An Enhanced Grasshopper Optimization Algorithm Applied to Feature Selection

Karl Robert F. Angeles[1], Earl B. Cuntapay[2], Raymund M. Dioses[3], Vivien A. Agustin[4], and Dan A. Michael Cortez[5]

[1,2]Student, College of Engineering and Technology, Pamantasan ng Lungsod ng Maynila

[3,4,5]Faculty, Computer Science Department, College of Engineering and Technology, Pamantasan ng Lungsod ng Maynila

Email: [1]krfangeles2018@plm.edu.ph

***Abstract*—** This paper introduces an enhanced version of the Grasshopper Optimization Algorithm with the aim to improve upon problems with slow convergence and the original algorithm not being suited to solving binary optimization problems. The solutions presented are to make use of q-Gaussian mutation in order to jump out of local optimum, thereby leading to faster convergence and the V4 transform function to map continuous values to binary values. The enhanced algorithm is applied to the optimization of feature selection. The proposed algorithm is tested against four other binary optimization algorithms and assessed on three different datasets. The results display that the proposed algorithm exceeds the compared techniques in terms of accuracy and minimizing features selected.

***Keywords*—** Binary Optimization, Feature Selection, Grasshopper Optimization Algorithm, q-Gaussian Mutation, Transfer function.

I. INTRODUCTION

In recent years, datasets with a vast number of characteristics and relatively few patterns have been generated. A large number of irrelevant and duplicated features may drastically decrease the accuracy of learnt models and slow down their learning rate. This issue, referred to as the curse of dimensionality in data mining techniques, increases the computational complexity of model construction (Moradi et al., 2016).

Feature Selection (FS) is a challenging problem in machine learning that seeks to reduce the number of features by deleting unnecessary, redundant, and noisy data while maintaining an acceptable degree of classification accuracy. This is accomplished by deleting features that are not relevant to the problem at hand (Mafarja et al., 2019).

Grasshopper Optimization Algorithm is a contemporary algorithm inspired by nature that imitates the swarming behavior of grasshoppers in the wild (Saremi et al., 2017). The initial version of GOA was created to solve continuous optimization challenges (Saremi et al., 2017). Nevertheless, several optimization problems (e.g., FS) contain discrete choice variables and search space.

In this paper, we propose the Enhanced Binary Grasshopper Optimization Algorithm (EBGOA) designed to tackle binary optimization problems. The V4 transfer function adapted from Mafarja et al. (2018) is used to convert continuous values into binary values. *q*-Gaussian mutation is applied to the positions generated by mutating the values thus allowing to jump out of local optimum. The results are obtained using three select UCI datasets and is compared against the Binary Bat Algorithm (BBA)(Nakamura et al., 2012), the Binary Cuckoo Search (BCS)(Pereira et al., 2014), the Binary Grasshopper Optimization Algorithm (BGOA-M)(Mafarja et al., 2018), and the Binary Particle Swarm Optimization Algorithm (BPSO)(Khanesar et al., 2008) to determine the proposed algorithm's robustness

.II. RELATED STUDIES

Using high-dimensional datasets makes the feature selection optimization problem more difficult. Traditional solutions prove ineffective in resolving this intricate situation. To solve this constraint, a number of swarm-based algorithms have been presented in the literature and are now employed to optimize the selection of features.

To overcome the feature selection problem, Hitchem et al. (2019) proposed a new binary grasshopper optimization algorithm NBGOA. It employs basic operations to update grasshopper positions after initializing them with binary values.

Twenty benchmark datasets from UCI datasets are tackled using the suggested NBGOA and five alternative methodologies from the literature.

Two innovative binary versions of the GOA method were suggested and used to FS problems in (Mafarja et al., 2019). The first strategy is based on transfer functions, while the second approach employs a unique mechanism that repositions the current solution by taking into account the position of the best solution so far. 25 standard UCI benchmark datasets were utilized to test the suggested techniques. The results were compared to 11 FS approaches found in the literature, including five wrapper-based metaheuristics and six filter-based methods.

A new nature-inspired feature selection technique based on bat behavior was proposed by Rodrigues et al. (2013). To determine the set of features that maximizes the accuracy in a validating set, the wrapper approach combines the strength of bat exploration with the speed of the Optimum-Path Forest classifier.

Vieira et al. (2013) proposed a modified binary particle swarm optimization (MBPSO) method for feature selection with simultaneous optimization of SVM kernel parameter setting, which they applied to predicting mortality in septic patients. It was put to the test on six databases, and it outperformed state-of-the-art approaches for PSO and produced similar or better results than GA.

In (Rodrigues et al., 2013) a new feature selection method based on the behavior of cuckoo birds named Binary Cuckoo Search was proposed. The tests were conducted on two datasets with the goal of detecting thefts in power distribution systems. Wrapper feature selection approaches minimize the number of features in the original feature set while simultaneously improving classification accuracy. A wrapper-feature selection approach based on the binary dragonfly algorithm is proposed in (Mafarja et al., 2017). When compared to other methodologies, the experimental results suggest that the BDA methodology performs better.

Jona and Nagaveti (2014), proposed a new hybrid metaheuristic dubbed Ant-Cuckoo Colony Optimization for feature selection in Digital Mammogram. It is a hybrid of Ant Colony Optimization (ACO) and Cuckoo Search (CS).

To distinguish between normal and abnormal mammograms, a Support Vector Machine (SVM) classifier with Radial Basis Kernel Function (RBF) is used in conjunction with the ACO. The experiments are carried out in the miniMIAS database. The novel hybrid algorithm's performance is compared to that of the ACO and PSO algorithms.

A binary version of the hybrid Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO). The wrapper-based method K-Nearest Neighbors (KNN) classifier with Euclidean separation matrices is used to find the best solutions. The results show that BGWOPSO outperforms the Binary Grey wolf optimization (BGWO), the Binary Particle Swarm Optimization (BPSO), the Binary Genetic Algorithm (BGA), and the Whale Optimization Algorithm with Simulated Annealing (WOASAT-2) when using several performance measures such as accuracy, BGWOPSO greatly exceeded Binary Grey wolf optimization (BGWO) in terms of finding the best optimal features and computational time (Al Tashi et al., 2019).

Yang et al. (2015) tested a binary-constrained variant of the Flower Pollination Algorithm (FPA) for feature selection, where the search space is a boolean lattice and each feasible solution, or string of bits, signals whether a feature will be employed to assemble the final set. Particle Swarm Optimization, Harmony Search, and the Firefly Algorithm have all been compared to the suggested method. PSO appears to have the fastest convergence process, while HS has the lowest computational cost.

Moth-flame optimization (MFO) is a swarm intelligent optimization algorithm that mimics the motion of moths recently proposed by (Zawbaa et al., 2016). The algorithm is used in the domain of machine learning for feature selection in the wrapper-based feature selection mode to discover the best feature combination. The evaluation stage in wrapper-based feature selection employs a machine learning technique. Despite being time-consuming, this strategy proved to be effective in terms of categorization accuracy. In this study, MFO is used as a strategy for finding optimal feature sets and maximizing classification performance. Furthermore, the results demonstrate that MFO outperforms GA and PSO, which are commonly used in wrapper-based feature selection.

In (Mafarja and Mirjalili, 2018), two incremental hill-climbing techniques (QuickReduct and CEBARKCC) are hybridized with the binary ant lion optimizer in a model called HBALO. The proposed method generates a pool of solutions (ants) at random, which is then enhanced by embedding the most useful features in the dataset that are selected by the two filter feature selection models.

A novel hybrid feature selection approach based on particle swarm optimization is proposed in (Parham and Mozhgan, 2016). To pick the less correlated and salient

feature subset, the suggested technique HPSO-LS uses a local search strategy incorporated in particle swarm optimization. The goal of the local search technique is to use correlation information to guide the particle swarm optimization search process in order to select distinct features. The results show that the proposed method improves classification accuracy over the filter and wrapper based feature selection methods.

III.GRASSHOPPER OPTIMIZATIONALGORITHM

A. Overview

The Grasshopper Optimization Algorithm, which was initially developed by Saremi et al. (2017), is one of the new nature-inspired and population-based algorithms that replicates the behavior of grasshopper swarms seen in natural environments. Exploration and exploitation of the search space are two phases that are required for optimization to be successful. In the larval stage, the swarm moves very slowly and the grasshoppers take very little steps. These are the primary characteristics of the swarm. Adult swarms, on the other hand, are characterized by movement that is both rapid and long-distance.

$$X_i = S_i + G_i + A_i \quad (1)$$

where Xi stands for the location of the ith grasshopper, Si for the social communication between grasshoppers, Gi for the pressure force exerted on the ith grasshopper, and Ai for the advection of wind.

$$S_i = \sum_{j=1, j\neq 1}^{N} S\left(\left|X_j - X_j\right|\right)\frac{X_j - X_i}{dij} \quad (2)$$

where N is the total number of grasshoppers in the swarm, dij is the distance in grasshoppers that separates the ith and jth grasshoppers, and S is a function that specifies the level of social cohesion and is determined using Eq. (3).

$$S(r) = fe^{\frac{-r}{l}} - e^{-2} \quad (3)$$

where f and l are two constants that indicate respectively the intensity of attraction and the attractive length scale, and r is a real value. Repulsion occurs in the interval [0 2.079]. When a grasshopper unit is 2.079 means that the current grasshopper is far away from another grasshopper, there is neither repulsion nor attraction. This is described as the encouragement distance. (Hichem et al., 2019). The authors (Saremi et al., 2017) make the assumption that the wind direction is always blowing in the direction of a target, and they do not take into account the gravity operator. Then Eq. (1) results to:

$$X_i^d = c\left(\frac{ub_d - ub_d}{2} S\left(\left|X_j^d - X_j^d\right|\right)\frac{X_j - X_i}{d_{ji}}\right) + T_d \quad (4)$$

where ubd denotes the upper bound in the dimension d and lbd denotes the lower bound in the dimension d, respectively. Td is the value of the dth dimension in the goal (the best solution found up to this point), and the coefficient c, which decreases the comfort zone proportionally to the number of iterations and is computed as follows in Eq. (5).

$$c = Cmax - l\frac{C_{max} - C_{min}}{L} \quad (5)$$

where Cmax denotes the highest possible value, Cmin denotes the lowest possible value, l is the iteration that is currently being performed, and L denotes the maximum number of iterations. In the study by Saremi et al. (2017), the authors set Cmax to 1 and Cmin to 0.00000001. Eq. (4) demonstrates that the future position of a grasshopper may be determined by taking into account both its present position as well as the positions of all of the other grasshoppers (first term in Eq. (4)), as well as the location of the target (second term).`

B. Pseudocode of the Grasshopper Optimization Algorithm

Initialize the swarm Xi (i=1,2,...,n)
Initialize the cmax, cmin, and maximum number of iterations
Calculate the fitness of each search agent
T=the best search agent
while (l< Max number of iterations)
  Update c using Eq. (5)
  for each search agent
    Normalize the distances between grasshoppers in [1,4]
    Update the position of the current search agent by the Eq. (4)
    Bring the current search agent back if it goes outside the boundaries
  end for
  Update T if there is a better solution
  l=l+1
end while
Return T

IV. ENHANCED BINARY GRASSHOPPER OPTIMIZATION ALGORITHM (EBGOA)

A. Transfer Function

A transfer function is used to map a continuous search space to a binary one, and the updating process is designed to switch positions of particles between 0 and 1 in binary search spaces (Mirjalili and Lewis, 2013).
The V4 transfer function used is adapted from Mirjalili and Lewis (2013), they found that V4 was capable of finding the best solution with a good convergence rate. Performing better than all other tested algorithms in 13 out of 25 benchmark functions. It was found that the proposed v-shaped family of transfer functions, especially the V4 function, has merit for use in binary algorithms. (Mirjalili and Lewis, 2013).

$$T(x) = \left|\frac{2}{\pi} arctan\left(\frac{\pi}{2}x\right)\right| \quad (6)$$

B. q-Gaussian Mutation

We employ the use of the q-Gaussian mutation in order to combat the algorithm being stuck in local optima. The q-Gaussian distribution is employed to generate new candidate solutions by mutation. A real parameter q, which defines the shape of the distribution, is encoded in the chromosome of individuals and is allowed to evolve (Tinos and Renato, 2008).

The shape of the q-Gaussian mutation distribution is controlled by a real parameter q. The control of the parameter q allows to smoothly and continuously change the shape of the distribution, as q is a real parameter and a small change in its value causes a small change in the shape of the mutation distribution (Tinos and Renato, 2008).

Self-adaptation causes a rise in the parameter q, which results in a greater number of long leaps (similar to the Cauchy mutation), which might assist the population escape from local optima and/or converge quicker to the optimal state. In later phases following environmental changes, the parameter q reaches small values, which enhances local search (like the Gaussian mutation) (Tinos and Renato, 2008).

The generalized Box-Müller algorithm provides a methodology for generating q-Gaussian random variates. The parameter $-\infty < q \leq 3$ is related to the shape of the tail decay (Thistleton et al., 2007). The specific values used for the parameter q being -0.5, 1 and 2 were inspired from Tinos and Renato (2008). Furthermore, Eq. (7) was adapted from Jie et al. (2018).

$$X_i^d = X_i \oplus QG(q) \quad (7)$$

C. Pseudocode of the Enhanced Binary Grasshopper Optimization Algorithm

Initialize the swarm Xi (i=1,2,...,n)
Initialize the cmax, cmin, and maximum number of iterations
Calculate the fitness of each search agent
T=the best search agent
while (l< Max number of iterations)
  Update c using Eq. (5)
  for each search agent
    Normalize the distances between grasshoppers in [1,4]
    Update the position of the current search agent by the Eq. (4)
    Apply *q*-Gaussian mutation by the equation (2.9)
    Bring the current search agent back if it goes outside the boundaries
  end for
  Apply V4 transfer function by the Eq. (6)
  Update T if there is a better solution
  l=l+1
end while
Return T

V. METHODOLOGY

A. Data Description

The performance of the proposed EBGOA is evaluated and compared alongside various algorithms in their application to feature selection problems. To verify the results, we chose three datasets obtained from the University of California at Irvine (UCI) machine learning repository. These datasets have been used in many studies of the machine learning area, however, the datasets used can only be categorized as small dimensional datasets due to the researchers' limitations regarding computing resources. Table 1 describes the properties of the datasets used.

Table I. *List of used Datasets*

No.	Dataset	# Feature	# Instances
1	Lymphography	18	148
2	SPECT	22	267
3	Wine	13	178

B. Benchmark Algorithm and Parameters Setting

The proposed EBGOA is measured and compared against four recent algorithms in the feature selection space. The Binary Bat Algorithm (BBA)(Nakamura et al., 2012), the Binary Cuckoo Search (BCS)(Pereira et al., 2014), the Binary Grasshopper Optimization Algorithm (BGOA-M)(Mafarja et al., 2018), and the Binary Particle Swarm Optimization Algorithm (BPSO)(Khanesar et al., 2008).

The results were obtained under similar parameters for all algorithms with a population size of 50 and with epochs set to 15. With the exception being the proposed EBGOA. This was done to measure the results under different values for *q* ranging from -0.5, 1 and 2.

5-Nearest-Neighbors algorithm was used for classification and results for all algorithms were recorded for 10 independent runs. All parameters used are listed under Table 2. The results were computed on a Lenovo ThinkPad T495 with an AMD Ryzen 3500U, 1.4GHz, 16 GB RAM running on Ubuntu 20.04.

Table II. *Initial Parameters of Tested Algorithms*

Parameter	Value
Epochs	15
Pop-size	50
Runs	5

C. Evaluation riteriaa

The evaluation criteria used, average classification accuracy (A_AAC) and average features selection size were sourced from (Hichel, et al., 2019). The average classification accuracy (A_ACC) is the average of solutions obtained from M runs of an optimization algorithm and can be formulated as in Eq. 8 where M is the number of runs of the optimization algorithm to select a feature subset. ACCi is the accuracy of the best solution obtained from the ith run (Hichel, et al., 2019).

$$A_{ACC} = \frac{1}{M}\sum_{i=1}^{M} ACC_1 \tag{8}$$

The average features selection size (AFSS) is the average number of selected features set obtained by the best solutions to the total number of features from M runs. This criterion can be calculated as in Eq. 9. The results of this criterion are reported in Table 5. where M is the number of runs of the optimization algorithm, Size(i) returns the number of features selected in the best solution from the ith run, and D is the size of the original dataset (Hichel, et al., 2019).

$$AFSS = \frac{1}{M}\sum_{i=1}^{M} \frac{Size(i)}{D} \tag{9}$$

VI. RESULTS

Table III. *Average Classification Accuracy and Standard Deviation Results*

Dataset	BBA		BCS		BGOA-M		BPSO		EBGOA	
	A_ACC	StD	A_ACC	StD	A_ACC	StD	A-ACC	StD	A_ACC	StD
Lymphography	0.895	0.102	0.891	0.007	0.910	0.006	0.889	0.010	0.914	0.012
SPECT	0.837	0.006	0.835	0.178	0.848	0.009	0.833	0.010	0.848	0.009
Wine	0.958	0.012	0.961	0.006	0.962	0.005	0.940	0.018	0.964	0.005

Table IV. *Average Selected Feature Ratio to the Total Number of Features*

Dataset	BBA	BSC	BGOA-M	BPSO	EBGOA
Lymphography	0.729	0.863	0.721	0.919	0.688
SPECT	0.636	0.690	0.645	0.672	0.609
Wine	0.631	0.754	0.692	0.892	0.739

Table V. *A_AAC, SD, and AFSS for EBGOA with Different q-values*

EBGOA				
Dataset	q	A_CC	StD	AFSS

SPECT	2	0.848	0.009	0.609
SPECT	1	0.856	0.007	0.618
SPECT	-0.5	0.861	0.003	0.559

The numerical results for the proposed EBGOA, as well as the other benchmark algorithms are presented here. Table 3 summarizes the findings for average classification accuracy, defined in Eq. 8. It is observed that the proposed EBGOA outperforms all the other benchmark algorithms, only matching with BGOA-M for the SPECT dataset. Table 4 summarizes the findings for average feature selection size, defined in Eq. 9. EBGOA was able to have less features selected compared to other algorithms, with the exception being on the Wine dataset. BBA was able to have the lowest average for selected features for the Wine dataset. Table 5 outlines the result for average classification accuracy and average feature selection size when modifying q in EBGOA for the SPECT dataset. It indicates that having q set to -0.5 achieves the best result across different values for q as well as other algorithms for the SPECT dataset in regards to average classification accuracy and average feature selection size. Also, with q being set to 1 also results in an accuracy higher than other algorithms as well as achieving a higher accuracy than with q being set to 2. With all other datasets, the value of q was set to 2, this implies that it's possible for the proposed EBGOA to receive an even higher accuracy across other datasets. In terms of average feature selection size, there were no notable differences with different values of q.

VII. CONCLUSION

In this paper, we proposed an Enhanced Binary Grasshopper Optimization Algorithm (EBGOA) applied to feature selection problems. The proposed EBGOA alongside four other algorithms are tested on three datasets sourced from the UCI machine learning repository. The results of all algorithms were compared in terms of average classification accuracy and average feature selection size. The proposed algorithm outperformed or equalled the other compared techniques in terms of average classification accuracy. Furthermore, modifying the parameter *q* to -0.5 in the proposed EBGOA resulted in a higher accuracy and lower selected features compared to other algorithms. However, in terms of average feature selection size, the proposed EBGOA scored slightly lower or matched the best result. For future work, a method to modify the parameter *q* dynamically during runtime may be explored. Furthermore, the algorithm's efficacy on varying sizes of datasets may also be investigated. The proposed algorithm's use on other binary optimization problems may also be explored.

ACKNOWLEDGMENT

The researchers would wish to express their deepest gratitude to Prof. Raymund Dioses for his invaluable and constructive guidance all throughout. The researchers would also wish to extend their thanks to the Pamantasan ng Lungsod ng Maynila - CET - CS department for their unwavering motivation and support. The researchers would also wish to express their appreciation for the continuous support of the university, Pamantasan ng Lungsod ng Maynila. Lastly, the researchers would like to thank their family and friends for their emotional and moral support that pushed the researchers to complete this paper.

REFERENCES

[1] Al-Tashi, Q., Abdul Kadir, S. J., Rais, H. M., Mirjalili, S., & Alhussian, H. (2019). Binary optimization using hybrid grey wolf optimization for feature selection. *IEEE Access*, *7*, 39496–39508. https://doi.org/10.1109/access.2019.2906757

[2] Hichem, H., Elkamel, M., Rafik, M., Mesaaoud, M. T., & Ouahiba, C. (2022). A new binary grasshopper optimization algorithm for feature selection problem. *Journal of King Saud University - Computer and Information Sciences*, *34*(2), 316–328. https://doi.org/10.1016/j.jksuci.2019.11.007

[3] Jona, J. B., & Nagaveni, N. (2014). Ant-cuckoo colony optimization for feature selection in Digital Mammogram. *Pakistan Journal of Biological Sciences*, *17*(2), 266–271. https://doi.org/10.3923/pjbs.2014.266.271

[4] Luo, J., Chen, H., zhang, Q., Xu, Y., Huang, H., & Zhao, X. (2018). An improved grasshopper optimization algorithm with application to financial stress prediction. *Applied Mathematical Modelling*, *64*, 654–668. https://doi.org/10.1016/j.apm.2018.07.044

[5] Mafarja, M., Aljarah, I., Faris, H., Hammouri, A. I., Al-Zoubi, A. M., & Mirjalili, S. (2019). Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Systems with Applications*, *117*, 267–286. https://doi.org/10.1016/j.eswa.2018.09.015

[6] Mafarja, M. M., Eleyan, D., Jaber, I., Hammouri, A., & Mirjalili, S. (2017). Binary dragonfly algorithm for feature selection. *2017 International*

Conference on New Trends in Computing Sciences (ICTCS). https://doi.org/10.1109/ictcs.2017.43

[7] Mahdavi, S., Rahnamayan, S., & Deb, K. (2018). Opposition based learning: A literature review. *Swarm and Evolutionary Computation*, *39*, 1–23. https://doi.org/10.1016/j.swevo.2017.09.010

[8] Meraihi, Y., Gabis, A. B., Mirjalili, S., & Ramdane-Cherif, A. (2021). Grasshopper optimization algorithm: Theory, variants, and applications. *IEEE Access*, *9*, 50001–50024. https://doi.org/10.1109/access.2021.3067597

[9] Moradi, P., & Gholampour, M. (2016). A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing*, *43*, 117–130. https://doi.org/10.1016/j.asoc.2016.01.044

[10] Nakamura, R. Y., Pereira, L. A., Rodrigues, D., Costa, K. A., Papa, J. P., & Yang, X.-S. (2013). Binary bat algorithm for feature selection. *Swarm Intelligence and Bio-Inspired Computation*, 225–237. https://doi.org/10.1016/b978-0-12-405163-8.00009-0

[11] Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. A. (2007). Quasi-oppositional differential evolution. 2007 IEEE Congress on Evolutionary Computation. https://doi.org/10.1109/cec.2007.4424748

[12] Rodrigues, D., Pereira, L. A., Almeida, T. N., Papa, J. P., Souza, A. N., Ramos, C. C., & Xin-She Yang. (2013). BCS: A binary cuckoo search algorithm for feature selection. *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*. https://doi.org/10.1109/iscas.2013.6571881

[13] Rodrigues, D., Yang, X.-S., de Souza, A. N., & Papa, J. P. (2014). Binary flower pollination algorithm and its application to feature selection. *Studies in Computational Intelligence*, 85–100. https://doi.org/10.1007/978-3-319-13826-8_5

[14] Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software*, *105*, 30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004

[15] Thistleton, W. J., Marsh, J. A., Nelson, K., & Tsallis, C. (2007). Generalized box–müller method for generating q-Gaussian random deviates. *IEEE Transactions on Information Theory*, *53*(12), 4805–4810. https://doi.org/10.1109/tit.2007.909173

[16] Tinos, R., & Shengxiang Yang. (2008). Evolutionary programming with Q-gaussian mutation for dynamic optimization problems. *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. https://doi.org/10.1109/cec.2008.4631036

[17] Vieira, S. M., Mendonça, L. F., Farinha, G. J., & Sousa, J. M. C. (2013). Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. *Applied Soft Computing*, *13*(8), 3494–3504. https://doi.org/10.1016/j.asoc.2013.03.021