0.0079	0.0061
10	Sonar	**0.0501**	0.0313	0.0636	0.0306	0.0626	0.0296	0.0890	0.0308	0.0979	0.0399	0.1007	0.0462	0.1650	0.0499	0.1083	0.0529	0.0847	0.0428	0.0819	0.0469
11	Breast EW	**0.1349**	0.0274	0.1424	0.0304	0.1468	0.0260	0.1543	0.0227	0.1582	0.0252	0.1581	0.0281	0.2007	0.0284	0.1641	0.0252	0.1565	0.0307	0.1505	0.0264
12	Soybean-small	**0.0006**	0.0002	0.0009	0.0006	0.0008	0.0004	0.0008	0.0004	0.0032	0.0003	0.0035	0.0006	0.0041	0.0007	0.0016	0.0010	0.0021	0.0006	0.0021	0.0006
13	Movement_libras	**0.1428**	0.0394	0.1537	0.0359	0.1580	0.0380	0.1647	0.0326	0.1771	0.0377	0.1734	0.0421	0.2120	0.0392	0.1800	0.0397	0.1585	0.0396	0.1611	0.0366
14	Spambase	**0.0732**	0.0072	0.0764	0.0078	0.0772	0.0065	0.0863	0.0090	0.0836	0.0083	0.0807	0.0084	0.1088	0.0116	0.0862	0.0104	0.0804	0.0100	0.0790	0.0081
15	Arrhythmia	**0.2880**	0.0437	0.2979	0.0354	0.3010	0.0450	0.3129	0.0433	0.3241	0.0447	0.3189	0.0452	0.3640	0.0427	0.3055	0.0713	0.2986	0.0475	0.3017	0.0440
16	Clean1	0.0483	0.0224	0.0586	0.0195	0.0535	0.0198	0.0793	0.0206	0.0777	0.0226	0.0685	0.0238	0.1166	0.0214	0.0705	0.0250	**0.0431**	0.0245	0.0585	0.0234
17	Hill valley	**0.3551**	0.0219	0.3626	0.0205	0.3655	0.0266	0.3726	0.0209	0.3861	0.0198	0.3794	0.0239	0.4104	0.0235	0.3760	0.0213	0.3632	0.0214	0.3769	0.0205
18	Leukemia	**0.0133**	0.0269	0.0244	0.0406	0.0157	0.0285	0.0200	0.0353	0.0668	0.0573	0.0734	0.0658	0.1039	0.0709	0.0201	0.0464	0.0816	0.0577	0.0729	0.0636
19	Colon	**0.0332**	0.0384	0.0588	0.0554	0.0639	0.0568	0.0636	0.0493	0.1322	0.0730	0.1298	0.0836	0.1801	0.0941	0.0562	0.0598	0.1286	0.0707	0.1290	0.0677
20	Arcene	**0.0617**	0.0378	0.0832	0.0370	0.0825	0.0370	0.0764	0.0336	0.1193	0.0458	0.1243	0.0410	0.1427	0.0494	0.0749	0.0298	0.1177	0.0463	0.1170	0.0415
21	Lymphoma	**0.0399**	0.0399	0.0417	0.0393	0.0434	0.0408	0.0421	0.0394	0.0580	0.0486	0.0616	0.0518	0.0697	0.0541	0.0519	0.0421	0.0612	0.0524	0.0577	0.0519
22	Smk_can_187	**0.2107**	0.0516	0.2143	0.0564	0.2179	0.0567	0.2313	0.0461	0.2896	0.0565	0.2863	0.0591	0.3272	0.0586	0.2137	0.0689	0.2835	0.0667	0.2845	0.0600
23	Tox_171	**0.1102**	0.0462	0.1420	0.0566	0.1263	0.0610	0.1469	0.0409	0.1847	0.0527	0.1753	0.0665	0.2690	0.0674	0.1673	0.0551	0.1650	0.0577	0.1558	0.0609
24	Coil20	**0.0072**	0.0051	0.0104	0.0064	0.0098	0.0063	0.0109	0.0077	0.0189	0.0106	0.0202	0.0112	0.0249	0.0115	0.0104	0.0072	0.0151	0.0084	0.0157	0.0077
	Avg Rank	**1.5000**		3.0000		3.4375		4.9688		7.1250		10.0000		6.0625		5.9063		5.9063		5.2500	
	Final Rank	**1.0000**		2.0000		3.0000		4.0000		8.0000		9.0000		10.0000		7.0000		6.0000		5.0000	

Table 3. Fitness – comparison over proposed methods vs. existing methods

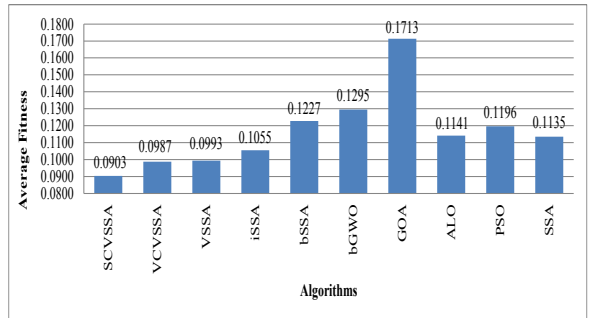


Figure 8. Comparison over average fitness

classification accuracy of 100 % and in such cases, the iSSA has failed to provide the least number of features instead, SCVSSA has chosen the least number of features. Even though SCVSSA has not been ranked one in terms of features selected, it is evident that the algorithm has provided a good reduction rate of features that provide the best accuracy which is the vital component.

5.1.2 Comparison over VCVSSA

The VCVSSA uses simple crossover despite the shuffled. This variant ranked 2, surpasses the other algorithms other than SCVSSA. The algorithm has an increased average accuracy of 15 % than the raw KNN algorithm with no feature selection. The algorithm has surpassed VSSA in over 62 % of the datasets, iSSA, ALO and SSA in over 83 % of the datasets, bSSA in over 91 % of the datasets, bGWO, GOA and KNN in over 100 % of the datasets in terms of accuracy. Sufficiently with the ordinary crossover, the algorithm could perform considerably well. However, the Shuffled crossover was proposed to improve the accuracy to some greater extent. Regarding the number of features selected, it has also managed to a good extent and is placed next to the VCVSSA. The overall performance of this variant can also be termed good when compared to the other algorithms than the VCVSSA.

5.1.3 Comparison over VSSA

The VSSA is the naïve model which did not use any crossover operator. Instead, it has used Equation (11) for acquiring the binary equivalent of a feature vector. The VSSA has the least proficiency among the proposed algorithms. But it is better than every other existing algorithm compared in Tables 3, 4 and 5.

The VSSA algorithm has outstepped the existing algorithms on 11 datasets in terms of accuracy. The algorithm has a higher accuracy for 20 of the datasets, i.e. 83 % over PSO and SSA. Likewise, higher accuracy over 87 % of the datasets has been achieved on iSSA and ALO, 91 % over bSSA and 100 % over bGWO, KNN and

Sl.	Algorithm	SCVSSA		VCVSSA		VSSA		iSSA		bSSA		bGWO		KNN		GOA		ALO		PSO		SSA	
	Dataset	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
No	**Wine**	0.9843	0.0174	**0.9861**	0.0175	0.9815	0.0184	0.9787	0.0215	0.9833	0.0187	0.9630	0.0295	0.7204	0.0606	0.9130	0.0377	0.9685	0.0270	0.9731	0.0258	0.9787	0.0189
2	**Hepatitis**	**0.9247**	0.0451	0.9226	0.0540	0.9237	0.0411	0.9129	0.0537	0.9097	0.0518	0.8892	0.0603	0.7430	0.0750	0.8419	0.0613	0.8903	0.0572	0.8763	0.0683	0.9226	0.0461
3	**Vehicle**	**0.7772**	0.0826	0.7614	0.0754	0.7579	0.0765	0.7558	0.0827	0.7579	0.0739	0.7013	0.0876	0.5354	0.1229	0.6503	0.1030	0.7118	0.0813	0.7403	0.0838	0.7632	0.0904
4	**Zoo**	0.9856	0.0223	0.9856	0.0256	0.9873	0.0248	0.9873	0.0214	**0.9889**	0.0205	0.9738	0.0376	0.8965	0.0640	0.9431	0.0421	0.9723	0.0330	0.9755	0.0308	0.9888	0.0241
5	**Heart disease**	**0.8790**	0.0332	0.8778	0.0339	0.8765	0.0289	0.8698	0.0306	0.8679	0.0299	0.8216	0.0583	0.6667	0.0416	0.7796	0.0549	0.8469	0.0451	0.8327	0.0602	0.8741	0.0338
6	**Wisconson**	**0.9881**	0.0093	0.9864	0.0097	0.9869	0.0102	0.9878	0.0095	0.9869	0.0098	0.9854	0.0098	0.6141	0.0413	0.9742	0.0141	0.9839	0.0106	0.9822	0.0108	0.9861	0.0104
7	**Ionosphere**	**0.9427**	0.0215	0.9380	0.0271	0.9352	0.0268	0.9315	0.0236	0.9014	0.0364	0.9014	0.0398	0.8380	0.0396	0.8704	0.0385	0.9239	0.0315	0.9155	0.0381	0.9169	0.0357
8	**Lung-cancer**	**0.9429**	0.0888	0.9286	0.0975	0.9190	0.0970	0.9190	0.0812	0.8714	0.1147	0.8563	0.1187	0.4971	0.1550	0.6775	0.1346	0.8857	0.1021	0.9048	0.1016	0.9190	0.0970
9	**Dermatology**	**0.9973**	0.0055	0.9955	0.0074	0.9955	0.0082	0.9923	0.0085	0.9955	0.0074	0.9923	0.0092	0.8599	0.0364	0.9644	0.0183	0.9878	0.0114	0.9919	0.0098	0.9964	0.0061
10	**Sonar**	**0.9524**	0.0319	0.9389	0.0311	0.9405	0.0298	0.9127	0.0315	0.9063	0.0405	0.9040	0.0466	0.7667	0.0645	0.8381	0.0506	0.8944	0.0537	0.9183	0.0432	0.9214	0.0473
11	**Breast EW**	**0.8678**	0.0276	0.8605	0.0314	0.8564	0.0260	0.8476	0.0231	0.8459	0.0257	0.8468	0.0283	0.7827	0.0293	0.8023	0.0285	0.8395	0.0249	0.8468	0.0311	0.8526	0.0266
12	**Soybean-small**	**1.0000**	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	0.9433	0.0626	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
13	**Movement_libras**	**0.8588**	0.0398	0.8486	0.0369	0.8435	0.0386	0.8361	0.0333	0.8264	0.0380	0.8310	0.0425	0.7639	0.0438	0.7907	0.0396	0.8222	0.0397	0.8440	0.0399	0.8417	0.0371
14	**Spambase**	**0.9322**	0.0073	0.9290	0.0080	0.9282	0.0070	0.9170	0.0092	0.9214	0.0082	0.9261	0.0085	0.8003	0.0131	0.8952	0.0118	0.9193	0.0104	0.9239	0.0102	0.9252	0.0082
15	**Arrhythmia**	**0.7126**	0.0446	0.7025	0.0360	0.6997	0.0456	0.6864	0.0436	0.6781	0.0452	0.6846	0.0457	0.5984	0.0527	0.6373	0.0432	0.6941	0.0720	0.7026	0.0480	0.7000	0.0445
16	**Clean1**	0.9549	0.0224	0.9455	0.0195	0.9503	0.0200	0.9229	0.0206	0.9271	0.0229	0.9375	0.0240	0.8715	0.0291	0.8872	0.0216	0.9340	0.0250	**0.9608**	0.0247	0.9455	0.0236
17	**Hill valley**	**0.6443**	0.0218	0.6370	0.0208	0.6343	0.0265	0.6254	0.0211	0.6155	0.0201	0.6240	0.0244	0.5579	0.0214	0.5904	0.0239	0.6222	0.0216	0.6379	0.0216	0.6240	0.0206
18	**Leukemia**	**0.9867**	0.0271	0.9756	0.0410	0.9844	0.0287	0.9800	0.0357	0.9378	0.0579	0.9311	0.0666	0.8711	0.0699	0.9000	0.0717	0.9800	0.0468	0.9222	0.0583	0.9311	0.0643
19	**Colon**	**0.9667**	0.0388	0.9410	0.0560	0.9359	0.0574	0.9359	0.0498	0.8718	0.0738	0.8744	0.0845	0.7564	0.0949	0.8231	0.0951	0.9436	0.0604	0.8744	0.0714	0.8744	0.0685
20	**Arcene**	**0.9383**	0.0381	0.9167	0.0373	0.9175	0.0372	0.9233	0.0341	0.8850	0.0462	0.8800	0.0412	0.8308	0.0494	0.8608	0.0499	0.9250	0.0301	0.8858	0.0467	0.8867	0.0419
21	**Lymphoma**	**0.9598**	0.0403	0.9582	0.0396	0.9565	0.0410	0.9578	0.0397	0.9466	0.0490	0.9431	0.0522	0.9142	0.0663	0.9346	0.0546	0.9481	0.0425	0.9426	0.0529	0.9464	0.0524
22	**Smk_can_187**	**0.7877**	0.0521	0.7842	0.0572	0.7807	0.0571	0.7667	0.0467	0.7132	0.0571	0.7167	0.0597	0.6474	0.0597	0.6746	0.0592	0.7851	0.0691	0.7184	0.0674	0.7175	0.0606
23	**Tox_171**	**0.8914**	0.0465	0.8590	0.0571	0.8752	0.0612	0.8533	0.0416	0.8190	0.0532	0.8295	0.0673	0.6381	0.0792	0.7333	0.0681	0.8333	0.0557	0.8381	0.0583	0.8476	0.0616
24	**Coil20**	**0.9941**	0.0050	0.9916	0.0064	0.9917	0.0064	0.9900	0.0077	0.9863	0.0107	0.9859	0.0112	0.9697	0.0139	0.9799	0.0116	0.9910	0.0072	0.9890	0.0085	0.9888	0.0077
	Avg Rank	**1.2800**		3.0400		3.0400		4.7200		6.4000		7.2800		11.0000		9.6400		6.2800		6.1200		4.9200	
	Final Rank	**1.0000**		2.0000		2.0000		4.0000		8.0000		9.0000		11.0000		10.0000		7.0000		6.0000		5.0000	

Table 4. Accuracy – comparison over proposed methods vs. existing methods

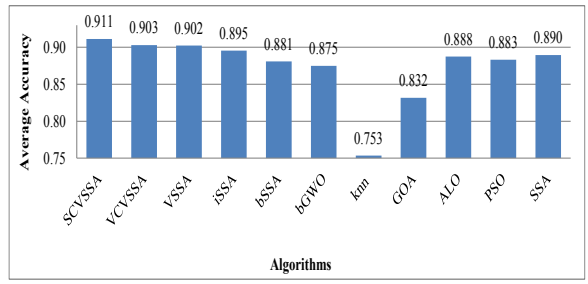


Figure 9. Comparison over average accuracy

GOA. In terms of the feature reduction, even though VSSA being ranked 4th, it has showcased a satisfactory rate of reduction in features.

5.1.4 Overall Analysis on Results with Meta-Heuristic Algorithms

Three variants were proposed, out of which one variant without crossover and the rest two with crossover operator have been proposed. All three algorithms have been compared over the three metrics of accuracy, fitness and number of features selected. Tables 3, 4 and 5 show that the variant SCVSSA that uses the shuffled crossover has gained better proficiency than the other two. Utilization of the Pseudo food has enhanced the search on exploration and exploitation stages where the leader is being re-positioned without being biased towards the best solution alone.

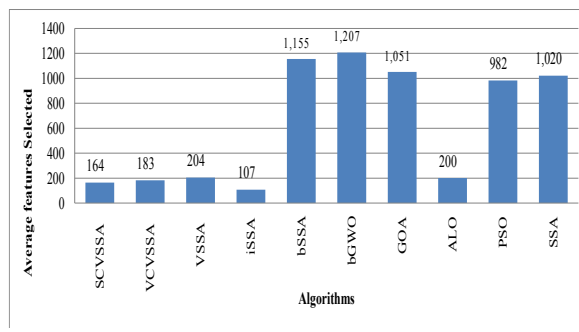


Figure 10. Comparison over the average number of features selected

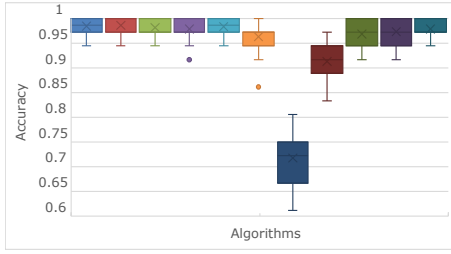
The comparison of all the datasets between the proposed algorithm and the existing algorithm is given in Figure 11 and Figure 12. The box plot depicts the median – a measure of centrality, and quartile ranges which aid the measures of dispersion, minimum and maximum values. On the algorithms such as bSSA and

Sl.	Algorithm	SCVSSA		VCVSSA		VSSA		iSSA		bSSA		bGWO		GOA		ALO		PSO		SSA	
	Dataset	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
1	**Wine**	**3.6**	1.0	4.4	1.5	4.3	1.5	**3.6**	1.0	4.8	1.2	6.8	1.9	5.6	1.5	6.1	2.5	4.6	1.4	4.3	1.4
2	**Hepatitis**	4.6	2.6	4.4	2.8	5.1	2.6	**3.1**	1.8	6.8	2.1	7.9	1.9	8.4	2.0	4.8	3.0	4.7	1.7	5.5	2.2
3	**Vehicle**	5.3	2.2	**4.9**	1.7	5.5	2.1	5.2	1.8	7.6	1.4	8.7	1.6	7.9	1.9	5.9	2.5	6.7	1.8	6.1	1.4
4	**Zoo**	4.6	1.8	4.9	2.3	4.6	2.4	**4.1**	2.0	6.1	1.7	7.6	1.7	7.7	1.6	5.8	2.9	5.9	1.7	5.1	1.5
5	**Heart disease**	4.5	1.5	4.4	1.3	5.0	1.4	**4.3**	1.1	5.1	1.4	6.8	1.2	5.5	1.6	4.4	1.4	4.8	1.4	4.9	1.4
6	**Wisconson**	4.3	1.5	4.4	1.2	4.4	1.2	**4.1**	1.2	4.8	1.4	6.2	1.3	5.0	1.0	5.6	1.8	4.4	1.2	4.5	1.1
7	**Ionosphere**	4.8	2.2	5.3	2.9	5.9	3.2	**4.2**	2.4	15.1	2.2	18.2	3.2	14.9	2.9	6.1	5.6	11.8	3.3	11.4	2.9
8	**Lung-cancer**	8.6	6.6	10.4	7.4	11.8	7.3	**6.1**	5.1	27.3	4.0	29.1	4.6	24.9	4.0	10.2	9.5	18.1	2.3	20.1	3.4
9	**Dermatology**	**12.7**	3.5	14.0	3.1	13.8	3.0	13.2	3.3	17.5	2.6	19.1	1.9	18.3	2.8	18.1	3.8	13.8	1.9	14.6	2.4
10	**Sonar**	17.9	7.3	18.9	5.8	22.0	7.6	**15.5**	5.1	31.0	3.0	34.1	4.4	28.1	4.1	23.0	8.4	22.4	3.6	24.6	3.3
11	**Breast EW**	11.7	4.5	12.5	4.6	13.5	4.0	**9.9**	4.4	16.4	2.7	18.6	2.4	14.5	3.0	14.9	6.0	13.9	2.2	13.4	2.7
12	**Soybean-small**	**2.0**	0.6	3.1	2.1	2.7	1.5	2.7	1.5	11.3	1.0	12.1	2.1	14.3	2.4	5.7	3.7	7.2	1.9	7.3	2.1
13	**Movement_libras**	26.7	8.0	34.7	10.2	27.8	8.1	**21.9**	10.0	46.7	4.4	54.8	3.2	43.5	4.4	35.6	16.7	36.8	4.9	38.9	4.8
14	**Spambase**	34.9	5.8	34.6	7.0	34.9	6.5	**23.6**	4.9	32.9	4.1	43.3	2.7	28.8	3.4	36.3	8.3	28.8	3.4	28.5	3.5
15	**Arrhythmia**	94.5	38.2	95.6	44.2	103.2	36.3	**69.0**	30.8	150.4	8.2	185.7	11.1	137.2	9.1	76.2	51.1	116.4	9.2	129.8	7.1
16	**Clean1**	59.9	21.6	76.6	24.4	72.6	19.7	**49.7**	23.3	91.8	6.2	111.2	8.6	80.8	6.4	86.4	28.3	71.1	6.0	76.4	6.7
17	**Hill valley**	**29.5**	12.0	32.7	15.7	34.8	20.4	**17.3**	12.4	54.5	5.6	71.3	5.3	49.3	5.2	20.2	17.0	46.7	4.9	46.7	5.5
18	**Leukemia**	**99.0**	118.9	154.6	147.4	228.2	360.0	116.0	149.2	3709.8	220.9	3680.4	263.1	3494.2	35.9	190.9	223.9	3220.5	53.7	3353.8	48.5
19	**Colon**	36.5	28.4	76.1	95.0	87.1	113.5	**26.4**	34.1	1047.3	57.3	1078.1	102.7	985.3	23.0	68.6	99.1	839.4	23.2	913.2	23.4
20	**Arcene**	644.8	462.4	714.7	658.4	805.1	1080.8	**490.2**	657.7	5446.1	253.9	5507.2	473.1	4966.0	50.8	658.0	604.6	4643.1	53.5	4832.1	54.7
21	**Lymphoma**	**57.5**	49.3	106.8	105.2	137.8	178.5	104.5	98.5	2069.5	126.4	2092.0	181.9	1973.2	26.7	195.9	299.8	1766.6	28.4	1877.2	37.4
22	**Smk_can_187**	1108.9	1706.4	1339.7	1979.9	1526.6	2375.0	**499.9**	1454.4	11153.1	341.2	11562.6	1069.4	9962.1	79.8	1847.6	3017.1	9565.0	84.5	9799.0	79.1
23	**Tox_171**	1541.2	623.8	1427.1	591.0	1585.0	763.3	**971.5**	728.9	3219.3	74.8	3780.2	193.7	2855.1	32.7	1339.2	982.5	2700.6	45.4	2812.8	34.2
24	**Coil20**	135.5	64.7	206.5	119.1	154.3	87.5	**111.2**	58.6	555.3	19.5	632.3	53.4	505.3	13.6	145.9	80.0	436.5	22.0	472.2	20.1
	Avg Rank	2.4		3.6		4.4		**1.3**		8.4		9.8		8.2		5.1		5.6		6.2	
	Final Rank	2.0		3.0		4.0		**1.0**		9.0		10.0		8.0		5.0		6.0		7.0	

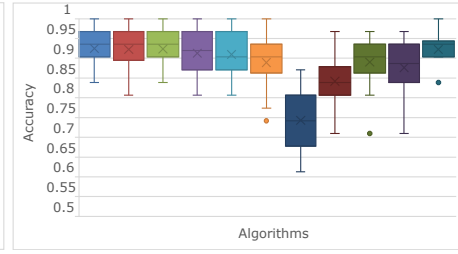
Table 5. Dimensions selected – comparison over proposed methods vs. existing methods

the bGWO the elongated whiskers and box sizes indicate a higher deviation from the median. This depicts the instability of the algorithm under various conditions and may fail to perform consistently under all conditions. From the plots, it is clearly visible that SCVSSA has less deviation and is more stable than the other algorithms in most of the datasets. The algorithm's minimum and maximum accuracies are not highly deviated from the median of the algorithm. Thus the algorithm can be termed more stable and has a good combination of exploration and exploitation under various conditions.

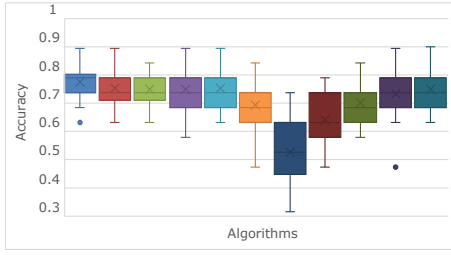
a) Color coding for the box plots



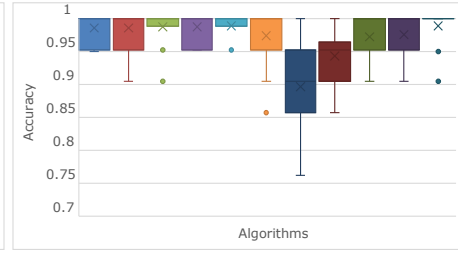
b) Wine



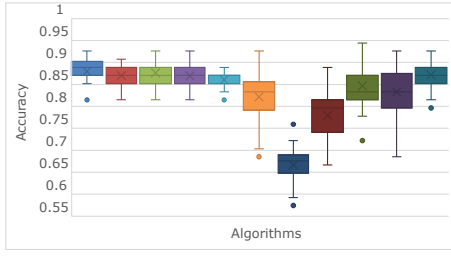
c) Hepatits



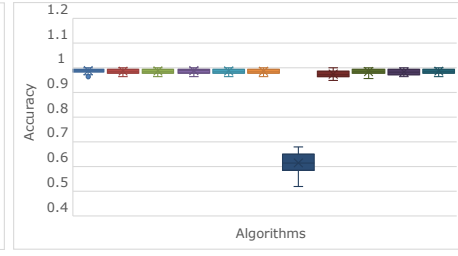
d) Vehicle



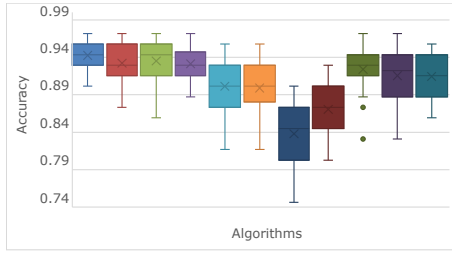
e) Zoo



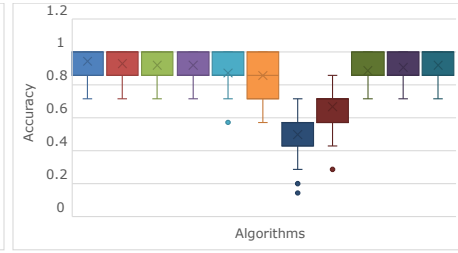
f) Heart disease



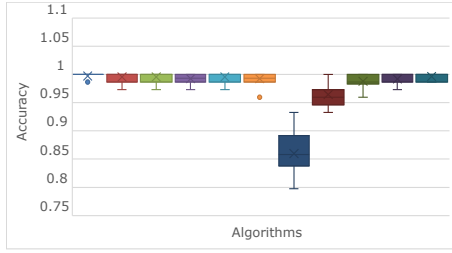
g) Wisconson



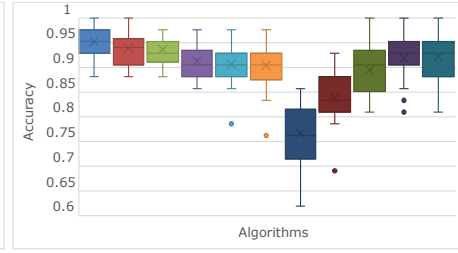
h) Ionosphere



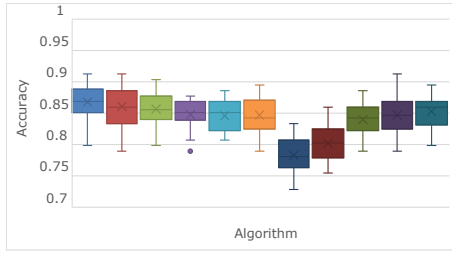
i) Lung-Cancer



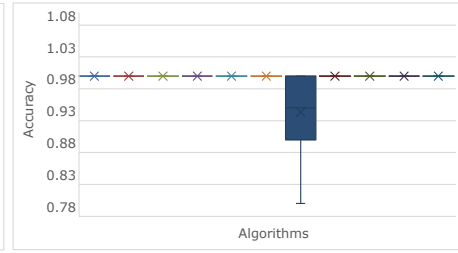
j) Dermatology



k) Sonar



l) Breast EW



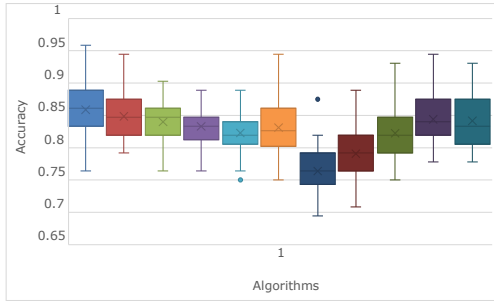
m) Soybean-small

Figure 11. Box plot comparison – proposed vs. existing meta heuristic algorithms over dataset (1–12)

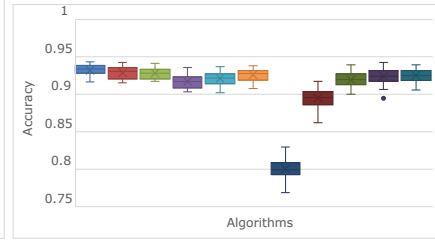
5.2 Comparison over Filter Methods

As discussed in Section 1, the filter models are independent of the classification algorithm apart from the wrapper models. These filter models are mostly used to rank the features based on its characteristics. Once the features are being ranked the first n feature would be chosen and the classification would be performed on it. The correlation-based feature selection CFS [17] uses the statistical measure of correlation to rank the feature. Other filter algorithms used for the comparison are Laplacian [38], F-score [39], relieff [40] and mutinffs [41]. The SCVSSA algorithm which is selected as the best algorithm from the previous findings is subjected to

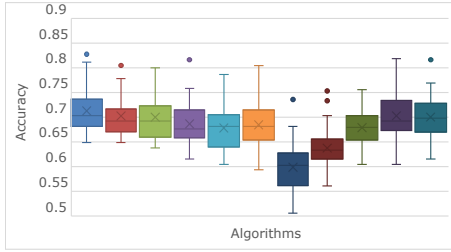
a) Color coding for the box plots



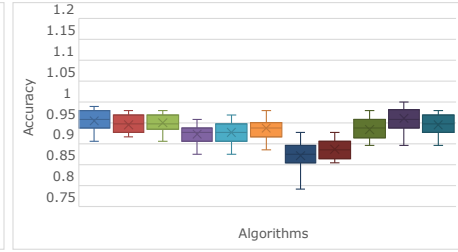
b) Movement libras



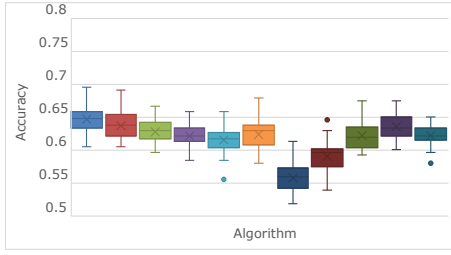
c) Spambase



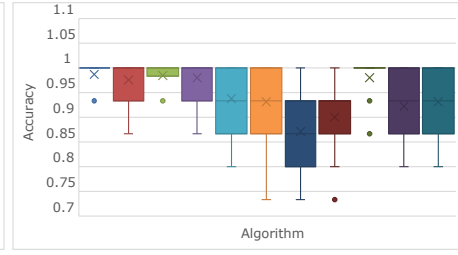
d) Arrhythmia



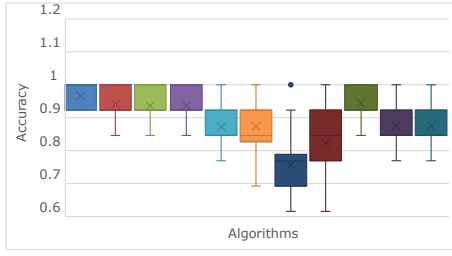
e) Clean1



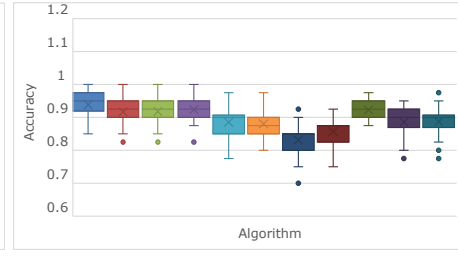
f) Hill valley



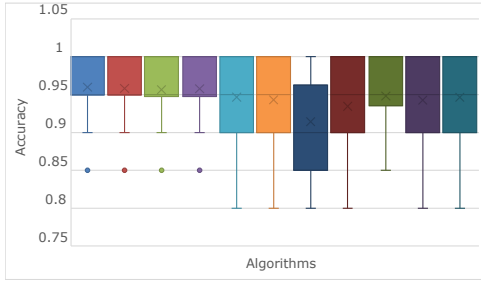
g) Leukemia



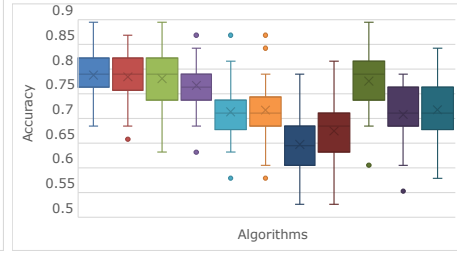
h) Colon



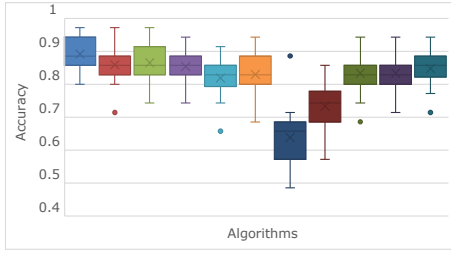
i) Arcene



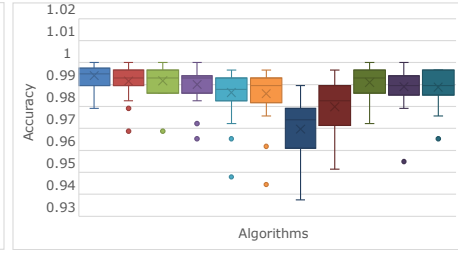
j) Lymphoma



k) Smk can 187



l) Tox 171



m) Coil20

Figure 12. Box plot comparison – proposed vs. existing meta heuristic algorithms over dataset (13-24)

test against the filter models. Once the features are ranked using the filter models the same number of n features which has been obtained by the SCVSSA has been filtered from the respective algorithms and subjected to the KNN classifier whose accuracy is enumerated in Table 6. From Table 5 it is visible that the SCVSSA has cleanly surpassed all the filter algorithms over all the datasets. From Figure 13 which summarizes the average accuracy of all the algorithms, it can be inferred that the SCVSSA is far better than the most commonly used filter models for feature selection.

6 CONCLUSION AND FUTURE WORK

In this paper, the SSA's performance has been improved by incorporating the vigilant mechanism adopted from the GWO. In addition to the above enhancement, two different crossover methods equivalent to the follower position update strategy of the SSA were applied. The main contribution of the paper is the adoption of a vigilant mechanism and shuffled crossover mechanism over the SSA. The effectiveness of this algorithm is tested by subjecting the proposed algorithms to the standard benchmark datasets downloaded from the UCI machine learning repository and ASU feature selection repository. The datasets were chosen such that they

Sl.	Algorithm	relieff		laplacian		f-Score		cfs		mutinffs		SCVSSA	
no.	Dataset	avg	std	avg	std	avg	std	avg	std	avg	std	avg	std
1	**Wine**	0.8190	0.1054	0.7362	0.0599	0.7352	0.1169	0.5324	0.1764	0.7333	0.1116	**0.9843**	0.0174
2	**Hepatitis**	0.7387	0.0773	0.7591	0.0775	0.7785	0.0756	0.7667	0.0671	0.7903	0.0591	**0.9247**	0.0451
3	**Vehicle**	0.4778	0.0858	0.5056	0.1224	0.4722	0.0782	0.3833	0.1490	0.4926	0.1146	**0.7772**	0.0826
4	**Zoo**	0.8161	0.1022	0.8223	0.1345	0.6279	0.1451	0.4689	0.2628	0.7729	0.1008	**0.9856**	0.0223
5	**Heart disease**	0.6877	0.0736	0.6519	0.0720	0.6790	0.0666	0.6179	0.1005	0.7111	0.0434	**0.8790**	0.0332
6	**Wisconson**	0.9417	0.0221	0.6064	0.0376	0.7806	0.1731	0.7498	0.1526	0.6507	0.1121	**0.9881**	0.0093
7	**Ionosphere**	0.8376	0.0560	0.8100	0.0370	0.8895	0.0278	0.7176	0.0979	0.9010	0.0356	**0.9427**	0.0215
8	**Lung-cancer**	0.4383	0.2352	0.4195	0.1644	0.4261	0.2052	0.5144	0.1699	0.4006	0.2209	**0.9429**	0.0888
9	**Dermatology**	0.7995	0.0752	0.8365	0.1060	0.7164	0.0746	0.5973	0.1673	0.8068	0.1061	**0.9973**	0.0055
10	**Sonar**	0.7268	0.0900	0.4992	0.0643	0.7797	0.0658	0.5260	0.0768	0.7707	0.0503	**0.9524**	0.0319
11	**Breast EW**	0.7336	0.0501	0.7003	0.0413	0.7106	0.0481	0.6850	0.0753	0.7103	0.0475	**0.8678**	0.0276
12	**Soybean-small**	0.5253	0.1386	0.3577	0.1628	0.3444	0.2486	0.4098	0.2148	0.4481	0.1384	**1.0000**	0.0000
13	**Movement_libras**	0.7440	0.0538	0.7093	0.0739	0.2796	0.0830	0.2319	0.0670	0.4519	0.1069	**0.8588**	0.0398
14	**Spambase**	0.8592	0.0194	0.8809	0.0206	0.8389	0.0136	0.7816	0.0469	0.8398	0.0129	**0.9322**	0.0073
15	**Arrhythmia**	0.5884	0.0508	0.5840	0.0387	0.5617	0.0432	0.5634	0.0481	0.5939	0.0467	**0.7126**	0.0446
16	**Clean1**	0.8312	0.0448	0.8316	0.0455	0.8463	0.0287	0.7958	0.0318	0.8470	0.0288	**0.9549**	0.0224
17	**Hill valley**	0.5326	0.0306	0.5620	0.0306	0.5317	0.0279	0.5565	0.0240	0.5263	0.0333	**0.6443**	0.0218
18	**Leukemia**	0.8119	0.1036	0.8000	0.1051	0.7929	0.1033	0.8143	0.1103	0.8071	0.1030	**0.9867**	0.0271
19	**Colon**	0.6639	0.1147	0.7250	0.1279	0.7556	0.1093	0.6639	0.1443	0.6972	0.1392	**0.9667**	0.0388
20	**Arcene**	0.8092	0.0498	0.8075	0.0595	0.8017	0.0704	0.7950	0.0628	0.7925	0.0743	**0.9383**	0.0381
21	**Lymphoma**	0.8004	0.1028	0.8347	0.0801	0.7907	0.1118	0.7516	0.0895	0.8394	0.1119	**0.9598**	0.0403
22	**Smk_can_187**	0.6297	0.0758	0.6477	0.0664	0.6162	0.0907	0.0000	1.0000	0.6423	0.0776	**0.7877**	0.0521
23	**Tox_171**	0.6578	0.1060	0.6569	0.0804	0.6598	0.0744	0.6471	0.0799	0.6569	0.0652	**0.8914**	0.0465
24	**Coil20**	0.9674	0.0100	0.9306	0.0189	0.9701	0.0133	0.7925	0.0508	0.9609	0.0142	**0.9941**	0.0050
	Avg Rank	3.30		3.87		4.00		5.08		3.65		**1.00**	
	Final Rank	2		4		5		6		3		**1**	

Table 6. Accuracy – comparison over filter models

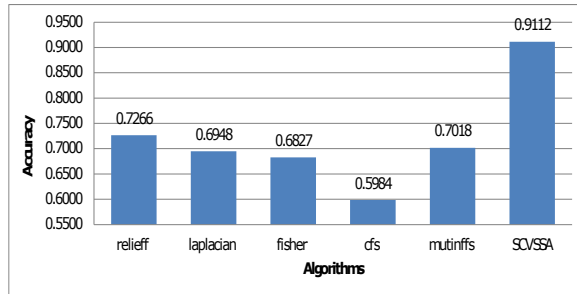


Figure 13. Comparison of average accuracy over filter models

possess various proportions of dimensions and a number of instances. To prove the proficiency of the VSSA and enhanced versions of VSSA, they were compared with the original SSA, its other hybrids and other promising meta-heuristic algorithms. The comparison and the analysis of results certainly portray that the SCVSSA could be adopted for feature selection to obtain good accuracy with the least number of features. The future direction of this work can be carried out by introducing and investigating the transfer function for the conversion of binary vectors. This wrapper model is well suited to be adopted as a pre-processing amenity for feature selection before applying a machine learning classifier.

REFERENCES

[1] Cai, J.—Luo, J.—Wang, S.—Yang, S.: Feature Selection in Machine Learning: A New Perspective. Neurocomputing, Vol. 300, 2018, pp. 70–79, doi: 10.1016/j.neucom.2017.11.077.

[2] Pedersen, M. E. H.—Chipperfield, A. J.: Simplifying Particle Swarm Optimization. Applied Soft Computing, Vol. 10, 2010, No. 2, pp. 618–628, doi: 10.1016/j.asoc.2009.08.029.

[3] Urahama, K.—Furukawa, Y.: Gradient Descent Learning of Nearest Neighbor Classifiers with Outlier Rejection. Pattern Recognition, Vol. 28, 1995, No. 5, pp. 761–768, doi: 10.1016/0031-3203(94)00142-9.

[4] Mirjalili, S.—Gandomi, A. H.—Mirjalili, S. Z.—Saremi, S.—Faris, H.—Mirjalili, S. M.: Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. Advances in Engineering Software, Vol. 114, 2017, pp. 163–191, doi: 10.1016/j.advengsoft.2017.07.002.

[5] Mirjalili, S.—Mirjalili, S. M.—Lewis, A.: Grey Wolf Optimizer. Advances in Engineering Software, Vol. 69, 2014, pp. 46–61, doi: 10.1016/j.advengsoft.2013.12.007.

[6] Man, K. F.—Tang, K. S.—Kwong, S.: Genetic Algorithms: Concepts and Applications [in Engineering Design]. IEEE Transactions on Industrial Electronics, Vol. 43, 1996, No. 5, pp. 519–534, doi: 10.1109/41.538609.

[7] Mafarja, M. M.—Mirjalili, S.: Hybrid Whale Optimization Algorithm with Simulated Annealing for Feature Selection. Neurocomputing, Vol. 260, 2017, pp. 302–312, doi: 10.1016/j.neucom.2017.04.053.

[8] Kashef, S.—Nezamabadi-pour, H.: An Advanced ACO Algorithm for Feature Subset Selection. Neurocomputing, Vol. 147, 2015, pp. 271–279, doi: 10.1016/j.neucom.2014.06.067.

[9] Kennedy, J.—Eberhart, R.: Particle Swarm Optimization. Proceedings of ICNN '95 – International Conference on Neural Networks, IEEE, Vol. 4, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[10] Rajabioun, R.: Cuckoo Optimization Algorithm. Applied Soft Computing, Vol. 11, 2011, pp. 5508–5518, doi: 10.1016/j.asoc.2011.05.008.

[11] Arunekumar, N. B.—Kumar, A.—Joseph, K. S.: Hybrid Bat Inspired Algorithm for Multiprocessor Real-Time Scheduling Preparation. 2016 International Conference on Communication and Signal Processing (ICCSP), IEEE, 2016, pp. 2194–2198, doi: 10.1109/ICCSP.2016.7754572.

[12] Marie-Sainte, S. L.—Alalyani, N.: Firefly Algorithm Based Feature Selection for Arabic Text Classification. Journal of King Saud University – Computer and Information Sciences, Vol. 32, 2020, No. 3, pp. 320–328, doi: 10.1016/j.jksuci.2018.06.004.

[13] Simon, D.: Biogeography-Based Optimization. IEEE Transactions on Evolutionary Computation, Vol. 12, 2008, No. 6, pp. 702–713, doi: 10.1109/TEVC.2008.919004.

[14] Mirjalili, S.—Lewis, A.: The Whale Optimization Algorithm. Advances in Engineering Software, Vol. 95, 2016, pp. 51–67, doi: 10.1016/j.advengsoft.2016.01.008.

[15] Güneş, S.—Polat, K.—Yosunkaya, S.: Multi-Class F-Score Feature Selection Approach to Classification of Obstructive Sleep Apnea Syndrome. Expert Systems with Applications, Vol. 37, 2010, No. 2, pp. 998–1004, doi: 10.1016/j.eswa.2009.05.075.

[16] Wold, S.—Esbensen, K.—Geladi, P.: Principal Component Analysis. Chemometrics and Intelligent Laboratory Systems, Vol. 2, 1987, No. 1-3, pp. 37–52, doi: 10.1016/0169-7439(87)80084-9.

[17] Hall, M. A.: Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning. Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), 2000, pp. 359–366.

[18] Oh, I. S.—Lee, J. S.—Moon, B. R.: Hybrid Genetic Algorithms for Feature Selection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, 2004, No. 11, pp. 1424–1437, doi: 10.1109/TPAMI.2004.105.

[19] Mistry, K.—Zhang, L.—Neoh, S. C.—Lim, C. P.—Fielding, B.: A Micro-GA Embedded PSO Feature Selection Approach to Intelligent Facial Emotion Recognition. IEEE Transactions on Cybernetics, Vol. 47, 2017, No. 6, pp. 1496–1509, doi: 10.1109/TCYB.2016.2549639.

[20] Panwar, L. K.—Reddy, S.—Verma, A.—Panigrahi, B. K.—Kumar, R.: Binary Grey Wolf Optimizer for Large Scale Unit Commitment Problem. Swarm and Evolutionary Computation, Vol. 38, 2018, pp. 251–266, doi: 10.1016/j.swevo.2017.08.002.

[21] Kennedy, J.—Eberhart, R. C.: A Discrete Binary Version of the Particle Swarm Algorithm. 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Vol. 5, 1997, pp. 4104–4108, doi: 10.1109/ICSMC.1997.637339.

[22] Emary, E.—Zawbaa, H. M.—Hassanien, A. E.: Binary Ant Lion Approaches for Feature Selection. Neurocomputing, Vol. 213, 2016, pp. 54–65, doi: 10.1016/j.neucom.2016.03.101.

[23] Mirjalili, S.—Lewis, A.: S-Shaped Versus V-Shaped Transfer Functions for Binary Particle Swarm Optimization. Swarm and Evolutionary Computation, Vol. 9, 2013, pp. 1–14, doi: 10.1016/j.swevo.2012.09.002.

[24] Faris, H.—Mafarja, M. M.—Heidari, A. A.—Aljarah, I.—Al-Zoubi, A. M.—Mirjalili, S.—Fujita, H.: An Efficient Binary Salp Swarm Algorithm with Crossover Scheme for Feature Selection Problems. Knowledge-Based Systems, Vol. 154, 2018, pp. 43–67, doi: 10.1016/j.knosys.2018.05.009.

[25] Mafarja, M.—Aljarah, I.—Heidari, A. A.—Faris, H.—Fournier-Viger, P.—Li, X.—Mirjalili, S.: Binary Dragonfly Optimization for Feature Selection Using Time-Varying Transfer Functions. Knowledge-Based Systems, Vol. 161, 2018, pp. 185–204, doi: 10.1016/j.knosys.2018.08.003.

[26] Ren, W.—Ma, D.—Han, M.: Multivariate Time Series Predictor with Parameter Optimization and Feature Selection Based on Modified Binary Salp Swarm Algorithm. IEEE Transactions on Industrial Informatics, Vol. 19, 2023, No. 4, pp. 6150–6159, doi: 10.1109/TII.2022.3198465.

[27] Hegazy, A. E.—Makhlouf, M. A.—El-Tawel, G. S.: Improved Salp Swarm Algorithm for Feature Selection. Journal of King Saud University – Computer and Information Sciences, Vol. 32, 2020, No. 3, pp. 335–344, doi: 10.1016/j.jksuci.2018.06.003.

[28] Aljarah, I.—Mafarja, M.—Heidari, A. A.—Faris, H.—Zhang, Y.—Mirjalili, S.: Asynchronous Accelerating Multi-Leader Salp Chains for Feature Selection. Applied Soft Computing, Vol. 71, 2018, pp. 964–979, doi: 10.1016/j.asoc.2018.07.040.

[29] Faraji, R.—Naji, H. R.: An Efficient Crossover Architecture for Hardware Parallel Implementation of Genetic Algorithm. Neurocomputing, Vol. 128, 2014, pp. 316–327, doi: 10.1016/j.neucom.2013.08.035.

[30] Kaya, M.: The Effects of Two New Crossover Operators on Genetic Algorithm Performance. Applied Soft Computing, Vol. 11, 2011, No. 1, pp. 881–890, doi: 10.1016/j.asoc.2010.01.008.

[31] Arora, S.—Anand, P.: Binary Butterfly Optimization Approaches for Feature Selection. Expert Systems with Applications, Vol. 116, 2018, pp. 147–160, doi: 10.1016/j.eswa.2018.08.051.

[32] Dheeru, D.—Taniskidou, E. K.: UCI Machine Learning Repository. 2017, `http://archive.ics.uci.edu/ml`.

[33] Li, J.—Cheng, K.—Wang, S.—Morstatter, F.—Trevino, R. P.—Tang, J.—Liu, H.: Feature Selection: A Data Perspective. ACM Computing Surveys (CSUR), Vol. 50, 2017, No. 6, Art. No. 94, doi: 10.1145/3136625.

[34] Emary, E.—Zawbaa, H. M.—Hassanien, A. E.: Binary Grey Wolf Optimization

Approaches for Feature Selection. Neurocomputing, Vol. 172, 2016, pp. 371–381, doi: 10.1016/j.neucom.2015.06.083.

[35] Saremi, S.—Mirjalili, S.—Lewis, A.: Grasshopper Optimisation Algorithm: Theory and Application. Advances in Engineering Software, Vol. 105, 2017, pp. 30–47, doi: 10.1016/j.advengsoft.2017.01.004.

[36] Mirjalili, S.: The Ant Lion Optimizer. Advances in Engineering Software, Vol. 83, 2015, pp. 80–98, doi: 10.1016/j.advengsoft.2015.01.010.

[37] Eberhart, R. C.—Shi, Y.: Particle Swarm Optimization: Developments, Applications and Resources. Proceedings of the 2001 Congress on Evolutionary Computation, IEEE, Vol. 1, pp. 81–86, doi: 10.1109/CEC.2001.934374.

[38] He, X.—Cai, D.—Niyogi, P.: Laplacian Score for Feature Selection. In: Weiss, Y., Schölkopf, B., Platt, J. (Eds.): Advances in Neural Information Processing Systems 18 (NIPS 2005). MIT Press, 2005, pp. 507–514.

[39] Song, Q.—Jiang, H.—Liu, J.: Feature Selection Based on FDA and F-Score for Multi-Class Classification. Expert Systems with Applications, Vol. 81, 2017, pp. 22–27, doi: 10.1016/j.eswa.2017.02.049.

[40] Robnik-Šikonja, M.—Kononenko, I.: Theoretical and Empirical Analysis of ReliefF and RReliefF. Machine Learning, Vol. 53, 2003, No. 1-2, pp. 23–69, doi: 10.1023/A:1025667309714.

[41] Zaffalon, M.—Hutter, M.: Robust Feature Selection by Mutual Information Distributions. Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UCI 2002), Morgan Kaufmann Publishers Inc., 2002, pp. 577–584.

N. B. Arunekumar received his B.E. in computer science and engineering from the Anna University and M.Tech. in computer science and engineering from the Pondicherry University in 2012 and 2016. He completed his Ph.D. at the Department of Computer Science and Engineering, Pondicherry University in 2022. He is working as Assistant Professor in the Department of Artificial Intelligence and Data Science, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India. His area of interests are optimization, machine learning and deep learning.

K. Suresh Joseph received his B.E. in computer science and engineering from the Bharathiar University in 1999 and M.E. in computer science and engineering at the University of Madras in 2003. He completed his Ph.D. in computer science and engineering at the Anna University in 2013. Currently, he is working as Associate Professor in the Department of Computer Science and Engineering, Pondicherry University. His area of interest are soft computing and NDN.

J. Viswanath is currently working as Assistant Professor at the Department of Artificial Intelligence and Data Science, Madanapalle Institute of Technology and Science (MITS), Andhrapradesh. He completed his Master's degree in software engineering in 2011 at the Hindustan University, Chennai. He also completed his Bachelor's degree in information technology at SRM University, Chennai. His area of interest are software engineering, artificial intelligence and data science.

A. Anbarasi has completed her Ph.D. in computer science and engineering at the Pondicherry University. She has completed her M.Tech. in distributed computing systems at the Pondicherry Engineering College and B.E. in computer science and engineering in the V.R.S. College of Engineering and Technology. Currently, she is working as Assistant Professor in the Department of Computing Technologies, at SRM Institute of Science and Technology, Kattankulathoor, TN, India.

N. Padmapriya completed her M.Sc. (five-year integrated programme) in statistics at the Pondicherry University in 2013. She qualified CSIR-NET in the year 2017. Currently, she is working as Assistant Professor in the Department of Statistics, Sri Sarada College for Women (Autonomous), Salem-16 and pursuing her Ph.D. in the Department of Statistics, Pondicherry University.