

Hybrid feature selection method based on particle swarm optimization and adaptive local search method

Malek Alzaqebah¹, Sana Jawarneh², Rami Mustafa A. Mohammad³, Mutasem K. Alsmadi⁴,
Ibrahim Al-marashdeh⁵, Eman A. E. Ahmed⁶, Nashat Alrefai⁷, Fahad A. Alghamdi⁸

^{1,6,7}Department of Mathematics, College of Science, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia

^{1,6,7}Basic and Applied Scientific Research Center, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia

²Computer Science Department, Community College Dammam, Imam Abdulrahman Bin Faisal University,
Dammam, Saudi Arabia

³Computer Information Systems Department, College of Computer Science and Information Technology,
Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia

^{4,5,8}Department of MIS, College of Applied Studies and Community Service, Imam Abdulrahman Bin Faisal University,
Dammam, Saudi Arabia

Article Info

Article history:

Received Sep 8, 2020

Revised Sep 21, 2020

Accepted Oct 6, 2020

Keywords:

Adaptive local search method
Feature selection
Particle swarm optimization
algorithm

ABSTRACT

Machine learning has been expansively examined with data classification as the most popularly researched subject. The accurateness of prediction is impacted by the data provided to the classification algorithm. Meanwhile, utilizing a large amount of data may incur costs especially in data collection and preprocessing. Studies on feature selection were mainly to establish techniques that can decrease the number of utilized features (attributes) in classification, also using data that generate accurate prediction is important. Hence, a particle swarm optimization (PSO) algorithm is suggested in the current article for selecting the ideal set of features. PSO algorithm showed to be superior in different domains in exploring the search space and local search algorithms are good in exploiting the search regions. Thus, we propose the hybridized PSO algorithm with an adaptive local search technique which works based on the current PSO search state and used for accepting the candidate solution. Having this combination balances the local intensification as well as the global diversification of the searching process. Hence, the suggested algorithm surpasses the original PSO algorithm and other comparable approaches, in terms of performance.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Malek Alzaqebah
Basic and Applied Scientific Research Center
Imam Abdulrahman Bin Faisal University
P.O. Box 1982, Dammam, Saudi Arabia
Email: maafehaid@iau.edu.sa

NOMENCLATURE

Acronyms

PSO : Particle Swarm Optimization

SSA : slap swarm algorithm

FFA : Firefly algorithm

FS : feature selection

CT : computed tomography

TS : Tabu search

BPSO : binary particle swarm optimization

BCO : bee colony optimization

LAHC : Late acceptance hill-climbing

ALS : adaptive local search

GA : Genetic algorithm

MFO : Moth-Flam optimization

ACO : ant colony optimization

NF : number of selected features

1. INTRODUCTION

Machine learning has become more prominent recently in many research fields, and this is due to the fast data growth and the need to meaningfully use them. Machine learning concerns discovering useful information from huge data using some machine learning techniques including anomaly detection, classification, and clustering [1, 2]. Accordingly, dimensionality can impede the machine learning process as it incurs high computational cost. Dimensionality is a major issue in machine learning, especially concerning datasets. A dataset comprises a set of examples representing information on a particular case in feature form, and dataset can have substantial dimensionality, aside from carrying features that are irrelevant and redundant, and noise of high level. Such a huge number of features could not be handled by traditional machine learning methods. Feature selection is therefore vital as a preprocessing phase as it decreases data dimensionality while also removing duplicating and useless features in the dataset [2-4]. Feature selection process aims to obtain the optimal set of useful features while maintaining good accurateness in representing the initial features of the dataset. In this regard, classification involves determining the class value of each sample from the available class pool [5, 6].

Feature selection techniques are divided into three categories according to the strategy of selection as follows: Filter, wrapper, and embedded techniques. Filter approaches do not require subsequent learning algorithms [7, 8], while wrapper techniques require the use of a learning algorithm [9, 10]. When compared with filter approach, wrapper approach possess more computational costs aside from showing an over-fitting risk. However, in embedded techniques, the features selection method is embedded within the model (s) training process [2, 4, 11], followed by the generation of an ideal group of features through the optimization of the objective function. Among the three mentioned types of feature selection, wrapper methods are chosen in this paper.

Metaheuristics optimization algorithms have shown good performance in the search for an optimal solution. Also, these algorithms are easy to implement and can solve a wide range of problems [12]. Among these metaheuristics algorithms are algorithms that are based on swarm intelligence. Swarm intelligence algorithms study the behavior of a collection of agents in self-organized societies, i.e. bees, ants, birds, and moths [13-17]. Techniques based on swarm intelligence have been widely used as a wrapper method for feature selection [18], for instance, bees algorithm [19], ant colony optimization (ACO) [20], butterfly optimization algorithm [21] and moth optimization algorithm [22].

Particle swarm optimization (PSO) was advised in Kennedy and Eberhart [23]. Such an algorithm relies on the behavior of social organisms that live in groups, as exemplified by birds and fish. PSO mimics the interaction between members in information sharing, and the application of PSO can be observed in various optimization domains and also together with other algorithms. To combine their advantages, filter-wrapper grounded upon the PSO feature selection technique was introduced in [24]. The filtering measure is applied in encoding the location of every particle, while the classification accuracy is utilized for the fitness purpose. As can be shown from the experiments, the suggested method was marginally better than binary PSO-based filtering method. On the other hand, the suggested method was yet to be compared with any wrapper algorithm and compared to the filter algorithm, the wrapper algorithm is generally superior in terms of classification performance.

In dealing with the FS problem, Ibrahim *et al.* [25] suggested hybrid optimization technique that comprises a combination of a slap swarm algorithm (SSA) and PSO. This combined method was called SSAPSO. The authors reported that this method improved the effectiveness of the exploitation and exploration phases [25].

PSO and firefly (FF) techniques were hybridized and called PSO-FF in [26] for the FS problem in the examination of childhood's normal "teratoid/rhabdoid" tumor (AT/RT) in brain MRI images and "hemochromatosis" in computed tomography (CT) images of liver. Meanwhile, in [27], the authors demonstrated the application of hybrid bio-inspired technique to the FS process. This proposed method is grounded upon 2 swarm intelligence techniques namely PSO and ACO. For the FS problem also, tabu search (TS) was combined with binary particle swarm optimization (BPSO) in [28]. In this study, BPSO functions as a local optimizer, whenever TS is executed for a particular generation in cancer classification, during gene expression. Somehow, the use of this approach is based on the smallest number of features which means that it may not be representative of the entire dataset. As such, the problem of the solution being stuck in local points may occur. Relevantly in [29], the application of a hybrid method comprising ACO, bee colony optimization (BCO), genetic algorithm (GA) and fuzzy C-means was demonstrated, with the aim of features selection from the mammogram images.

In the current research, the PSO technique is combined with an adaptive local search approach to quickly attain suitable solutions for the problem by combining the advantage of the exploration provided by PSO algorithm and the exploitation ability by the provided by the local search method. PSO algorithm ensures the diversity of the solutions while the adaptive local search method exploits the solutions to obtain

the possible ideal solution. This combination increases the flexibility of the PSO algorithm to enhance the capability to exploit the solutions in searching space while the possible ideal solution can be quickly found. The proposed adaptive local searching method is relies on the late acceptance hill-climbing algorithm [30, 31], and this method is free from parameter tuning, whereby the parameters are tuned through the search of PSO algorithm which makes the algorithm more fixable. PSO algorithm sends the solution to be exploited together with the current iteration and the number of tries used to improve the solutions. Among the most significant features of the proposed algorithm is that it takes advantage of population-based algorithms that preserve the diversity of the solutions and local search algorithms that exploit the solution very fast. The results generated by the suggested algorithm are contrasted against the traditional PSO algorithm and with other contemporary approaches.

The structure of the article is as follows: Section 2 details the standard particle swarm optimization (PSO), followed by section 3 that examines and elaborates the suggested algorithm namely “particle swarm optimization with adaptive local search method”. Then, section 4 reviews the empirical results, and section 5 reviews the study conclusions along with several suggestions to be considered in future studies.

2. PARTICLE SWARM OPTIMISATION ALGORITHM

The PSO algorithm is created by Eberhart and Kennedy in [23], and this algorithm mimics the communication behavior of a group of agents, for instance, birds flocking and fish schooling. In the PSO algorithm, a group of agents denotes the solutions (particles) of the problem and the swarm represents a population of solutions.

PSO algorithm begins by generating random solutions for each particle and assigning them an initial velocity. Particles travel within the searching space in order to search for the ideal solution. Here, the location of every particle is updated based on its knowledge and its adjoining particles. As the particles are moving, their current position i is symbolized by a vector $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, whereby D denotes the search space’s dimensionality. Meanwhile, their velocity i is symbolized by $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. A predefined maximum velocity confines the velocity of the particles whereby v_{max} and $v_{id}^t \in [-v_{max}, v_{max}]$. Further, the best past position of a particle is documented as the personal best and it is symbolized as $pbest$. Accordingly, the best location achieved by the swarm is called “global best” or “ $gbest$ ”. PSO searches and finds the ideal solution by updating the particles using (1) and (2) is used to calculate the moving velocity as follow [23]:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (1)$$

$$v_{id}^{t+1} = w \times v_{id}^t + c_1 \times r_{1i} \times (p_{id} - x_{id}^t) + c_2 \times r_{2i} \times (p_{gd} - x_{id}^t) \quad (2)$$

In which, t symbolizes the t^{th} iteration in the evolutionary process; $d \in D$ symbolizes the d^{th} dimension within the searching space; w signifies the weight of inertia; c_1 and c_2 denote the acceleration constants; r_{1i} and r_{2i} are random values dispersed homogeneously in $[0, 1]$; and p_{id} and p_{gd} symbolize the elements of “ $pbest$ ” and “ $gbest$ ” in the d^{th} dimension. Figure 1 depicts the pseudo-code of PSO.

```

Initialize the population (swarm) with n solutions (particles)
while (termination creation is not met) do
  for i=1: n
    xi ← particle i
    if (f(xi) > pbest) // f calculates the objective
      function(maximization)
        pbest = f(xi)
      end if
    gbes ← best xi among all solutions in the population
    vi ← calculate the velocity based on Equion 2
    update the solutions(particles) based on Equion 1
  end for
end while

```

Figure 1. Particle swarm optimization (PSO) algorithm’s pseudo-code

3. PROPOSED PSO ALGORITHM WITH ADAPTIVE LOCAL SEARCH

3.1. Solution depiction

The solution of features selection is depicted as a vector of N feature (number of features within a data set), and the contents in this vector are either 0 or 1, whereby 0 means unselected feature and 1 means selected feature. PSO algorithm changes the values in the vector to improve classification accuracy; this study uses classification accuracy as an objective function to be maximized. Accordingly, the classification algorithm used in the present study is discussed in the ensuing section. The following figure shows the representation of PSO algorithm for feature selection. For demonstration purposes, suppose that we have a solution for a dataset with 5 features; the selected features are first and third, and hence, the solution will be [1,0,1,0].

3.2. Adaptive local search

The local searching method works relies on “Late acceptance hill-climbing” (LAHC) [16, 30]. LAHC algorithm works based on memory (*list*) with length (L) to save the objective values of the solutions produced during the search. The acceptance of any new coming solutions depends on the assessment of the new solution with the last one saved within the list at the L^{th} step. The worst solution is accepted providing that the value of the possible solution is equal to or better than the value within the list L of index v (virtual starting of the list). v is computed by dividing the current number of the iterations I by the length of L (e.g., see figure line# 12), and after that the value in L of index v becomes the possible solution. Otherwise, the worse solution will not be accepted. In this regard, the “physical” list stays static. However, its “virtual” beginning v is dynamically computed as a division remainder of the number of iterations I by the length of list L ($v=I \bmod L$). The full pseudocode of late acceptance hill climbing is presented in Figure 2 [30].

1	$s \leftarrow$ initial solution;
2	list: LAHC List;
3	NumOfIte= number of iterations;
4	L =the length of the list;
5	Calculates objective value $f(s)$
6	for all $k \in \{0 \dots L-1\}$ do List $k \leftarrow f(s)$
7	$I \leftarrow 0$;
8	Do until a chosen stopping condition
9	Calculates the objective value $f(s^*)$;
10	//Calculate v where v is the virtual beginning
11	$v = I \% L$; where “%” is the remainder of integer division.
12	if $f(s^*) \geq \text{List}_v$ or $f(s^*) \geq f(s)$
13	$s \leftarrow s^*$ //accept the solution
14	end if
15	Inserts objective value to the list List $v \leftarrow f(s)$
16	$I \leftarrow I+1$;
17	end Do

Figure 2. The pseudo-code of late acceptance hill climbing [30]

The pseudocode of the offered algorithm is depicted in Figure 3. In our proposed algorithm, the stopping condition is set by counting the idle steps (idelsteps) or the maximum iteration number is attained, where the idle steps is increased by one if the algorithm couldn't further improve the local best solution (Pbest) see Figure 3 line# 7. The adaptive local search (ALS) is performed if the random number between 0 and 1 is greater than 0.75. This percentage was chosen experimentally to avoid applying the local search for every solution and to avoid long processing time. Another condition is applied to ensure that ALS is applied when the solution is getting stuck in local optima, or when no further improvement is possible, a worse solution is accepted in this stage to skip from getting stuck in local optima see Figure 3 line# 15. Three parameters should be provided to LAHC algorithm; the iterations number (NumOfIte), the list size (L), and the solution (x_i) see Figure 3 line# 17.

The adaptive local search method uses an adaptive method to set these parameters as follows: First, the number of iterations (NumOfIte) is calculated by multiplying the of idle steps number (idelsteps) with the current number of PSO iteration (PsoIter). This is to provide more iterations in local search at the end of a search of PSO to promote more iteration at the final stage. Secondly, in terms of list size, a list from the PSO

algorithm (PSOList, similar to the LAHC list) that keeps the objective value of each solution is provided to local search see Figure 3 lines# 7&19. Hence, the history of the solutions' objective values will be saved in the PSOList. This will benefit from the previous experiments and save initialization time of the list of ALS.

```

1 Initialize the population (swarm) with n solutions (particles)
2 PsoIter ← set Pso iteration to 0, maxIter←maximum iterations
3 idelsteps←0
4 while (idelsteps < maxidesteps OR PsoIter < maxIter) do
5   for i←1: n
6     xi ← particle i
7     PSOListi ← f(xi)
8     if (f(xi) > pbest)//f calculates the objective function(maximization)
9       pbest ← f(xi)
10      idelsteps←0
11     else
12       idelsteps ← idelsteps +1
13     end if
14   //----- Start Adaptive local search (ALS) -----
15   if (random (0,1)≥0.75 and idelsteps >PsoIter)/////
16     NumOfIte= idelsteps * PsoIter;// number of ALS iterations
17     Start ALS(xi, PSOListi, NumOfIte)
18     S ← xi;
19     list ← PSOListi;/////
20     L←the length of the list;
21     Calculates initial cost function f(s)
22     I ← 0;
23     while (I<NumOfIte) do
24       s* ← add/remove one random feature from s
25       Calculates its cost function f(s*);
26       v = I % L;
27       if f(s*) ≥ Listv or f(s*) ≥ f(s*)
28         accept candidate (s ← s*)
29       Inserts cost value into the list Listv←f(s)
30       I ← I+1;
31     end while
32     return s
33   end local search
34   //----- End Adaptive local search (ALS) -----
35   if (f(s) > pbest)//s is the solution returned from ALS
36     pbest ← f(s)
37     xi ← s// update the population with solution from ALS
38     idelsteps←0
39   else
40     idelsteps ← idelsteps +1
41   end if
42   gbest←best xi among all solutions in the population
43   vi←calculate the velocity based on Equion 2
44   update the solutions(particles) based on Equion 1
45   PsoIter ← PsoIter+1
46 end for
47 end while

```

Figure 3. The pseudo-code of the proposed PSO algorithm with adaptive local search

4. EMPIRICAL EVALUATION RESULTS AND DISCUSSION

This part of the article looks into the effectiveness and strength of the suggested PSO with the adaptive local search algorithm (PSO_ALS). Further, this study compared PSO_ALS with other population-based algorithms, and among the algorithms compared in this study include GA, MFO and FFA. Accordingly, the tests carried out in this study involved the use of 8 datasets comprising various characteristics.

The following Table 1 presents the eight datasets utilized in this study. These commonly used standard datasets were obtained from the UCI data source [32], and in fact, they have been used in several well-confirmed studies. Among the primary attributes of these datasets are as follows: number of attributes (features), number of examples (Instances), the number of possible class values. Table 1 shows the details.

For the purpose of this work, the instances in the datasets were splitted into two groups of testing and training. In specific, 80% of the instances were utilized in training, while the other 20% were used in

testing. The use of this division was proposed in Friedman *et al.* [33]. The runs and experiments were performed using a system with the following specifications: Intel CPU i5-5200U 2.2 GHz and a RAM of 8.0 GB. The parameter values of the suggested algorithm are depicted in Table 2. Accordingly, these values have been identified based on the results obtained from 10 warming up experimental runs, and as can be observed, Table 2 shows better settings of the algorithm's parameters, which generate better accuracy.

Table 1. The employed datasets

Dataset name	Features	Instances	Class
German	20	1000	2
WBC	10	699	2
SpectF	44	267	2
Sonar	60	208	2
Ionosphere	34	351	2
Heart	13	270	2
WDBC	31	569	2
Parkinsons	23	197	2

Table 2. Better settings of the algorithm's parameters

Parameter	Value
move rate(w1)	0.5
population size	20
Max idle steps	50
Max number of Iterations	200
Limit search range (vmax)	4

Table 3 displays the number of selected features (NF), the best-attained accuracy (ACC) utilized in the comparison between PSO_ALS algorithm and other states of the art algorithms namely GA, MFO, FFA, and PSO. Average Accuracy results of GA, MFO, FFA, and PSO were compared with those of PSO_ALS. The results in Table 3 demonstrate the superiority of the PSO_ALS algorithm when contrasted with other techniques in terms of attained accuracy with 75%, and comparable (same accuracy) with 12.5%.

Also, GA failed to obtain superior accuracy result in comparison with MFO and PSO algorithms which obtain 1 same Accuracy result for WDBC dataset. Table 3 also shows that the performance of PSO-ALS algorithm supersedes other algorithms when it comes to the number of features just in 2 datasets. However, MFO algorithm attains the best outcomes in 4 datasets.

Table 3. Comparison of the best accuracy of GA, MFO, FFA, PSO and PSO_ALS

Dataset	GA			MFO			FFA			PSO			PSO_ALS		
	NF	ACC	Avg-ACC	NF	ACC	Avg-ACC	NF	ACC	Avg-ACC	NF	ACC	Avg-ACC	NF	ACC	Avg-ACC
German	11	78.00	77.26	12	78.63	77.66	8	77.88	77.03	13	78.38	77.88	15	81.50	79.87
Heart	6	88.03	84.81	5	88.42	88.42	5	88.42	87.33	5	88.42	87.29	6	90.74	89.02
Ionosphere	20	87.86	87.21	15	89.64	88.75	9	88.21	87.75	19	89.64	89.18	13	90.35	88.95
Parkinsons	14	88.42	86.94	10	88.42	87.46	7	88.50	87.16	10	89.00	87.20	6	92.30	89.39
SpectF	33	83.20	79.16	14	85.97	80.94	14	84.55	80.35	24	84.13	80.91	20	87.38	85.92
Sonar	36	84.52	82.03	12	86.32	85.26	17	83.90	82.81	33	86.28	84.64	24	85.00	84.51
WDBC	15	98.75	95.22	8	98.96	96.95	12	98.75	96.59	13	98.96	96.70	10	98.96	97.98
WBC	6	98.17	97.86	7	98.17	98.17	6	98.17	97.86	6	98.17	97.88	5	99.27	99.01

Results of average Accuracy by the same algorithm are also displayed in Table 3. As can be observed, the PSO_ALS algorithm proposed in this study achieved six best average results, particularly in the following: German, heart, WDBC, Parkinsons, SpectF and WBC datasets. Meanwhile, PSO shows the best average in Ionosphere dataset, while MFO shows the best average in the Sonar dataset. The details are shown in Table 3, whereby the accuracies with the highest average are bolded in the table above.

The significance of the obtained results can be determined through the Mann Whitney test as was demonstrated by McKnight and Najab [22]. Table 4 accordingly shows the Mann Whitney statistical test's p-values according to values of suitability. From these statistical tests, the spotted differences and improvements are proven to be considered meaningful. The excellence of PSO_ALS regarding average accuracy over other comparable algorithms is proven in Table 4. As the table is showing, the proposed algorithm is significant statistically for most of the cases excepting some cases.

Table 4. The Mann Whitney test p-values for the results of average accuracy

Datasets	GA	MFO	FFA	PSO
WBC	0.100	0.485	0.200	0.315
SpectF	0.100	0.012	0.002	0.210
German	0.002	0.057	0.028	0.057
WDBC	0.000	0.013	0.022	0.012
Heart	0.200	0.281	0.200	0.015
Sonar	0.551	0.125	0.465	0.001
Parkinsons	0.013	0.342	0.012	0.231
Ionosphere	0.013	0.001	0.114	0.485

*p ≥ 0.05

Table 5 presents the Mann Whitney test's levels of marginal significance (p-values) based on the features number. As can be seen from the table, the observed differences between algorithms of PSO_ALS and GA for all datasets are significant statistically and for most other competitor methods are significant statistically excluding for the PSO algorithm. Figure 4 shows a comparison for the best accuracy between the GA, FFA, MFO, PSO, and PSO_ALS algorithms, and can be realize that PSO_ALS clearly achieves the best results in most datasets.

Table 5. The Mann Whitney test p-values for the results of selected features

Datasets	GA	FFA	MFO	PSO
WBC	0.001	0.012	0.436	0.912
SpectF	0.001	0.853	0.019	0.063
German	0.004	1.000	0.105	0.143
WDBC	0.000	0.105	0.002	0.853
Heart	0.001	0.003	0.000	0.631
Sonar	0.002	0.000	0.315	0.631
Parkinsons	0.000	0.218	0.002	0.089
Ionosphere	0.002	0.009	0.143	0.002

*p ≥ 0.05

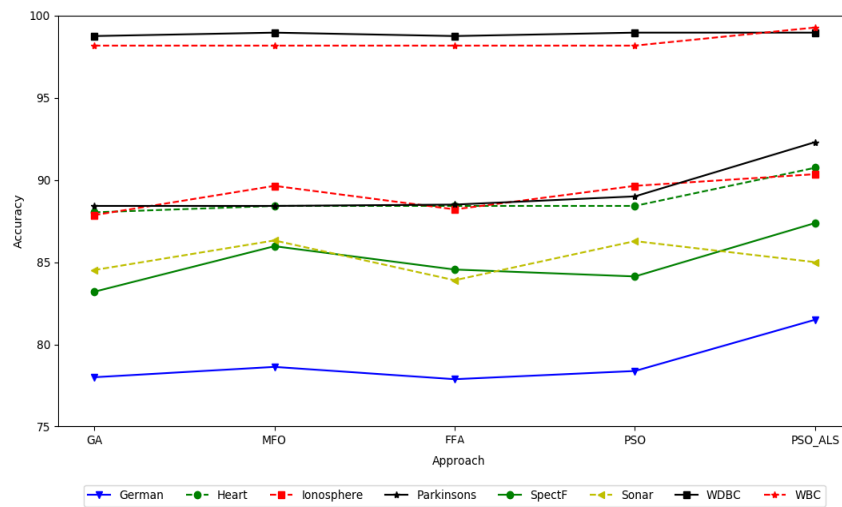


Figure 4. Accuracy comparison between PSO, GA, MFO, FFA, and PSO_ALS

Figure 5(a) proves that the PSO algorithm in some points keeps not improving the solution in several iterations, which means it's got stuck in local optima in some points. This can be exemplified by the first graph shown in Figure 5 which demonstrates the behavior of PSO and PSO_ALS for the Heart dataset. As shown in Figure 5(a), the solution is not improved from iterations number 110 to 180. Contrariwise, in Figure 5(b) graph for PSO_ALS applied to a similar dataset, the algorithm converges smoothly and generates superior results because the behavior of the adaptive local search accepts the worst solution to skip the algorithm out from local optima. Further, in the PSO algorithm at iteration# 120, the algorithm demonstrates its ability in generating an accuracy of 87%. However, in PSO_ALS similar accuracy is produced after Iteration# 170. Thus, it can be said that the application of the adaptive local search.

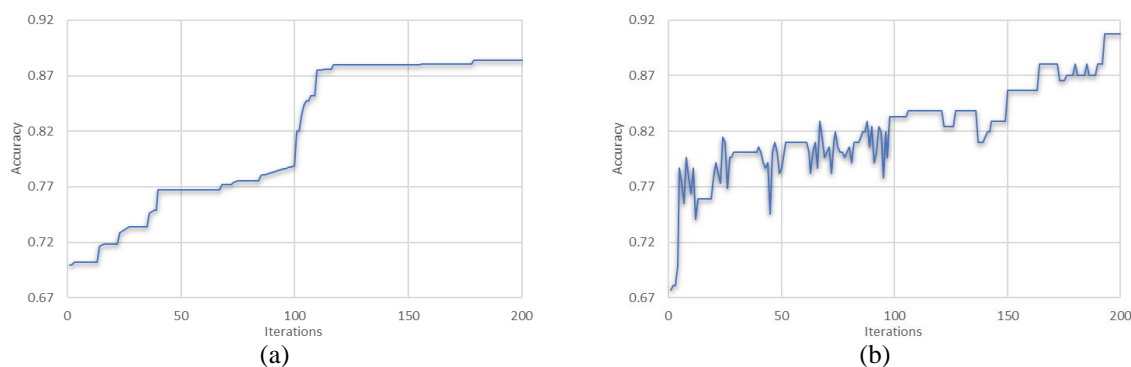


Figure 5. The convergence behavior of the (a) PSO, (b) PSO_ALS algorithms on heart dataset

In accepting candidate solution that has less accuracy in order to jump out from local optima and balance the local intensification and global diversification of the search, that has been seen from the final accuracy produced by both algorithms. For most of the datasets considered in this research, the technique of PSO_ALS yielded equal or improved results of accuracy and selected features number. Somehow, it should be noted that not all the differences observed are significant statistically when compared to other competitors.

Accordingly, Table 6 shows a comparison of the best results obtained by PSO_ALS algorithm against some solutions reported in the literature, involving the use of eight tested datasets. For the purpose of this study, accuracy is utilized as the main goal when comparing the performance of the algorithms. In Table 6, the highest Accuracy is shown in bold. Table 6 shows that PSO_ALS proposed in this study achieved values that are highly comparable to those of most competitors in term of accuracy. However, PSO_ALS shows superior performance in some datasets, when compared to other algorithms.

Table 6. Comparing PSO_ALS with contemporary approaches

Dataset	PSO_ALS	Best-known result	Source
German	81.50	78.63	Alzaqebah <i>et al.</i> [22]
Heart	90.74	88.42	Alzaqebah <i>et al.</i> [22]
Ionosphere	90.35	89.90	Mafarja <i>et al.</i> [34]
Parkinsons	92.30	92.00	Kumar and Kumar [35]
SpectF	87.38	86.38	Alzaqebah <i>et al.</i> [22]
Sonar	85.00	91.20	Mafarja <i>et al.</i> [34]
WDBC	98.96	98.96	Alzaqebah <i>et al.</i> [22]
WBC	99.27	98.35	Alzaqebah <i>et al.</i> [22]

5. CONCLUSION AND FUTURE WORKS

In the current article, the application of the PSO Algorithm with the adaptive local search method (PSO_ALS algorithm) was demonstrated for features selection problem. In the current work, the algorithms are employed to a benchmark of 8 standard UCI datasets. The PSO_ALS algorithm results were contrasted against these generated from the four approached found in the literature. The method proposed in this work demonstrated the performance superiority when compared with other equivalent methods, by balancing local intensification and global diversification of the search through the application of PSO algorithm that finds the best global solution within the search space and adaptive local search method in exploring the local search space. The utilization of the adaptive local search technique improves the results of the suggested algorithm. PSO_ALS shows performance that is superior to other comparable approaches, and also to the basic PSO algorithm.

REFERENCES

- [1] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1136-1159, 2012.
- [2] P. Birkle, et al., "Machine Learning-based Approach for Automated Identification of Produced Water Types from Conventional and Unconventional Reservoirs," *Petroleum Geostatistics*, vol. 2019, no. 1, pp. 1-5, 2019.
- [3] J. Huang, G. Li, Q. Huang, and X. Wu, "Joint feature selection and classification for multilabel learning," *IEEE transactions on cybernetics*, vol. 48, no. 3, pp. 876-889, 2017.

- [4] P. Zhu, Q. Xu, Q. Hu, C. Zhang, and H. Zhao, "Multi-label feature selection with missing labels," *Pattern Recognition*, vol. 74, pp. 488–502, 2018.
- [5] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Systems with Applications*, vol. 57, pp. 117–126, 2016.
- [6] M. Al-Batah, S. Mrayyen, and M. Alzaqebah, "Arabic Sentiment Classification using MLP Network Hybrid with Naive Bayes Algorithm," *Journal of Computer Science. Science Publications*, vol. 14, no. 8, pp. 1104–1114, 2018.
- [7] V. Bolón-Canedo, N. Sánchez-Marono, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Information Sciences*, vol. 282, pp. 111–135, 2014.
- [8] W. Gao, L. Hu, P. Zhang, and F. Wang, "Feature selection by integrating two groups of feature evaluation criteria," *Expert Systems with Applications*, vol. 110, pp. 11–19, 2018.
- [9] D. Rodrigues, et al., "A wrapper approach for feature selection based on bat algorithm and optimum-path forest," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2250–2258, 2014.
- [10] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Applied Soft Computing*, vol. 62, pp. 441–453, 2018.
- [11] L. Jian, J. Li, K. Shu, and H. Liu, "Multi-label informed feature selection," *IJCAI*, pp. 1627–1633, 2016.
- [12] K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, pp. 2191–2233, 2019.
- [13] A. E. Hassanien and E. Emary, "Swarm intelligence: principles, advances, and applications," *CRC Press*, 2018.
- [14] M. Alzaqebah, S. Jawarneh, H. M. Sarim, and S. Abdullah, "Bees Algorithm for Vehicle Routing Problems with Time Windows," *International Journal of Machine Learning and Computing*, vol. 8, pp. 234–240, 2018.
- [15] T. Zhang, S. Wang, W. Tian, and Y. Zhang, "ACO-VRPTWRV: A new algorithm for the vehicle routing problems with time windows and re-used vehicles based on ant colony optimization," in *Sixth International Conference on Intelligent Systems Design and Applications*, 2006, pp. 390–395.
- [16] M. Alzaqebah and S. Abdullah, "An adaptive artificial bee colony and late-acceptance hill-climbing algorithm for examination timetabling," *Journal of Scheduling*, vol. 17, no. 3, pp. 249–262, 2014.
- [17] M. Alzaqebah and S. Abdullah, "Hybrid bee colony optimization for examination timetabling problems," *Computers & Operations Research*, vol. 54, pp. 142–154, 2015.
- [18] L. Brežočnik, I. Fister, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: a review," *Applied Sciences*, vol. 8, p. 1521, 2018.
- [19] O. Alomari and Z. A. Othman, "Bees algorithm for feature selection in network anomaly detection," *Journal of applied sciences research*, vol. 8, pp. 1748–1756, 2012.
- [20] Y. Wan, M. Wang, Z. Ye, and X. Lai, "A feature selection method based on modified binary coded ant colony optimization algorithm," *Applied Soft Computing*, vol. 49, pp. 248–258, 2016.
- [21] M. Alweshah, et al., "The monarch butterfly optimization algorithm for solving feature selection problems," *Neural Computing and Applications*, pp. 1–15, 2020.
- [22] M. A. Alzaqebah, N. Alrefai, E. Ahmed, S. Jawarneh, and M. Alsmadi, "Neighborhood search methods with Moth Optimization algorithm as a wrapper method for feature selection problems," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 4, pp. 3672–3684, 2020.
- [23] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95-International Conference on Neural Networks*, 1995, pp. 1942–1948.
- [24] M. A. Esseghir, G. Goncalves, and Y. Slimani, "Adaptive particle swarm optimizer for feature selection," *International Conference on Intelligent Data Engineering and Automated Learning*, 2010, pp. 226–233.
- [25] R. A. Ibrahim, et al., "Improved salp swarm algorithm based on particle swarm optimization for feature selection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 3155–3169, 2019.
- [26] B. Thamarachelvi and G. Yamuna, "Hybrid firefly swarm intelligence based feature selection for medical data classification and segmentation in SVD-NSCT domain," *International Journal of Advanced Research*, vol. 4, pp. 744–760, 2016.
- [27] K. Menghour and L. Souici-Meslati, "Hybrid aco-pso based approaches for feature selection," *Int J Intell Eng Syst*, vol. 9, no. 3, pp. 65–79, 2016.
- [28] L. Y. Chuang, C. H. Yang, and C. H. Yang, "Tabu search and binary particle swarm optimization for feature selection using microarray data," *Journal of computational biology*, vol. 16, no. 12, pp. 1689–1703, 2009.
- [29] K. Thangavel and C. Velayutham, "Mammogram image analysis: Bio-inspired computational approach," *Proc. of the International Conference on Soft Computing for Problem Solving (SocProS 2011)*, 2012, pp. 941–955.
- [30] E. K. Burke and Y. Bykov, "The late acceptance hill-climbing heuristic," *University of Stirling, Tech. Rep.*, 2012.
- [31] M. Alzaqebah, et al., "Self-Adaptive Bee Colony Optimisation Algorithm for the Flexible Job Shop Scheduling Problem," *International Journal of Operational Research*, 2020.
- [32] A. Frank and A. Asuncion, "UCI machine learning repository," vol. 15, p. 22, 2011, [Online] Available: <http://archive.ics.uci.edu/ml>.
- [33] J. Friedman, T. Hastie, and R. Tibshirani, "The elements of statistical learning," *Springer series in statistics New York*, vol. 1, no. 10, 2001.
- [34] M. Mafarja, et al., "Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems," *Knowledge-Based Systems*, vol. 145, pp. 25–45, 2018.
- [35] R. N. Kumar and M. A. Kumar, "A novel feature selection algorithm with Dempster Shafer fusion information for medical datasets," *International Journal of Applied Engineering Research*, vol. 12, no. 14, pp. 4205–4212, 2017.