International Journal of Electrical and Computer Engineering (IJECE)
Vol. 12, No. 4, August 2022, pp. 4315~4326
ISSN: 2088-8708, DOI: 10.11591/ijece.v12i4.pp4315-4326

Cuckoo algorithm with great deluge local-search for feature selection problems

Mutasem Khalil Alsmadi[1], Malek Alzaqebah[2,3], Sana Jawarneh[4], Sami Brini[2,3], Ibrahim Al-Marashdeh[1], Khaoula Briki[2,3], Nashat Alrefai[2,3], Fahad Ali Alghamdi[1], Maen T. Al-Rashdan[5]
[1]Department of Management Information Systems, College of Applied Studies and Community Service, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia
[2]Department of Mathematics, College of Science, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia
[3]Basic and Applied Scientific Research Center, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia
[4]Computer Science Department, The Applied College, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia
[5]Faculty of Ccience and Information Technology, Jadara University, Irbid, Jordan

Article Info

Article history:

Received Oct 16, 2020
Revised Mar 25, 2022
Accepted Apr 18, 2022

Keywords:

Classification
Cuckoo search
Feature selection
Great deluge
Metaheuristic optimization

ABSTRACT

Feature selection problem is concerned with searching in a dataset for a set of features aiming to reduce the training time and enhance the accuracy of a classification method. Therefore, feature selection algorithms are proposed to choose important features from large and complex datasets. The cuckoo search (CS) algorithm is a type of natural-inspired optimization algorithms and is widely implemented to find the optimum solution for a specified problem. In this work, the cuckoo search algorithm is hybridized with a local search algorithm to find a satisfactory solution for the problem of feature selection. The great deluge (GD) algorithm is an iterative search procedure, that can accept some worse moves to find better solutions for the problem, also to increase the exploitation ability of CS. The comparison is also provided to examine the performance of the proposed method and the original CS algorithm. As result, using the UCI datasets the proposed algorithm outperforms the original algorithm and produces comparable results compared with some of the results from the literature.

Corresponding Author:

Mutasem Khalil Alsmadi
Department of MIS, College of Applied Studies and Community Service, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, 31441, City of Dammam, Saudi Arabia
Email: mkalsmadi@iau.edu.sa

1. INTRODUCTION

Classification is considered as one of the machine learning tasks, which have been widely used recently to categorize the data into classes [1]–[5]. Classification techniques predict the classes of the data instances based on a given set of data fields (features). Using the original number of features may be time-consuming and may mislead the classification process, so the methods for feature selection chose a minimum set of features that lead to better learning accuracy and less computational cost. The methods for feature selection are separated into three categories [6]: i) filter-based methods that use statistical approaches to assess the correlation between features and the class, ii) wrapper-based methods: assess the selected features' subset using a machine learning algorithm, and iii) embedded-based methods: combines the advantage of wrapper methods and filter-based methods [6]–[8].

Feature selection is an nondeterministic polynomial (NP)-problem because of its high-dimensional space [9]–[12] where the exhaustive search is unfeasible. To perform the feature selection task, an efficient search algorithm is required. Swarm intelligence is a group of population-based algorithms that contain algorithms inspired by the social insects/animals' behaviors, which are called nature-inspired optimization algorithms [6], [13]–[17].

Several nature-inspired optimization algorithms were implemented for solving feature selection problems using the wrapper-based method, since its simple, natural representation and efficient in global search [6], [7], [18]–[22]. Particle swarm optimization mimics the behavior of birds and is utilized to find the best set of features [8], [23]–[25]. Ant colony optimization simulates the ants' behavior in searching for food and it has been used for feature selection problems [26]–[32]. Cat swarm optimization is inspired by cats searching for their prey [33], [34]. Grey wolf optimization algorithm depends on how the wolf pack behaves (the hierarchy and hunting) [35], [36]. The behavior of genuine moths in looking for light sources is mimicked by moth flame optimization algorithms [37]–[40].

In Yang and Deb [41] the authors presented the cuckoo search (CS) algorithm for continuous optimization problems [42], [43], CS is based on the attractive cuckoo bird's breeding method. CS algorithm was effectively proposed to problems from different domains such as mobile robot navigation [44] and reliability–redundancy allocation [45]. CS algorithm has some advantages compared with other nature-inspired optimization algorithms such as it explores some elitism types. Also, in CS, randomness is more useful as a move size, where it is heavy-tailed with any likely large move size. And, because there are fewer parameters to tune than with genetic algorithms and particle swarm optimization, it may be easier to adapt to a broader range of optimization problems [42], [46]. Similarly, CS has global optima achievement and rapid convergence. A binary CS (BCS) proposed for feature selection problems in [46], [47]. But there is a limitation in the CS algorithm, which is it has a slow convergence speed [47]. Modified CS algorithm with rough sets is proposed by [48] for feature selection problem, In the modified version some cuckoo species use the obligate brood parasitic behavior and some birds the Lévy flight behavior. The dimensionality of the datasets required an efficient search algorithm to discover the optimum features' subset for better prediction, thus we propose a modified CS algorithm with great deluge algorithm (GD) as local search, to overcome the slow convergence speed of the CS algorithm along with avoiding the CS algorithm getting trapped in local optima.

This paper is organized: the cuckoo search algorithm is presented in section 2 along with the proposed approaches and great deluge algorithm. The results with some discussions are listed in section 3. Comparison of CS algorithm with other results from the literature are stated in section 4. The conclusion and future research are stated in section 5.

2. METHOD

The methodology presented here contains information about the original CS algorithm. Followed by the modification of CS for feature selection problem and description of great deluge as a local-search algorithm. Finally, the details of CS with great deluge algorithm.

2.1. Cuckoo search algorithm (CS)

CS algorithm was firstly proposed in [41]. The behavior of force brood parasitic for cuckoo inspired the authors in [41] to develop a CS algorithm. This behavior starts from the cuckoo laying the eggs in another small bird's nest (host), normally the cuckoo's eggs hatch before the eggs of the host, then the cuckoo chick discovers an outlandish egg and decided to through other eggs. The representation of the CS algorithm is: the population is represented by the nest, the solutions in the population are represented by eggs and the new solutions that are produced using Levy-flight are the cuckoo's eggs. Then the new solution is compared to the other solutions and the best solutions are replaced by the worst solutions. The three main rules in the CS algorithm are stated [41]: i) each cuckoo puts one egg in the random nest, ii) the higher quality nests are set aside and considered for further improvement, and iii) the number of nests is predetermined, for each nest, the cuckoo chick finds an outlandish egg using the probability between 0 and 1. Then the host chooses to abandon the nest or throw the egg.

The levy-flight presented in (1) is used by the CS algorithm for creating the new solutions.

$$x_{i+1} = x_i + \alpha \oplus \mathrm{Lvy}(\lambda) \tag{1}$$

The produced new solution for (cuckoo i) is xi, α is greater than 0, which represents the move size, λ represents a constant of distribution of levy. Usually, the left side term in (1) denotes the random move where the next location is based on the current location and the left term in (1) is the probability of a transition, ⊕ representing the entry-wise multiplication. The move size here is multiplied by a number chosen at random with a distribution of levy.

The random move using levy-flight is efficient for exploring all search regions, using its move length is longer. Levy-flights allow a random move, and the random move length is given by a levy distribution as shown in (2) [41].

$$Lvy \sim u = t - \lambda, (1 < \lambda \leq 3 \quad (2)$$

The process of the CS algorithm is presented in Figure 1, firstly the algorithm starts with initializing the population with the number of host nests, after that in every iteration a randomly selected cuckoo (solution) for generating a new solution using levy-flights.

```
Set Nests:the number of nests
Initialize the Population with Nests;
Set CSNiter: the number of Iterations;
i=0;
for i=0 to CSNiter do:
   Co= a randomly selected cuckoo
   Co*= levy-flight(Co);//produce a new solution for Co
   Fun_Co*=FitnessFunction(Co*);
   RN=Random nest among Nests
   Fun_N= FitnessFunction(Co);
   if Fun_Co* < Fun_N
      RN = Co*;// Replace the solution
   endif
   remove the worst nests and replace new nests using levy-flight
   i.e. The solutions of fraction pa∈[0,1] are abandon
   output the best nests;
endwhile
```

Figure 1. The CS algorithm Pseudo-code [41]

2.2. Great deluge algorithm

Among algorithms that are based on water behavior [49], the great deluge (GD) algorithm was firstly proposed by Dueck in 1993 [50], GD uses an acceptance criterion for accepting the neighbor solutions. GD simulates the hill climber path in a great deluge while trying to maintain his feet dry. GD accepts the neighbor solutions with worse objective value based on the water level (a boundary value). The level value starts reducing with the decay rate during the search process. Reducing the value of level encourages the working solution to consistently reduce till convergence.

The whole process of the GD algorithm is represented in Figure 2, the algorithm starts with initializing the parameters then the iterative process starts, in every iteration, producing k neighboring solutions from the input (current) solution (Co), line-8 Figure 2. The produced neighboring solution is accepted if it's better than the current solution or less than or equal to the water level (boundary), this condition helps the GD to avoid getting trapped in local optima.

```
1   Function GD(Co): The input solution
2   Initialize K:# of neighborhood solutions
3   Initialize rain-Speed
4   Initialize GDIter: # of iterations:
5   water-Level ← f(Co)
6   decay-Rate ← f(Co) × rainSpeed/GDIter
7   i=0
8   while(i< GDIter)
9       Co* ← produce-Neighbour(Co,K)
10      if f(Co*)  < f(Co) or f(Co) ≤ water-Level
11              Co ← Co*//accept the solution
12      endif
13      water-Level ← water-Level – decay-Rate
14      i++
15  endwhile
16  output Co
```

Figure 2. Pseudocode of the great deluge algorithm [51]

2.3. Cuckoo search with great deluge algorithms

CS algorithm in this section is utilized to select the best subset of features. The solution can be represented for the feature selection problem as a matrix with size N containing 0 and 1, where N is the whole features number in the given dataset, 0 indicates that the feature is not chosen, while 1 indicates that it is chosen. CS algorithm disuse solutions based on a fraction, and produces a new solution, at an early stage of the CS algorithm process, disusing the solution may be time-consuming and solutions didn't improve, and not enough iterations left to start improving a new solution, so an updating strategy before desertion the solutions are required to improve the solutions by accepting the worse neighbor solutions.

The levy-flight is used by the CS algorithm to produce a new solution, we propose updating a strategy that uses the great deluge algorithm with two neighborhood strategies, to avoid the CS algorithm from getting stuck in the local optima and to speed up the convergence. Figure 3 represents the process of cuckoo search algorithm with great deluge algorithm. The neighborhood strategies can be explained [52]. Let's consider the solution is Co=[0, 0, 1, 1, 0, 1, 1, 0, 1, 0], so the neighborhood strategies are: i) move neighborhood: chooses a feature at random and move its position to a new random position and ii) swap neighborhood: chooses two features at random and swap values.

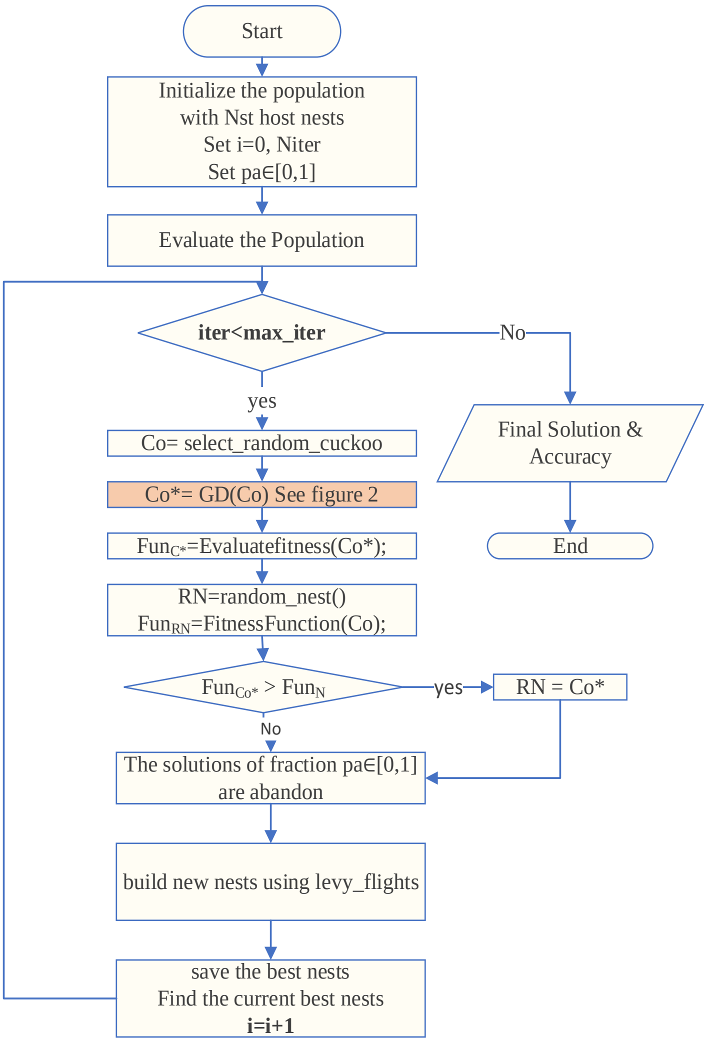


Figure 3. The process of CS algorithm with GD

In feature selection, two objectives should be taken into account to produce a good solution for the problem, where the accuracy should be maximized as much as possible with minimizing the number of selected features. thus, the k-nearest neighbor classifier (KNN) [53], used to produce the mean accuracy using 10-fold-cross-validation [54], and the input features are given by the algorithm as a solution. So, the objective function (OF) in equation 3 is considered both objectives (maximize accuracy while reducing the number of features selected) [53].

$$\mathrm{OF} = \alpha E + \beta \left|\frac{S}{N}\right| \quad (3)$$

Where the value of α is a parameter between 0 and 1 and β=1–α, E is the rate of error given by the KNN classifier. S is the selected features' number and N is the features' total number.

3. RESULTS AND DISCUSSION

The performance of this work is tested in this section using 9 UCI datasets which are used in several well-confirmed research. These datasets are presented in Table 1 [55]. Experimental results based on different values of parameters show the CS algorithm's final parameter settings as presented in Table 2. In the original CS algorithm, parameters *pa*, α, and λ initialized firstly based on [56]. The findings of this study have been implemented using a personal computer with the specifications: Intel i5-2.30 GHz Processor and RAM of 8.0 GB. And, the results are conducted over 10 runs. The datasets are split into 80 training and 20 testings [55].

Table 1. UCI datasets used

	Dataset	Features	Instances
1	German	20	1000
2	Breastcancer	9	699
3	Spect	22	267
4	Krvskp	36	3196
5	Ionosphere	34	351
6	Sonar	60	208
7	Lymphography	148	18
8	Tic-tac-toe	9	958
9	Wdbc	30	569

Table 2. Final parameters settings for CS algorithm

Algorithm	Parameter Name	Value
CS	*Pa*	0.3
	a	1
	λ	1.5
	Size of population	10
	CS Iterations (CSNiter)	100
GD	GD Iterations (*GDIter*)	100
	#neighborhood solutions (K)	2
	rain-Speed	0.5

3.1. Comparison between CS algorithm and CS with local search (CS_GD)

The results of the original CS algorithm presented in this section are evaluated and compared with the proposed approach (CS_GD) to show its effectiveness. Table 3 shows the results, where the best results for each dataset are represented by bold font. These two algorithms are compared based on the testing mean accuracy in 10 folds cross-validation, the average accuracy of 10 runs, the selected features, and the time in seconds taken to finish the process.

Table 3. Results comparison between CS algorithm and the proposed CS_GD

Dataset	CS				CS_GD			
	Accuracy	Average Accuracy	#Features Selected	Average Time	Accuracy	Average Accuracy	#Features Selected	Average Time
German	74.0	70.8	**7**	35.7	**75.5**	72.5	9	48.8
Breastcancer	**92.9**	92.9	3	15.3	**92.9**	92.9	**3**	28.0
Spect	**83.3**	75.7	16	10.2	**83.3**	76.9	**11**	15.1
Krvskp	96.6	94.7	14	161.6	**97.5**	95.1	**10**	232.5
Ionosphere	85.9	82.8	7	10.5	**87.3**	83.4	**6**	16.4
Sonar	88.1	76.9	20	9.2	**90.5**	78.3	**11**	14.2
Lymphography	**80.0**	69.7	5	6.8	**80.0**	72.7	**4**	10.5
Tic-tac-toe	**89.1**	89.1	9	16.5	**89.1**	89.1	**9**	30.7
Wdbc	92.1	91.0	8	13.1	**93.0**	90.1	**6**	23.9

The CS_GD algorithm presented better accuracies for 5 datasets and produces 4 similar accuracies compared with the CS algorithm, and the results show that CS_GD 8 datasets have fewer selected features. Based on the average computation time represented in Table 3 shows that the CS_GD algorithm needs slightly more time to complete the process, nonetheless this extra time is worth it to produce better results. The behavior of the CS_GD algorithm is presented in Figure 4 for Lymphography and German datasets, where the objective function (OF) value is presented (3), in Figure 4, the number of iterations is represented by the x-axis and the y-axis represents the objective function.

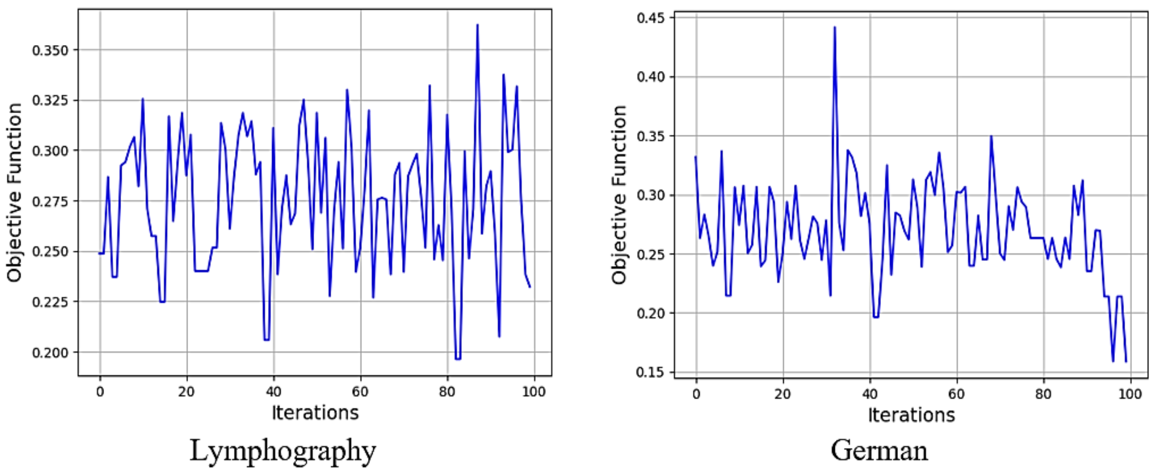


Figure 4. The convergence behavior of the current solution of CS and CS_GD

Using the great deluge algorithm, the solution is accepted in some cases based on the water level, so the worst solution is accepted in some iterations to which helps the algorithm to escape from getting trapped in local the optima and getting better solutions that improve the objective function, also GD speed up the convergence behavior. Boxplots of accuracies produced by the CS and CS GD algorithms are compared and exhibited in Figures 5 and 6 to investigate the reliability and stability of the findings. Each box shows the median which is represented by the middle line in the box, while the top and bottom lines represent the minimum and maximum values, respectively. The box plots for breast cancer and tic-tac-toe datasets show the middle, top, and bottom lines as one line, which means that a similar result is represented for all runs, but other box plots illustrate that the variance of maximum and minimum values in most of the datasets are as acceptable and small, which represents the reliability and stability of the results.

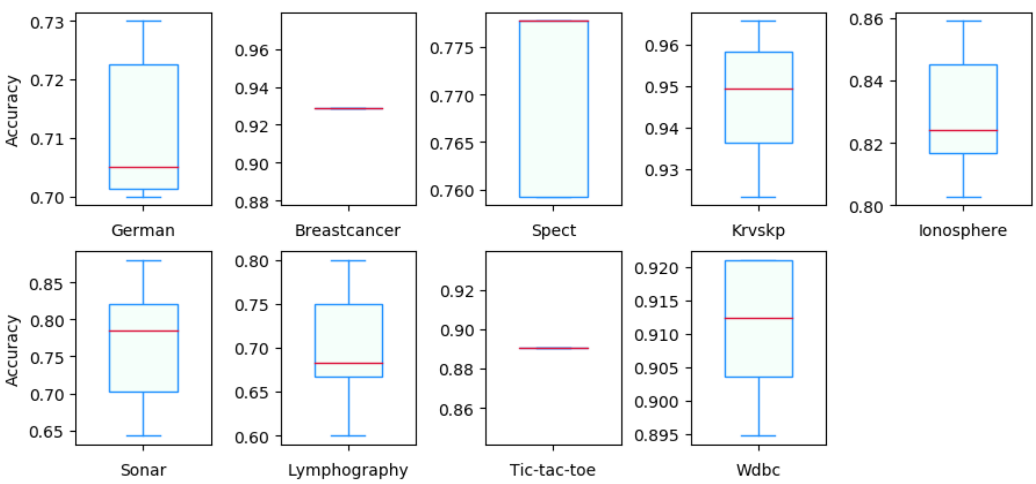


Figure 5. Boxplots of CS algorithm for all datasets

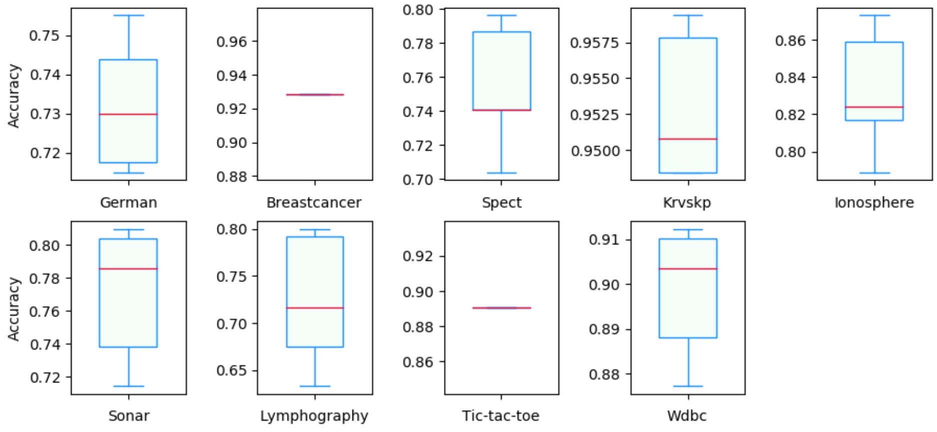


Figure 6. Boxplots of GD_CS algorithm for all datasets

3.2. Comparison between CS_GD and other nature-inspired algorithms

Three nature-inspired algorithms (bat algorithm (BAT), particle swarm optimization (PSO), and firefly optimization algorithm (FFO)) are compared with the superior algorithm from the previous section (CS_GD), The comparison is represented in Table 4, based on the average accuracy produced by 10 independent runs and the selected features' number. As shown in Table 4 the average accuracies for CS_GD outperform the PSO, BAT, and FFO algorithms for 7 datasets out of 9 datasets, 2 of them are similar; for the Lymphography dataset, it has the same average accuracy in CS_GD and BAT algorithms and for Tic-tac-toe have also the same average accuracy between CS_GD and FFO algorithms. Figure 7 represents the visual results using the column chart to view the differences between the algorithms.

Table 4. Results of CS_GD, PSO, BAT and FFO algorithms

Dataset	CS_GD		PSO		BAT		FFO	
	Average Accuracy	#Features Selected	Average Accuracy	#Features Selected	Average Accuracy	#Features Selected	Average Accuracy	#Features Selected
German	72.5	9	68.5	16	70.1	8	71.5	9
Breastcancer	92.9	3	95.6	4	95.1	2	93.1	3
Spect	76.9	11	73.9	24	74.6	7	74.1	18
Krvskp	95.1	10	89.3	21	62.8	8	94.4	15
Ionosphere	83.4	6	80.4	18	82.5	10	81.3	10
Sonar	78.3	11	78.1	40	74.8	16	76.4	27
Lymphography	72.7	4	71.7	9	72.7	7	68.3	4
Tic-tac-toe	89.1	9	81.5	5	66.5	1	89.1	9
Wdbc	90.1	6	92.1	17	90.3	8	91.1	10

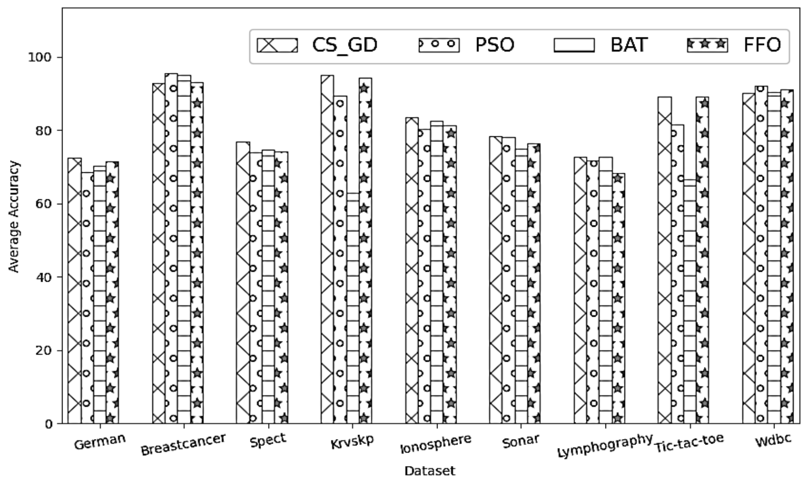


Figure 7. The comparison between CS_GD and other nature-inspired algorithms

The t-test is used to assess the significance of the acquired findings by calculating the difference between the means of two groups. Table 5 provide the p-values that were obtained after applying the t-test for CS_GD and other algorithms' average accuracy results. These statistical tests show that the observed improvements and differences are significant. Where the detected differences between the CS_GD and FFO are significant (p-value≤0.05) for 6 out of 8 datasets, the same number for the PSO algorithm.

Table 5. T-test of CS_GD with other algorithms

Dataset	PSO	BAT	FFO
German	0.0070	0.0493	0.0120
Breastcancer	4.82E-05	0.0024	0.0839
Spect	0.2232	0.2799	0.0070
Krvskp	0.0072	4.18E-09	0.1527
Ionosphere	0.0188	0.2300	0.0204
Sonar	0.4504	0.1452	0.0239
Lymphography	0.3610	0.5000	0.0136
Tic-tac-toe	0.0009	3.81E-08	--
Wdbc	0.0072	0.4181	0.0805

3.3. Comparison of CS GD algorithm with other results from the literature

For the nine datasets studied in this work, the comparison between the best results achieved using the CS_GD method and the best-known solutions from the literature is presented in this section. Table 6 compares the CS_GD algorithm with the most well-known findings of other algorithms from the literature. When assessing the algorithms' performance, accuracy is considered the primary goal. The highest level of precision is shown in Table 6. As seen in Table 6, the CS_GD algorithm outperformed the other literature results in 5 out of 9 datasets in terms of accuracy and has a comparable result with other comparators.

Table 6. Results of CS_GD Algorithm compared with some literature results

Dataset	CS_GD	Literature results	Taken From
German	75.5	81.50	[24]
Breast cancer	92.9	98.00	[57]
Spect	83.3	82.60	[57]
Krvskp	97.5	96.80	[57]
Ionosphere	87.3	79.8	[58]
Sonar	90.5	86.70	[59]
Lymphography	80.0	85.30	[60]
Tic-tac-toe	89.1	80.80	[57]
WDBC	93.0	97.00	[24]

4. CONCLUSION AND FUTURE WORK

In this work, a hybridized CS algorithm with GD algorithm was introduced for feature selection problem. Where, two objectives were considered for the feature selection problem, minimizing the number of selected features and maximizing the prediction accuracy as possible. Thus; to achieve these objectives an effective algorithm is necessary to be used to find an optimum solution of the problem. CS algorithm needs to be modified to improve the convergence speed of the algorithm and to produce good results, thus GD algorithm is proposed to enhance the solutions and to provide faster convergence to CS, due to the ability of the GD algorithm to accept some worse moves to find better solutions. Using nine UCI datasets the efficiency of the algorithm was exposed, the proposed method effectively finds good solutions compared with some other nature-inspired algorithms and other comparable state-of-the-art methods. Our future work is to find an automatic parameter tuning approach to set the parameters used and to investigate the proposed algorithm in other domains.

ACKNOWLEDGMENTS

Many thanks to the Deanship of Scientific Research at the Imam Abdulrahman Bin Faisal University. Where this research is funded by Imam Abdulrahman Bin Faisal University, under grant number 2019-416-ASCS.

REFERENCES

[1] I. H. Sarker, "Machine learning: algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, May 2021, doi: 10.1007/s42979-021-00592-x.

[2] M. K. Alsmadi, M. Tayfour, R. A. Alkhasawneh, U. Badawi, I. Almarashdeh, and F. Haddad, "Robust features extraction for general fish classification," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 6, pp. 5192–5204, Dec. 2019, doi: 10.11591/ijece.v9i6.pp5192-5204.

[3] M. Wang, Z. Wei, M. Jia, L. Chen, and H. Ji, "Deep learning model for multi-classification of infectious diseases from unstructured electronic medical records," *BMC Medical Informatics and Decision Making*, vol. 22, no. 1, Dec. 2022, doi: 10.1186/s12911-022-01776-y.

[4] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021.

[5] H. Wang, R. Czerminski, and A. C. Jamieson, "Neural networks and deep learning," in *The Machine Age of Customer Insight*, Emerald Publishing Limited, 2021, pp. 91–101.

[6] B. H. Nguyen, B. Xue, and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining," *Swarm and Evolutionary Computation*, vol. 54, May 2020, doi: 10.1016/j.swevo.2020.100663.

[7] S. Fong, S. Deb, X.-S. Yang, and J. Li, "Feature selection in life science classification: metaheuristic swarm search," *IT Professional*, vol. 16, no. 4, pp. 24–29, Jul. 2014, doi: 10.1109/MITP.2014.50.

[8] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A new feature selection method to improve the document clustering using particle swarm optimization algorithm," *Journal of Computational Science*, vol. 25, pp. 456–466, Mar. 2018, doi: 10.1016/j.jocs.2017.07.018.

[9] A. A. Albrecht, "Stochastic local search for the feature set problem, with applications to microarray data," *Applied Mathematics and Computation*, vol. 183, no. 2, pp. 1148–1164, Dec. 2006, doi: 10.1016/j.amc.2006.05.128.

[10] G. Zhang, P. Yu, J. Wang, and C. Yan, "Feature selection algorithm for high-dimensional biomedical data using information gain and improved chemical reaction optimization," *Current Bioinformatics*, vol. 15, no. 8, pp. 912–926, Jan. 2021, doi: 10.2174/1574893615666200204154358.

[11] J. Li, S. Fong, R. K. Wong, R. Millham, and K. K. L. Wong, "Elitist binary wolf search algorithm for heuristic feature selection in high-dimensional bioinformatics datasets," *Scientific Reports*, vol. 7, no. 1, Dec. 2017, doi: 10.1038/s41598-017-04037-5.

[12] A. J. Tallon-Ballesteros, L. Correia, and S. Fong, "Hybrid scatter and ant search feature subset selection: applications in classification problems," in *2021 Second International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, Nov. 2021, pp. 150–153, doi: 10.1109/IDSTA53674.2021.9660823.

[13] 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, no. 3, pp. 236–240, Jun. 2018, doi: 10.18178/ijmlc.2018.8.3.693.

[14] M. Alzaqebah, S. Abdullah, and S. Jawarneh, "Modified artificial bee colony for the vehicle routing problems with time windows," *SpringerPlus*, vol. 5, no. 1, Dec. 2016, doi: 10.1186/s40064-016-2940-8.

[15] W. Korani and M. Mouhoub, "Review on nature-inspired algorithms," *Operations Research Forum*, vol. 2, no. 3, Sep. 2021, doi: 10.1007/s43069-021-00068-x.

[16] J. Zhao, D. Tang, Z. Liu, Y. Cai, and S. Dong, "Spherical search optimizer: a simple yet efficient meta-heuristic approach," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9777–9808, Jul. 2020, doi: 10.1007/s00521-019-04510-4.

[17] M. K. Alsmadi *et al.*, "An enhanced particle swarm optimization for ITC2021 sports timetabling," *Computers, Materials and Continua*, vol. 72, no. 1, pp. 1995–2014, 2022, doi: 10.32604/cmc.2022.025077.

[18] M. Alzaqebah, S. Jawarneh, R. M. A. Mohammad, M. K. Alsmadi, and I. ALmarashdeh, "Improved multi-verse optimizer feature selection technique with application to phishing, spam, and denial of service attacks," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 13, no. 1, pp. 76–81, 2021.

[19] M. Sharawi, H. M. Zawbaa, E. Emary, H. M. Zawbaa, and E. Emary, "Feature selection approach based on whale optimization algorithm," in *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*, Feb. 2017, pp. 163–168, doi: 10.1109/ICACI.2017.7974502.

[20] R. A. Khurma, I. Aljarah, A. Sharieh, and S. Mirjalili, "EvoloPy-FS: an open-source nature-inspired optimization framework in python for feature selection," in *Evolutionary machine learning techniques*, 2020, pp. 131–173.

[21] D. Gupta, A. K. Ahlawat, A. Sharma, and J. J. P. C. Rodrigues, "Feature selection and evaluation for software usability model using modified moth-flame optimization," *Computing*, vol. 102, no. 6, pp. 1503–1520, Jun. 2020, doi: 10.1007/s00607-020-00809-6.

[22] R. Guha, B. Chatterjee, S. K. Khalid Hassan, S. Ahmed, T. Bhattacharyya, and R. Sarkar, "Py_FS: A python package for feature selection using meta-heuristic optimization algorithms," in *Computational Intelligence in Pattern Recognition*, 2022, pp. 495–504.

[23] Y. Lu, M. Liang, Z. Ye, and L. Cao, "Improved particle swarm optimization algorithm and its application in text feature selection," *Applied Soft Computing*, vol. 35, pp. 629–636, Oct. 2015, doi: 10.1016/j.asoc.2015.07.005.

[24] M. Alzaqebah *et al.*, "Hybrid feature selection method based on particle swarm optimization and adaptive local search method," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 2414–2422, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2414-2422.

[25] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, and S. Wang, "An improved particle swarm optimization for feature selection," *Journal of Bionic Engineering*, vol. 8, no. 2, pp. 191–200, Jun. 2011, doi: 10.1016/S1672-6529(11)60020-6.

[26] H. M. Rais and T. Mehmood, "Dynamic ant colony system with three level update feature selection for intrusion detection," *International Journal of Network Security*, vol. 20, no. 1, pp. 184–192, 2018.

[27] H. Arafat, R. M. Elawady, S. Barakat, N. M. Elrashidy, and W. S. Email, "Using rough set and ant colony optimization in feature selection," *International Journal of Emerging Trends and Technology in Computer Science*, vol. 2, no. 1, pp. 148–155, 2013.

[28] H. Liang, Z. Wang, and Y. Liu, "A new hybrid ant colony optimization based on brain storm optimization for feature selection," *IEICE Transactions on Information and Systems*, vol. E102.D, no. 7, pp. 1396–1399, Jul. 2019, doi: 10.1587/transinf.2019EDL8001.

[29] M. Paniri, M. B. Dowlatshahi, and H. Nezamabadi-pour, "Ant-TD: Ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection," *Swarm and Evolutionary Computation*, vol. 64, Jul. 2021, doi: 10.1016/j.swevo.2021.100892.

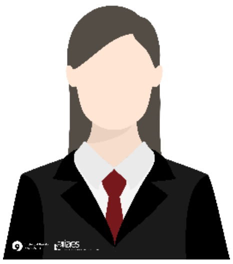
[30] P. Shunmugapriya and S. Kanmani, "A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid)," *Swarm and Evolutionary Computation*, vol. 36, pp. 27–36, Oct. 2017, doi:

10.1016/j.swevo.2017.04.002.
[31] H. Peng, C. Ying, S. Tan, B. Hu, and Z. Sun, "An improved feature selection algorithm based on ant colony optimization," *IEEE Access*, vol. 6, pp. 69203–69209, 2018, doi: 10.1109/ACCESS.2018.2879583.
[32] B. Chen, L. Chen, and Y. Chen, "Efficient ant colony optimization for image feature selection," *Signal Processing*, vol. 93, no. 6, pp. 1566–1576, Jun. 2013, doi: 10.1016/j.sigpro.2012.10.022.
[33] K.-C. Lin, K.-Y. Zhang, Y.-H. Huang, J. C. Hung, and N. Yen, "Feature selection based on an improved cat swarm optimization algorithm for big data classification," *The Journal of Supercomputing*, vol. 72, no. 8, pp. 3210–3221, Aug. 2016, doi: 10.1007/s11227-016-1631-0.
[34] K.-C. Lin, Y.-H. Huang, J. C. Hung, and Y.-T. Lin, "Modified cat swarm optimization algorithm for feature selection of support vector machines," in *Lecture Notes in Electrical Engineering*, vol. 301, Springer Netherlands, 2014, pp. 329–336.
[35] C. Shen and K. Zhang, "Two-stage improved Grey Wolf optimization algorithm for feature selection on high-dimensional classification," *Complex and Intelligent Systems*, Jul. 2021, doi: 10.1007/s40747-021-00452-4.
[36] D. Jitkongchuen and P. Phaidang, "Grey wolf algorithm with borda count for feature selection in classification," in *2018 3rd International Conference on Control and Robotics Engineering (ICCRE)*, Apr. 2018, pp. 238–242, doi: 10.1109/ICCRE.2018.8376472.
[37] R. A. Khurma, I. Aljarah, and A. Sharieh, "Rank based moth flame optimisation for feature selection in the medical application," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2020, pp. 1–8, doi: 10.1109/CEC48606.2020.9185498.
[38] A. Abuhamdah, M. Alzaqebah, S. Jawarneh, A. Althunibat, and M. Banikhalaf, "Moth optimisation algorithm with local search for the permutation flow shop scheduling problem," *International Journal of Computer Applications in Technology*, vol. 65, no. 3, 2021, doi: 10.1504/IJCAT.2021.116008.
[39] M. Alzaqebah, N. Alrefai, E. A. E. Ahmed, S. Jawarneh, and M. K. 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, Aug. 2020, doi: 10.11591/ijece.v10i4.pp3672-3684.
[40] M. Yarlagadda, K. Gangadhara Rao, and A. Srikrishna, "Frequent itemset-based feature selection and rider moth search algorithm for document clustering," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 4, pp. 1098–1109, Apr. 2022, doi: 10.1016/j.jksuci.2019.09.002.
[41] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009-Proceedings*, 2009, pp. 210–214, doi: 10.1109/NABIC.2009.5393690.
[42] M. S. Al-Batah, S. Mrayyen, and M. Alzaqebah, "Arabic sentiment classification using MLP network hybrid with naive Bayes algorithm," *Journal of Computer Science*, vol. 14, no. 8, pp. 1104–1114, Aug. 2018, doi: 10.3844/jcssp.2018.1104.1114.
[43] X. S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, 2010, doi: 10.1504/IJMMNO.2010.035430.
[44] P. K. Mohanty and D. R. Parhi, "Cuckoo search algorithm for the mobile robot navigation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8297, no. 1, Springer International Publishing, 2013, pp. 527–536.
[45] G. Kanagaraj, S. G. Ponnambalam, and N. Jawahar, "A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems," *Computers and Industrial Engineering*, vol. 66, no. 4, pp. 1115–1124, Dec. 2013, doi: 10.1016/j.cie.2013.08.003.
[46] X.-S. Yang, "Bat algorithm and cuckoo search: a tutorial," in *Studies in Computational Intelligence*, vol. 427, Springer Berlin Heidelberg, 2013, pp. 421–434.
[47] D. Rodrigues *et al.*, "BCS: A binary Cuckoo search algorithm for feature selection," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, May 2013, pp. 465–468, doi: 10.1109/ISCAS.2013.6571881.
[48] M. A. El Aziz and A. E. Hassanien, "Modified cuckoo search algorithm with rough sets for feature selection," *Neural Computing and Applications*, vol. 29, no. 4, pp. 925–934, Feb. 2018, doi: 10.1007/s00521-016-2473-7.
[49] B. A. Aldeeb *et al.*, "Hybrid intelligent water drops algorithm for examination timetabling problem," *Journal of King Saud University-Computer and Information Sciences*, Jun. 2021, doi: 10.1016/j.jksuci.2021.06.016.
[50] G. Dueck, "New optimization heuristics," *Journal of Computational Physics*, vol. 104, no. 1, pp. 86–92, Jan. 1993, doi: 10.1006/jcph.1993.1010.
[51] E.-G. Talbi, *Metaheuristics*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2009.
[52] M. Alzaqebah, S. Jawarneh, M. Alwohaibi, M. K. Alsmadi, I. Almarashdeh, and R. Mustafa A. Mohammad, "Hybrid brain storm optimization algorithm and late acceptance hill climbing to solve the flexible job-shop scheduling problem," *Journal of King Saud University-Computer and Information Sciences*, Sep. 2020, doi: 10.1016/j.jksuci.2020.09.004.
[53] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54–65, Nov. 2016, doi: 10.1016/j.neucom.2016.03.101.
[54] M. Alwohaibi, M. Alzaqebah, N. M. Alotaibi, A. M. Alzahrani, and M. Zouch, "A hybrid multi-stage learning technique based on brain storming optimization algorithm for breast cancer recurrence prediction," *Journal of King Saud University-Computer and Information Sciences*, May 2021, doi: 10.1016/j.jksuci.2021.05.004.
[55] "UCI machine learning repository." http://archive.ics.uci.edu/ml/index.php (accessed Dec. 25, 2021).
[56] M. Alzaqebah *et al.*, "Memory based cuckoo search algorithm for feature selection of gene expression dataset," *Informatics in Medicine Unlocked*, vol. 24, 2021, doi: 10.1016/j.imu.2021.100572.
[57] M. Mafarja *et al.*, "Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems," *Knowledge-Based Systems*, vol. 145, pp. 25–45, Apr. 2018, doi: 10.1016/j.knosys.2017.12.037.
[58] E. Hato, "Impact of feature selection for data classification using naive Bayes classifier," *Journal of Physics: Conference Series*, vol. 1879, no. 2, May 2021, doi: 10.1088/1742-6596/1879/2/022088.
[59] F. Han, W.-T. Chen, Q.-H. Ling, and H. Han, "Multi-objective particle swarm optimization with adaptive strategies for feature selection," *Swarm and Evolutionary Computation*, vol. 62, Apr. 2021, doi: 10.1016/j.swevo.2021.100847.
[60] T. M. T. A. Hamid, R. Sallehuddin, Z. M. Yunos, and A. Ali, "Ensemble based filter feature selection with harmonize particle swarm optimization and support vector machine for optimal cancer classification," *Machine Learning with Applications*, vol. 5, Sep. 2021, doi: 10.1016/j.mlwa.2021.100054.

BIOGRAPHIES OF AUTHORS

Mutasem Khalil Alsmadi is currently an associate professor at the Faculty of Applied Studies and Community Service, Department of Management of Information Systems, Imam Abdurrahman Bin Faisal University. He received his B.S. degree in Software engineering in 2006 from Philadelphia University, Jordan, his M.Sc. degree in intelligent systems in 2007 from University Utara Malaysia, Malaysia, and his Ph.D. in Computer Science from The National University of Malaysia. He has published more than one hundred papers in the image processing and algorithm optimization areas. His research interests include artificial intelligence, pattern recognition, algorithms optimization and computer vision. He can be contacted at email: mksalsmadi@gmail.com.

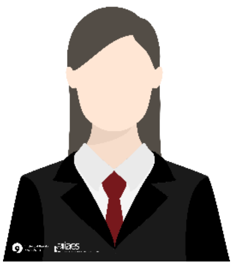
Malek Alzaqebah obtained his B.Sc. in computer science from Al-Balqa Applied University, Jordan and his master degree specializing in information technology from University Utara Malaysia (UUM). He did his Ph.D. in computer science at University Kebangsaan Malaysia (UKM). Now, he is an assistant professor at the Department of Mathematics, Faculty of Science, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. His research interest involves data mining and metaheuristic algorithms for combinatorial optimization problems such as university timetabling, job shop scheduling, vehicle routing and search-based software engineering. He can be contacted at email: maafehaid@iau.edu.sa.



Sana Jawarneh obtained her BSc in computer engineering from Yarmouk University and her PhD in computer science at University Kebangsaan Malaysia. Now, she is an assistant professor at Department of computer science, The applied college, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. Her research interest falls under meta-heuristic algorithms in various optimization problems. Email: sijawarneh@iau.edu.sa.

Sami Brini obtained his B.Sc. in computer science from Faculty of Science of Monastir, Monastir University, Tunisia and his master's degree in computer science from University of Sousse, Tunisia. He prepared his thesis in Tunisia that covers Artificial Intelligence and NLP. He is currently a Faculty Member at the Department of Mathematics, Faculty of Science, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. His research interest involves data analysis and security, computational physics and modeling, machine learning, computing and artificial intelligence. He can be contacted at email: ssbrini@iau.edu.sa.

Ibrahim Abood Al-Marashdeh is currently an associate professor at the Faculty of Applied Studies and Community Service, Department of Management of Information Systems, Imam Abdurrahman Bin Faisal University. He received his B.Sc from University of Jordan in 2001. His MSc degree in information and communication technology in in 2007 from University Utara Malaysia, and his PhD in Software engineering/Industrial computing from The National University of Malaysia award in 2012. He has published number of research papers and his research include: human computer interaction, image processing and algorithm optimization areas and software development and usability, requirements analysis, mobile government and mobile learning. he can be contacted at email: ibramars@gmail.com.



Khaoula Briki obtained his BSc in computer science from the Higher Institute of Applied Sciences and Technology of Sousse (ISSATS), University of Sousse her master's degree specializing in Decision Making from Higher Institute of Computer Science and Management of Kairouan (ISIGK), University of Kairouan. Now, she is a Lecturer at the Department of Mathematics, Faculty of Science, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. She is currently Head of the Learning Outcomes Measurement Unit, College of Science. Prior to this, she was Head of Statistical Analysis and Measurement Unit, College of Science. Her main research interests and experiences are in diverse areas of machine intelligence, data mining and data analysis. She can be contacted at email: kabriki@iau.edu.sa.

Nashat Alrefai received the B.Sc. in programming and system analysis from Jordan University of Science and Technology. He got his Master's degree in computer information systems from the University of Banking and Financial Sciences and he is a Ph.D. candidate in University Technology Malaysia (UTM). He is a lecturer at the Department of Mathematics, Faculty of Science, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia, and work as researcher in basic and applied scientific research center. His research focuses on deep learning, machine learning, artificial intelligence, bioinformatics and data mining. Email: nalrefai@iau.edu.sa.

Fahad Ali Alghamdi is currently an assistant at the Faculty of Applied Studies and Community Service, Department of Management of Information Systems, Imam Abdurrahman Bin Faisal University, Saudi Arabia. He received his B.S. degree in computer science in 2002 from Imam Abdurrahman Bin Faisal University, Saudi Arabia, his M.Sc. degree in computer science in 2009 from University Wollongong, Australia, and his Ph.D. in Computer Science in 2017 from Aberystwyth University in UK. His research interests include network, applied computer software, cloud computing and online learning. He is currently the Vice Dean of Faculty of Applied Studies and Community Service, Imam Abdurrahman Bin Faisal University, Saudi Arabia. He can be contacted at email: faghamdi@iau.edu.sa.

Maen T. Al-Rashdan received the BEng. degree in Electronic engineering from Yarmouk University of Jordan, in 2001, and the M.Sc. degree in Computer and Communication Engineering from the National University of Malaysia (UKM), in 2004 and completed his PhD in Computer and Communication Engineering, electrical, electronic and system engineering department, UKM, Malaysia, in 2011. His research has been focused on Smart Selection Hierarchical Mobile Internet Protocol Version Six with Load Balancing and Mobility Anchor Points Queue Management. He is currently working as Assisstant Professor at Jadara University. He has published several articles in top journals and international conferences. He can be contacted at Email: dr.maen@jadara.edu.jo.