 doi:10.20944/preprints202001.0318.v1

Efficient Binary Symbiotic Organisms Search Algorithm Approaches for Feature Selection Problems

Hekmat Mohmmadzadeh[1], Farhad Soleimanian Gharehchopogh[2]

[1,2]Departent of Computer Engineering, Urmia Brach, Islamic Azad University, Urmia, IRAN.

Abstract

Feature selection is one of the main data preprocessing steps in machine learning. Its goal is to reduce the number of features by removing extra and noisy features. Feature selection methods must consider the accuracy of classification algorithms while performing feature reduction on a dataset. Meta-heuristic algorithms are the most successful and promising methods for solving this issue. The symbiotic organisms search algorithm is one of the successful meta-heuristic algorithms which is inspired by the interaction of organisms in the nature called Parasitism Commensalism Mutualism. In this paper, three engulfing binary methods based on the symbiotic organisms search algorithm are presented for solving the feature selection problem. In the first and second methods, several S-shaped and V-shaped transfer functions are used for binarizing the symbiotic organisms search algorithm, respectively. These methods are called BSOSS and BSOSV. In the third method, two new operators called BMP and BCP are presented for binarizing the symbiotic organisms search algorithm. This method is called EBSOS. The third approach presents an advanced binary version of the coexistence search algorithm with two new operators, BMP and BCP, to solve the feature selection problem, named EBSOS. The proposed methods are run on 18 standard UCI datasets and compared to base and important meta-heuristic algorithms. The test results show that the EBSOS method has the best performance among the three proposed approaches for binarization of the coexistence search algorithm. Finally, the proposed EBSOS approach was compared to other meta-heuristic methods including the genetic algorithm, binary bat algorithm, binary particle swarm algorithm, binary flower pollination algorithm, binary grey wolf algorithm, binary dragonfly algorithm, and binary chaotic crow search algorithm. The results of different experiments showed that the proposed EBSOS approach has better performance compared to other methods in terms of feature count and accuracy criteria. Furthermore, the proposed EBSOS approach was practically evaluated on spam email detection in particular. The results of this experiment also verified the performance of the proposed EBSOS approach. In addition, the proposed EBSOS approach is particularly combined with the classifiers including SVM, KNN, NB and MLP to evaluate this method performance in the detection of spam emails. The obtained results showed that the proposed EBSOS approach has significantly improved the accuracy and speed of all the classifiers in spam email detection.

Keywords: *Efficient Binary Symbiotic, Feature selection, Classification, Optomization.*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 January 2020 doi:10.20944/preprints202001.0318.v1

1. Introduction

In real-world problems, the existence of datasets with high dimensionality and also useless and extra data has made the process of analyzing these data challenging. Feature selection is one of the preprocessing steps in machine learning which can remove useless and irrelevant features from a dataset and find the ultimate subset of important features which leads to the better performance of machine learning algorithms[1, 2]. In fact, feature selection is an important and common method in data mining and machine learning for dimensionality reduction by eliminating irrelevant and redundant information from the dataset for achieving the optimal feature subset which leads to an increase in the speed and accuracy of classification algorithms[2, 3]. However, obtaining the optimal feature subset is posed as a complex optimization problem and conventional methods are unable to solve this problem. In fact, the goal of feature selection is finding a set of m features from the full set of n features which improves the performance of the learning algorithm in terms of learning speed or classification accuracy.

Until now, two frameworks, including search-based feature selection and correlation-based feature selection, have been proposed for solving the feature selection problem efficiently[2]. Search strategy and evaluation criterion are the two key components in the first set of methods. The search strategy specifies how the solution is generated for an optimal feature subset. Each generated solution is evaluated using a specified criterion. Of course, in this strategy, search methods try to work better in later iterations and the subset generation and evaluation process are repeated until a stopping criterion is met. Unlike the first set of methods, in the second set, the abundance, relationship, and correlation between features are used for identifying

useless and extra features. In other studies[3, 4], however, feature selection approaches fall into two main categories: filter-based methods and coating-based methods. The filter-based method usually uses the correlation between data for finding the optimal feature subset. Filter-based methods are independent of the classification algorithm and work relatively fast. However, coating-based algorithms involve the classification algorithm in the evaluation criterion in order to present an optimal solution for the feature selection method. In figure 1., the overview of filter-based and coating-based methods is presented.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 January 2020 doi:10.20944/preprints202001.0318.v1

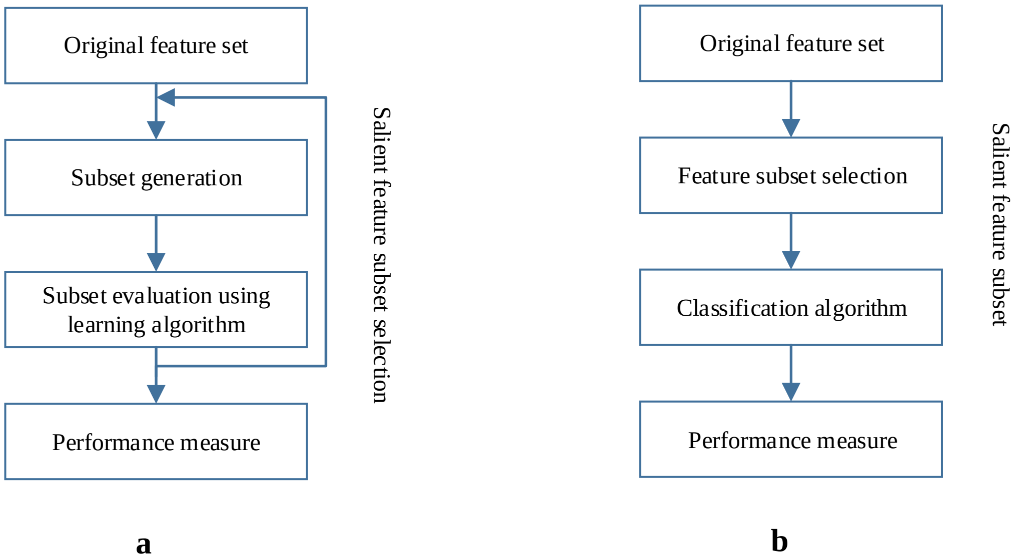

Fig. 1. Action diagram (a) wrapper approach (b) filter approach[5].

The researchers have found out[4-9] that coating-based methods obtain better results compared to filter-based methods because they utilize the classification algorithm in their evaluation model. Coating models take advantage of meta-heuristic algorithms. Of course, nowadays, metaa-heuristic algorithms have proven themselves useful for most complex computational and optimization problems. These efficient and reliable methods are for finding near-optimal solutions with a reasonable computational cost. Of course, most of these algorithms are inspired by the behavior of creatures, animals' hunting, or nature[10, 11]. At the beginning of meta-heuristic algorithm generations, they use exploration to generate new solutions and try to gradually decrease exploration as the generation comes closer to its end. On the other hand, they use exploitation to generate new solutions around the solutions they have already discovered. Therefore, meta-heuristic algorithms use the two exploration and exploitation operations to prevent being trapped in a local minimum and converge to the target.

Symbiotic organisms search is one of the successful and promising meta-heuristic algorithms inspired by the encounters of animals in nature which was presented by Prayogo and Cheng in 2014. This algorithm has three separate and powerful processes called mutualism, commensalism, and parasitism which improve the solutions found in the population. Also, there are no parameters for tuning these two actions between exploration and exploitation. For this reason, it has been able to solve most optimization problems successfully. Due to the novelty and superior results of this algorithm in solving optimization problems, we were encouraged to present three different binary variations of this algorithm in this paper for solving binary optimization problems. In this paper, three binary coating-based symbiotic organisms search methods are presented for solving the feature selection problem. In the first method, we used multiple Sigmoid transfer functions (S-shaped) to move the symbiotic organisms search algorithm in the binary space. Then, we used another transfer function called the V-shaped transfer function to move the symbiotic organisms search algorithm in the binary space. In both of these methods, we used a simple transfer function to transform a continuous space to binary. We did this to show that a transfer function can be used with the least amount of modification to the operations of the symbiotic organisms search algorithm to present a coating-based method for moving in the binary space. Furthermore, in the third method, two new operators called BMP and BCP were presented for making the advanced binary version of the symbiotic organisms search algorithm called EBSOS. In the EBSOS approach, our goal is to apply some changes to the structure of the operators of the symbiotic organisms search algorithm based on various new operators while keeping the rules present in the base symbiotic organisms search algorithm. In this method, of course, we have tried for exploration and exploitation to be upheld.

In the rest of this paper, binary versions of various meta-heuristic algorithms are reviewed in section 2. In section 3, the fundamental concepts of the symbiotic organisms search algorithm and its steps are explained in detail. In section 4, the three proposed binary approaches based on the symbiotic organisms search algorithm are presented. In section 5, the efficiency and performance of the proposed approaches will be tested. In the final section of the paper, the overall conclusion and future work will be presented.

2. Previous Work

In this section, we will review the papers about feature selection. Of course, we have comprehensively described the methods of transforming continuous meta-heuristic algorithms to binary in table 1. and described the differences and operators of our proposed methods in the end as well. Since our proposed method is a coating-based method, we will mostly review papers which are coating-based and have used different transfer functions. In 2013, a particle swarm algorithm based on two V-Shaped and S-shaped transfer functions was presented by Mirjalili et al.[12]. The proposed method was run on 2005 benchmark functions and the results of the proposed method were promising compared to other methods. In another research in 2013, a binary cuckoo search algorithm was presented for feature selection[13]. In this study, only the S-shaped transfer function was used for transforming the continuous space to binary. Finally, the proposed method was run on two datasets which showed that the proposed method performs better than base binary algorithms like binary bat algorithm, binary particle swarm, binary firefly algorithm, and binary gravitational search algorithm.

Table 1: *Investigating some research on the problem of feature selection with binary continuous space conversion methods*

Reference	Authors	Meta-heuristic algorithm	Suggested method for conversion
[12]	(Mirjalili & Lewis, 2013)	Particle swarm optimization	V-shaped S-shaped
[13]	(Rodrigues et al., 2013)	Cuckoo search algorithm	S-shaped
[14]	(Mirjalili, Mirjalili, & Yang, 2014)	Bat algorithm	V-shaped S-shaped
[8]	(Emary, Zawbaa, & Hassanien, 2016b)	Grey wolf optimization	S-shaped Crossover
[7]	(Emary, Zawbaa, & Hassanien, 2016a	Ant lion optimization	V-shaped S-shaped Crossover

[15]	(Mafarja & Mirjalili, 2018)	Whale optimization algorithm	Crossover Mutation
[6]	(Faris et al., 2018)	Binary salp swarm algorithm	Types of-V-shaped Types of-S-shaped
[4]	(Arora & Anand, 2019)	Butterfly optimization algorithm	V-shaped S-shaped
[1]	(Mafarja et al., 2019)	Grasshopper optimisation algorithm	V-shaped S-shaped Crossover Mutation
-	This Article	Symbiotic Organisms Search	Types of-V-shaped Types of-S-shaped New Operator(BCP) New operator(BMP) Crossover Mutation

In[14], a binary bat algorithm based on two V-shaped and S-shaped transfer functions was presented by Mirjalili et al. in 2014. Experiments were carried out on 22 benchmark functions and the results showed that the binary bat algorithm performed significantly better compared to the genetic and particle swarm algorithms. Also, the proposed algorithm performed better in real-world problems as well.

In 2016, two binary grey wolf optimization approaches were presented for feature selection[8]. In the first approach, the composition operator is used for updating the operators of the grey wolf optimization algorithm. In the second approach, the sigmoidal function is used to move the grey wolf optimization algorithm in the binary space and finally, random thresholding is carried out to convert the solutions to binary. To evaluate and compare the proposed and other methods, 18 different datasets from the UCI repository were used. The simulation results indicated the superiority of the first method more. Also, in another research, a binary ant lion algorithm was presented for feature selection[7]. In this research, two types of approaches were studied for binarizing the binary ant lion algorithm. In the first approach, the composition operator was used and in the second approach, the S-shaped and V-shaped transfer functions were used. The experiments we applied on 21 datasets and the results showed that the proposed algorithm based on the composition operator has presented acceptable results.

Mafarja and Mirjalali presented two different approaches of the whale optimization algorithm in 2018 for coating feature selection. In this study, genetic operators are used to binarize the whale optimization algorithm[15]. Furthermore, in an approach called WOA-T, the tournament selection and in an approach called WOA-R, the roulette wheel operator is used. In the main method, the mutation and crossover operators are used simultaneously to move the whale optimization algorithm. Finally, the proposed methods are tested on standard datasets and compared to filter-based and coating-based algorithms. The results verify the superiority of the proposed algorithms. In another research in 2018, the efficient binary Salp swarm algorithm with the composition method was presented by Faris et al. for solving the feature selection problem[6]. Two different approaches

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 January 2020 doi:10.20944/preprints202001.0318.v1

were presented in this study as well. In the first approach, eight binary transfer functions are used for converting the continuous version of the Salp swarm algorithm to binary. In the second approach, the composition operator was used to replace the ordinary operator and increasing the exploration behavior of the algorithm in addition to the transfer functions. Finally, different tests verify the superiority of the proposed algorithms.

In the most recent research in 2019, Arora and Anand presented two binary approaches to the impulse optimization algorithm[4]. In the first approach, the S-shaped transfer function is used to transform the continuous space to binary while in the second approach, the V-shaped transfer function is used for transforming the continuous space to binary. To evaluate and compare the performance of the proposed algorithms, more than 21 datasets from the UCI repository were used. Experimental results showed that the approach based on the S-shaped transfer function performs better than the V-shaped transfer function. In addition, the proposed method has performed better compared to other algorithms in terms of improving the classification accuracy. In another research, two different approaches of the Grasshopper Optimization Algorithm were presented by Mafarja et al. for solving the feature selection problem[1]. The first approach is based on the Sigmoid and V-shaped transfer functions and are named BGOA-S and BGOA-V respectively. The second proposed approach combines the best obtained solutions and also a mutation operator is utilized for increasing the exploration phase in the BGOA algorithm. Finally, the second approach is called BGOA-M. the proposed methods were evaluated using 25 standard UCI datasets and compared to 8 coating-based meta-heuristic approaches and six well-known filter-based methods. Test results show the advantage of BGOA and BGOA-M methods compared to other similar techniques.

As seen in table 1, researchers have used different meta-heuristic algorithms for solving binary problems, including feature selection, and in most studies, it is tried to use transfer or transform functions in the main procedures of each meta-heuristic algorithm to move them in the binary space. In some versions, they have only used an S-shaped transfer function while in others, they have used only the V-shaped transfer function. Of course, in some studies, both the S-shaped and V-shaped functions have been used for presenting the binary version of meta-heuristic algorithms[4, 16, 17]. Of course, in different studies, different versions of the S-shaped and V-shaped functions are used simultaneously. Finally, some researchers have used the mutation and crossover operators for presenting the binary version of meta-heuristic functions[4, 16, 17]. However, each one of the S-shaped and V-shaped functions might have its advantages and one might outperform the other in an algorithm depending on the procedures of the algorithm. Also, the mutation and crossover operators can be suitable operators for transforming continuous meta-heuristic algorithms to the binary version. However, if its exploration and exploitation are not tuned correctly, it will lead to the poor performance of the considered algorithm and occasional premature or slow convergence. Therefore, in our proposed method, we have considered the symbiosis search as the proposed method, which is a powerful algorithm for solving optimization problems, and used a different version of the S-shaped and V-shaped functions simultaneously for moving in the binary search space. In addition, in the third method, two new operators called BMP and BCP are presented for binarizing the symbiosis organisms search algorithm.

3. Symbiotic Organisms Search Algorithm

 doi:10.20944/preprints202001.0318.v1

In this section, we will briefly study the symbiotic organisms search algorithm. The readers can use references[18, 19] for more reading and advantages and limitations. The symbiotic organisms search algorithm is a meta-heuristic inspired by the opposition of organisms presented by Cheng and Prayogo in 2014 for solving optimization problems[19]. This algorithm starts working with an initial population called the ecosystem. In this ecosystem, a group of organisms is randomly generated in the search space and each organism is considered a solution for solving the optimization problem. Each one of the solutions or organisms has a fitness attributed to it. Furthermore, there are no parameters in this algorithm for tuning exploration and exploitation and this is done automatically. Also, since it uses optimal solutions of the current neighbor and the global solution through two commensalism and mutualism steps, it has good exploitation[18].

The main point in this algorithm is the way new solutions are generated. New solution generation is done by emulating the relationship or interaction of two organisms in the ecosystem. In this algorithm, the most prevalent or popular symbiosis relationship between two organisms in the environment is simulated which includes mutualism, commensalism, and parasitism. Figure 2. presents the most common symbiosis relationships in the environment, i.e. mutualism, commensalism, and parasitism.

(a): Mutualism	**(b):** Commensalism	**(c):** Parasitism

Fig. 2. Three main stages of symbiotic[18]

In figure 2., the main three steps of symbiosis are presented. In the mutualism step, both the organisms profit while in commensalism, only one organism profits but not the other. Finally, in the parasitism state of life, one organism profits while the other one gets harmed. In the following, first, the flowchart of the symbiosis organisms search algorithm is presented in figure 3. Then, each step of the algorithm is described comprehensively.

Mutualism Stage: in this stage, two organisms start a relationship and both will profit from this relationship. The relationship between bees and flowers can be mentioned which is presented in figure 2. part (a). In this step of the symbiotic organisms search algorithm, two organisms called X_i and X_j are chosen from the ecosystem on random and both organisms profit or update according to equations 1 and 2.

$$X_{inew} = X_i + rand \times (X_{best} - Mutual\_Vector \times BF_1) \tag{1}$$

$$X_{jnew} = X_j + rand \times (X_{best} - Mutual\_Vector \times BF_2) \tag{2}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 January 2020 doi:10.20944/preprints202001.0318.v1

$$Mutual\_Vector = \frac{X_i + X_j}{2} \tag{3}$$

In equations 1 and 2, BF_1 and BF_2 represent the profit coefficient of the two organisms and each organism's profit might be different than the other. Therefore, in this algorithm, BF_1 and BF_2 are determined randomly between 1 and 2. Also, *rand* is a random vector of numbers between zero and one. $Mutual\_Vector$, the mutual vector, represents the relationship between organisms X_i and X_j. Also, X_{best} represents the best organism in the ecosystem.

Commensalism Stage: in this stage, two organisms start a relationship and one of the organisms will profit from this relationship while the other one is not affected at all. The relationship between sticky fish and sharks can be mentioned which is presented in figure 2. part (b). In this step of the symbiotic organisms search algorithm, two organisms name X_i and X_j are chosen from the ecosystem on random and organism X_j is updated according to equation 4.

$$X_{inew} = X_i + rand(-1.1) \times (X_{best} - X_j) \tag{4}$$

In equation 4, rand is a random vector of numbers between -1 and 1 while X_{best} represents the best organism in the ecosystem.

Parasitism Stage: in this stage, two organisms enter a relationship where one of the organisms profits from this relationship while the other is damaged by it. The relationship between the Malaria disease which is transmitted to humans by Malaria mosquitos which is presented in figure 2. part (c). In the symbiotic organisms search algorithm, an organism X_j is chosen on random and like the Malaria mosquito, it acts like a parasite by creating an artificial parasite "Parasite-Vector". The parasite is created in the space by the multiplication of organism X_j. Parasite-Vector tries to replace X_j in the ecosystem. When Parasite-Vector works better than organism X_j, the organism X_j must be removed from the ecosystem and replaced by the Parasite-Vector. Otherwise, Parasite-Vector does not affect organism X_j in any way.

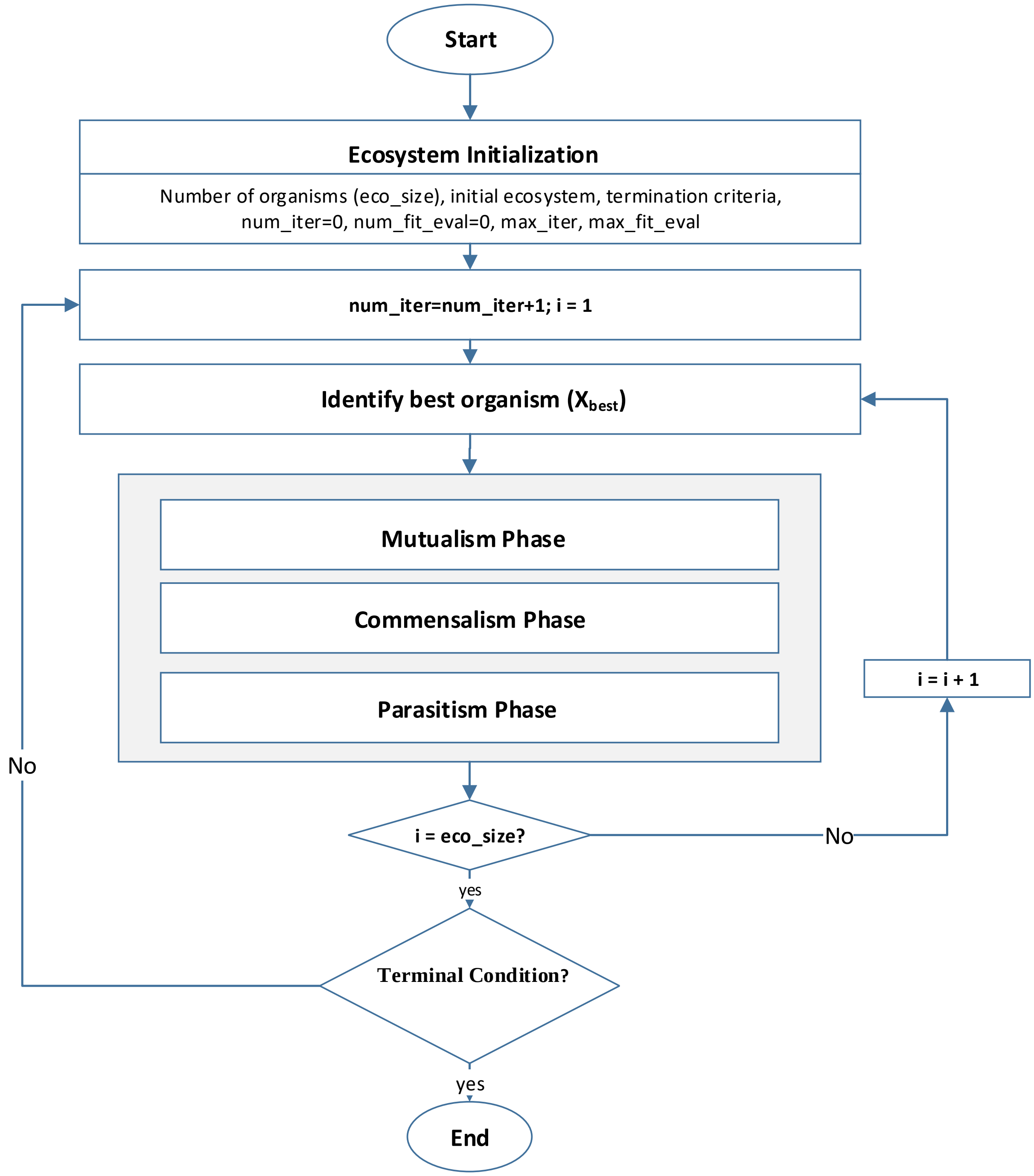

Fig. 3. Flowchart of the SOS algorithm[18, 20]

4. Proposed Method

We described the continuous symbiotic organisms search algorithm used for solving continuous optimization problems in section 3. In this section, our goal is to present different binary versions of the symbiotic organisms search algorithm for solving binary problems. In the first two versions, we used the S-shaped and V-shaped transfer functions which are two of the most important and most successful transfer functions from the continuous to the binary state. In addition, we will present a different version by making some modifications to the structure of the symbiotic organisms search algorithm procedures. In this version, some new operators are used. In the rest of this section, we describe the proposed method in three different subsections. In section 4.1, a binary version of the symbiotic organisms search algorithm based on the Sigmoid function will be presented. In this approach, four different functions are used for moving the symbiotic organisms search algorithm in the binary space. Finally, after experiments, an S-shaped function is considered as the transfer function from the continuous space to the binary space. In section 4.2, a binary version of the symbiotic organisms search algorithm based on the V-shaped function will be presented. In this approach, four different V-shaped functions are presented for moving the symbiotic organisms search algorithm in the binary space. Finally, after some experiments, a V-shaped function is used as the final transfer function between the continuous and binary space. In section 4.3, a different binary version based on altering the procedures of the symbiotic organisms search algorithm will be presented which uses new operators presented to improve the exploration and exploitation of the proposed algorithm. Finally, in section 4.4, a valid multi-objective function is presented for feature reduction and improving the classification accuracy.

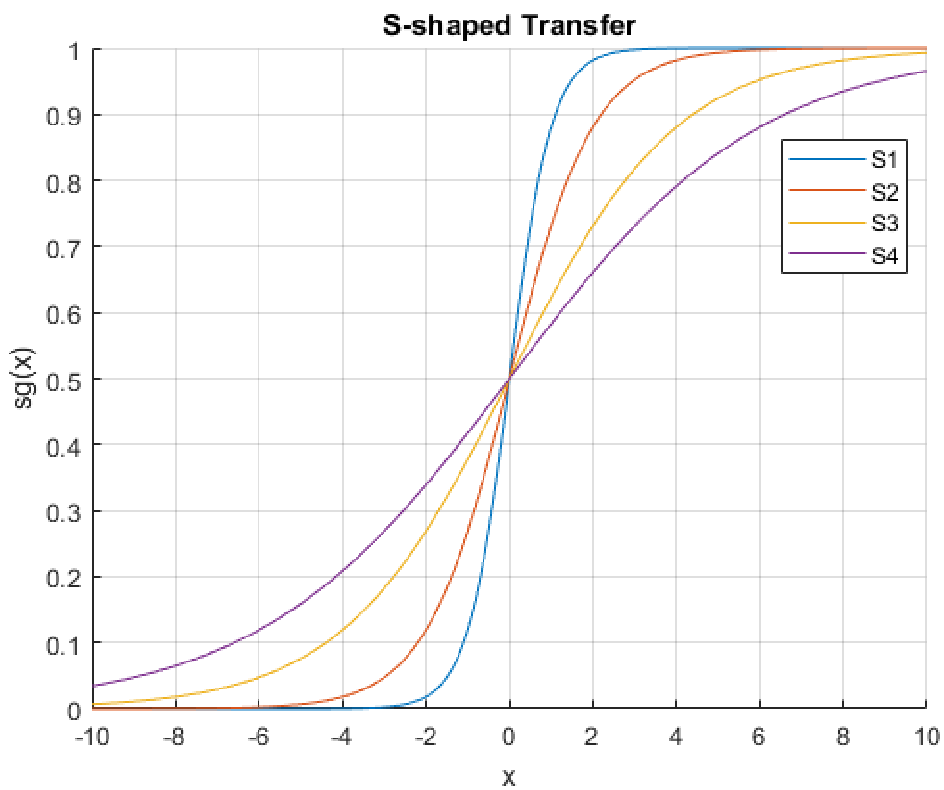

Fig. 4. Graphical representation of the S-shaped transfer function types

4.1 Binary Symbiotic Organisms Search Algorithm based on the S-shaped base function (BSOSS)

In this section, the new BSOSS approach for solving the feature selection problem using multiple S-shaped functions for binarizing the symbiotic organisms search algorithm is presented. The

Sigmoid or S-shaped function[4, 12, 21] is a transfer function which has been proven to be effective for transforming the continuous space to binary by many researchers. We used four well-known functions for binarization as well. These famous Sigmoid functions are presented in table 2 along with their formula. Also, the graphical state of these four functions is presented comprehensively in figure 4.

Table 2: *Types of Sigmoid Transfer Function*

Function name	Transfer function	Transfer function in coexistence search algorithm
S1	$sg(x) = \frac{1}{1 + e^{-2x}}$	$sg\left(SOS_i^d(t)\right) = \frac{1}{1 + e^{-2SOS_i^d(t)}}$
S2	$sg(x) = \frac{1}{1 + e^{-x}}$	$sg\left(SOS_i^d(t)\right) = \frac{1}{1 + e^{-SOS_i^d(t)}}$
S3	$sg(x) = \frac{1}{1 + e^{\left(-\frac{x}{2}\right)}}$	$sg\left(SOS_i^d(t)\right) = \frac{1}{1 + e^{\left(-\frac{SOS_i^d(t)}{2}\right)}}$
S4	$sg(x) = \frac{1}{1 + e^{\left(-\frac{x}{3}\right)}}$	$sg\left(SOS_i^d(t)\right) = \frac{1}{1 + e^{\left(-\frac{SOS_i^d(t)}{3}\right)}}$

Therefore, in the BSOSS approach, four S1, S2, S3, and S4 transfer functions are used for transforming the continuous symbiotic organisms search algorithm to the binary form. In table 2, SOS_i^d is the continuous value of solution i among the population of the symbiotic organisms search algorithm in dimension d at iteration t. According to the output obtained from figure 4., it is seen that the output of four S-shaped functions are continuously between 0 and 1. After using four S-shaped functions, thresholding is carried out and the best case in meta-heuristic algorithms is to use a random function for thresholding. Finally, in the S-shaped functions, an organism can be updated in the next iteration using equation 5.

$$SOS_i^d(t+1) = \begin{cases} 0 & if\ rand(0.1) < sg\left(SOS_i^d(t)\right) \\ 1 & if\ rand(0.1) \geq sg\left(SOS_i^d(t)\right) \end{cases} \tag{5}$$

In equation 5., SOS_i^d is the position of solution i in the population at iteration t in dimension d of the symbiotic organisms search algorithm. Also, $rand(0.1)$ is a number between zero and one from a uniform distribution. According to this equation, all the solutions present in the symbiotic organisms search algorithm will be transformed to binary. In the following, we added four S-shaped functions to the symbiotic organisms search algorithm as presented in the pseudo-code in figure 5.

BSOSS Algorithm
01: Define S-Shaped Transfer function S1,S2,S3,S4 according to Table(2)
02: Determining initial parameter
03: Generate an ecosystem X_i (i=1... EcoSize) with random $X_i \in$ random 0,1
04: Calculate Objective function according to Equations (10)
05: Select one S-Shaped Transfer[1]
06: **while (t < MaxIt)**

07: **For** i = 1: EcoSize
08: Find the X_{best} in the ecosystem
09: **% Mutualism Phase**
10: Randomly select X_j ($X_j \neq X_i$)
11: Determine $Mutual\_Vector$ and BF_1 , BF_2
12: calculation X_{inew} and X_{jnew} according to Equations (1) and (2)
13: Convert Continuous(X_{inew}) to Binary(BX_{inew}) using transfer S-shaped according to Equations (5) and Table(2)
14: Convert Continuous(X_{jnew}) to Binary(BX_{jnew}) using transfer S-shaped according to Equations (5) and Table(2)
15: Replace X_i with BX_{inew}(if BX_{inew} gives better fitness) and X_j with BX_{jnew}(if BX_{jnew} gives better fitness)
13: **%Commensalism Phase**
16: Randomly select X_j ($X_j \neq X_i$)
17: calculation X_{inew} according to Equation (4)
18: Convert Continuous(X_{inew}) to Binary(BX_{inew}) using transfer S-shaped according to Equations (5) and Table(2)
19: Replace X_i with BX_{inew}(if BX_{inew} gives better fitness)
20: **% Parasitism Phase**
21: Randomly select X_j ($X_j \neq X_i$)
22: Generate Parasite_Vector from organism X_i
23: Convert Continuous(Parasite_Vector) to Binary using transfer S-shaped according to Equations (5) and Table(2)
24: Replace X_j with BParasite_Vector (if BParasite_Vector gives better fitness)
25: **end for**
26: Save X_{best} in each iteration
27: **End while**
28: Print X_{best}

Fig. 5. Pseudo-code of BSOSS approach

In figure 5., the pseudo-code of the BSOSS approach based on four S-shaped functions is presented. According to line (01) of the pseudo-code, first, each transfer function is defined in the simulation environment. Then, setting the parameters and generating the initial population is done randomly in lines (02:03). In line (04), the definition of the target feature selection function defined in subsection 4.3 is implemented. In line (05), one of the S-shaped transfer functions gets selected for transforming the continuous space to binary. In lines (06:28), the main loop of the BSOSS approach which includes mutualism, commensalism, parasitism, and binarization phases is run. In lines (12:13, 18, and 23), new changes are made so that two X_{inew} and X_{jnew} solutions are transformed to the binary space before being evaluated by the target function and two new solutions BX_{inew} and BX_{jnew} are created. In line (18), X_{inew} generated in the commensalism step is transformed to the binary space before being evaluated by the target function and the new solution called BX_{inew} is created. In line (23), the Parasite_Vector created in the parasitism step is transformed to the binary space before being evaluated by the target function and the new solution called BParasite_Vector is generated.

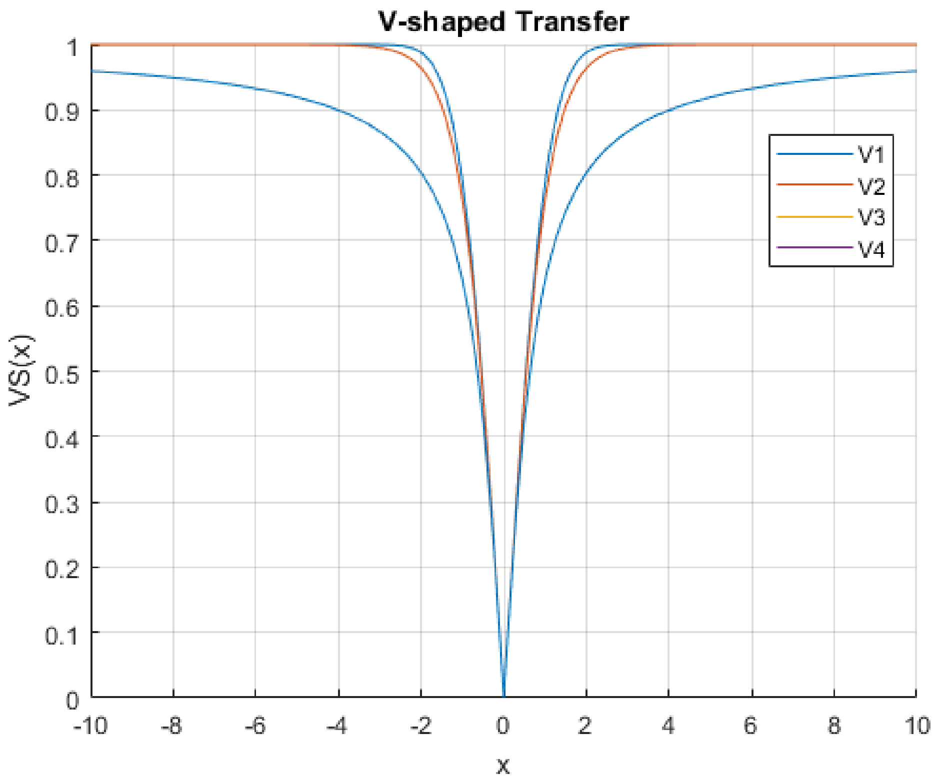

Fig. 6. Graphical view of the V-shaped transfer function types

4.2 Binary Symbiotic Organisms Search Algorithm Based on the V-shaped Transfer Function (BSOSV)

In this section, the new BSOSV approach consisting of multiple V-shaped transfer functions is presented for binarizing the symbiotic organisms search algorithm for solving the feature selection problem. The Tan hyperbolic or V-shaped function is another transfer function for binarizing meta-heuristic algorithms presented by Rashedi et al. in 2010[22] and has been approved by many researchers[1, 10, 12, 22]. In this paper, we used four well-known V-shaped functions for binarization. These four famous V-shaped functions are presented in table 3. along with their equations. Also, the graphical state of these functions is comprehensively presented in figure 6.

Table 3: *V-shaped transfer function types*

Function name	Transfer function	Transfer function in coexistence search algorithm
V1	$VS(x) = \left\|\mathrm{erf}\,(\frac{\sqrt{\pi}}{2}x)\right\|$	$VS\left(SOS_i^d(t)\right) = \left\|\mathrm{erf}\,(\frac{\sqrt{\pi}}{2}SOS_i^d(t))\right\|$
V2	$VS(x) = \|\tanh\,(x)\|$	$VS\left(SOS_i^d(t)\right) = \left\|\tanh\,(SOS_i^d(t))\right\|$

V3	$VS(x) = \left\lvert \frac{x}{\sqrt{1+x^2}} \right\rvert$	$VS\left(SOS_i^d(t)\right) = \left\lvert \frac{SOS_i^d(t)}{\sqrt{1+SOS_i^d(t)^2}} \right\rvert$
V4	$VS(x) = \left\lvert \frac{2}{\pi} arc\ \tan\left(\frac{\pi}{2}x\right) \right\rvert$	$VS\left(SOS_i^d(t)\right) = \left\lvert \frac{2}{\pi} arc\ \tan\left(\frac{\pi}{2} SOS_i^d(t)\right) \right\rvert$

In the BSOSV approach, four V1, V2, V3, and V4 transfer functions are used for transforming the continuous symbiotic organisms search algorithm to the binary form. We will act the same way for this transfer function as we did for the S-shaped transfer function where after applying four V-shaped transfer functions, thresholding takes place. Finally, in the V-shaped functions, an organism can be updated in the next iteration using equation 6.

$$SOS_i^d(t+1) = \begin{cases} 0 & if\ rand(0.1) < VS\left(SOS_i^d(t)\right) \\ 1 & if\ rand(0.1) \geq VS\left(SOS_i^d(t)\right) \end{cases} \tag{6}$$

All the details of equation 6. are like equation 5. with only the difference that we will use the V-shaped transfer function here. Later, we added the four V-shaped functions to the symbiotic organisms search algorithm as presented by the pseudo-code in figure 7.

BSOSV Algorithm

```
Define V-Shaped Transfer function V1,V2,V3,V4 according to Table(3)
Determining initial parameter
Generate an ecosystem X_i (i=1... EcoSize)  with random X_i ∈ random 0,1
Calculate Objective function according to Equations (10)
Select one S-Shaped Transfer{V1,V2,V3,V4}
while (t < MaxIt)
    For i = 1: EcoSize
        Find the  X_best in the ecosystem
        % Mutualism Phase
        Randomly select X_j (X_j ≠ X_i)
        Determine Mutual_Vector and BF_1 , BF_2
        calculation X_inew and X_jnew according to Equations (1) and (2)
        Convert Continuous(X_inew) to Binary(BX_inew)  using transfer V-shaped according to Equations (6) and Table(3)
        Convert Continuous(X_jnew) to Binary(BX_jnew)  using transfer V-shaped according to Equations (6) and Table(3)
        Replace X_i with BX_inew(if BX_inew gives better fitness) and X_j with BX_jnew(if BX_jnew gives better fitness)
        %Commensalism Phase
        Randomly select X_j (X_j ≠ X_i)
        calculation X_inew  according to Equation (4)
        Convert Continuous(X_inew) to Binary(BX_inew)  using transfer V-shaped according to Equations (6) and Table(3)
        Replace X_i with BX_inew(if BX_inew gives better fitness)
        % Parasitism Phase
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 January 2020 doi:10.20944/preprints202001.0318.v1

Randomly select X_j ($X_j \neq X_i$)
Generate Parasite_Vector from organism X_i
Convert Continuous(Parasite_Vector) to Binary using transfer V-shaped according to Equations (6) and Table(2)
Replace X_j with BParasite_Vector (if BParasite_Vector gives better fitness)
end for
Save X_{best} in each iteration
End while
Print X_{best}

Fig. 7. BSOSV approach pseudo-code

In figure 7., the pseudo-code of the BSOSV approach based on four V-shaped functions is presented. Description of this pseudo-code is similar to the pseudo-code of the BSOSS approach with the difference that in this approach, four V-shaped functions including V1, V2, V3, and V4 are used for transforming the continuous symbiotic organisms search algorithm to the binary form.

4.3 Efficient Binary SOS (EBSOS)

In this part, the Efficient Binary Symbiotic Organisms Search algorithm is presented. In this paper, we have named this approach EBSOS. In this approach, some major changes have been applied to transform the mutualism, commensalism, and parasitism steps from the continuous form to the binary form. Transforming the mutualism step from continuous to binary is so that first, the binary mutual vector (BMV) is presented and then, the binary for of the mutualism step along with its changes and pseudo-code is named BMP. Transforming the commensalism step from the continuous form to the binary form is such that organism X_i follows two general rules in this step. In the first rule, organism X_i moves more toward X_{best} and in the second rule, organism X_i takes solution X_j into consideration and in this step, a new operator called BCP is defined. Transforming the parasitism step from continuous to the binary form is such that first, solution X_i is considered as the Parasite_Vector. Then, some dimensions of Parasite_Vector are chosen on random and these random indices are saved in idx_random. Finally, entries in these random indices are refilled with random numbers in [0, 1]. In the following, each new step of the algorithm is described in detail and its equations are presented. Also, the pseudo-code is presented in figure 8.

EBSOS Algorithm
Determining initial parameter
Generate an ecosystem X_i (i=1... EcoSize) with random $X_i \in$ random 0,1
Calculate Objective function according to Equations (10)
while (t < MaxIt)
For i = 1: EcoSize
Find the X_{best} in the ecosystem
% Mutualism Phase
Randomly select X_j ($X_j \neq X_i$)
Determine $Binary\ Mutual\_Vector(BMV)$ and BF_1 , BF_2 according to Fig.9
calculation X_{inew} and X_{jnew} according to Binary Mutualism Phase(BMP) Fig.10
Replace X_i with BX_{inew}(if X_{inew} gives better fitness) and X_j with X_{jnew}(if X_{jnew} gives better fitness)
%Commensalism Phase

Randomly select X_i ($X_j \neq X_i$)
calculation X_{inew} according to Binary Commensalism Phase(BCP) Fig.12
Replace X_i with BX_{inew}(if BX_{inew} gives better fitness)
% Parasitism Phase
Randomly select X_j ($X_j \neq X_i$)
Generate Parasite_Vector from organism X_i according to Equations (7:9) and Fig.13
Replace X_j with Parasite_Vector (if Parasite_Vector gives better fitness)
end for
Save X_{best} in each iteration
End while
Print X_{best}

Fig. 8. EBSOS approach pseudo-code

4.3.1 Changing the Assistance Stage From continuous to binary Mode

In this step, two organisms enter a relationship they will both profit from. The mutual vector is the first thing that needs to change in this step so that it is usable for the binary problem. Here, we first define the mutual vector such that it contains all the mutual points. Then, non-mutual points are chosen randomly from either one of X_i or X_j solutions. This operation is called the Binary Mutual Vector (BMV) and is presented in detail in figure 9.

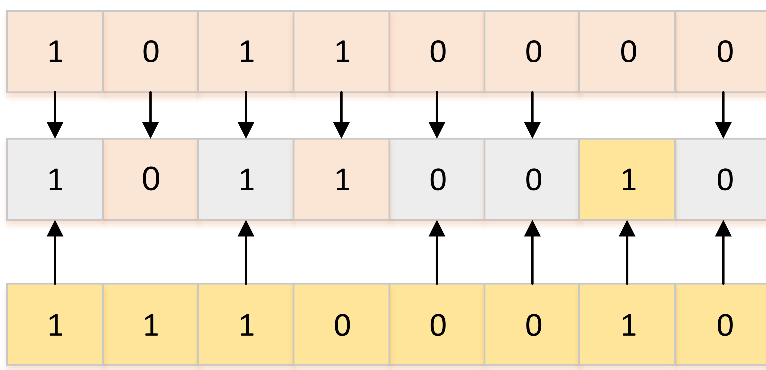

Fig. 9. Reciprocal vector in binary mode (BMV)

According to figure 9, the new BMV operator is used for creating the mutual vector in the binary form. The next step is using BF_1 and BF_2 for creating or improving new solutions. In this step, we have considered BF_1 and BF_2 as the composition coefficients of the mutual vector with X_{best}. This way, values of BF_1 and BF_2 determine the amount of composition with X_{best} in the binary form. Finally, after obtaining the mutual vector and combining it with the best solution, new solutions X_{inew} and X_{jnew} are sometimes randomly affected by the new vector. Overall, we defined a new BMP operator in this step and we have presented it in the pseudo-code in figure 10.

Binary Mutualism Phase: BMP
X_j= Randomly select X_j ($X_j \neq X_i$)
BMV= calculation Mutual_Vector according to Fig.9
CX_i= HybridBMV($BMV.X_{best}.BF_1$)
CX_j= HybridBMV($BMV.X_{best}.BF_2$)
For k=1 to dim

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 January 2020 doi:10.20944/preprints202001.0318.v1

```
    IF ( rand> 0.5 )
        X_inew(k) = cX_i(k)
    Else
        X_inew(k) = X_i(k)
    End if
For l=1  to dim
    IF ( rand> 0.5 )
        X_jnew(l) = cX_j(l)
    Else
        X_jnew(l) = X_j(l)
    End if
End for
```

Fig. 10. Cooperation step in binary form(BMP)

As seen in figure 10, first, organism X_j is chosen randomly and then, organism X_i and X_j are combined with organism X_{best} according to BF_1 and BF_2 and create two new solutions called cX_i and cX_j. These two solutions will cause fundamental changes in organisms X_i and X_j. Since we face the binary space in the new BMP operator, X_{inew} will use solution cX_i with a 50 percent chance. Otherwise, it uses solution X_i. Also, X_{jnew} will use solution cX_j with a 50 percent chance and otherwise, it will use solution X_j. Of course, we have used a new suitable function for the binary form which is presented in figure 11.

```
01: Function HybridBMV(BMV, X_best,BF)
02:    n=dim(BMV);
03:    X_new=empty array(1,n);
04:    For k=1:n
05:        IF(rand<BF)
06:        X_new (k)= X_best (k);
07:        else
08:        X_new (k)=BMV(k);
09:        End IF
10:    End For
11:    End Function
```

Fig. 11. New suitable function HybridBMV to combine function by BF

As presented in figure 11, in the first line, the best solution function gets the intended mutual vector and BF or profit from the input. Then, it tries to use the best solution and the binary mutual vector (BMV) to create new solutions according to BF. In line 4, the condition is set according to BF where the higher BF is, the new solution will use the best solution more. Otherwise, it uses the binary mutual vector (BMV) to create new solutions. Of course, variable BF here is between zero and one.

4.3.2 Transforming the commensalism step from continuous to binary

In this step, two organisms begin a relationship where one organism profits from this relationship and the other one is not affected by it. In the continuous form of mutualism, it is tried for organism X_i to move toward two X_j and X_{best} solutions. Also, multiplication by a random number between -

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 January 2020 doi:10.20944/preprints202001.0318.v1

1 and 1 adds more random moves to this step. We ran the continuous form of the SOS algorithm on MATLAB software. A sample solution along with its movement is depicted in figure 12.

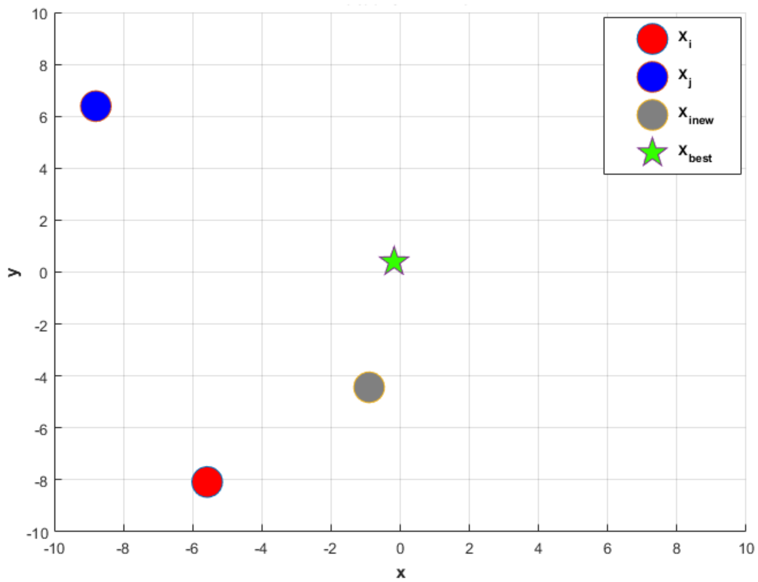

Fig. 12. An example of the X_i movement in the food phase

As seen in figure 12, organism X_i follows two general rules. In the first rule, organism X_i moves more towards X_i while in the second rule, organism X_i considers solution X_j at the same time. We considered these rules in the binary commensalism step as well and defined them as a new operator called BCP. This operator is presented in the new pseudo-code in figure 13.

Binary Commensalism Phase: BCP

X_j= Randomly select X_j ($X_j \neq X_i$)
X_{best}= Find the X_{best} in the ecosystem
For k=1 to dim
    IF (rand> 0.5)
        $X_{inew}(k) = X_{best}(k)$
    ElseIf (rand> 0.5)
        $X_{inew}(k) = X_j(k)$
    Else
        $X_{inew}(k) = X_i(k)$
    End if
End for

Fig. 13. Binary Commensalism Phase (BCP)

As seen in figure 13, a random organism X_j is selected on random first. Then, organism X_i improves using this organism X_j and X_{best} according to the two specified rules. In this new BCP operator, since we are faced with the binary space, the new solution uses X_{best} with a 50 percent chance. Otherwise, it might stay unchanged with a 50 percent chance or it uses solution X_j.

4.3.3 Transforming the parasitism step from continuous to binary

In the continuous symbiotic organisms search algorithm, the parasite or Parasite_Vector is created in the space by the multiplication of organism X_i. Parasite_Vector tries to replace X_j in the ecosystem. Our binary version works similar to the continuous form but with the difference that Parasite_Vector grows in the binary space and tries to choose a random number between zero and one and leads to fundamental changes in the Prasite_Vector. To understand the parasitism step better, we have depicted it as figure 14.

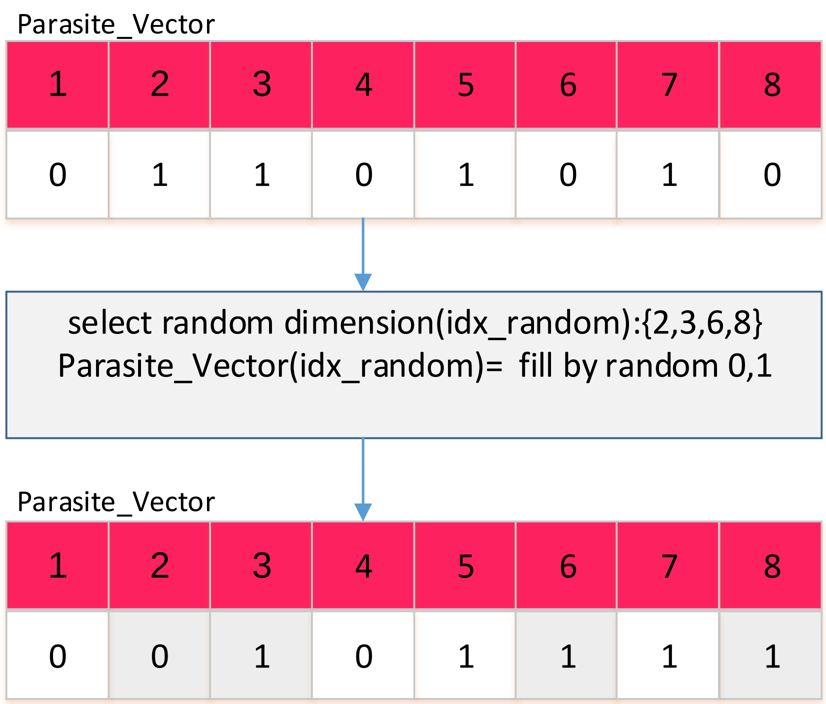

Fig. 14. Binary parasitism phase

In figure 14, first, solution X_i is considered as the Prasite_Vector. Then, some dimensions of Parasite_Vector are chosen randomly and these random indices are saved in idx_random. Finally, entries corresponding to these random indices are replaced with random numbers in the [0, 1] interval. In this example, assuming we have a vector of dimension 8, indices 2, 3, 6, and 8 are chosen randomly and saved in idx_random. Finally, the binary parasitism phase can be defined as follows using equation 9:

$$\text{Parasite} - \text{Vector} = X_i \tag{7}$$

$$\text{idx}_{\text{random}} = \text{select random dimension}(\text{Parasite} - \text{Vector}) \tag{8}$$

$$\text{Parasite} - \text{Vector}(\text{idx}_{\text{random}}) = \text{random } 0.1(1.\text{size}(\text{idx}_{\text{random}})) \tag{9}$$

Furthermore, in this section, we have used the mutation operator to increase the exploration in the proposed approach which is applied to Parasite_Vector at the final step in order to achieve a better result.

4.3 Target Function

The multi-objective function proposed for balancing the number of selected features in each solution (minimum) and classification accuracy (maximum) is presented in this section. This objective function is used in equation 10 for evaluating a solution in meta-heuristic algorithms.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|N|} \quad (10)$$

Where $\alpha\gamma_R(D)$ represents the classification error of a classifier and |R| shows that the selected subset is multi-linear. Also, |N| is the number of all features available in the dataset while parameter α is the importance of classifier quality and parameter β is the length of the subset. The values of these two parameters are calculated according to $\alpha \in [0, 1]$ and β = (1 -α) which were adopted from research[7]. In this research, the initial value of α is considered to be 0.99. therefore, the value of β will be 0.01. Most researchers[9, 15, 23-25] use the simplest classification method, i.e. KNN[26]. We used this classifier for defining the objective function in the feature selection problem as well.

4 Evaluation and Results

In this section, we have evaluated the proposed BSOSS, BSOSV, and EBSOS methods. All the tests in this section were run in MATLAB software on a system with a X GHz Core i5 processor and 8 gigabytes of RAM. To evaluate the proposed method, 18 feature selection datasets from the UCI[27] repository are used. The features of each dataset are comprehensively presented in table 4. Also, base algorithms like genetic algorithm[28], binary bat algorithm[29], binary particle swarm algorithm[30], binary flower pollination algorithm[31], binary grey wolf algorithm[8], binary dragonfly algorithm[32], and binary chaotic crow search algorithm[33] will be used for comparison with the final proposed approach (EBSOS).

Table 4. *Dataset description*

ID	Dataset	No. of features	No. of instances	No. of classes	Missing values	Type
D1	Abalone	8	4177	29	No	Life
D2	Breast Cancer Wisconsin	9	699	2	Yes	Life
D3	BreastEW	30	569	2	No	Life
D4	Dermatology	34	366	6	Yes	Life
D5	Germen credit	24	1000	2	No	Business
D6	Glass identification	10	214	6	No	Physical
D7	Hepatitis	19	55	2	Yes	Clinical
D8	Indian Liver Patient Dataset	10	583	2	No	Clinical
D9	IonosphereEW	34	351	2	No	Physical
D10	Lung Cancer	56	32	3	Yes	Clinical
D11	Lymphography	18	148	4	No	Life
D12	SPECT	22	267	2	No	Clinical
D13	Statlog (heart)	13	270	2	No	Clinical
D14	Steel Plates Faults	33	1941	7	NO	Physical
D15	Thoracic surgery	17	470	2	No	Clinical
D16	Waveform	21	5000	3	No	Physical
D17	WineEW	13	178	3	No	Physical
D18	Zoo	17	101	2	No	Life

In the rest of this section, we will first compare the methods based on the S-shaped function which are named S1, S2, S3, and S4 and then choose one of them which works better than the others as the final method based on the S-shaped function. Then, we will first compare the methods based on the V-shaped function called V1, V2, V3, and V4 and we will choose the one which works better than the others as the final method based on the V-shaped function. finally, we will compare the BSOSS, BSOSV, and EBSOS approaches. In the end, we will choose a method as the final approach and compare it to other methods. In addition to this, in the last section, we will present an applied study on an email span dataset to further evaluate the performance of the proposed algorithm.

5.1 Evaluation of S-shaped methods

In this section, four S-shaped methods including S1, S2, S3, S4 are implemented on 18 datasets and the results are compared, and finally an S-shaped method is selected to compare the three approaches proposed in section (5-3). In these tests, the iterations and population are 50 and 10, respectively. We evaluate four S-shaped methods including S1, S2, S3, S4 in terms of mean feature number, classification accuracy and objective function convergence rate. In Table 5, four S-shaped methods including S1, S2, S3, S4 are shown in terms of the mean number of features.

Table 5: *Comparison of four S-shaped methods based on the criterion of mean number of features*

Dataset	BSOS_S1	BSOS_S2	BSOS_S3	BSOS_S4
D1	5.1	4.7	4.9	**3.9**
D2	5.3	5.4	5.5	**4.9**
D3	22.6	16.8	16.7	**14.3**
D4	25.4	22.7	20.4	**18.2**
D5	17.9	15.5	14	**13.8**
D6	3.3	3.2	2.8	**2.7**
D7	12.3	10.8	**10.1**	10.7
D8	5	4.3	**3.4**	3.9
D9	19.8	17.9	15.8	**14.6**
D10	33.4	29.9	30.1	**26.3**
D11	12.6	10.5	**9.7**	10
D12	12.8	12.9	11.9	**10.2**
D13	6.9	5.5	6	**5.1**
D14	22.2	20	17.1	**16.8**
D15	8.7	6	5.5	**5.5**
D16	17.4	15.8	14.4	15.6
D17	7.3	6.9	**6.2**	7.1
D18	9.2	8.4	9.1	**8.3**

Table 5 shows the four methods based on the S-shaped transfer function in terms of the mean number of selected features with iteration number of 50. This experiment shows that the S4 performs best in terms of the average number of selected features, and the S3 sometimes does. But the S1 and S2 models have a relatively modest performance. In Table 6, four S-shaped methods including S1, S2, S3, S4 are presented in terms of accuracy criteria.

Table 6: *Comparison of four S-shaped methods in terms of classification accuracy*

Dataset	BSOS_S1	BSOS_S2	BSOS_S3	BSOS_S4
D1	**0.2166**	0.2162	0.2162	0.2166
D2	0.9629	0.9629	0.9629	**0.9629**
D3	**0.9579**	0.9509	0.9544	**0.9579**
D4	0.9727	0.9781	0.9727	**0.9781**
D5	0.714	0.716	0.72	**0.734**
D6	1	1	1	**1**
D7	**0.7308**	**0.7308**	0.7179	0.6795
D8	0.6678	0.6781	0.6815	**0.6849**
D9	0.9034	0.9091	0.9205	**0.9205**
D10	0.9375	0.9375	0.9375	**0.9375**
D11	**0.8514**	0.8514	0.8378	0.8378
D12	0.709	0.7015	0.7164	**0.7313**
D13	0.7778	0.7852	0.7778	**0.8148**
D14	0.7497	**0.7662**	0.7446	0.7611
D15	0.8012	0.8255	0.8128	**0.8426**
D16	**0.8013**	0.7868	0.7888	0.7788
D17	0.9551	0.9551	0.9551	0.9438
D18	0.9412	0.9412	0.9412	**0.9412**

Table (6) shows the results associated with four methods which are based on the S-shaped transfer function in terms of classification accuracy with iteration 50. The test shows that the S4 is the best in terms of classification accuracy, and the S1 also performs better. However, the S2 and S3 models exhibit relatively poor performance in classification accuracy. Based on the results of feature selection and classification accuracy, it can be said that the S4 is the best model for both feature selection and classification accuracy. But other models have lost their performance in terms of accuracy and average selection. As a result, the S2 model is chosen as the final V-shaped approach.

5.2 Evaluation of V-shaped methods

In this section, four V-shaped methods, namely V1, V2, V3, V4, are implemented on 18 datasets and the results are compared to each other, and finally a final V-shaped method is selected to compare the three approaches proposed in section (5-3). In these experiments, the number of iterations and population is set to 50 and 10, respectively. As in Section (5-1) in the S-shaped method, here we evaluate four V-shaped methods including V1, V2, V3, V4 in terms of mean number of feature and classification accuracy. The following four V-shaped methods including V1, V2, V3, V4 are compared in terms of the mean number of features and the results are shown in Table (7).

Table 7: *Comparison of four S-shaped methods based on the criterion of mean number of features*

Dataset	BSOS_V1	BSOS_V2	BSOS_V3	BSOS_V4
D1	4.8	4.4	4.5	**4.3**
D2	6.1	6.1	6.9	**5.9**
D3	16.6	17.2	**15.8**	18.3
D4	20.5	21.9	**17.3**	21.6
D5	13.9	12.8	13.6	**11.5**
D6	6.2	7	**3.4**	6.8
D7	9.3	9.3	**7.9**	10.9
D8	5.6	4.5	6.2	**4.2**
D9	16.6	**16.5**	16.8	19.5
D10	28.2	**27.8**	28.9	33.9
D11	9	10	9.8	**8.8**
D12	11.9	**10.4**	10.9	12.9
D13	7.1	7.1	7	**6.2**
D14	**17.2**	17.7	17.3	19.4
D15	7.2	**7.1**	8.7	8.7
D16	16.5	16.7	**10.4**	18.1
D17	6.5	6.5	6.7	**6.2**
D18	9.4	9.3	7.1	10.3

The results obtained based on Table (7) for comparing four V-shaped methods in terms of mean number of features show that the V4 and V3 models are the best in terms of the average number of selected features. Of course, the S1 and S2 models have shown average performance. The following four V-shaped methods including V1, V2, V3, V4 are compared in terms of accuracy criteria and their results are shown in Table (8).

Table 8: *Comparison of four S-shaped methods in terms of classification accuracy*

Dataset	BSOS_V1	BSOS_V2	BSOS_V3	BSOS_V4
D1	**0.2217**	0.2217	0.219	**0.2217**
D2	0.9629	0.9629	0.9486	**0.9657**
D3	0.9439	0.9368	0.9158	**0.9509**
D4	0.9563	**0.9672**	0.7213	0.9454
D5	0.704	0.684	0.706	0.674
D6	**0.9813**	0.9813	0.972	**0.9813**
D7	**0.6154**	0.6026	0.5641	**0.6154**
D8	0.6849	0.6986	0.6644	**0.7021**
D9	**0.9205**	0.9034	0.8693	0.9148
D10	**0.875**	0.875	0.875	**0.875**
D11	0.7568	0.7568	0.6486	**0.7973**
D12	**0.7388**	0.7239	0.6045	0.7164
D13	0.7481	**0.8011**	0.6593	0.7407
D14	0.8074	0.7312	0.6375	**0.8074**
D15	**0.8638**	0.8255	0.8213	0.834
D16	0.7644	0.7804	0.6644	**0.7864**

D17	**0.9326**	0.9213	0.9326	0.809
D18	0.9804	0.9804	0.9412	**1**

The results associated with four methods based on the V-shaped transfer function in terms of classification accuracy (Table 8) show that the V4 model is the best in terms of classification accuracy criteria and then the S1 model is the best in terms of classification accuracy criteria. However, the S2 and S3 models have a relatively modest performance. Based on the results of feature selection and classification accuracy, it can be said that the S4 is the best model for both feature selection and classification accuracy, and on the other hand, the S2 model performs better in feature selection, but in terms of accuracy, the S1 model has also performed better in classification accuracy, but has lost performance in terms of feature selection. At the end, the model S3 exhibits moderate performance. In Figures 14 to 16, the results of each transfer function convergence rate are shown.

5.3 Comparison and evaluation of three proposed approaches (BOSS, BSV, EBSOS)

In this section, we examine three proposed approaches BSOSS, BSOSV and EBSOS in detail. In this paper, the S4 transfer function is used in the BSOSS approach which is based on the BSOSS function regarding the section (1.5) tests results, and also the V4 transfer function is used in the BSOSV approach which is based on the BSOSV function regarding the results of section (2.5) as the final method. The purpose of this experiment is to compare the three proposed approaches and select one final method as the proposed approach for the next section, namely one of the proposed approaches BSOSS, BSOSV, and EBSOS as one of the proposed methods to compare with other meta-heuristic methods in Section (4-5). The following three approaches are compared in terms of the criterion of mean number of features and their results are presented in Table 9. Population number and iterations are set to 10 and 60, respectively.

Table 9: *Comparison of the three proposed approaches BOSS, BSSV, BESOS in terms of mean number of features*

Dataset	BSOSS	BSOSV	EBSOS	BEST Approaches
D1	**6**	**6**	**6**	☐ **BSOSS** ☒ **BSOSV** ☒ **EBSOS**
D2	5.5	5.5	5	☐ **BSOSS** ☐ **BSOSV** ☒ **EBSOS**
D3	14.7	15.2	9	☒ **BSOSS** ☐ **BSOSV** ☒ **EBSOS**
D4	19	20.3	15	☐ **BSOSS** ☐ **BSOSV** ☒ **EBSOS**
D5	13.4	13.1	11	☐ **BSOSS** ☐ **BSOSV** ☒ **EBSOS**
D6	4.5	4.3	3	☐ **BSOSS** ☐ **BSOSV** ☒ **EBSOS**
D7	7.8	9.2	6.7	☐ **BSOSS** ☐ **BSOSV** ☒ **EBSOS**
D8	4.5	3.9	4	☐ **BSOSS** ☒ **BSOSV** ☐ **EBSOS**
D9	15.9	16.9	13	☐ **BSOSS** ☐ **BSOSV** ☒ **EBSOS**
D10	24.2	27.6	18	☐ **BSOSS** ☐ **BSOSV** ☒ **EBSOS**
D11	10.2	11.8	8	☐ **BSOSS** ☐ **BSOSV** ☒ **EBSOS**
D12	8.5	11.5	5.1	☐ **BSOSS** ☐ **BSOSV** ☒ **EBSOS**
D13	6.1	5.8	7	☐ **BSOSS** ☒ **BSOSV** ☐ **EBSOS**
D14	16.2	17.6	11.9	☐ **BSOSS** ☐ **BSOSV** ☒ **EBSOS**
D15	4.8	6.8	7	☒ **BSOSS** ☐ **BSOSV** ☐ **EBSOS**

D16	15	15.8	14	☐ BSOSS ☐ BSOSV ☒ EBSOS
D17	6	6	6	☒ BSOSS ☒ BSOSV ☒ EBSOS
D18	9.6	10.8	8	☐ BSOSS ☐ BSOSV ☒ EBSOS

The results of the three proposed approaches in terms of the mean number of features presented in Table (9) show that the EBSOS approach performed the best in terms of feature selection, with 18 datasets able to perform 83% more successfully than the other two approaches. Of course, in addition to feature selection, the classification accuracy criterion should also be considered, which is compared with the three proposed approaches in terms of classification accuracy and the results are shown in Table (10).

Table 10: *Comparison of the three proposed approaches BOSS, BSSV, BESOS in terms of classification accuracy*

Dataset	BSOSS	BSOSV	EBSOS	BEST Approaches
D1	0.2198	0.2169	0.2198	☒ BSOSS ☐ BSOSV ☒ EBSOS
D2	0.9657	0.9657	0.9657	☒ BSOSS ☒ BSOSV ☒ EBSOS
D3	0.9509	0.9474	0.9649	☐ BSOSS ☐ BSOSV ☒ EBSOS
D4	0.9617	0.9508	0.9836	☐ BSOSS ☐ BSOSV ☒ EBSOS
D5	0.706	0.704	0.726	☐ BSOSS ☐ BSOSV ☒ EBSOS
D6	0.9907	0.9813	0.9907	☐ BSOSS ☒ BSOSV ☐ EBSOS
D7	0.6923	0.641	0.7436	☐ BSOSS ☐ BSOSV ☒ EBSOS
D8	0.6986	0.6781	0.7021	☐ BSOSS ☐ BSOSV ☒ EBSOS
D9	0.8807	0.858	0.9205	☐ BSOSS ☐ BSOSV ☒ EBSOS
D10	0.875	0.875	0.875	☒ BSOSS ☒ BSOSV ☒ EBSOS
D11	0.8514	0.8378	0.8784	☐ BSOSS ☐ BSOSV ☒ EBSOS
D12	0.6493	0.6269	0.7388	☐ BSOSS ☐ BSOSV ☒ EBSOS
D13	0.7852	0.7333	0.8222	☐ BSOSS ☐ BSOSV ☒ EBSOS
D14	0.7508	0.7106	0.9907	☐ BSOSS ☐ BSOSV ☒ EBSOS
D15	0.834	0.8255	0.8383	☐ BSOSS ☐ BSOSV ☒ EBSOS
D16	0.7864	0.7804	0.7956	☐ BSOSS ☐ BSOSV ☒ EBSOS
D17	0.9551	0.9213	0.9663	☐ BSOSS ☐ BSOSV ☒ EBSOS
D18	1	1	1	☒ BSOSS ☒ BSOSV ☒ EBSOS

Results associated with three proposed approaches in terms of classification accuracy criteria in Table (10) show that the EBSOS approach performed best in terms of classification accuracy, with 18 datasets able to perform 95% more successfully than the other two approaches. Consequently, the results obtained in terms of the criterion of accuracy and average number of selected features, the EBSOS approach proves its remarkable superiority over the other two methods and can be chosen as a final method for comparison with other algorithms. However, in the remainder of this section, we evaluated three proposed approaches in terms of objective function convergence rate in order to show which method is better in terms of convergence than the other methods, and the comparison results are shown in Figures (15) and (16) in terms of the objective function convergence rate.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 January 2020 doi:10.20944/preprints202001.0318.v1

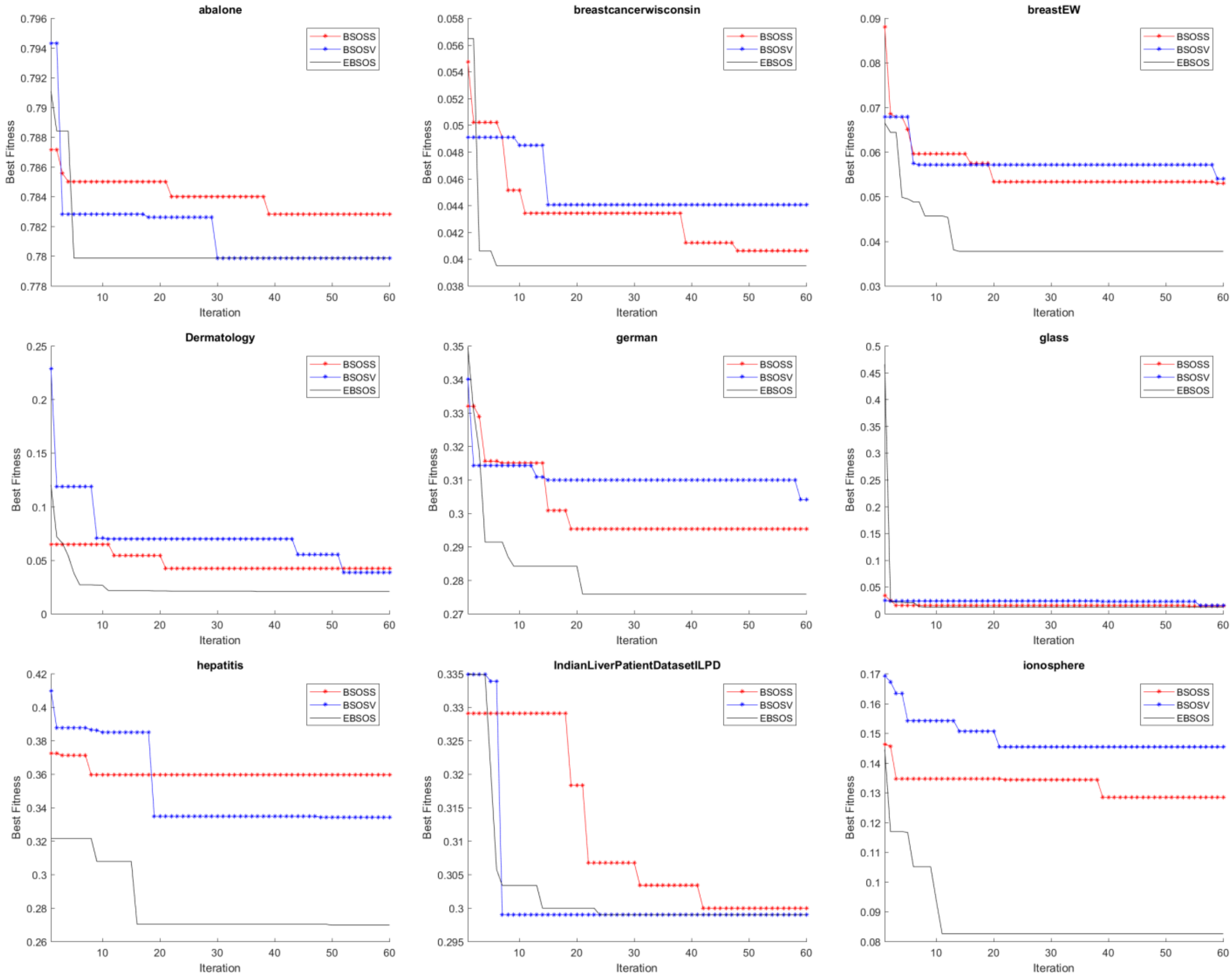

Fig. 15. Comparison of the three proposed approaches in terms of objective function convergence on dataset D1: D9

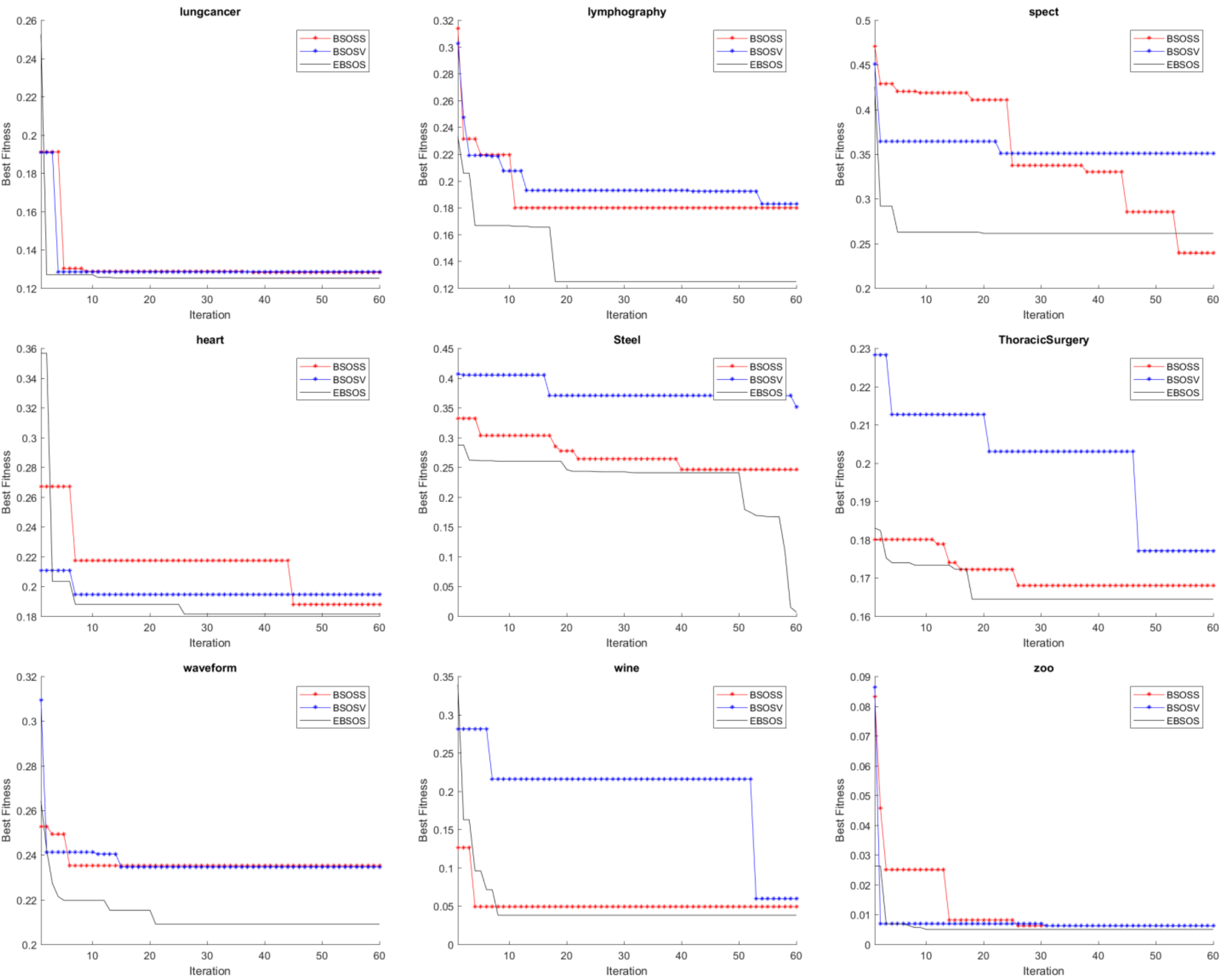

Fig. 16. Comparison of the three proposed approaches in terms of objective function convergence on dataset D10: D18

The results of the three proposed approaches in terms of the objective function convergence in Figures (15) and (16) show that the EBSOS approach has been able to achieve objective function convergence goals in addition to the features accuracy and average. From the results obtained in terms of criteria of accuracy and average number of selected features as well as the convergence of the objective function EBSOS approach proves its remarkable superiority in two other ways and can be selected as a final method for comparison with other algorithms. In section (4.5) we compared the EBSOS approach with more benchmarks with powerful meta-algorithms including GA, BBA, BPSO, BFPA, BGWO, BDA, BCCSA. All experiments confirm the remarkable superiority of the EBSOS approach in most statistical criteria.

5.4 Comparing with other approaches

In this section, the proposed EBSOS approach is implemented on 18 datasets with other meta-heuristic algorithms such as GA, BBA, BPSO, BFPA, BGPA, BGWO, BDA, BCCSA, and then, are compared in terms of average feature selection criteria, classification accuracy, objective function convergence as well as Statistical criteria including best, worst, average and standard deviation. In this section, the experiments in iterations 40 and 80 are intended to compare the algorithms with fewer and more iterations. In all experiments in this section, the population is also considered 10. In the following, the proposed EBSOS approach is compared with other meta-heuristic algorithms in terms of statistical criteria and other criteria in iteration 40, and its results are shown in Tables 11 to 12 and Figures 17 and 18.

Table 11: *Comparing the proposed EBSOS approach with other meta-heuristic algorithms in terms of statistical criteria with iteration number of 40*

Dataset	cirita	GA	BBA	BPSO	BFPA	BGWO	BDA	BCCSA	EBSOS
D1	Best	0.7798	0.7902	0.7857	0.7852	0.7905	0.7798	0.7903	**0.7798**
	Mean	0.7798	0.8279	0.7894	0.7963	0.7937	0.7915	0.7903	**0.7798**
	Worst	0.7798	0.8412	0.8179	0.8087	0.8042	0.7949	0.7903	**0.7798**
	Std	**0**	0.0102	0.0101	0.007	0.0049	0.0035	**0**	**0**
D2	Best	0.0338	0.0378	0.0321	0.035	0.0378	0.0321	0.0497	**0.0321**
	Mean	0.0338	0.1341	0.0389	10.057	0.0402	0.0364	0.0497	**0.0321**
	Worst	0.0338	0.2596	0.0678	100	0.0445	0.0734	0.0497	**0.0321**
	Std	**0**	0.074	0.0142	31.6027	0.002	0.013	**0**	**0**
D3	Best	0.0488	0.072	0.054	0.0647	0.0512	0.0454	0.0612	**0.0415**
	Mean	0.0536	0.1217	0.056	0.0774	0.0516	0.0656	0.0612	**0.0413**
	Worst	0.0564	0.1593	0.0651	0.0894	0.0518	0.0829	0.0612	**0.0415**
	Std	0.0037	0.0302	0.0044	0.008	0.0002	0.0117	**0**	**0.0002**
D4	Best	0.0332	0.0537	0.0372	0.0543	0.0383	0.0323	0.0555	**0.0198**
	Mean	0.0333	0.2375	0.0448	0.1327	0.0403	0.0591	0.0555	**0.0198**
	Worst	0.0335	0.3862	0.0696	0.2006	0.0549	0.1459	0.0555	**0.0198**
	Std	0.0001	0.0851	0.0125	0.0579	0.0051	0.0323	**0**	**0**
D5	Best	0.3103	0.3238	0.2869	0.3151	0.2945	0.2948	0.3405	**0.2814**
	Mean	0.3156	0.3891	0.292	0.3509	0.2981	0.3006	0.3405	**0.2814**
	Worst	0.3163	0.4828	0.3297	0.3773	0.3044	0.321	0.3405	**0.2814**
	Std	0.0019	0.042	0.0133	0.0209	0.0045	0.0099	**0**	**0**
D6	Best	0.0123	0.0143	0.0123	0.0133	0.0123	0.0123	0.0143	**0.0113**
	Mean	0.0123	0.4505	0.0138	0.2678	0.0135	0.0203	0.0143	**0.0113**
	Worst	0.0123	0.6959	0.0215	0.4944	0.0143	0.0318	0.0143	**0.0113**
	Std	**0**	0.2398	0.0028	0.1787	0.0006	0.006	**0**	**0**
D7	Best	0.3083	0.3469	0.3083	0.3601	0.321	0.3083	0.385	**0.2306**
	Mean	0.3083	0.4584	0.3085	0.4252	0.328	0.3656	0.385	**0.2306**
	Worst	0.3083	0.5235	0.3088	0.5003	0.386	0.4611	0.385	**0.2306**
	Std	**0**	0.039	0.0003	0.0506	0.0204	0.0528	**0**	**0**
D8	Best	0.2742	0.3091	0.2786	0.3037	0.2908	0.2661	0.3169	**0.2661**
	Mean	0.2742	0.4017	0.2792	0.3564	0.2992	0.302	0.3169	**0.2661**
	Worst	0.2742	0.6181	0.2844	0.3789	0.3091	0.3766	0.3169	**0.2661**
	Std	**0**	0.0803	0.0018	0.0183	0.0072	0.0405	0	**0**
D9	Best	0.1063	0.1089	0.1048	0.1299	0.1006	0.0986	0.1237	**0.0811**
	Mean	0.1126	0.1546	0.1133	0.1569	0.1023	0.1385	0.1237	**0.0797**
	Worst	0.1178	0.1826	0.1326	0.1684	0.106	0.151	0.1237	**0.0876**
	Std	0.0051	0.0153	0.0092	0.0077	0.0025	0.0085	**0**	0.0071
D10	Best	0.1892	0.1289	0.1894	0.1291	0.1291	0.1271	0.1901	**0.064**
	Mean	0.1895	0.3804	0.2019	0.3032	0.1355	0.2945	0.1901	**0.0641**
	Worst	0.1896	0.4367	0.2514	0.3782	0.1913	0.3748	0.1901	**0.0641**

	Std	0.0001	0.0457	0.0261	0.0639	0.0196	0.0773	**0**	**0**
D11	Best	0.1511	0.2447	0.165	0.1923	0.2475	0.1527	0.2464	**0.1115**
	Mean	0.1671	0.3784	0.1745	0.3348	0.2479	0.2131	0.2464	**0.1115**
	Worst	0.1789	0.5496	0.2581	0.4582	0.248	0.2876	0.2464	**0.1115**
	Std	0.0078	0.0783	0.0294	0.0628	0.0003	0.0403	**0**	**0**
D12	Best	0.2562	0.2853	0.2484	0.2889	0.258	0.241	0.3005	**0.2327**
	Mean	0.2806	0.3981	0.2607	0.3633	0.2631	0.3006	0.3005	**0.2327**
	Worst	0.2862	0.4608	0.2922	0.4104	0.2723	0.3657	0.3005	**0.2327**
	Std	0.0093	0.053	0.02	0.0306	0.0068	0.0457	**0**	**0**
D13	Best	0.2165	0.2011	0.2018	0.2319	0.2092	0.1945	0.2458	**0.1945**
	Mean	0.2165	0.382	0.2108	0.3717	0.2169	0.2203	0.2458	**0.1945**
	Worst	0.2165	0.4628	0.2613	0.4029	0.2246	0.3199	0.2458	**0.1945**
	Std	**0**	0.0533	0.0201	0.0295	0.0081	0.0444	**0**	**0**
D14	Best	0.2597	0.3757	0.2523	0.3667	0.3748	**0.0172**	0.3362	0.2495
	Mean	0.2597	0.3984	0.2716	0.403	0.3771	**0.1779**	0.3362	0.2495
	Worst	0.2597	0.4074	0.4	0.4123	0.3945	0.409	0.3362	**0.2495**
	Std	**0**	0.009	0.0458	0.0066	0.0061	0.1568	**0**	**0**
D15	Best	0.1506	0.1409	0.1409	0.1748	0.1409	0.1396	0.1783	**0.1396**
	Mean	0.1628	0.3112	0.1858	0.2163	0.1416	0.1927	0.1783	**0.1396**
	Worst	0.1657	0.5706	0.4497	0.2771	0.1421	0.5285	0.1783	**0.1396**
	Std	0.0046	0.1365	0.0988	0.03	0.0003	0.1216	**0**	**0**
D16	Best	0.2112	0.2347	0.2139	0.2312	0.2113	**0.2096**	0.2242	0.2114
	Mean	0.212	0.3281	0.2177	0.2637	0.2149	0.2159	0.2242	**0.2114**
	Worst	0.2185	0.39	0.249	0.3277	0.2177	0.2285	0.2242	**0.2114**
	Std	0.0023	0.04	0.011	0.0328	0.002	0.0079	**0**	**0**
D17	Best	0.0506	0.0483	0.0269	0.0587	0.0618	0.0372	0.0936	**0.0253**
	Mean	0.0506	10.2549	0.0404	0.3073	0.0716	0.1486	0.0936	**0.0253**
	Worst	0.0506	100	0.0928	0.3748	0.0737	0.3797	0.0936	**0.0253**
	Std	**0**	31.5334	0.0232	0.0954	0.0035	0.1474	**0**	**0**
D18	Best	0.0263	0.062	0.062	0.0639	0.0257	0.0438	0.0632	**0.0257**
	Mean	0.0263	0.3228	0.0934	0.1147	**0.0261**	0.0535	0.0632	0.0257
	Worst	0.0263	0.5849	0.3162	0.2185	**0.0263**	0.0645	0.0632	0.0257
	Std	**0**	0.1654	0.0804	0.0433	0.0003	0.0102	**0**	**0**

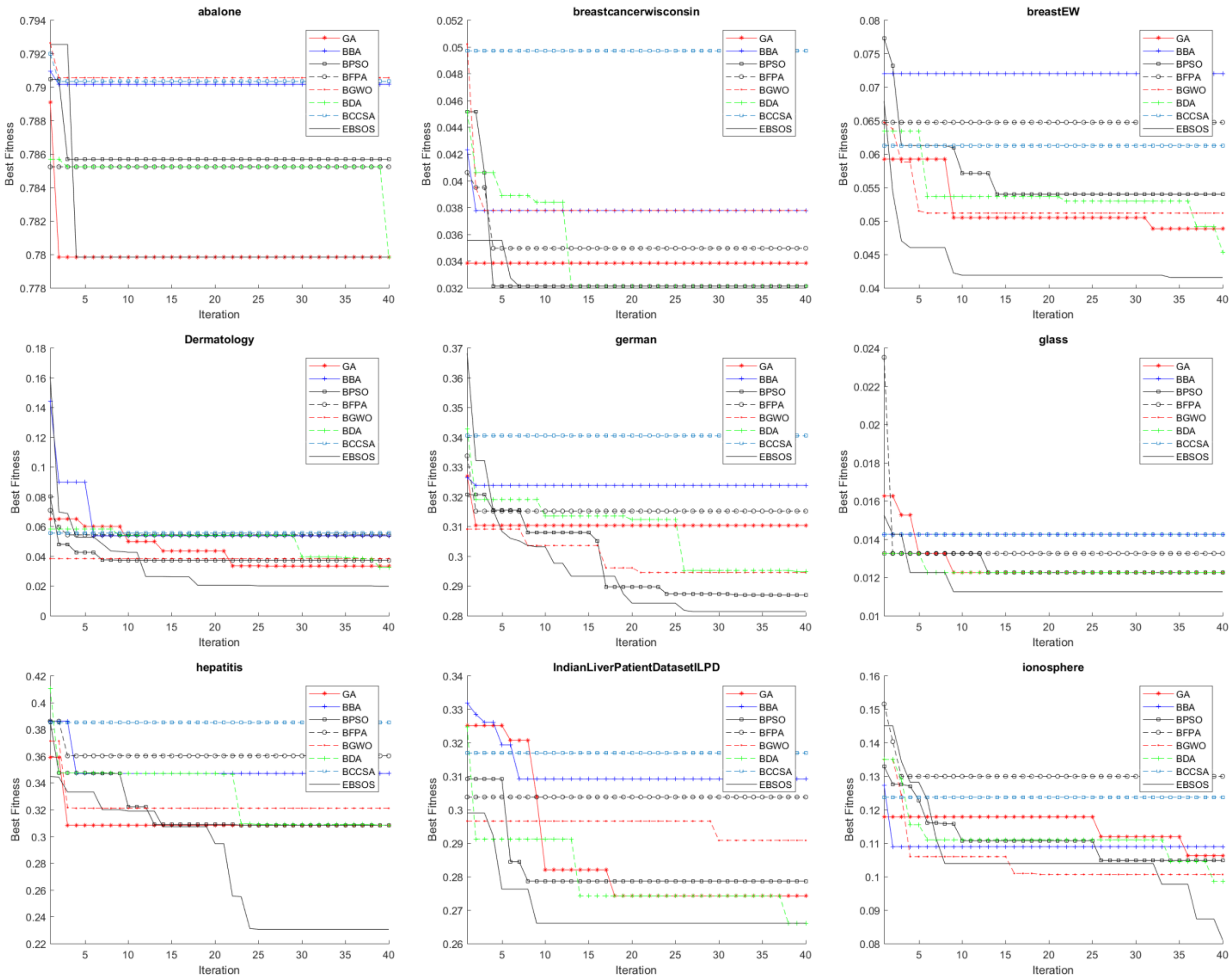

Fig. 17. Comparing the three proposed approaches in terms of objective function convergence on D1: D9 dataset with iteration 40

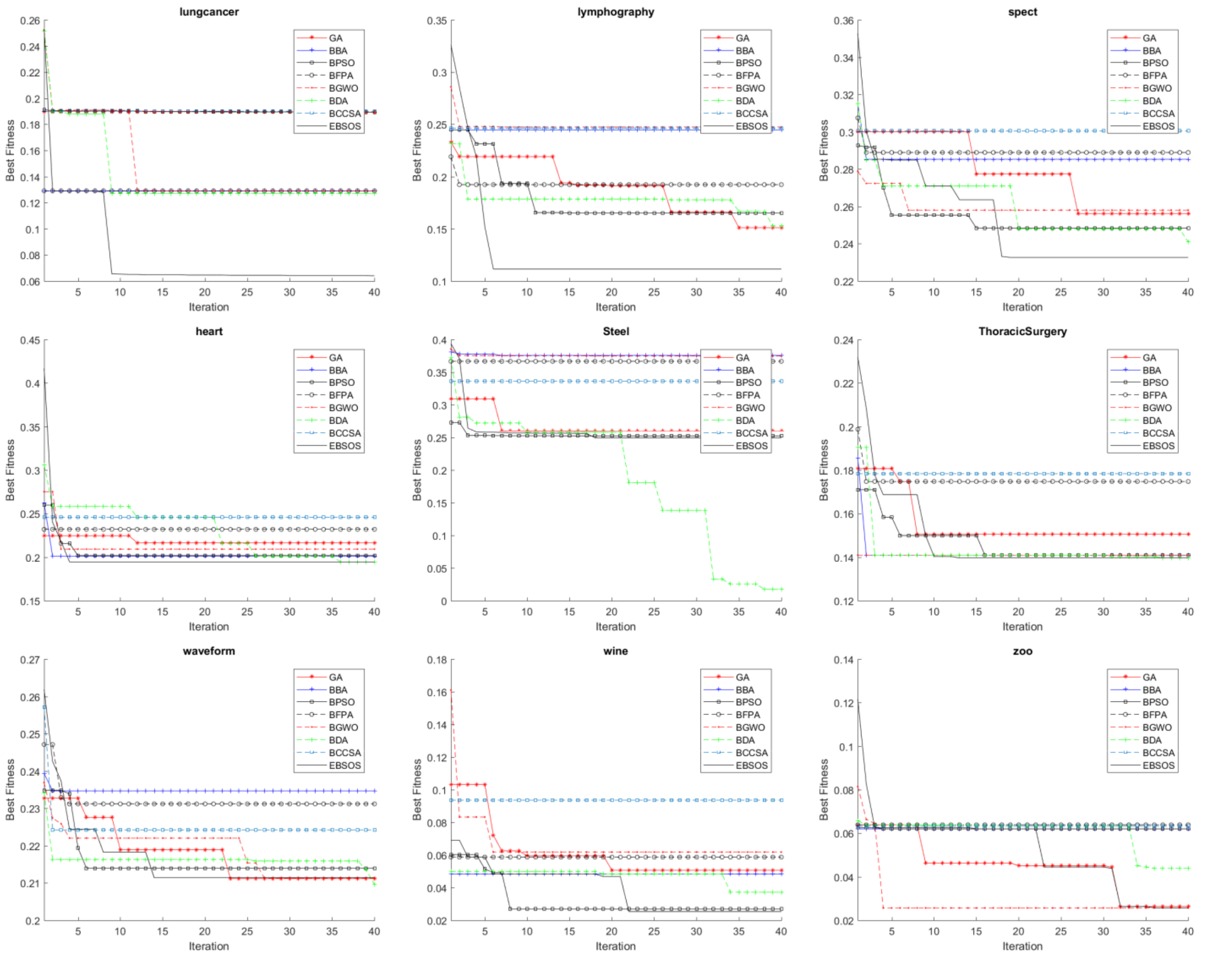

Fig. 18. Comparing the three proposed approaches in terms of objective function convergence on D10: D18 dataset with iteration 40

Comparison of the proposed EBSOS approach with other meta-algorithms in terms of statistical criteria with iteration of 40 and convergence rate in Table (11) and Figures (17) and (18) shows that the proposed EBSOS approach is a powerful method to solve the feature selection problem. In the 18 datasets, it was able to perform 98% more successfully than other algorithms in terms of statistical criteria, including best, worst, average and standard deviation, and the convergence rate of the proposed EBSOS approach is better than other meta-heuristic algorithms. The proposed EBSOS approach with other meta-heuristic algorithms is then implemented in terms of the mean number of features and the results are presented in Table 12. Population number and iterations were considered 10 and 40 in this experiment.

Table 12: *Comparison of EBSOS proposed approach with other meta-heuristic algorithms for mean number of features with iteration 40*

Dataset	GA	BBA	BPSO	BFPA	BGWO	BDA	BCCSA	EBSOS
D1	5	**4.8**	3.881	5.5	5.2	4.7	8	**4.8**
D2	5	4.8	4.6269	5	7.4	5.8	**4**	**4**
D3	12.7	9.6	16.7733	18.7	19.3	12.4	17	**9.3**
D4	21.1	13.9	17.1872	20.9	21.1	18.8	23	**12.2**
D5	12.9	**8.1**	11.5664	16.3	13.1	9.5	14	10
D6	3	2.7	4.5849	5	4.2	3.6	5	**2**
D7	7	7.6	9.2968	12.1	8.2	7.4	8	**4**
D8	**3**	3.3	4.6672	6.8	4.9	5.3	5	5
D9	17.4	11.2	15.432	20.8	17.1	12	19	**9.1**
D10	21.7	16.6	27.4564	34.5	31.1	20.4	25	**12.6**
D11	9.4	**4.4**	8.4831	10.2	12.7	10.3	10	8
D12	11	8	10.4258	14.2	14.9	9.5	11	**7.9**
D13	**5**	**5**	6.3034	8.5	5.5	5.2	**5**	**5**
D14	16	11.3	16.9174	21.6	20.2	13.9	16	**9**
D15	9	5.9	6.8494	11.1	4.2	4.2	9	**2.3**
D16	15.7	**8.5**	11.8806	14	18.9	15.9	21	14
D17	8	4.1	6.4633	9	7.7	5.2	6	**4**
D18	11	4	7.1981	9.6	10.7	7.9	8	9.8

The results of comparing the proposed EBSOS approach with other meta-heuristic algorithms in terms of the mean number of features show that the EBSOS approach performed the best in terms of feature selection, so that in the 18 datasets were able to be 72% more successful than Other meta-algorithms such as GA, BBA, BPSO, BFPA, BGWO, BDA, BCCSA operate. Of course, in addition to feature selection, the classification accuracy criterion should also be taken into account, which is then compared with the proposed EBSOS approach with other meta-heuristic algorithms in terms of accuracy criterion with iteration 40 and the results are shown in Table (13).

Table 13: *Comparing the proposed EBSOS Approach with other meta-heuristic algorithms with iteration 40*

Dataset	GA	BBA	BPSO	BFPA	BGWO	BDA	BCCSA	EBSOS
D1	**0.2186**	0.1516	0.2102	0.1956	0.2065	0.2186	0.2118	**0.2186**
D2	0.9714	0.8914	0.9743	0.96	0.9686	**0.9743**	0.9543	**0.9743**
D3	0.9544	0.8456	0.9509	0.9298	0.9544	0.9579	0.9439	**0.9614**
D4	0.9727	0.8907	0.9672	0.7923	0.9672	0.9727	0.9508	**0.9836**
D5	0.692	0.608	0.714	0.664	0.708	0.706	0.662	**0.72**
D6	**0.9907**	0.3832	**0.9907**	0.9813	**0.9907**	**0.9907**	**0.9907**	**0.9907**
D7	0.6923	0.6026	0.6923	0.6282	0.6795	0.6923	0.6154	**0.7692**
D8	0.726	0.6267	0.7226	0.6575	0.7123	**0.7363**	0.6849	**0.7363**
D9	0.8977	0.8693	0.8977	0.8466	0.9034	0.9034	0.8807	**0.9205**
D10	0.8125	0.6875	0.8125	0.75	0.875	0.875	0.8125	**0.9375**
D11	0.8514	0.7162	0.8378	0.6622	0.7568	0.8514	0.7568	**0.8919**
D12	0.7463	0.6866	0.7537	0.6045	0.7463	0.7612	0.7015	**0.7687**
D13	0.7852	0.6815	0.8	0.6296	0.7926	0.8074	0.7556	**0.8074**
D14	0.7425	0.6087	0.7497	0.5953	0.6272	**0.9866**	0.6653	0.7508
D15	0.8511	0.783	**0.8596**	0.8298	0.8596	**0.8596**	0.8255	**0.8596**
D16	0.7944	0.7072	0.7916	0.7496	0.7952	**0.796**	0.7836	0.7932
D17	0.9551	0.5843	0.9775	0.7865	0.9438	0.9663	0.9101	**0.9775**
D18	**0.9804**	0.8431	0.9412	0.902	0.9804	0.9608	0.9412	0.9608

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 January 2020 doi:10.20944/preprints202001.0318.v1

The results of comparing the proposed EBSOS approach with other meta-heuristic algorithms with iteration number of 40 in terms of the classification accuracy criterion presented in Table (13) show that the EBSOS approach has the best performance in terms of classification accuracy, as it was able to be 89% more successful than other meta-algorithms including GA, BBA, BPSO, BFPA, BGWO, BDA, BCCSA in 18 datasets. Consequently, the results obtained in terms of accuracy criterion and average number of selected features, the EBSOS approach has been able to prove its superiority over robust basic meta-heuristic methods in feature selection. In the following section, we evaluate the EBSOS approach and other comparative meta-heuristic algorithms in terms of statistical criteria and other criteria with higher number of iterations (iteration 80), while the results are shown in Tables (14) to (15) and Figures (19) and (20).

Table 14: *Comparing the proposed EBSOS approach with other meta-heuristic algorithms in terms of Statistical Criteria with iteration 80*

Dataset	cirita	GA	BBA	BPSO	BFPA	BGWO	BDA	BCCSA	EBSOS
D1	Best	0.7872	0.7866	0.7857	0.7793	0.7938	0.7793	0.7868	**0.7793**
	Mean	0.7872	20.657	0.7857	0.8022	0.795	0.7809	0.7868	**0.7793**
	Worst	0.7872	100	0.7857	0.826	0.7979	0.7952	0.7868	**0.7793**
	Std	0	41.8174	0	0.0109	0.0016	0.005	0	**0**
D2	Best	**0.0384**	0.0434	0.0434	0.0406	0.0395	0.0389	**0.0384**	**0.0384**
	Mean	**0.0384**	0.0819	0.0456	0.0564	0.0438	0.0389	**0.0384**	**0.0384**
	Worst	**0.0384**	0.1063	0.0536	0.0723	0.0491	0.0389	**0.0384**	**0.0384**
	Std	**0**	0.0189	0.0042	0.0084	0.0034	**0**	**0**	**0**
D3	Best	0.0391	0.0588	0.0498	0.0695	0.0518	0.0419	0.0574	**0.0384**
	Mean	0.0391	0.101	0.0506	0.0919	0.0535	0.0537	0.0574	**0.0384**
	Worst	0.0391	0.1281	0.0568	0.1099	0.0553	0.0662	0.0574	**0.0384**
	Std	**0**	0.0166	0.0022	0.0138	0.0016	0.0096	0	**0**
D4	Best	0.0209	0.0856	0.0363	0.0714	0.0296	0.0269	0.0687	**0.0201**
	Mean	0.0209	0.1587	0.0363	0.122	0.0298	0.0291	0.0687	**0.0201**
	Worst	0.0209	0.2671	0.0366	0.2226	0.0302	0.038	0.0687	**0.0201**
	Std	0	0.0542	0.0001	0.0524	0.0003	0.0045	0	**0**
D5	Best	0.2881	0.3071	0.2775	0.314	0.2957	0.2664	0.3187	**0.2675**
	Mean	0.2881	0.3772	0.2787	0.3436	0.3041	0.2725	0.3187	**0.2675**
	Worst	0.2881	0.4017	0.2878	0.3885	0.3274	0.3024	0.3187	**0.2675**
	Std	0	0.0191	0.0032	0.0229	0.0108	0.0118	0	**0**
D6	Best	0.004	0.008	0.004	0.006	0.004	**0.003**	0.007	**0.003**
	Mean	0.004	0.4832	0.0048	0.1754	0.0063	0.0033	0.007	**0.003**
	Worst	0.004	0.7422	0.0123	0.4676	0.0163	0.005	0.007	**0.003**
	Std	0	0.1925	0.0026	0.1803	0.0035	0.0007	**0**	**0**
D7	Best	0.3332	0.3728	0.3712	0.3866	0.3718	0.2565	0.4088	**0.2554**
	Mean	0.3639	0.5016	0.3713	0.4869	0.3721	0.3429	0.4088	**0.2554**
	Worst	0.3723	0.5997	0.3718	0.5653	0.3723	0.4982	0.4088	**0.2554**
	Std	0.0127	0.0563	0.0002	0.0429	0.0003	0.0958	0	**0**
D8	Best	0.3247	**0.2994**	0.3057	0.3363	0.3678	**0.2994**	0.3329	**0.2994**
	Mean	0.3247	0.3837	0.3108	0.3763	0.3688	0.2998	0.3329	**0.2994**
	Worst	0.3247	0.4122	0.3474	0.3929	0.3698	0.3037	0.3329	**0.2994**
	Std	0	0.0158	0.013	0.0085	0.0005	0.0014	0	**0**
D9	Best	0.0938	0.111	0.0879	0.1231	0.1012	0.0826	0.1237	**0.0826**
	Mean	0.0992	0.1414	0.0912	0.1561	0.1044	0.0966	0.1237	**0.0825**
	Worst	0.1003	0.1619	0.1214	0.1749	0.1071	0.111	0.1237	**0.0826**
	Std	0.0019	0.0144	0.0106	0.0113	0.0029	0.0121	**0**	0.0002
D10	Best	0.128	0.1883	0.1271	0.1289	0.0683	0.0032	0.1291	**0.003**
	Mean	0.1284	0.307	0.1272	0.2539	0.0748	0.0652	0.1291	**0.003**

	Worst	0.1286	0.3763	0.1277	0.3766	0.1305	0.189	0.1291	**0.003**
	Std	0.0003	0.0543	0.0002	0.0712	0.0196	0.0714	**0**	**0**
D11	Best	0.1795	0.2051	0.2174	0.2213	0.1823	0.1377	0.2374	**0.1237**
	Mean	0.1795	0.4279	0.2401	0.2861	0.1901	0.1539	0.2374	**0.1237**
	Worst	0.1795	0.6165	0.2971	0.3679	0.1956	0.2174	0.2374	**0.1237**
	Std	**0**	0.1203	0.0367	0.049	0.0067	0.0307	**0**	**0**
D12	Best	0.3001	0.3495	0.2774	0.3379	0.2784	0.2779	0.3439	**0.277**
	Mean	0.3001	0.4086	0.2844	0.3914	0.2847	0.3192	0.3439	**0.277**
	Worst	0.3001	0.4469	0.3227	0.4418	0.2936	0.3804	0.3439	**0.277**
	Std	**0**	0.0239	0.0152	0.0286	0.0049	0.0445	**0**	**0**
D13	Best	0.2262	0.2393	0.2312	0.2841	0.1659	0.1937	0.2532	**0.1652**
	Mean	0.2262	0.3728	0.2414	0.4087	0.2069	0.1937	0.2532	**0.1652**
	Worst	0.2262	0.4601	0.3331	0.4689	0.2254	0.1937	0.2532	**0.1652**
	Std	**0**	0.0523	0.0322	0.0437	0.0169	**0**	**0**	**0**
D14	Best	0.2485	0.1679	0.3133	0.4214	0.3017	0.2447	0.2916	**0.0021**
	Mean	0.2485	0.4206	0.3133	0.4402	0.3021	0.2483	0.2916	**0.0021**
	Worst	0.2485	0.4473	0.3136	0.445	0.3023	0.2789	0.2916	**0.0021**
	Std	**0**	0.0438	0.0001	0.0074	0.0002	0.0108	**0**	**0**
D15	Best	0.1464	0.183	0.1428	0.1855	0.1464	0.1409	0.1693	**0.1409**
	Mean	0.1464	0.2843	0.148	0.2309	0.1546	0.1435	0.1693	**0.1409**
	Worst	0.1464	0.4088	0.1819	0.2795	0.1687	0.1542	0.1693	**0.1409**
	Std	**0**	0.0798	0.0121	0.0289	0.0094	0.0047	**0**	**0**
D16	Best	0.23	0.2435	0.2413	0.2309	0.2223	0.2124	0.2345	**0.2088**
	Mean	0.23	0.3276	0.2458	0.2793	0.2235	0.2166	0.2345	**0.2088**
	Worst	0.23	0.357	0.2621	0.319	0.2278	0.225	0.2345	**0.2088**
	Std	**0**	0.0234	0.0073	0.0257	0.0018	0.0055	**0**	**0**
D17	Best	0.061	0.0714	0.0602	0.0506	0.0506	**0.0491**	0.0729	0.0499
	Mean	0.061	0.259	0.0603	0.2774	0.052	**0.0492**	0.0729	0.0497
	Worst	0.061	0.2915	0.061	0.3629	0.0618	**0.0499**	0.0729	0.0506
	Std	**0**	0.0258	0.0002	0.0627	0.0035	0.0002	**0**	0.0005
D18	Best	0.1403	0.1227	0.1221	0.1215	0.1033	0.0996	0.1597	**0.0996**
	Mean	0.1403	0.3297	0.1283	0.183	0.1037	0.0997	0.1597	**0.0996**
	Worst	0.1403	0.6237	0.1622	0.3143	0.1046	0.1002	0.1597	**0.0996**
	Std	**0**	0.1311	0.0135	0.0478	0.0004	0.0003	**0**	**0**

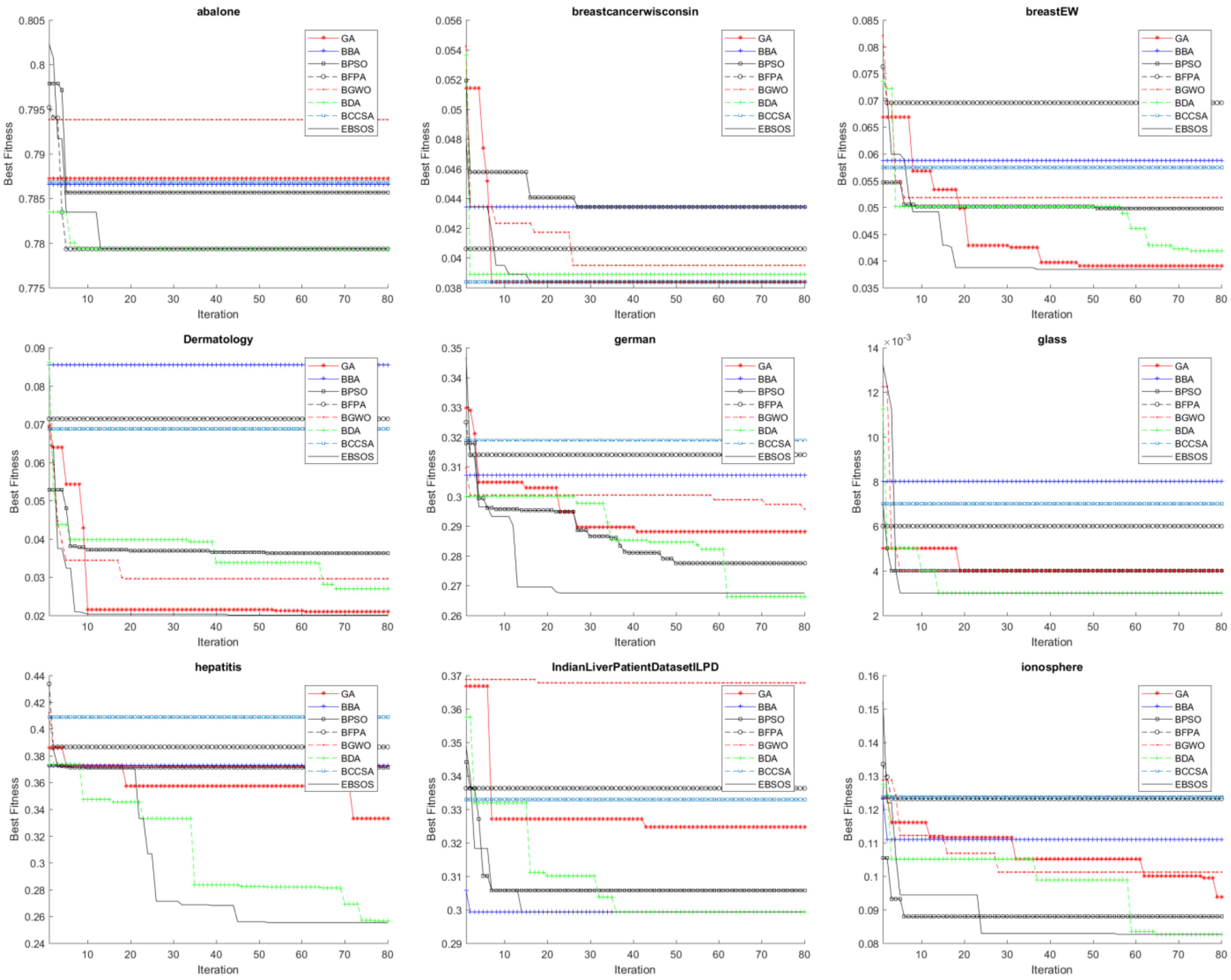

Fig. 19. *Comparing three proposed approaches in terms of objective function convergence on D1: D9 dataset with iteration 80*

 doi:10.20944/preprints202001.0318.v1

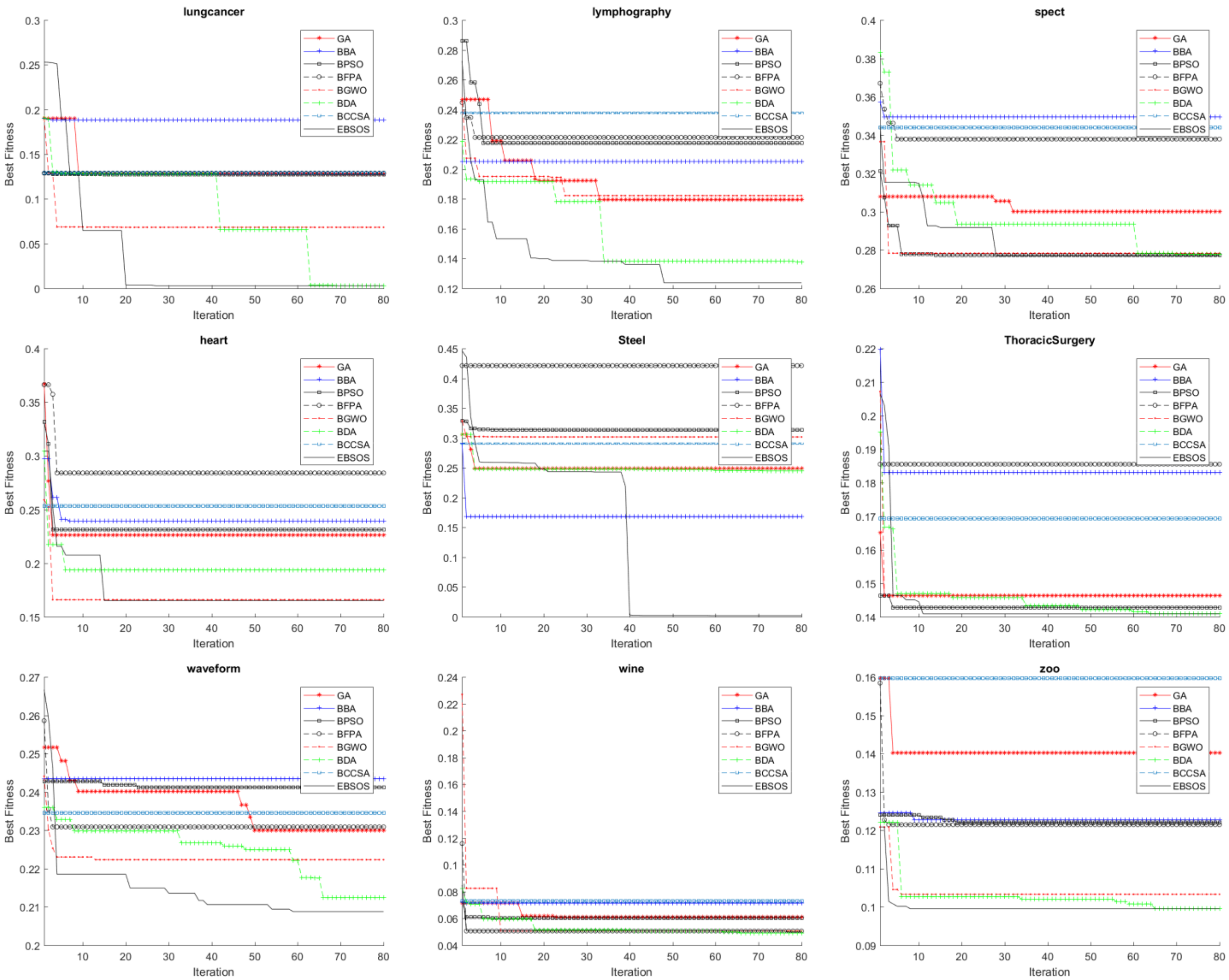

Fig. 20. *Comparing three proposed approaches in terms of objective function convergence on D10: D20 dataset with iteration 80*

Comparison of the proposed EBSOS approach with other meta-heuristic algorithms in terms of statistical criteria with iteration 80 and convergence rate in Table (14) and Figures (19) and (20), respectively, shows that the proposed EBSOS approach is a powerful method in solving feature selection problem. , In which 18 datasets were able to perform about 95% more successful than other algorithms in terms of statistical criteria, including best, worst, average and standard deviation, and the convergence rate of the proposed EBSOS approach was better than other meta-heuristic algorithms. In the following, the proposed EBSOS approach is compared with other meta-heuristic algorithms in terms of average number of features and the results are shown in Table (15). In this experiment, the population number is 10 and the number of iterations is set to 80.

Table 15: *Comparing the proposed EBSOS approach with other meta-heuristic algorithms in terms of mean number of features with iteration 80*

Dataset	GA	BBA	BPSO	BFPA	BGWO	BDA	BCCSA	EBSOS
D1	5	3.1	4.6884	5.1	6.8	3.1	4	**3**
D2	**4**	4.2	4.7266	5.2	5.3	7	**4**	**4**
D3	13	12.4	14.6098	19.2	19.8	11.1	16	**11**
D4	16	13.4	16.0741	21.2	27.9	18.1	**13**	**13**
D5	12	**6.6**	12.7753	15.4	15.1	12.1	14	**6.6**
D6	4	3.9	4.3107	6.1	5.4	3.3	7	**3**
D7	6.6	5.4	8.6005	11.2	7.7	5.3	5	**3**
D8	6	3.6	4.765	5.7	6	3.1	4	**2.8**
D9	14.1	14.1	13.5614	23.8	20.3	13	19	**12.8**
D10	26.1	21.3	25.2913	36	37.5	18.5	30	**17**
D11	10	**4.4**	7.6959	11.7	14.7	7.3	18	6
D12	10	**6.6**	11.1363	14.3	11.9	9.8	9	8
D13	8	6.1	5.5605	8.9	5.8	**4**	5	**4**
D14	16	12.3	15.6164	21	21.2	10.3	17	**7**
D15	5	6	7.1731	10.2	6	3.1	8	**3**
D16	14	11.3	12.7082	14.4	16.3	11.6	21	**8**
D17	7	7.1	6.4885	9.2	8.3	**6.1**	8	**6.1**
D18	7	5.7	9.7567	10.2	10.7	4.3	7	**4**

The results of Table (15) show that the EBSOS approach has the best performance in terms of feature selection, with 18 datasets being about 89% more successful than other meta-heuristic algorithms including GA, BBA, BPSO, BFPA, BGWO, BDA , BCCSA. Of course, in addition to feature selection, the classification accuracy criterion must also be considered, which is then compared with the proposed EBSOS approach with other meta-heuristic algorithms in terms of accuracy criterion with iteration 80 and the results are shown in Table (16).

Table 16: *Comparing the proposed EBSOS approach with other meta-heuristic algorithms with iteration 80*

Dataset	GA	BBA	BPSO	BFPA	BGWO	BDA	BCCSA	EBSOS
D1	0.2111	0.1663	0.214	0.2036	0.207	0.2166	0.2103	**0.2166**
D2	0.9657	0.94	0.9629	0.9514	0.9657	**0.9686**	0.9657	0.9611
D3	0.9649	0.9228	0.9544	0.9123	0.9544	0.9614	0.9474	**0.9649**
D4	0.9836	0.8852	0.9672	0.8962	0.9781	0.9781	0.9344	**0.9836**
D5	0.714	0.618	0.726	0.67	0.708	0.736	0.684	**0.734**
D6	1	0.5514	1	0.9907	1	1	1	**1**
D7	0.6667	0.5256	0.6282	0.5769	0.6282	0.7436	0.5897	**0.7436**
D8	0.6781	0.6336	0.6952	0.6267	0.6336	0.6986	0.6678	**0.6986**
D9	0.9091	0.8807	0.9148	0.8352	0.9034	0.9205	0.8807	**0.9205**
D10	0.875	0.75	0.875	0.8125	0.9375	1	0.875	**1**
D11	0.8243	0.4324	0.7838	0.6216	0.8243	0.8649	0.7703	**0.8784**
D12	0.7015	0.6119	0.7239	0.6567	0.7239	0.7239	0.6567	**0.7239**
D13	0.7778	0.6	0.7704	0.6074	0.837	0.8074	0.7481	**0.837**
D14	0.7539	0.5994	0.6869	0.5572	0.7013	0.7559	0.7106	**1**
D15	0.8553	0.8128	0.8596	0.7787	0.8553	0.8596	0.834	**0.8596**
D16	0.7744	0.6764	0.764	0.7096	0.7836	0.7912	0.7732	**0.7968**
D17	0.9438	0.7303	0.9438	0.7416	0.9551	0.9551	0.9326	**0.9551**
D18	0.8627	0.7647	0.8824	0.8235	0.902	0.902	0.8431	**0.902**

The results of Table (16) show that the EBSOS approach has the best performance in terms of classification accuracy, with 18 datasets being about 95% more successful than other meta-heuristic algorithms such as GA, BBA, BPSO, BFPA, BGWO, BDA, BCCSA. From the results obtained in

 doi:10.20944/preprints202001.0318.v1

terms of the criterion of accuracy and the average number of selected features, the EBSOS approach has been able to prove its remarkable superiority over robust basic meta-heuristic methods in feature selection. Therefore, all experiments in this section, in terms of mean criteria of feature selection, cluster accuracy, objective function convergence rate as well as statistical criteria including best, worst, mean and standard deviation of EBSOS's proposed approach over other meta-algorithms such as GA, BBA, BPSO BCFSA, BFPA, BGWO, BDA, BCCSA in the discussion of feature selection proved well. In addition, the proposed EBSOS approach is evaluated in spam email detection in a specific and functional way in the next section (5.5).

5.5 Applied study on Email data

The proposed EBSOS approach is implemented in the previous section on 18 valid UCI datasets. Results showed that the proposed EBSOS approach is significantly superior to other meta-heuristic algorithms in selecting fewer features and classification accuracy as well as other statistical criteria; Given the robust results of the proposed algorithm on valid UCI datasets, we were motivated to implement our proposed EBSOS approach on spam email datasets and compared criteria such as accuracy, sensitivity and accuracy of the clusters for performance. We used a valid spam mail database called Spambase to perform this test. This dataset contains 4601 records and 58 attributes and the last attribute is concerned with the class. There are also 4601 records of 2788 regular emails and 1813 spam emails [30-34]. We split the spam email dataset into test (30%) and training (70%) datasets. We also considered the initial population number of all algorithms equal to 10 and the number of iterations to 20. This section uses well-known criteria such as accuracy, sensitivity and accuracy formulated in relations (11), (12) and (13), respectively:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{12}$$

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

Each of these criteria is defined in the range of [0.1], with a higher number indicating better classification quality and accuracy. In this section, four experiments have been carried out, each of which uses the proposed EBSOS approach with the SVM, KNN, NB, and MLP classifiers to detect spam emails, as well as three implementations are considered for each category for better evaluation and are compared in terms of accuracy, sensitivity and accuracy. In this section, in the all experiments the initial population number is 10 and the proposed approach iteration number is 20. In the following, four experiments are presented with combination of different classifiers and the proposed EBSOS approach. In addition, in all experiments the speed reduction rate of each classifier is shown using the proposed EBSOS approach to evaluate the rate of acceleration of the classifier algorithms using the proposed EBSOS approach.

In the first experiment of this subsection, an evaluation of the proposed EBSOS approach in terms of the accuracy, sensitivity, and accuracy of KNN spam mail detection with three different implementations is shown in Figure (21) and Table (17).

Table 17: *Comparing the evaluation of the proposed EBSOS approach in terms of accuracy, sensitivity and precision to email spam detection with KNN classification and with three different implementations*

Run	Run1		Run2		Run3		Run4	
Algorithm	KNN	KNN-EBSOS	KNN	KNN-EBSOS	KNN	KNN-EBSOS	KNN	KNN-EBSOS
Accuracy	0.81159	**0.92391**	0.80725	0.932610	0.80797	**0.94130**	0.800720	**0.93116**
Precision	0.82921	**0.92749**	0.83929	0.939140	0.82598	**0.94279**	0.81893	**0.93420**
Sensitivity	0.85958	**0.94856**	0.84330	0.949340	0.85345	**0.95995**	0.85384	**0.95324**
Time(s)	0.24002	**0.16056**	0.22163	0.073975	0.22004	**0.12194**	0.27435	**0.11589**

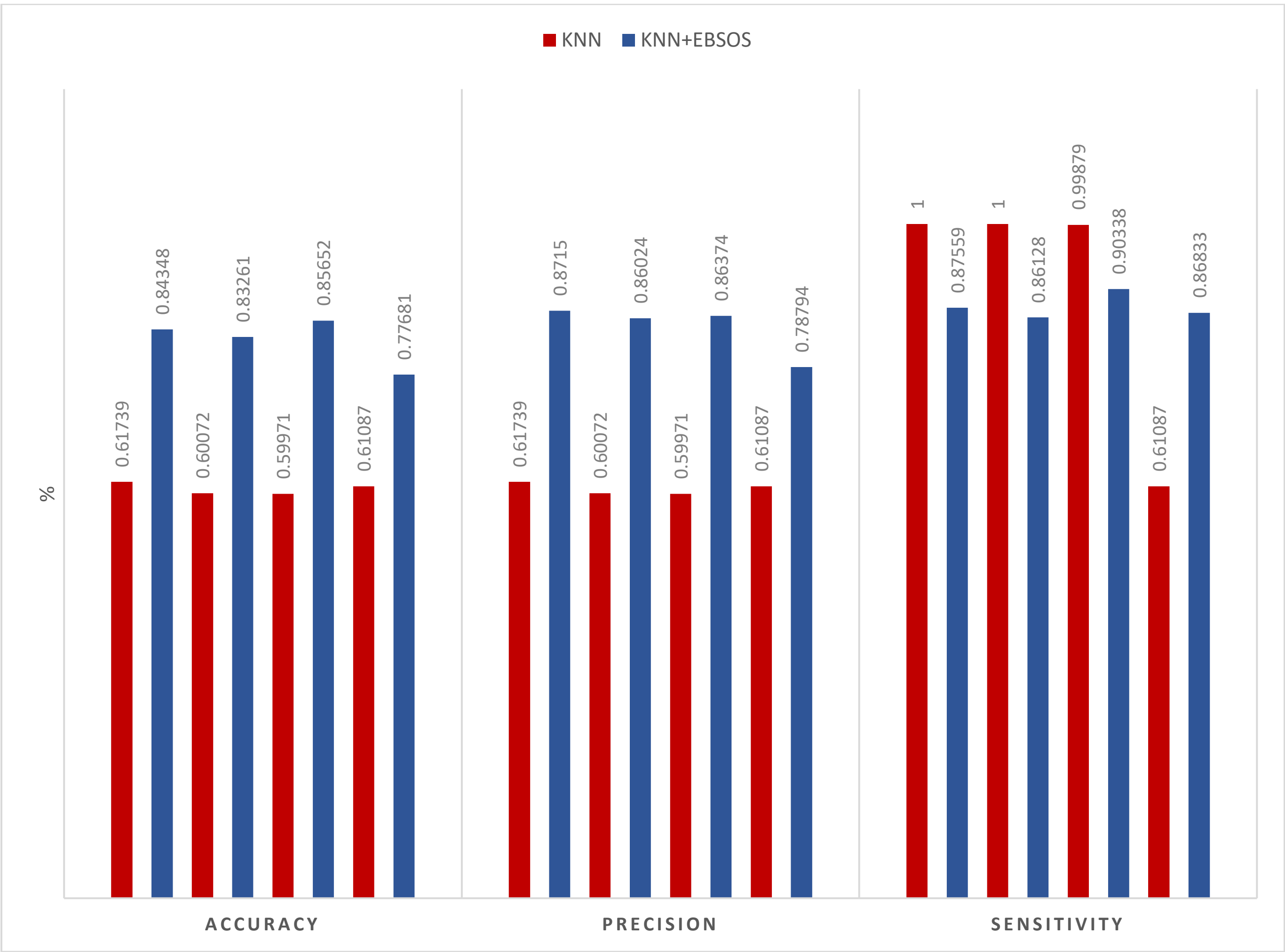

Fig. 21. Comparison of EBSOS proposed approach in terms of accuracy, sensitivity and precision to email spam detection with KNN classification and with three different implementations

The test results presented in Figure (21) show that the proposed EBSOS approach has been able to improve the KNN classification in terms of accuracy, sensitivity, and accuracy by improving the accuracy of the algorithm up to 60%. In addition, the proposed EBSOS approach has reduced the time of this classifier by 50%. In the second test, the evaluation of the proposed EBSOS approach in terms of accuracy, sensitivity and validity of spam email detection with NV blocking with three different implementations is shown in Figure (22) and Table (18).

Table 18: *Comparing the evaluation of the proposed EBSOS approach in terms of accuracy, sensitivity and precision to email spam detection with NV classification and with three different implementations*

Run	Run1		Run2		Run3		Run4	
Algorithm	NV	NV-EBSOS	NV	NV-EBSOS	NV	NV-EBSOS	NV	NV-EBSOS
Accuracy	0.5471	**0.88406**	0.53188	**0.80362**	0.52826	**0.88551**	0.55362	**0.88406**
Precision	0.97531	**0.89636**	**0.93914**	0.86135	0.97596	**0.91778**	**0.95582**	0.90698
Sensitivity	0.27687	**0.88406**	0.25814	**0.81628**	0.23910	**0.89399**	0.28233	**0.89818**
Time(s)	6.26630	**0.084185**	13.6611	**0.072005**	6.12060	**0.10822**	6.36180	**0.11589**

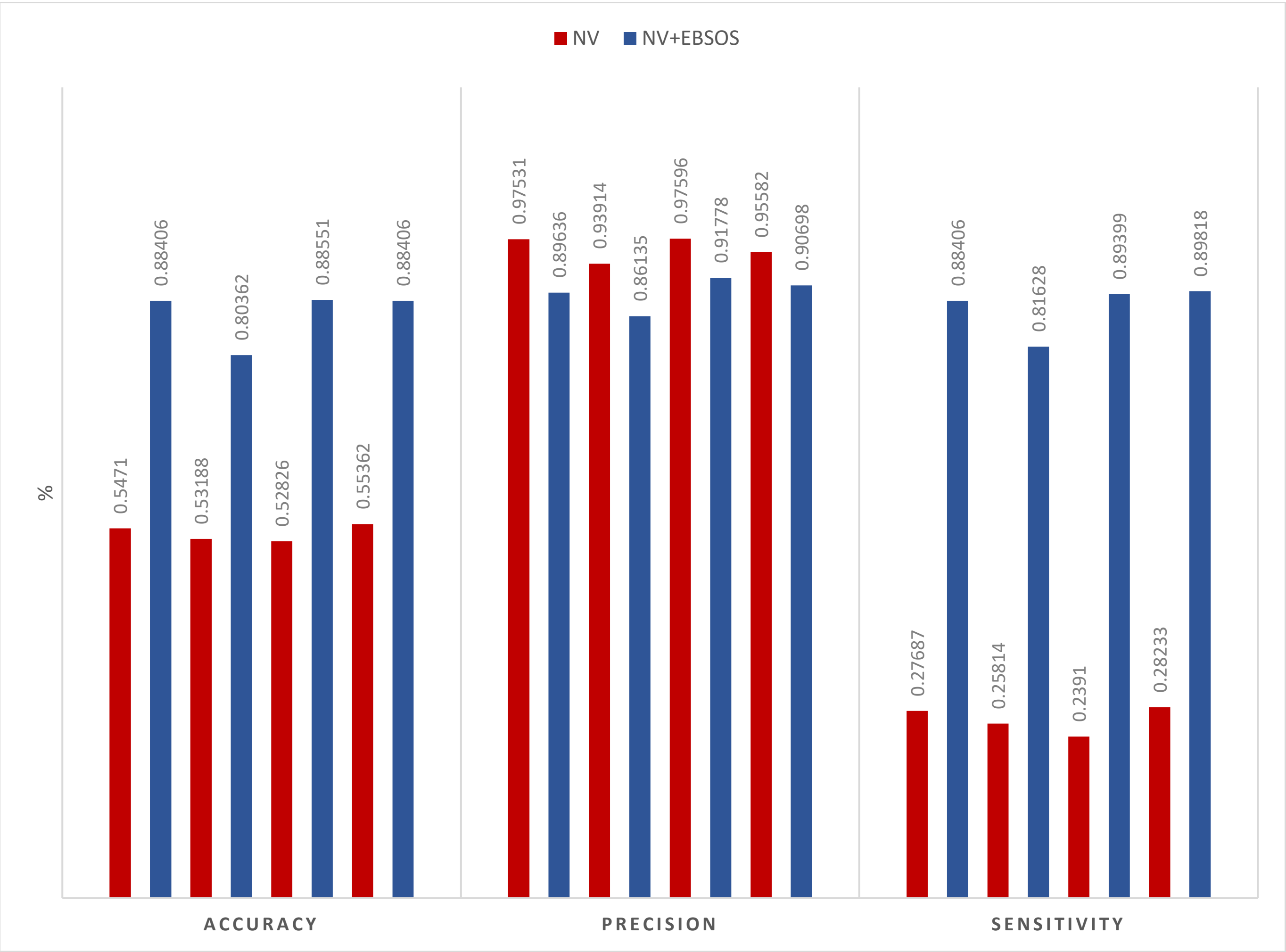

Fig. 22. Comparison of EBSOS proposed approach in terms of accuracy, sensitivity and precision to email spam detection with NV classification and with three different implementations

The test results presented in Figure (22) show that the proposed EBSOS approach has improved the NV classification in terms of accuracy and sensitivity. It improved the accuracy of the algorithm up to 38%. However, in terms of accuracy, it has not been able to improve this algorithm, but the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 January 2020 doi:10.20944/preprints202001.0318.v1

proposed EBSOS approach has significantly reduced the time of this classifier. In the third experiment, the evaluation of the proposed EBSOS approach in terms of accuracy, sensitivity and accuracy of spam mail detection with SVM cluster with three different implementations is shown in Figure (23) and Table (19).

Table 19: *Comparing the evaluation of the proposed EBSOS approach in terms of accuracy, sensitivity and precision to email spam detection with SVM classification and with three different implementations*

Run	Run1		Run2		Run3		Run4	
Algorithm	SVM	SVM-EBSOS	SVM	SVM -EBSOS	SVM	SVM-EBSOS	SVM	SVM-EBSOS
Accuracy	0.80870	**0.91637**	0.82319	**0.82101**	0.81667	**0.89710**	0.83188	**0.86594**
Precision	0.76503	**0.88773**	**0.77715**	0.85018	0.76952	**0.90695**	0.79037	**0.88399**
Sensitivity	0.98805	**0.87899**	0.99282	**0.85526**	0.99400	0.92437	0.98819	0.89965
Time(s)	3.06500	**0.12239**	1.58810	**0.09803**	1.15270	**0.08885**	1.24620	**0.068051**

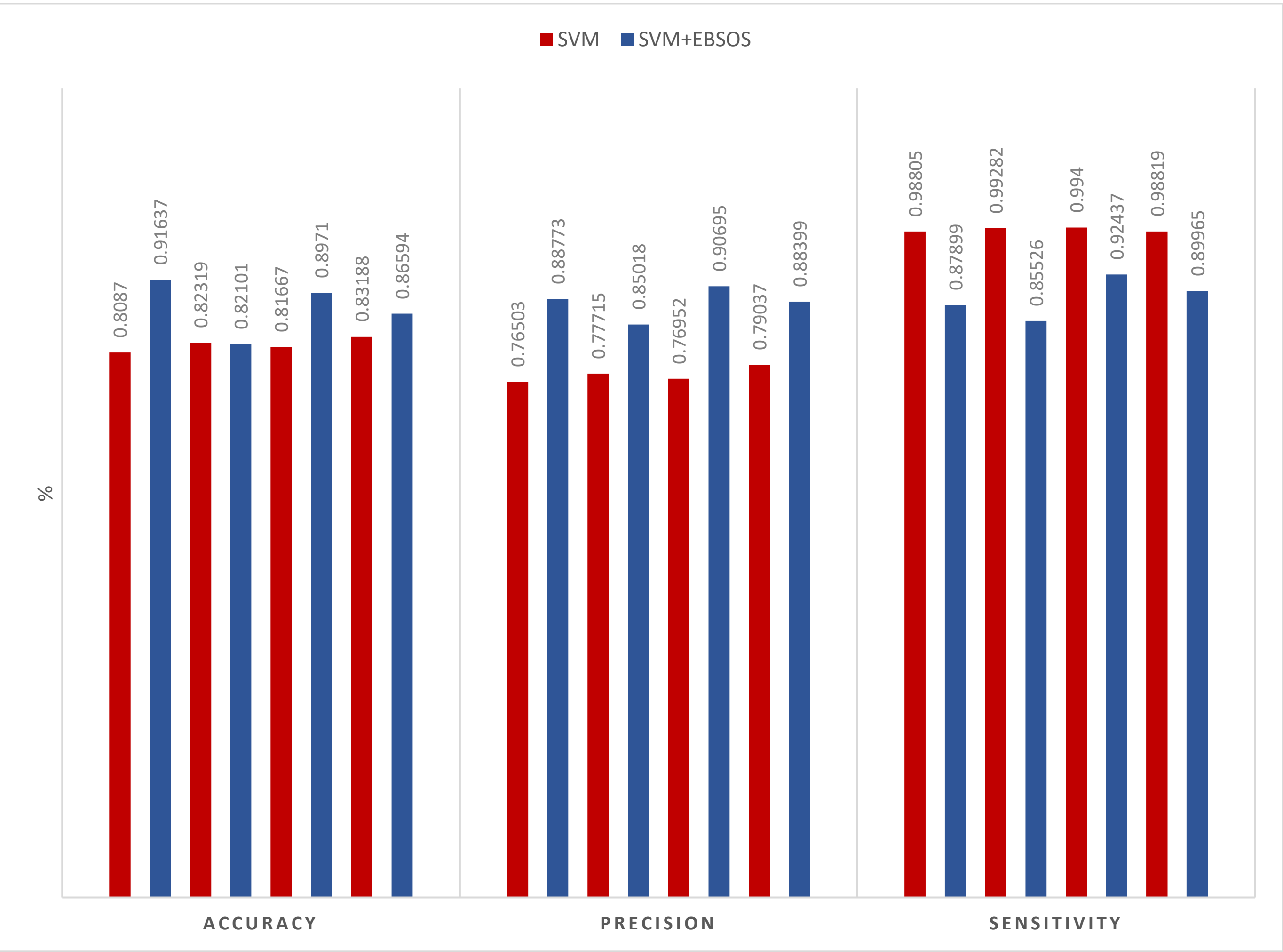

Fig. 23. Comparison of EBSOS proposed approach in terms of accuracy, sensitivity and precision to email spam detection with SVM classification and with three different implementations

The test results depicted in Figure (23) show that the proposed EBSOS approach has been able to improve the SVM classification in terms of accuracy and accuracy up to 11%. Although, in terms of accuracy, it has not been able to improve this algorithm, but the proposed EBSOS approach has significantly reduced the time of this classifier. In the fourth experiment, the proposed EBSOS approach is evaluated in terms of accuracy, sensitivity and validity of spam mail detection with MLP classifier with three different implementations, which is shown in Figure (24) and Table (20).

Table 20: *Comparing the evaluation of the proposed EBSOS approach in terms of accuracy, sensitivity and precision to email spam detection with MLP classification and with three different implementations*

Run	Run1		Run2		Run3		Run4	
Algorithm	MLP	MLP -EBSOS	MLP	MLP -EBSOS	MLP	MLP-EBSOS	MLP	MLP -EBSOS
Accuracy	0.61739	**0.84348**	0.60072	**0.83261**	0.59971	**0.85652**	0.61087	**0.77681**
Precision	0.61739	**0.8715**	0.60072	**0.86024**	0.59971	**0.86374**	0.61087	**0.78794**
Sensitivity	**1**	0.87559	1	0.86128	**0.99879**	0.90338	0.61087	0.86833
Time(s)	1.75	**0.050827**	37.9551	**0.057872**	24.4771	**0.022344**	88.8671	**0.026075**

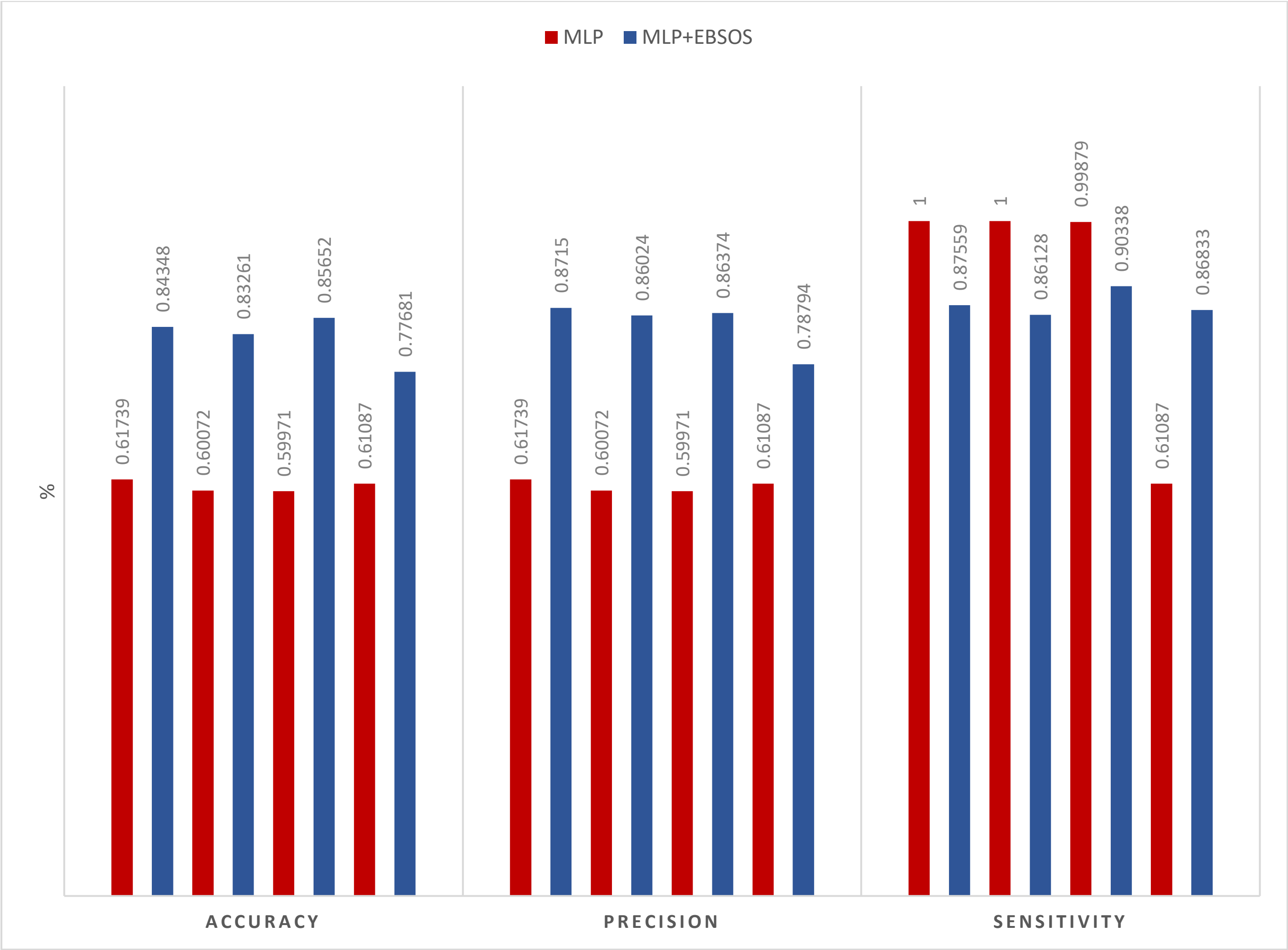

Fig. 24. Comparison of EBSOS proposed approach in terms of accuracy, sensitivity and precision to email spam detection with MLP classification and with three different implementations

The test results shown in Figure (24) indicate that the proposed EBSOS approach has improved the MLP classifier in terms of accuracy and validity by improving the accuracy of the algorithm up to 26%. However, in terms of sensitivity it has not been able to improve this algorithm, but the proposed EBSOS approach has significantly reduced the time of this classifier. The overall results of comparing the proposed EBSOS approach in terms of accuracy, sensitivity, and validity of spam mail detection with different categories show that the proposed EBSOS approach is a powerful method in feature selection to increase the speed and accuracy of the categories in all contexts. As in all experiments, the proposed EBSOS approach has been able to significantly reduce all classifiers time, and has also improved the accuracy of all clusters by 10 to 60%.

6. Conclusion and Future Works

Today, due to the vast amount of information, feature selection methods are one of the important steps in pre-processing information that aims to reduce dimensions by eliminating redundant features. However, the methods of feature selection should also consider the classifier accuracy while selecting the important feature and removing the redundant features. As a result, feature selection methods have a direct impact on the accuracy and speed of machine learning classifiers. Recent meta-heuristic algorithms have attracted the attention of many researchers because of their simplicity and random nature, as well as successful and promising methods for feature selection.

The coexistence search algorithm, which is inspired by the opposition of organisms in nature, is one of the most successful meta-heuristic algorithms. In this paper, three different approaches of the coexistence search algorithm, BSOSS, BSOSV and EBSOS, are presented to solve the feature selection problem. In the BSOSS approach, several S-shaped transfer functions are used to binarize the algorithm, and in the BSOSV approach several V-shaped transfer functions are used to binarize the algorithm. Finally, in the EBSOS approach, an advanced version of the coexistence search algorithm with two new operators, BMP and BCP, are presented to binarize the coexistence search algorithm.

Eventually, the three proposed approaches have been simulated, at first, the four BSOSS models are compared, and finally the S4 model is selected as the final BSOSS method with respect to feature selection results and classification accuracy, and the V4 model with respect to feature selection results. The classification accuracy is selected as the final BSOSV method and then the three proposed approaches BSOSS, BSOSV and EBSOS are compared with each other and the EBSOS approach is selected as the final proposed method. The proposed EBSOS approach is compared with other meta-heuristics methods such as genetic algorithm, bat binary algorithm, binary particle swarm algorithm, flower pollinator binary algorithm, gray wolf binary algorithm, dragonfly binary algorithm and chaotic-based binary search algorithm. The results of various experiments showed that the proposed EBSOS approach performs better in terms of number of features and accuracy than other methods. In addition, the proposed EBSOS approach is evaluated in spam emails detection in specific and functional applications in combination with SVM, KNN, NB, and MLP classifiers. These tests results showed that the proposed EBSOS approach significantly improved the accuracy and speed of all classifiers.

Acknowledgment

We appreciate the support of this research from the computer company of Apameh system and Vice-Chancellor for Research of Urmia Branch, Islamic Azad University, and we also thanks to Dr.

Bagherzadeh, Dr Masdari, and Dr. Majidzadeh, whos have been working hard to accomplish this research.

7. Refrences

1. Mafarja, M., et al., *Binary grasshopper optimisation algorithm approaches for feature selection problems.* Expert Systems with Applications, 2019. **117**: p. 267-286.
2. Li, Y., T. Li, and H. Liu, *Recent advances in feature selection and its applications.* Knowledge and Information Systems, 2017. **53**(3): p. 551-577.
3. Liu, H. and L. Yu, *Toward integrating feature selection algorithms for classification and clustering.* IEEE Transactions on Knowledge & Data Engineering, 2005(4): p. 491-502.
4. Arora, S. and P. Anand, *Binary butterfly optimization approaches for feature selection.* Expert Systems with Applications, 2019. **116**: p. 147-160.
5. Kabir, M.M., M. Shahjahan, and K. Murase, *A new hybrid ant colony optimization algorithm for feature selection.* Expert Systems with Applications, 2012. **39**(3): p. 3747-3763.
6. Faris, H., et al., *An efficient binary salp swarm algorithm with crossover scheme for feature selection problems.* Knowledge-Based Systems, 2018. **154**: p. 43-67.
7. Emary, E., H.M. Zawbaa, and A.E. Hassanien, *Binary ant lion approaches for feature selection.* Neurocomputing, 2016. **213**: p. 54-65.
8. Emary, E., H.M. Zawbaa, and A.E. Hassanien, *Binary grey wolf optimization approaches for feature selection.* Neurocomputing, 2016. **172**: p. 371-381.
9. Mafarja, M.M. and S. Mirjalili, *Hybrid whale optimization algorithm with simulated annealing for feature selection.* Neurocomputing, 2017. **260**: p. 302-312.
10. Al-Madi, N., H. Faris, and S. Mirjalili, *Binary multi-verse optimization algorithm for global optimization and discrete problems.* International Journal of Machine Learning and Cybernetics, 2019: p. 1-21.
11. Shayanfar, H. and F.S. Gharehchopogh, *Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems.* Applied Soft Computing, 2018. **71**: p. 728-746.
12. Mirjalili, S. and A. Lewis, *S-shaped versus V-shaped transfer functions for binary particle swarm optimization.* Swarm and Evolutionary Computation, 2013. **9**: p. 1-14.
13. Rodrigues, D., et al. *BCS: A binary cuckoo search algorithm for feature selection*. in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*. 2013. IEEE.
14. Mirjalili, S., S.M. Mirjalili, and X.-S. Yang, *Binary bat algorithm.* Neural Computing and Applications, 2014. **25**(3-4): p. 663-681.
15. Mafarja, M. and S. Mirjalili, *Whale optimization approaches for wrapper feature selection.* Applied Soft Computing, 2018. **62**: p. 441-453.
16. Hussien, A.G., et al., *S-shaped binary whale optimization algorithm for feature selection*, in *Recent trends in signal and image processing*. 2019, Springer. p. 79-87.
17. De Souza, R.C.T., et al. *A V-Shaped Binary Crow Search Algorithm for Feature Selection*. in *2018 IEEE Congress on Evolutionary Computation (CEC)*. 2018. IEEE.
18. Gharehchopogh, F.S., H. Shayanfar, and H. Gholizadeh, *A comprehensive survey on symbiotic organisms search algorithms.* Artificial Intelligence Review, 2019: p. 1-48.
19. Cheng, M.-Y. and D. Prayogo, *Symbiotic organisms search: a new metaheuristic optimization algorithm.* Computers & Structures, 2014. **139**: p. 98-112.
20. Prasad, D. and V. Mukherjee, *A novel symbiotic organisms search algorithm for optimal power flow of power system with FACTS devices.* Engineering Science and Technology, an International Journal, 2016. **19**(1): p. 79-89.

21. Mirjalili, S. and S.Z.M. Hashim, *BMOA: binary magnetic optimization algorithm.* International Journal of Machine Learning and Computing, 2012. **2**(3): p. 204.
22. Rashedi, E., H. Nezamabadi-Pour, and S. Saryazdi, *BGSA: binary gravitational search algorithm.* Natural Computing, 2010. **9**(3): p. 727-745.
23. Liao, T.W. and R. Kuo, *Five discrete symbiotic organisms search algorithms for simultaneous optimization of feature subset and neighborhood size of KNN classification models.* Applied Soft Computing, 2018. **64**: p. 581-595.
24. Mafarja, M., et al., *Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems.* Knowledge-Based Systems, 2018. **145**: p. 25-45.
25. Rajamohana, S. and K. Umamaheswari, *Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection.* Computers & Electrical Engineering, 2018. **67**: p. 497-508.
26. Altman, N.S., *An introduction to kernel and nearest-neighbor nonparametric regression.* The American Statistician, 1992. **46**(3): p. 175-185.
27. Asuncion, A. and D. Newman, *UCI machine learning repository*. 2007.
28. Sivanandam, S. and S. Deepa, *Genetic algorithm optimization problems*, in *Introduction to genetic algorithms*. 2008, Springer. p. 165-209.
29. Yang, X.-S., *A new metaheuristic bat-inspired algorithm*, in *Nature inspired cooperative strategies for optimization (NICSO 2010)*. 2010, Springer. p. 65-74.
30. Mabodi, K., Yusefi, M., Zandiyan, S., Irankhah, L., & Fotohi, R. (2020). Multi-level trust-based intelligence schema for securing of internet of things (IoT) against security threats using cryptographic authentication. The Journal of Supercomputing, 1-25.
31. Seyedi, B., & Fotohi, R. (2020). NIASHPT: a novel intelligent agent-based strategy using hello packet table (HPT) function for trust Internet of Things. The Journal of Supercomputing, 1-24.
32. Fotohi, R. (2020). Securing of Unmanned Aerial Systems (UAS) against security threats using human immune system. Reliability Engineering & System Safety, 193, 106675.
33. Fotohi, R., Firoozi Bari, S., & Yusefi, M. (2020). Securing Wireless Sensor Networks Against Denial-of-Sleep Attacks Using RSA Cryptography Algorithm and Interlock Protocol. International Journal of Communication Systems, 33(4), e4234.
34. Kennedy, J., *Particle swarm optimization.* Encyclopedia of machine learning, 2010: p. 760-766.
35. Yang, X.-S. *Flower pollination algorithm for global optimization*. in *International conference on unconventional computing and natural computation*. 2012. Springer.
36. Mirjalili, S., *Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems.* Neural Computing and Applications, 2016. **27**(4): p. 1053-1073.

33\. Sayed, G.I., A.E. Hassanien, and A.T. Azar, *Feature selection via a novel chaotic crow search algorithm.* Neural Computing and Applications, 2019. **31**(1): p. 171-188.

34\. Sakkis, G., et al., *Stacking classifiers for anti-spam filtering of e-mail.* arXiv preprint cs/0106040, 2001.