A Dynamic Locality Multi-Objective Salp Swarm Algorithm for Feature Selection

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: http://ees.elsevier.com

A dynamic locality multi-objective salp swarm algorithm for feature selection

Ibrahim Aljarah [a], Maria Habib [a], Hossam Faris [a], Nailah Al-Madi [f], Ali Asghar Heidari [b,g], Majdi Mafarja [c], Mohamed Abd Elaziz [d], Seyedali Mirjalili [e]

[a] King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan
[b] School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran
[c] Department of Computer Science, Birzeit University, Birzeit, Palestine
[d] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig, Egypt
[e] Centre for Artificial Intelligence Research and Optimisation Torrens University Australia, Fortitude Valley, Brisbane, 4006 QLD, Australia
[f] Department of Computer Science, Princess Sumaya University for Technology, Amman, Jordan
[g] Department of Computer Science, School of Computing, National University of Singapore, Singapore

ARTICLE INFO

Keywords
Wrapper feature selection
Salp swarm algorithm
Optimization
Multi-objective
Classification

ABSTRACT

Developing intelligent analytical tools requires pre-processing data and finding relevant features that best reinforce the performance of the predictive algorithms. Feature selection plays a significant role in maximizing the accuracy of machine learning algorithms since the presence of redundant and irrelevant attributes deteriorates the performance of the learning process and increases its complexity. Feature selection is a combinatorial optimization problem that can be formulated as a multi-objective optimization problem with the purpose of maximizing the classification performance and minimizing the number of irrelevant features. It is considered an NP hard optimization problem since having a number of (n) features produces a large search space of size (2^n) of different permutations of features. An eminent type of optimizer for tackling such an exhausting search process is evolutionary, which mimic evolutionary processes in nature to solve problems in computers. Salp Swarm Algorithm (SSA) is a well-established metaheuristic that was inspired by the foraging behavior of salps in deep oceans and has proved to be beneficial in estimating global optima for optimization problems. The objective of this article is to promote and boost the performance of the multi-objective SSA for feature selection. Therefore, it proposes an enhanced multi-objective SSA algorithm (MODSSA-lbest) that adopts two essential components: the dynamic time-varying strategy and local fittest solutions. These components assist the SSA algorithm in balancing exploration and exploitation. Thus, it converges faster while avoiding locally optimal solutions. The proposed approach (MODSSA-lbest) is tested on 13 benchmark datasets and compared with the well-regarded Multi-Objective Evolutionary Algorithms (MOEAs). The results show that the MODSSA-lbest achieves significantly promising results versus its counterpart algorithms.

1. Introduction

Owing to the rapid progress of industrial engineering and machine learning, the dimension of the offered datasets is extremely increased (Shen et al., 2016). Having large datasets exposes the process of extracting useful information to several challenges Zhao, Chen, Hu, Min, and Jiang (2018). Some features of the datasets are detected as irrelevant and redundant, which leads to performance degradation of the learning algorithms (Ibrahim, Elaziz, Ewees, Selim, & Lu, 2018), yet it is a challenging concern to comprehend whether the features can be detected as relevant or not (Wang et al., 2017; Zhao et al., 2014). One of the approaches for addressing the problem of high-dimensionality is performing feature selection (FS), which is known as a pre-analysis stage. FS help us to select only the relevant features that best maximize the performance of the learning methods and decreases the time required for learning them (Dash & Liu, 1997; Mafarja, Aljarah, Heidari, Hammouri, et al., 2018). However, articulating FS is a challenging task since the features might be dependant and have strong interactions (Gheyas & Smith, 2010). It is crucial to search for the valuble features that can represent the hidden physiognomies of the classes (Chen, Zhang, Luo, Xu, & Zhang, 2020; Wang & Chen, 2020). Large search domains make the process of FS a more perplexing task since the size of the search space increases exponentially with the increasing number of features (Guyon & Elisseeff, 2003). For example, if we have n dimensions, then, there can be 2^n possible solutions. Hence, using an exhaustive search algorithms for FS case is not a

*corresponding author
E-mail addresses: i.aljarah@ju.edu.jo (I. Aljarah); hossam.faris@ju.edu.jo (H. Faris); n.madi@psut.edu.jo (N. Al-Madi); as_heidari@ut.ac.ir, aliasghar68@gmail.com, aliasgha@comp.nus.edu.sg, t0917038@u.nus.edu (A.A. Heidari); mmafarja@birzeit.edu (M. Mafarja); meahmed@zu.edu.eg (M.A. Elaziz); ali.mirjalili@gmail.com (S. Mirjalili)

https://doi.org/10.1016/j.cie.2020.106628
Received 20 March 2020; Received in revised form 12 June 2020; Accepted 21 June 2020
Available online xxx
0360-8352/© 2020.

true choice (Chandrashekar & Sahin, 2014); since they are computationally expensive. Generally, FS methods require two components: First, it is the fitness equation, which is utilized to judge the excellence of the selected features. Second, it is the heuristic way utilized to deal with it and explore the feasible area of the domain to realize the optimal set of features (Zhao et al., 2019). Broadly speaking, two directions of FS methods are available: the filter-based and the wrapper-based. The filter-based methodologies can select the features subsets freely without being constrained to the predictors. Filtering-based FS methods are like Information Gain (IG) (Quinlan, 1986), ReliefF (Yu & Liu, 2003), and the Gain Ratio (Witten, Frank, Hall, & Pal, 2016). Unlike the filter-based approaches, wrapper-based methodologies need to utilize some predictors to estimate the superiority of the detected features (Ma et al., 2017; Zhu, Ong, & Dash, 2007). These approaches include the neural network-based methods (Setiono & Liu, 1997), the Sequential Backward Selection (SBS) (Gheyas & Smith, 2010), and the Sequential Forward Selection (SFS) (Bermejo, Gámez, & Puerta, 2009).

In particular, several search procedures can be applied to the application domains to detect the optimal subset of features: one method is the greedy search and the other choice is the random search (Langley, 1994). In the greedy search techniques, all appointments of the features are produced and assessed, which makes this scheme timewasting. Meanwhile, methods that are based on the random search techniques explore the domain, randomly. However, there are also some drawbacks and limits. For instance, there is a chance for stagnation issues and the case of high time complexity.

To alleviate the drawbacks of the previously mentioned FS methods, researchers proposed to utilize the metaheuristic paradigms. Metaheuristic methods are global optimization approaches that follow the physical, biological, and animal social activities in wildlife (Mirjalili & Lewis, 2016). They have the ability to explore the search space globally and locally when applied to FS problems. Some examples of metaheuristics are Particle Swarm Optimization (PSO) (Banks, Vincent, & Anyakoha, 2008; Eberhart & Kennedy, 1995), Genetic Algorithms (GAs) (Han & Kim, 2002), Differential Evolution (DE) (Han & Kim, 2002), AntLion Optimization (ALO) (Mafarja & Mirjalili, 2018; Mirjalili, 2015), Grey Wolf Optimizer (GWO) (Ibrahim, Elaziz, & Lu, 2018; Mirjalili, Mirjalili, & Lewis, 2014), Artificial Bee Colony (ABC) (Karaboga & Akay, 2009), Harris Hawks Optimization (HHO) (Heidari et al., 2019), and Slime Mould Algorithm (SMA) (Li, Chen, Wang, Heidari, & Mirjalili, 2020)

Metaheuristics can be divided into two categories: single-objective and multi-objective. They can be configured and formulated to solve problems in any of the two schemes depending on the nature of the problems and the computational resources. In the former category, there is one optimization function, where the search algorithm seeks to find an optimal solution. In other words, considering a classification problem where the objective $F(x)$ is to maximize the classifier performance by maximizing its classification accuracy $g(x)$, therefore $F(x) = g(x)$. In single-objective optimization, the solutions are single real numbers, while comparing the best among all is achieved by a simple comparison operation. On the other hand, the latter treats the problem as having at least two optimization objectives. For example, assuming a classification problem that has two objectives, which are the maximization of the accuracy $g(x)$, and the maximization of the features reduction rate $d(x)$. Thus, it is a multi-objective optimization problem, where the objective $F(x)$ is denoted by $F(x) = [g(x), d(x)]$. The solutions of a multi-objective optimization algorithm are represented as vectors, therefore, deciding the best among all cannot be achieved by simple comparison, but rather using the dominance relationship, as will be discussed further on.

Generally speaking, FS problems can be formulated to have two main objectives: minimizing the number of final features, and the error function. Therefore, it can be represented as a Multi-objective Optimization Problem (MOP). The approaches of MOPs produce a set of solutions called non-dominated (ND) solutions, which are used to balance all the objectives (Coello, Lamont, & Van Veldhuizen, 2007; Mirjalili, Jangir, & Saremi, 2017), and reach a satisfactory solution. In terms of the MOP, the solution is called the ND solution, if there is no solution better than it in terms of the objective function. Additionally, the set of all ND solutions is called the Pareto Front. Based on the properties of the MOP methods, they have been applied to several applications; such as Cloud Computing (Yao, Ding, Jin, & Hao, 2017), Big Data (Ferranti, Marcelloni, Segatori, Antonelli, & Ducange, 2017), Data Mining (Bandaru, Ng, & Deb, 2017), and other (Antonio & Coello, 2017). However, there are few MOP methods that have been applied for feature selection.

Further, the performance of the MOP approaches is affected by the behavior of the metaheuristic techniques. So, there are several MOP solutions that have been proposed based on metaheuristic techniques, such as PSO, GWO, and others. In the same context, the multi-objective salp swarm algorithm (MOSSA) that proposed in (Mirjalili, Gandomi, et al., 2017), aims to solve a benchmark of MOPs. The MOSSA is an extension of the SSA algorithm that mimics the swarming behavior of salps. Based on these behaviors, the SSA has been applied to different applications such as the optimization problems (Rizk-Allah, Hassanien, Elhoseny, & Gunasekaran, 2019), and the prediction of the activities of the chemical compounds (Hussien, Hassanien, & Houssein, 2017). As well as, the SSA was applied for the feature selection as in (Aljarah, Mafarja, et al., 2018; Faris et al., 2018; Ibrahim, Ewees, Oliva, Elaziz, & Lu, 2018; Sayed, Khoriba, & Haggag, 2018). However, these SSA-based methods treating the FS problem as a single-objective optimization problem by aggregating the two objectives and using parameters to balance among them. So, it is hard to determine the Pareto optimal set since it requires to run the algorithm many times, while changing the balance between the two objectives.

Nonetheless, most of the multi-objective feature selection techniques stagnate at a locally optimal solution, which, in consequence, leads to a poorly convergence toward optimal solutions and degrades the quality of the final solutions. Also, according to the No-Free-Lunch theorem which assumes that the optimization techniques do not have the ability to solve all problems by the same quality (Wolpert & Macready, 1997). Therefore, this motivated us to propose an alternative wrapper multi-objective feature selection method based on a modified version of the MOSSA. Where the novel proposed approach uses an evolutionary dynamic component alongside a local best component.

The aim of the dynamic component is to make the salps' movement more vital depending on an adaptive number of leaders that is variable with time.

The main contributions of the proposed method are:

- Improving the search process of the MOSSA using the Dynamic and local best operators.
- Modifying the MOSSA to handle the feature selection problem using transfer functions.
- Evaluating the performance of the modified MOSSA on 13 UCI benchmarks datasets to investigate its efficiency for the FS problem.
- To the best of our knowledge, this is the first time the proposed MOSSA method is used as a feature selection technique to improve the classification performance.

The rest of the paper is organized as follows. Section 2 provides a review of related works. Section 3 represents a background of the theories of multi-objective optimization, the SSA algorithm, including single-objective SSA and multi-objective SSA. Section 4 shows a description of the problem and the proposed methodology. Section 5 describes the experimental setup. Section 6 discusses the obtained results. Finally, Section 7 presents a concluding summary and potential future works.

2. Review of related works

This section presents a review of research studies in the area of evolutionary feature selection. It covers the single-objective and the multi-objective approaches in various applications.

One of the early implementations of evolutionary feature selection is the adoption of the GA algorithm. The proposed approach is used for searching for the optimal feature subset for different optimization problems. For instance, Yang and Honavar (1998) demonstrated the feasibility of the GA algorithm and Neural Networks (NN) in seeking the optimal subset of features. The NN method was used to assess the selected subset of features by the GA algorithm. However, the authors depended merely on the fitness measure to evaluate the proposed classification approach, while other metrics (e.g., precision, sensitivity, required execution time, and other measures) are important as well. Nonetheless, the authors did not provide any statistical tests to quantify the significant improvements of the proposed approach. Additionally, authors in Kim, Y, Street, and Menczer (2000) proposed an unsupervised evolutionary feature selection approach. In which, they adopted an evolutionary local selection algorithm (ELSA) with the k-means algorithm. ELSA is used for searching for the best subset of features, while the k-means method assesses the quality of potential features subsets. However, the proposed approach was not applied to real or benchmark datasets. As well as, the results were weakly presented. In Tan, Teoh, Yu, and Goh (2009), a hybrid approach of the GA algorithm and Support Vector Machines (SVMs) was proposed for attribute selection.Even though the authors were focusing on improving the classification performance by selecting the relevant features, the performance of the learning algorithms is highly influenced by the settings of the hyperparameters of SVMs, which is not considered in this study. The evaluation results of the performance were good in terms of the average classification accuracy when applied to 5 UCI datasets Lichman (2013). However, testing the proposed approach on a wider-set of benchmark datasets raises its reliability and robustness. Several studies have used evolutionary algorithms for features (genes) selection in the case of microarray data analysis. For instance, Chuang, Yang, Wu, and Yang (2011) utilized the GA algorithm with either the SVMs or the k-Nearest Neighbor (KNN) for gene selection, respectively. In which, 11 microarray datasets were utilized. The performance was assessed by using the classification accuracy based on hold-one-out cross-validation. Even the authors achieved relatively good results, but they relied mainly on the accuracy measure for the evaluation, while they did not consider the data imbalanced problem with the multi-class classification. In addition, they did not perform any statistical test to substantiate the strengths of the designed approach.

Schiezaro and Pedrini (2013) proposed an evolutionary feature selection approach for classification based on the Artificial Bee Colony (ABC) algorithm. The designed algorithm achieved a considerable increase in the classification accuracy with a significant decrease in the selected number of features when compared with other evolutionary algorithms. However, it will be more convincing if the authors included more metrics other than the accuracy for the performance evaluation. Further, Ghosh, Datta, and Ghosh (2013) designed a self-adaptive differential evolution algorithm for a supervised feature selection procedure. In which, a fuzzy KNN has utilized for the evaluation of a set of candidate features subsets. The implemented method is tested on three datasets in the context of hyperspectral images and compared with the other four evolutionary algorithms. It exhibited promising results regarding the classification accuracy and kappa coefficient. Although the datasets were imbalanced, the authors should include other measures that are insensitive to the imbalanced problem. Also, Chen, Chen, and Chen (2013) designed an evolutionary feature selection based on Ant Colony Optimization (ACO) algorithm, where it applied for image feature selection. Fifteen datasets were utilized for testing the proposed approach using a weighted-sum evaluation criterion that depends on the recall, precision, and the number of selected features. The experiments performed and compared over 5-folds and 10-folds cross-validation settings. The proposed algorithm showed very good improvement in comparison with the GA algorithm and other algorithms. However, since the researchers followed a single-objective approach to address the feature selection problem, the comparison with multi-objective approaches could have been explored. Furthermore, the authors compared the designed approach with two variants of the GA; however, they did not consider other well-established evolutionary algorithms.

Whereas, Xue, Zhang, Browne, and Yao (2015) provided a comprehensive survey of evolutionary computation for FS. The survey covered the search techniques, evaluation criteria, applications, and the number of objectives. Upon that, the GA and PSO algorithms were the most studied algorithms by the researchers. In addition, the authors reported existing challenges such as scalability, computational cost, and search algorithms.

Faris, Aljarah, and Mirjalili (2016) proposed a new mechanism to handle the feature selection problem based on the Multi-Verse Optimizer (MVO) algorithm. The proposed approach outperformed PSO, GA, DE, Firefly Algorithm (FA), Cuckoo Search (CS), Back-Propagation (BP), and the Levenberg-Marquardt algorithms. When it tested on nine UCI medical datasets, it achieved outperforming results in terms of convergence speed and gained the highest statistical ranking. Moreover, Mafarja, Aljarah, Heidari, Hammouri, et al. (2018) used another search algorithm for building a wrapper-based feature selection method based on the Grasshopper Optimization Algorithm (GOA), which achieved significantly better results. In this work, the authors followed a weighted-sum approach to solve a multi-objective problem as a single-objective, for which the weights parameters of the fitness function were assigned constant real values. Even these weighting parameters significantly affect the algorithm's performance. Also, an evolutionary feature selection approach based on the GOA algorithm and SVM designed by (Aljarah et al., 2018). It obtained promising results regarding the classification accuracy when tested on eighteen benchmark datasets. Additionally, Faris et al. (2018) proposed an approach for FS using the SSA. The proposed algorithm experimented with 8 Transfer Function (TF) and a crossover operator, where it achieved significantly better results than other 5 evolutionary algorithms (i.e., binary GWO, binary PSO, binary Gravitational Algorithm, binary Bat Algorithm, and GA algorithm). Worthwhile, diverse evolutionary algorithms were utilized for feature selection as the implementation of the Dragonfly Algorithm (DA) in Mafarja, Aljarah, Heidari, Faris, et al. (2018), fruit fly optimization (FFO) in (Zhang et al., 2020), and the use of the Crow Search Algorithm (CSA) in De Souza, dos Santos Coelho, De Macedo, and Pierezan (2018), where they demonstrated remarkable ability in reducing the number of redundant features and improving the classification performance. Another study Sayed, Hassanien, and Azar (2019) used a chaotic crow search algorithm for feature selection. The results of the evaluation on 20 benchmark datasets were very effective, and significant when compared with other 9 evolutionary algorithms. Furthermore, Ibrahim, Ewees, Oliva, Elaziz, and Lu (2019) adopted a hybrid of the SSA and PSO algorithms for feature selection. The new approach is tested versus the classical SSA on 15 benchmark functions, and against different four evolutionary algorithms on UCI datasets. The hybrid of SSA and PSO outperformed other algorithms in terms of accuracy and f-measure.

All the articles mentioned above were handling the problem of FS as a single-objective or a weighted-sum single-objective. However, feature selection problems have multiple criteria to optimize by nature which will hypothetically improve the quality of potential solutions. The classical weighted-sum (composite) approach faces challenges in converging and finding optimal solutions due to the need to optimize its weights to find an optimal solution on average. Nonetheless, it has a higher complexity of function evaluations as the problem scales highly while requiring multiple runs to optimize a single-objective function.

In the literature, very few have formulated the problem of feature selection as a multi-objective optimization problem using the logic of multi-criteria optimization. However, an early study has proposed a multi-objective feature selection technique based on the multi-objective GA and NN algorithm by Emmanouilidis, Hunter, and MacIntyre (2000). In this study, the number of features and the sum-squared error were considered the objectives. However, it was an initial general implementation of an evolutionary multi-objective framework for feature selection. Pappa, Freitas, and Kaestner (2002) formulated the GA algorithm with a sort of Decision Tree (DT) that is known by (C4.5) for multi-objective feature selection. In which, the error rate and the size of the obtained trees were the objectives. In addition, Coello and Lechuga (2002) created an evolutionary multi-objective particle swarm optimization (MOPSO), which is then used for feature selection in various research studies Xue, Zhang, and Browne (2012), Xue, Cervante, et al. (2014), Zhang, Gong, and Cheng (2015), Behravan et al. (2016), Zhang, Gong, Sun, and Guo (2017), Amoozegar and Minaei-Bidgoli (2018). Nonetheless, Wang and Huang (2009) presented the implementation of Non-dominated Sorting GA (NSGA-II) Deb, Agrawal, Pratap, and Meyarivan (2000) for feature selection, where the inter-correlation and intra-correlation were the objective functions. Remarkably, the proposed method outperformed many other utilized evolutionary single-objective approaches for feature selection. Tan, Lim, and Cheah (2014) proposed a multi-objective approach for feature selection. In which, the proposed approach depends on a modified micro GA algorithm and neural networks for optimizing two objectives: the number of features and the classification accuracy. Although this method is evaluated just on two datasets, the results were very good. Comparing the proposed approach with other multi-objective evolutionary algorithms can further validate the effectiveness of this method. Moreover, few studies afterward have been implemented a multi-objective GA for feature selection, such as in Khan and Baig (2015),Zhu, Liang, Chen, and Ming (2017), Spolaôr, Lorena, and Diana Lee (2017). In the literature, various recent multi-objective evolutionary algorithms were proposed and utilized to address feature subsets optimization. For example, Xue, Fu, and Zhang (2014, 2015) implemented a multi-objective differential evolutionary algorithm (MOEA/D) for feature selection. Wang, Li, and Li (2015) presented a multi-objective evolutionary approach for feature selection, which is the Strength Pareto Evolutionary Algorithm (SPEA-II) that aims to maximize the relevance of features and minimize the redundancy of them. Whereas, Dickson, Wang, Dong, and Wen (2015) proposed a multi-objective gravitational search algorithm for feature selection, which showed a remarkable reduction in error rate. In Emary, Yamany, Hassanien, and Snasel (2015), the authors presented a multi-objective GWO for feature selection that has powerful strength in producing optimal solutions. Also, Hancer, Xue, Zhang, Karaboga, and Akay (2018) constructed a multi-objective ABC algorithm for feature selection that is combined with a non-dominated sorting approach. Yet, Kiziloz, Deniz, Dokeroglu, and Cosar (2018) designed a new multi-objective evolutionary algorithm based on the teaching-learning optimization algorithm for feature selection, which showed significant enhancement when compared with NSGA-II, MOPSO, greedy search, and scatter search algorithms. On the other hand, Soliman, Abou-El-Enien, Emary, and Khorshid (2018) created a multi-objective moth-flame optimization algorithm for feature selection. Even though it is achieved good results, the authors did not compare its performance with other state-of-the-art multi-objective evolutionary algorithms. The performance of multi-objective evolutionary algorithms has been debated and reviewed by many studies as presented in Antonio and Coello (2017),Tian, Cheng, Zhang, Li, and Jin (2019), Yu, Lu, and Yu (2018).

To sum up, the performance of the multi-objective SSA has not been investigated in the literature for performing feature selection. Even in Alresheedi, Lu, Elaziz, and Ewees (2019), the authors introduced an improved multi-objective SSA algorithm for optimizing the placement of virtual machines on the cloud. In which, the objectives were to maximize the time before the host shutdown and minimize the service level agreement violations. Therefore, the objective of this article is to examinegenerally the performance of the multi-objective SSA in optimizing the process of feature selection.

3. Preliminaries

This section describes briefly the main concepts utilized in the proposed approach, which are the multi-objective optimization, the conventional SSA optimizer, and the multi-objective variant of SSA.

3.1. Multi-objective optimization

The real-world applications demand experts to take optimal decisions based on different objectives that have different orientations which conflict with each other. To illustrate, in the literature there are different real problems such as the big data optimization (Wang et al., 2018), remote sensing (Ma, Zhong, He, & Zhang, 2018), image segmentation (Sarkar, Das, & Chaudhuri, 2017), and feature selection (Xue, Zhang, & Browne, 2013). The multi-objective optimization methods have been used to resolve these kinds of problems, which have a high ability to determine the solutions that balance the conflict between the objectives (Coello, Lamont, & Van Veldhuizen, 2007; Mirjalili, Jangir, & Saremi, 2017).

In general, the mathematical definition of the multi-objective optimizations can be formulated as follows: considering that there is a set of M objectives, the minimization of the objectives can be represented as (Mirjalili, Jangir, et al., 2017):

$$\begin{aligned} &\textit{Minimize}\ \ F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})]^T \\ &\textit{subject to}:\ \ g_i(\mathbf{x}) \geqslant 0, i = 1, 2, \dots, C_g \\ &h_j(\mathbf{x}) = 0, j \\ &\quad = 1, 2, \dots, C_h, LB_i \leqslant \mathbf{x}_i \leqslant UB_i, \quad i \\ &\quad = 1, 2, \dots, n \end{aligned} \tag{1}$$

where $f_M(\mathbf{x})$ represents the M-th objective function. $g_i(x)$, and $h_j(x)$ are the inequality and equality constraints, respectively. The number of variables (dimension of the given problem), the number of equality constraints, and the number of inequality constraints are represented by n, C_h and C_g, respectively. LB_i and UB_i represent the lower and the upper boundaries of the variable x_i, respectively.

In MOP, existing M objectives makes the comparison process between the solutions to determine the best of them is difficult when using the traditional relational operators. Unlike MOP, the single-objective optimization can determine the best solution using these relational operators, for example, if the solution $\mathbf{x}$ is smaller than the solution $\mathbf{y}$ then $\mathbf{x}$ is the best solution. Therefore, to find the best solutions that trade-off the multi-criterion and the conflict between the objective functions, the dominance concepts are used (Auer, Chiang, Ortner, & Drugan, 2016; Reddy & Kumar, 2015).

The solution $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is dominant (i.e., better than) $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$ if and only if the following criteria is met.

$$\forall i:\ f_i(\mathbf{x}) \leqslant f_i(\mathbf{y}) \wedge \exists j: f_j(\mathbf{x}) < f_j(\mathbf{y}) \tag{2}$$

where $i, j \in \{1, 2, \dots, M\}$, based on Eq. (2), the solution dominates the other solutions if it is better at least in one objective. Meanwhile, the solutions that are not dominated by any other solutions are called Pareto-optimal (non-dominated) solutions, which represent the solution for a multi-objective problem. The set of all the non-dominated solutions is called Pareto optimal set (*PS*), and it is defined as:

$$PS = \{\mathbf{x} | \mathbf{x} \ is \ Pareto \ Optimal\} \tag{3}$$

The projection of the solutions of *PS* in the objective space is referred to as Pareto optimal front (*PF*) and it is defined as:

$$PF = \{f(\mathbf{x}) | \mathbf{x} \in PS \ and \ f(\mathbf{x}) \in R^M\} \tag{4}$$

From the previous definitions in Eqs. (2)–(4), the comparison between the solutions of the multi-objective problems becomes more suitable and easier. Based on the concepts of the multi-objective problem, feature selection can be formulated to involve two objectives. Since the FS aims to minimize the classification error rate and the number of features, simultaneously.

3.2. Salp Swarm Algorithm (SSA)

The foraging styles of salps has inspired Mirjalili, Gandomi, et al. (2017) to propose a swarm-based method according to supposed chains of search agents and a considered food source based on fitness values (optimum solution). In this technique, we can divide the agents' set into two classes, one sole salp is termed the leader, and the other agents will be the follower salps. The location of the leader is located at the head of the chain, and the other followers will make the body of the chain (Abd Elaziz, Heidari, Fujita, & Moayedi, 2020). A simple chain is shown in Fig. 1.

3.2.1. Single-Objective SSA

In the single-objective version, the set of agents X involves N salps with d-dimensions, which is prepared as in Eq. (5):

$$X_i = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^N & x_2^N & \dots & x_d^N \end{bmatrix} \tag{5}$$

To obtain the food source, the position vector of the chief agent (leader) is updated using Eq. (6):

$$x_j^1 = \begin{cases} F_j + c_1 \left(\left(ub_j - lb_j\right) c_2 + lb_j\right) & c_3 \geqslant 0.5 \\ F_j - c_1 \left(\left(ub_j - lb_j\right) c_2 + lb_j\right) & c_3 < 0.5 \end{cases} \tag{6}$$

where x_j^1 is the leader's location and F_j is the food source in the j^{th} dimension, ub_j and lb_j show the limits of j^{th} dimension, c_2 and c_3 denote random numbers in [0, 1], and c_1 works as an adaptive parameter, as presented in Eq. (7):

$$c_1 = 2e^{-\left(\frac{4t}{L}\right)^2} \tag{7}$$

where t shows the iteration counter and L is the maximum iterations. The c_1 parameter is so important to have balanced stability in terms of explorative and exploitative trends. Follower salps are updated using Eq. (8), where $i \geqslant 2$ and x_j^i denotes the position of i^{th} salp at the j^{th} dimension. The pseudo-code of the single-objective SSA is represented in Algorithm 1.

$$x_j^i = \frac{x_j^i + x_j^{i-1}}{2} \tag{8}$$

Algorithm 1 Pseudo-code of the single-objective SSA

```
Input: Number of agents, number of iterations
Output: The best salp and its fitness
Generate the initial salps $x_i (i = 1, 2, \dots, n)$
while (Termination condition is not met) do
 Evaluate the fitness values
 Find the fittest salp and save it as the food source $F$
 Update $c_1$ using Eq. (7)
 for(any salp ($x_i$)) do
 if ($i == 1$) then
  Update the location vector of best salp by Eq. (6)
 else
  Update the locations of follower salps by Eq. (8)
 Update all salps using the legal limits
 Return back those salps that go over the true limits of feature space.
Return $F$
```

As it is presented in Algorithm 1, SSA initializes the agents and then, it obtains the fitness (objective) results of current salps. Based on

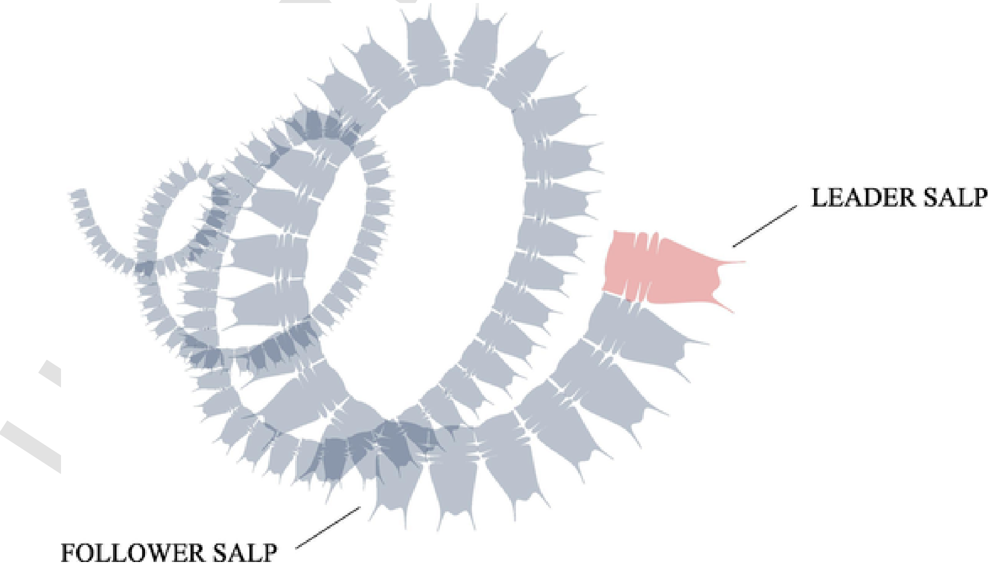


Fig. 1. A salp chain.

the loop condition, all agents evolve their positions based on the best salp. To analysis all regions inside the searching area, the variable c_1 is updated in each level based on rule in Eq. (7). Using equation in Eq. (6), SSA generates the best agent, while using the logic in Eq. (8), algorithm updates the follower salps. SSA will repeat its evolutionary steps until the stopping condition reached and finding the best results (Abd Elaziz, Heidari, Fujita, & Moayedi, 2020).

3.2.2. Multi-Objective SSA (MOSSA)

A solution for a multi-objective task is different from the solution to a single-objective case, which we call the solutions of a multi-objective case as the *Pareto optimal set*. The multi-objective variant of SSA was originally proposed by Mirjalili, Gandomi, et al. (2017) in the original work of SSA.

When solving multi-objective cases, two main questions should be responded. First, in MOSSA, we need to store several salps as the leaders in dealing with a multi-objective scenario. Second, the SSA needs to update the food source in each iteration with regard to the best agent (leader), but in a multi-objective scenario, we cannot separate a single best salp. In MOSSA, to address the first problem, a repository of food sources is added to the SSA. A series of non-dominated salps are stored in this repository. During the optimization, any agent is compared with all the residents that are available in the repository based on the mechanisms of *Pareto dominance*. When a salp can dominate only a single agent in the repository, it needs to be swapped. In another case, if a salp has dominated a set of agents, we need to remove all of them and add that salp to the repository. Now, if at least one of the residents dominates an agent in the newly generated swarm, we need to discard it, quickly. When the salp is non-dominated compared to entire repository residents, we will include it in the archive. When we face a full repository, one of the similar non-dominated salps is eliminated from the repository.

For addressing the second problem, an accepted approach is to pick a solution (source food) from a set of non-dominated salps having the least crowded neighborhood. This mechanism can be performed based on the same ranking method and roulette wheel selection (RWS) utilized in the repository maintenance operator Wang, Gao, and Chen (2018). During the operation of the archive maintenance, the salps with better ranks (crowded neighborhood) have more chances to be selected. While the less populated neighborhood (the inferior rank) for a non-dominated salp located in the repository, the more chance to be selected as the source of food. The pseudo-code of MOSSA is represented in Algorithm 2.

Algorithm 2 Pseudo-code of multi-objective SSA

```
Input: Number of agents, maximum capacity of the archive (repository), number of iterations
Output: Repository
Generate the initial set of randomly-generated salps $x_i (i = 1, 2, \dots, n)$
while (Termination condition is not met) do
  Evaluate the fitness values of all search agents
  Obtain the non-dominated solutions
  Update the repository based on the attained non-dominated search agents
  if the repository is full then
  Perform the repository maintenance process to eliminate one repository resident
  Insert the non-dominated agent to the repository
  end if
  Select a food source F from repository
  Update $c_1$ using Eq. (7)
  for (any salp ($x_i$)) do  if ($i <= n/2$) then
   Update the status (position vector) of leader salp by Eq. (6)
  else ($i > n/2$ and $i < n + 1$)
   Update the status (position vector) of follower salps by Eq. (8)
  end if
  end for
  Return back those search agents that go over the legal limits of feature space.
 end while
Return repository
```

4. The proposed approach

The objective of this section is to introduce a wrapper feature selection strategy that efficiently finds a subset of features that minimizes the classification error rate. The proposed approach (which will be referred to as MODSSA-lbest) is designed around a modified version of the binary version of MOSSA for performing the search process. The modifications are the dynamic updating mechanism of the leaders and the adoption of the local best strategy. The former divides the SSA population into several sub-swarms with multiple leaders to perform more exploratory-oriented search at the first stages and then decreases over time to be more exploitation-oriented. Also, the c_1 coefficient is configured to change over time for providing a smoother transition from exploration to exploitation. The latter enhances the exploitation capability of salps by integrating their personal best experiences during the search process into the position updating equation. So, the salps do not depend only on their current experience and their neighbor experience, but further to consider their best-obtained solutions throughout the search history. In addition, the proposed algorithm used an S-shaped transfer function to transform the algorithm from searching in a continuous space to search in a discrete space. The general workflow of the proposed approach is depicted in Fig. 2. The following subsections present the detail of the proposed approach.

4.1. Dynamic Time-Varying MOSSA (MODSSA)

Population-based metaheuristics look to balance between diversification (exploration) and intensification (exploitation) in order to avoid the stagnation at local optimal solutions (X. Xu & Chen, 2014; Y. Xu et al., 2019). Hence, the algorithm performs more exploration at the first stages while later in the process tends to exploit the current regions more. In the earlier implementation of SSA, the population is divided into one leading salp and the follower salps. In which, the followers adjust their movement regarding other salps, while the leader depends on the source food *F*. However, this increases the likelihood of trapping in local regions. Therefore, splitting the swarm into multi-

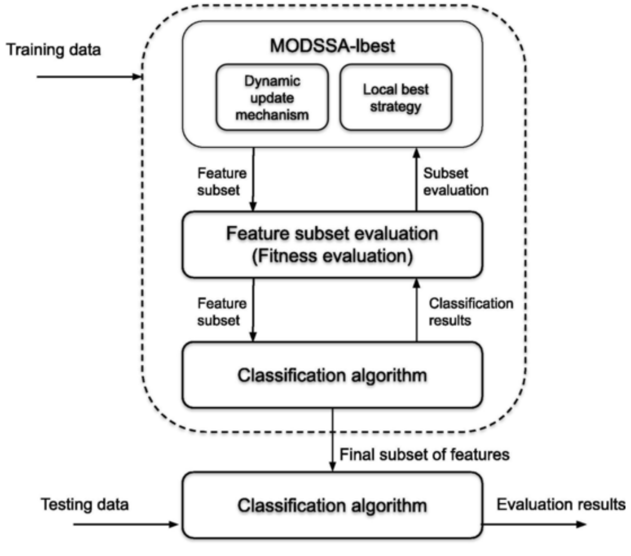


Fig. 2. Abstract flowchart of the proposed MODSSA-lbest..

ple sub-swarms with multi-leaders decreases the chance of the stuck in local solutions while at the same time brings more exploration. Initially, the swarm of salps is divided into several sub-swarms, where each sub-swarm has a leader salp and follower salps. Over time, the number of leaders increases while the number of followers decreases. Hence, the number of leaders (L) is defined in Eq. 9 , and the number of followers (Q) is described as in Eq. 10, where N is the number of salps in the population, i is the salp's index, and L is the number of the maximum iterations. The presence of multiple leaders promotes the movement of salps to be more adaptive and diversified during the search process.

$$L = ceil(N \times \frac{i}{L+1}) \tag{9}$$

$$Q = N - L \tag{10}$$

Initially, the population is created by producing a set of random salps. At each iteration, the fitness of each salp is calculated and the fittest salp is set as the food source F. The c_1 parameter is updated by Eq. 7 so the positions of the followers and leader salp will be updated accordingly. The c_1 coefficient plays a significant role in enhancing the explorative behaviors of the SSA algorithm. Designing a time-varying approach for c_1 equips the SSA algorithm with the needed flexibility in searching intensively globally at the beginning then decreasing gradually at the ending search periods. Eq. 7 shows an exponential behavior that first allows the algorithm to explore powerfully the search space then declines over the course of iterations.

Indeed, the implementation of multiple leaders and the continuous update of c_1 relying on time allow the population to relocate dynamically during the search. So, it can converge to the optimal solutions more smoothly.

4.2. A modified MODSSA with local best

A well-designed metaheuristic algorithm has the ability to balance between performing local and global search operations, so as to avoid the stagnation at local optimal solutions. In the early proposed version of the SSA, each follower salp updates its position regarding its current position and the position of the following salp, as illustrated in Eq. 8. Meanwhile, the leader salp adjusts its position using Eq. 6 and depending on the source food position F. Therefore, to enhance the exploitation capability of the salps, the best personal solution for each salp is also considered in modifying their positions. This is done by equipping each salp with internal memory to store its past experiences. Hence, the new updating procedure, which is presented by Eq. 11, is based on the difference between the previous best solution and the current solution.

$$x_j^i = \begin{cases} F_j + c_1\left(\left(Pbest_j^i - x_j^i\right) \times c_2\right) & c_3 \geqslant 0.5 \\ F_j - c_1\left(\left(Pbest_j^i - x_j^i\right) \times c_2\right) & c_3 < 0.5 \end{cases} \tag{11}$$

where the $Pbest_j^i$ is the personal best solution of a salp (i) at the dimension (j). As the personal best of a salp is obtained by using the dominance relationship. Thus, whenever a new, better solution is found, the Pbest of the salp will be updated. Essentially, the salps update their positions depending on their personal best and a threshold value. The threshold value is expressed by the maximum Traveling Distance Rate (TDR). TDR is set initially as in Eq. 12, which relies on the current (l) and the maximum iterations (L).

$$TDR = \frac{l}{L+1} \tag{12}$$

The TDR parameter equips the algorithm to perform more exploitation at the local regions of each salp at each sub-swarm, which consistently increases with the increasing number of leaders. Fig. 3 illustrates

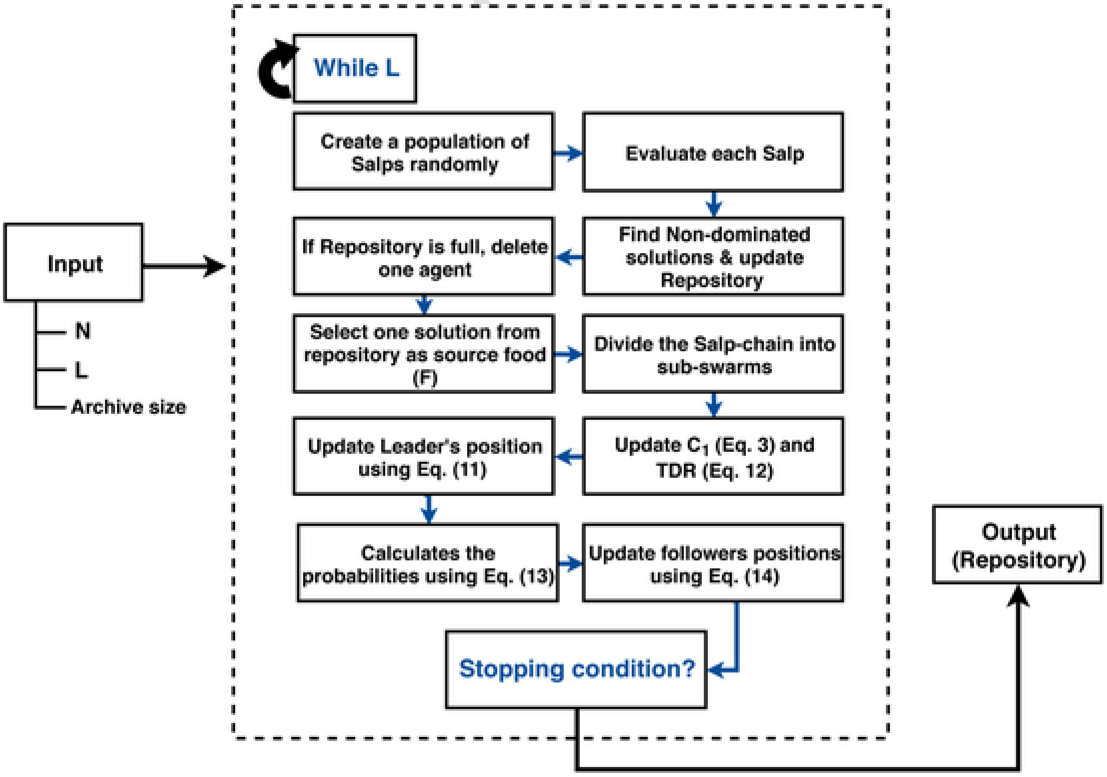

While L
Create a population of Salps randomly
Evaluate each Salp
Input
N
L
Archive size
If Repository is full, delete one agent
Find Non-dominated solutions & update Repository
Select one solution from repository as source food (F)
Divide the Salp-chain into sub-swarms
Update Leader's position using Eq. (11)
Update C1 (Eq. 3) and TDR (Eq. 12)
Calculates the probabilities using Eq. (13)
Update followers positions using Eq. (14)
Output (Repository)
Stopping condition?

Fig. 3. Flowchart of the proposed MODSSA-lbest.

the structural steps of the proposed methodology. Handling feature selection problems demands the transforming of the continuous search space into binary search space. The following subsection discusses the implemented binary SSA algorithm.

4.3. The binary SSA

Primarily, the SSA algorithm was developed to address continuous problems. Since the feature selection problem is a discrete problem, then the SSA algorithm is transformed to search in discrete or binary search spaces. A fundamental approach to convert continuous algorithms into binary one is the use of the transfer function (TF). TF is a mathematical operator that produces a probability for the input value being 1 or 0. A popular family of TFs is the V-shaped and S-shaped functions Mirjalili and Lewis (2013). Eq. 13 shows the S1 type of S-shaped transfer functions, which utilized in this research for features encoding. In which, j is the j^{th} element of salp i at iteration t.

$$T(x_j^i(t)) = \frac{1}{1 + e^{-2*x_j^i(t)}} \tag{13}$$

Using the produced probability of Eq. 13, the follower salps use it to update their positions in the next iteration, as in Eq. 14. Where i is the current element at the d^{th} dimension.

$$x_i^d(t+1) = \begin{cases} 1 & \text{if } rand \geqslant T(x_j^d(t)) \\ 0 & \text{if } rand < T(x_i^d(t)) \end{cases} \tag{14}$$

Each salp in the population represents a potential solution, which is a subset of features. The length of the salp's dimension corresponds to the total number of features. Thus, each dimension is a feature. The salps positions that were updated by Eq. 11, they are encoded using the S1 transfer function. Therefore, the output of Eq. 13 is used to create an encoded vector of a salp's position representing the selected subset of features, as it is given by Eq. 14. The generated encoded vectors transmitted to a machine-learning algorithm to evaluate their performance and to optimize the objective functions.

5. Experiments setup

5.1. Datasets

In order to assess the performance of the proposed algorithm, several benchmark datasets were deployed. Thirteen datasets were drawn from the UCI machine learning repository with different properties from various fields. Mainly, the datasets vary in the number of attributes and instances, while three of them are high-dimensional datasets (Glioma, Colon, and Leukemia). Table 1 presents the utilized datasets, the number of classes, the number of features, and the number of data instances.

Table 1
Summary of utilized datasets.

#	Dataset	No. of features	No. of instances	No. Classes
1	Nci9	9712	60	9
2	Glioma	4434	50	4
3	Lymphography	18	148	4
4	PenglungEW	325	73	7
5	WaveformEW	40	5000	3
6	Zoo	16	101	7
7	Exactly	13	1000	2
8	Exactly2	13	1000	2
9	HeartEW	13	270	2
10	SonarEW	60	208	2
11	SpectEW	22	267	2
12	Colon	2000	62	2
13	Leukemia	7129	72	2

5.2. Evaluation measures

Various evaluation measures were utilized including error rate, g-mean, recall, specificity, and the number of features. The error rate is the rate of incorrect predictions or classifications made by the learning algorithm. It is defined as the total number of incorrectly classified instances over the total number of instances in the dataset (Eq. 15), where TP is the true positive, TN is the true negative, FP is the false positive, and FN is the false negative.

$$Error\ rate = \frac{FP + FN}{TP + TN + FP + FN} \tag{15}$$

G-mean: is the geometric mean. This measure indicates the tradeoff between the classification performances on the majority class and the minor class (see Eq. 16).

$$G - mean = \sqrt{specificity \cdot recall} \tag{16}$$

Recall: is the true positive rate which corresponds to how much the learning algorithm can recognize the positively-classified records of data. It is defined in Eq. 17.

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

Specificity: it is known by the true negative rate that presents the ability of the learning algorithm in recognizing the records of data with the negative class. It is described as in (Eq. 18).

$$Specificity = \frac{TN}{TN + FP} \tag{18}$$

The number of features: corresponds that the superior classifier algorithm is the one that has the minimal most significant number of features which results in the minimum classification error rate.

5.3. Experimental settings

All experiments have been implemented on MATLAB R2019a, and conducted on a computer, where the processor is Intel Core(TM) i5, 6 GB of memory, the hard drive is 500 GB, and the operating system is Windows 10.

The proposed MODSSA-lbest is compared with traditional multi-objective evolutionary algorithms that are all adopted as wrapper feature selection methods. The used algorithms are the Multi-Objective Particle Swarm Optimization (MOPSO), the Non-dominated Sorting Genetic Algorithm (NSGA-II) Deb et al. (2000), and the Decomposition based Multi-Objective Evolutionary Algorithm (MOEA/D) Zhang and Li (2007). As they are wrapper methods, all the used algorithms integrate the K-Nearest Neighbor (KNN) with k = 5 as the learning algorithm for its simplicity and fast execution time Mohemmed and Zhang (2008). All used algorithms have experimented on the same environmental conditions. Table 2 presents the parameters settings of used multi-objective evolutionary algorithms.

The datasets divided into training and testing sets with (70%, 30%), respectively. Mainly, the training set is split into 5 subsets using the 5 folds cross-validation. The five subsets are used to perform an internal evaluation of the generated subsets of features. On the other hand, at the final stage, the testing set is utilized for assessing the performance of the produced non-dominated subsets of features that are located in the repository. Hence, the candidate solutions in the repository are evaluated on the testing set and the average of them is reported.

The experiments repeated 30 times to ensure if the results are statistically reasonable. For all multi-objective evolutionary algorithms,

Table 2
The initial settings of used classical MOEAs.

Algorithm	Parameter	Value
MOPSO	c_1, c_2	[1, 2]
	w, w_{damp}	[0.50, 0.99]
	No. of grids	7
	α	0.1
	β	2
	γ	2
NSGA-II	Crossover probability	0.9
	Mutation probability	0.4
MOEA/D	No. of sub-problems	population size (N)
	γ	0.5
	Neighbors ratio (T)	min(max(ceil(0.15*N,2),15))

the maximum number of agents in the population is 30, the maximum number of iteration is 50, and the repository size is 100.

6. Results and discussions

This section discusses the obtained results of the proposed algorithm (MODSSA-lbest) versus the standard MOSSA and other well-regarded multi-objective evolutionary algorithms (i.e., MOPSO, NSGA-II, and MOEA/D).

6.1. The comparison with MOSSA

Table 3 compares the performance of the proposed MODSSA-lbest with the Multi-Objective SSA (MOSSA) Mirjalili, Gandomi, et al. (2017) for 13 benchmarks datasets. The assessment is conducted regarding the average error rate, g-mean, recall, specificity, and the number of features. In regard to the average error rate, MODSSA-lbest accomplished a lower error in 62% of the datasets in comparison with MOSSA. Where it outperformed MOSSA at Glioma, Lymphography, PenglungEW, WaveformEW, Exactly, HeartEW, SpectEW, Colon, and Leukemia by obtaining (0.0140, 0.226, 0.230, 0.248, 0.272, 0.205, 0.198, 0.225, and 0.185), respectively, with feasible standard deviation values. However, at Exactly2 and WaveformEW, both algorithms; MODSSA-lbest and MOSSA had an equal error rate of (0.290 and 0.248), respectively.

While looking at the average g-mean, MODSSA-lbest outperformed MOSSA in 62% of the datasets (Nci9, Glioma, Lymphography, PenglungEW, Exactly, Exactly2, HeartEW, and Colon). The MODSSA-lbest had a maximum average g-mean at PenglungEW with 96.8%, which is considerably better than MOSSA that obtained 83.8%. Also, at Lymphograrphy, MOSSA extremely failed to achieve well which performed 0.0 in terms of g-mean. Whereas, MODSSA-lbest did significantly better by having (0.769). Nonetheless, at the Zoo dataset, even that MOSSA achieved slightly better than MODSSA-lbest by obtaining a maximum of 0.975, yet MODSSA-lbest could achieve 0.940. Overall, MODSSA-lbest can perform significantly better in terms of g-mean in most of the cases.

In terms of average recall, MODSSA-lbest outperformed MOSSA in 38% of the datasets; at Nci9, Glioma, Lymphography, PenglungEW, and Exactly which achieved 0.994, 0.750, 0.840, 0.955, and 0.865, respectively. Markedly, at Leukemia, MODSSA-lbest and MOSSA accomplished superb results by obtaining 1.000 of the true positive rate (recall). At Colon, SonarEW, HeartEW, Zoo, and waveformEW, even that MODSSA-lbest did not achieve better than MOSSA, but it accomplished somewhat close results justified by the standard deviation values. For instance, at SonarEW, MOSSA gained an average 0.799 with standard deviation equals to (± 0.027). Whereas, MODSSA-lbest had 0.755 with standard deviation equals to (± 0.033).

Interestingly, regarding the true negative rate (specificity), MODSSA-lbest accomplished better in 69% of the datasets. Where it obtained 1.000 at Glioma and 0.989 at PenglungEW. In addition, both MODSSA-lbest and MOSSA did the same at SonarEW by holding 0.908. However, both MOSSA and MODSSA-lbest achieved relatively the same having (0.980, 0.989) at PenglungEW and (0.870, 0.8733) at WaveformEW, respectively. Obviously, at Colon, MODSSA-lbest had approximately double the performance of MOSSA. Where the former had 0.463 and the latter had 0.823. Similarly, is at Leukemia, where MOSSA has 0.376, and MODSSA-lbest obtained 0.630.

Moreover, investigating the reduction in features is expressed by reporting the average number of features. At Nci9, both MODSSA-lbest and MOSSA have decreased highly the number of features from 9712 to 5642.52 with a Reduction Rate (RR) is 42% in regard to MOSSA, and decreased to 5753.19 for MODSSA-lbest (RR = 41%). While for Glioma, the number of selected features is reduced nearly to the half, where the dataset features demoted from 4434 to 2689.5 at MOSSA, and to 2565.1 at MODSSA-lbest. Likewise, is at SonarEW, where the RR = 50% for MOSSA, and RR = 48.6% for MODSSA-lbest. For Lymphography, the selected features have been decreased to less than half with RR equals 59%. PenglungEW, had RR = 44.6% at MOSSA, and equals to 44.4% for MODSSA-lbest. Similarly, WaveformEW dataset has minimized the number of features from 40 to 21.52 in the case of MOSSA, and to 21.47 for MODSSA-lbest. In addition, at Exactly, Exactly2, HeartEW, and SpectEW, the features diminished to less than half of the number of the original features. In which, MODSSA-lbest has a better RR at HeartEW and SpectEW. Also, at Colon and Leukemia, MODSSA-lbest had a superior RR, which obtained 40.9% for Colon and 45.8% for Leukemia. Furthermore, Table 3 provides the results of the Wilcoxon non-parametric statistical test for reinforcing whether the difference in the performance of the algorithms is significant. This test is based on the g-mean measure, and with the significance level alpha is set to 5%. The probabilities (p-values) produced by the Wilcoxon are less than 0.05 in 85% of the datasets. This indicates that the null hypothesis is strongly rejected at the 5% significance level. Therefore, the proposed approach MODSSA-lbest is significantly different from the standard method MOSSA, and the g-mean results obtained by the proposed method are statistically significant and were not achieved by chance for the used datasets.

Fig. 4 depicts the box-plots for comparing the proposed MODSSA-lbest with the MOSSA algorithm. The horizontal axis shows the algorithm and the vertical axis is the error rate. The start and the end of the boxes present the first and third quartiles, while the middle line is the median. It is clear from the graph that MODSSA-lbest accomplished stunning minimized error results at Glioma, PenglungEW, and Leukemia, as it is shown by sub-figures (b, d, and m), respectively. However, it still achieves dramatically well with the obvious differences in the error rate at Colon and SpectEW. Yet, MODSSA-lbest performed slightly better than MOSSA at Lymphography (c), WaveformEW (e), Exactly (g), Exactly2 (h), and HeartEW (i). Where there is some overlapping between the boxes which indicates that the two algorithms achieved similarly at some runs. Nonetheless, at Nci9 (a), Zoo (f), and SonarEW (j), the box-plots denote that MOSSA obtained better results than MODSSA-lbest.

Essentially, evaluating a multi-objective algorithm depends mainly on how much it can produce diverse, non-dominated solutions. In Fig. 5, the average Pareto front figures are characterized by presenting the number of features at the horizontal axis, and the error rate at the vertical axis. Whereas, the lines represent the Pareto non-dominated solutions obtained and preserved at the repository. The red lines denote the MODSSA-lbest algorithm, while the black one is the MOSSA. Obviously, at Glioma (b), PenglungEW (d), SonarEW (k), Colon (l), and Leukemia (m), MODSSA-lbest outperformed MOSSA in obtaining superior Pareto solutions by having the best number of features with

Table 3
Comparison between MODSSA_lbest and MOSSA used datasets in terms of error rate, g-mean, recall, specificity, and features ratios. For Wilcoxon test, when ($P \geqslant 0.05$) is underlined.

Datasets	Algorithm	Measure	Error rate	G-mean (P-values)	Recall	Specificity	Features#
Nci9	MOSSA	AVG	**0.576**	0.456 (2.872E-11)	0.337	**0.640**	**5642.516**
		STD	0.020	0.044	0.069	0.021	513.014
	MODSSA_lbest	AVG	0.734	**0.785**	**0.994**	0.624	5753.193
		STD	0.010	0.031	0.030	0.020	366.379
Glioma	MOSSA	AVG	0.489	0.649 (2.872E-11)	0.667	0.637	2689.489
		STD	0.028	0.025	0.048	0.002	233.635
	MODSSA_lbest	AVG	**0.140**	**0.866**	**0.750**	**1.000**	**2565.064**
		STD	0.007	0.000	0.000	0.000	167.484
multirow490ptLymphography	MOSSA	AVG	0.229	0.000 (2.872E-11)	0.000	**1.000**	**7.224**
		STD	0.032	0.000	0.000	0.000	0.983
	MODSSA_lbest	AVG	**0.226**	**0.769**	**0.840**	0.713	7.400
		STD	0.050	0.048	0.078	0.054	1.189
multirow490ptPenglungEW	MOSSA	AVG	0.408	0.838 (5.228E-11)	0.720	0.980	**180.044**
		STD	0.016	0.032	0.060	0.010	12.424
	MODSSA_lbest	AVG	**0.230**	**0.968**	**0.955**	**0.989**	180.791
		STD	0.016	0.032	0.057	0.009	16.156
multirow490ptWaveformEW	MOSSA	AVG	**0.248**	**0.782**(1.833E-01)	**0.703**	0.870	21.518
		STD	0.012	0.012	0.016	0.008	1.709
	MODSSA_lbest	AVG	**0.248**	0.776	0.691	**0.873**	**21.474**
		STD	0.018	0.018	0.023	0.013	1.712
multirow490ptZoo	MOSSA	AVG	**0.097**	**0.975**(4.639E-05)	**0.996**	**0.955**	7.318
		STD	0.037	0.020	0.009	0.036	0.837
	MODSSA_lbest	AVG	0.126	0.940	0.972	0.917	**6.685**
		STD	0.044	0.038	0.028	0.054	0.867
Exactly	MOSSA	AVG	0.293	0.537 (3.219E-01)	0.851	0.397	**4.705**
		STD	0.023	0.052	0.036	0.050	0.424
	MODSSA_lbest	AVG	**0.272**	**0.547**	**0.865**	**0.431**	4.983
		STD	0.030	0.070	0.035	0.066	0.476
Exactly2	MOSSA	AVG	**0.290**	0.176 (1.479E-05)	**0.950**	0.072	**4.948**
		STD	0.015	0.114	0.037	0.049	1.680
	MODSSA_lbest	AVG	**0.290**	**0.298**	0.902	**0.133**	5.887
		STD	0.023	0.096	0.043	0.048	1.581
HeartEW	MOSSA	AVG	0.223	0.757 (1.336E-03)	**0.840**	0.690	5.626
		STD	0.019	0.029	0.018	0.048	1.085
	MODSSA_lbest	AVG	**0.205**	**0.786**	0.824	**0.758**	**5.567**
		STD	0.029	0.033	0.033	0.046	1.155
SonarEW	MOSSA	AVG	**0.144**	**0.851** (9.845E-06)	**0.799**	**0.908**	**29.976**
		STD	0.015	0.015	0.027	0.026	2.874
	MODSSA_lbest	AVG	0.153	0.827	0.755	**0.908**	30.834
		STD	0.018	0.020	0.033	0.019	2.241
SpectEW	MOSSA	AVG	0.209	**0.745** (5.709E-09)	**0.699**	0.807	9.481
		STD	0.017	0.031	0.041	0.023	0.986
	MODSSA_lbest	AVG	**0.198**	0.670	0.522	**0.895**	**9.142**
		STD	0.014	0.061	0.061	0.016	1.426
Colon	MOSSA	AVG	0.333	**0.878**(2.872E-11)	**0.785**	0.463	1182.899
		STD	0.014	0.019	0.019	0.031	81.673
	MODSSA_lbest	AVG	**0.225**	0.784	0.753	**0.823**	**1181.788**
		STD	0.024	0.029	0.027	0.053	90.767

Table 3 (*Continued*)

Datasets	Algorithm	Measure	Error rate	G-mean (P-values)	Recall	Specificity	Features#
Leukemia	MOSSA	AVG	0.227	0.613 (2.872E-11)	**1.000**	0.376	3904.030
		STD	0.002	0.003	0.000	0.005	241.901
	MODSSA_lbest	AVG	**0.185**	**0.794**	**1.000**	**0.630**	**3862.20**
		STD	0.006	0.008	0.000	0.012	246.339

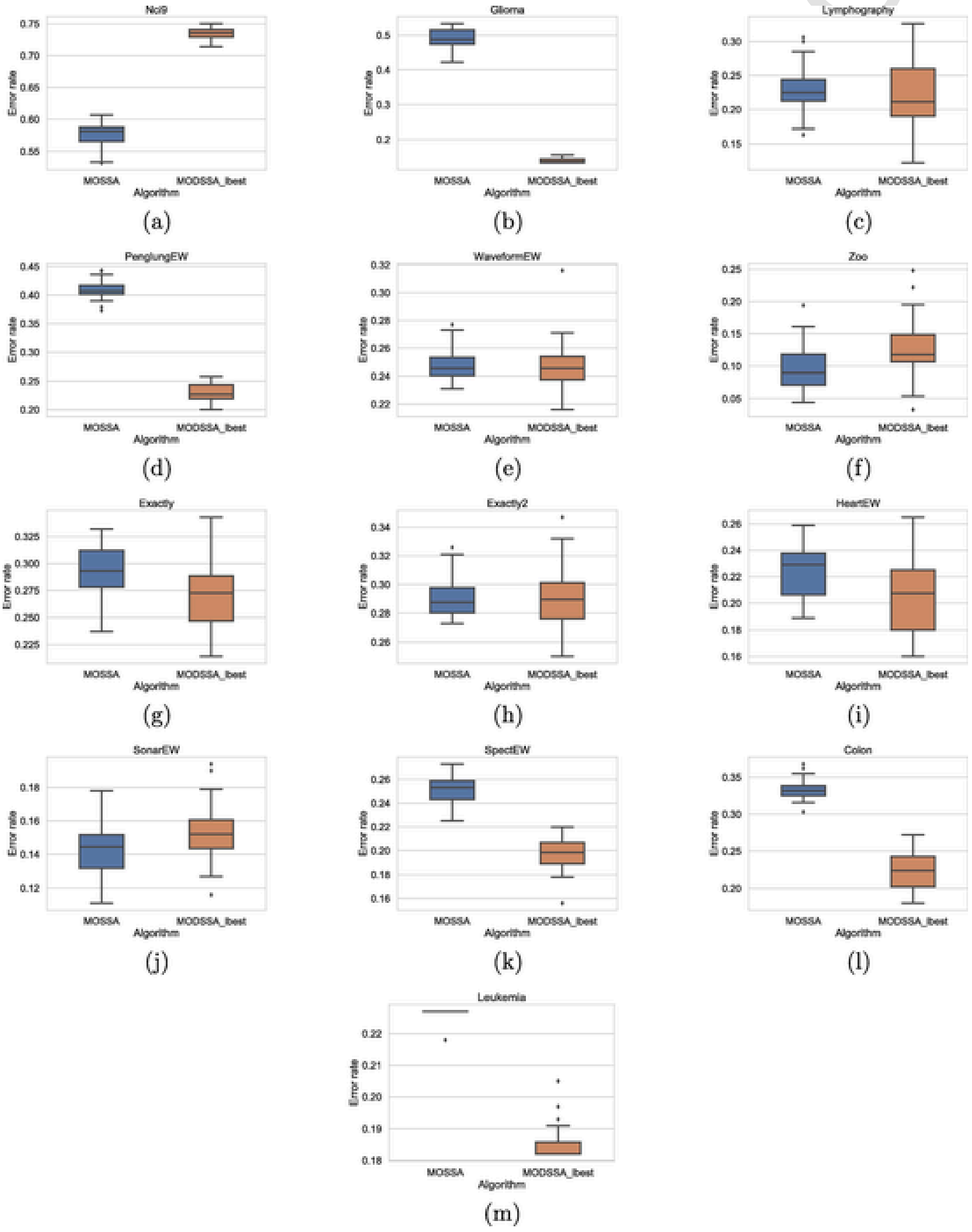
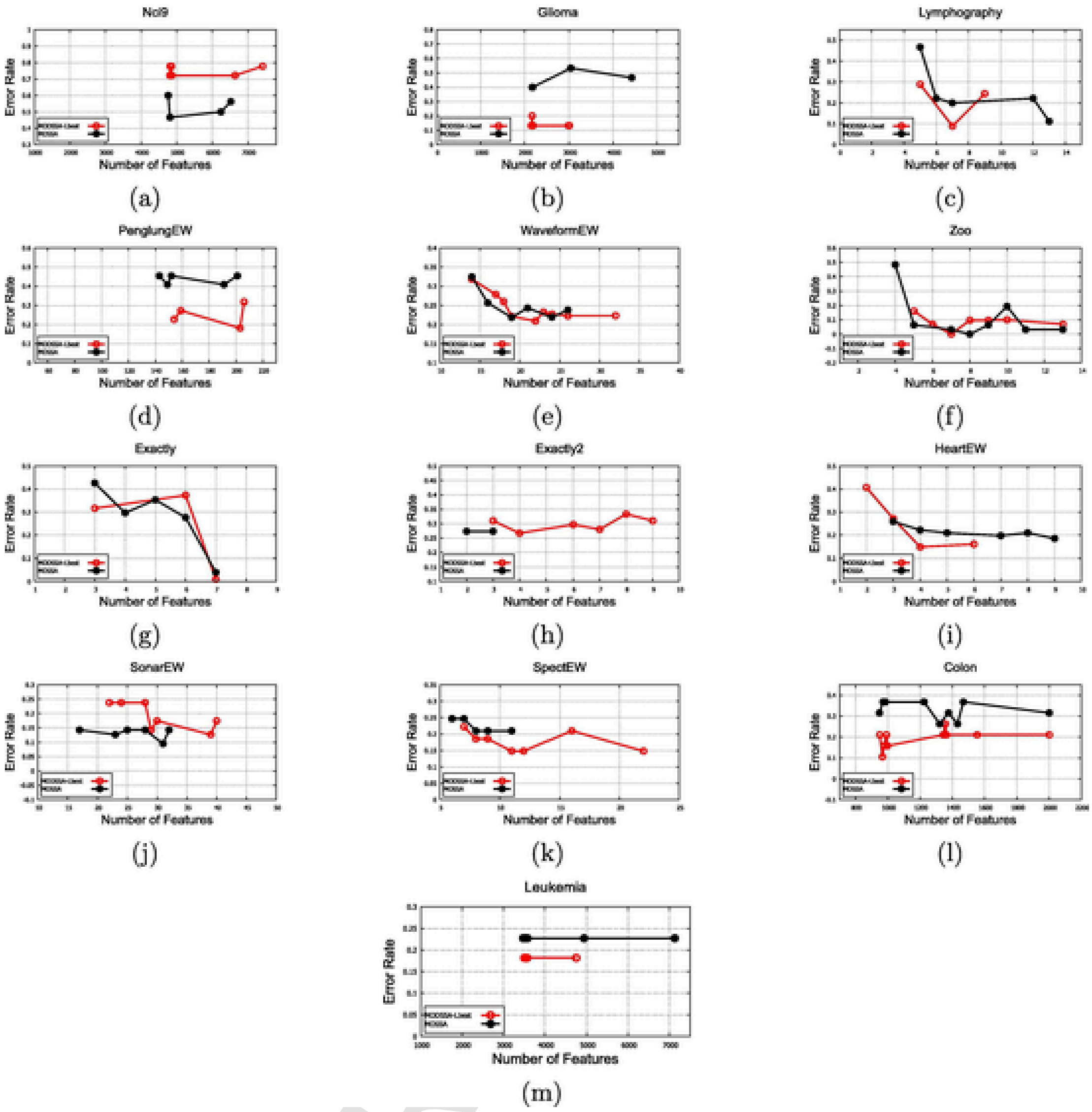


Fig. 4. Comparison of average error rate for MOSSA and MODSSA-Lbest over all datasets using box-plots.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m)

Fig. 5. The Pareto front plots that compare MODSSA-lbest versus MOSSA over all datasets.

the minimal error rate. However, at Lymphography, WaveformEW, Zoo, Exactly, HeartEW (c, e, f, g, and i), respectively, the produced solutions fluctuate, which means at some runs the Pareto solutions from MODSSA-lbest were with higher quality than the solutions generated from MOSSA, whilst at other points, they are worse. In other words, at WaveformEW dataset, first MOSSA produced better solutions, then it is performance declined and MODSSA-lbest could obtain superior results over the course of runs. Even that, MOSSA accomplished well and better than MODSSA-lbest at Exactly2 (h) and SonarEW (j).

6.2. The comparison with MOEAs

Table 4 discusses the performance of MODSSA-lbest versus MOPSO, NSGA-II, and MOEA/D, regarding the average error rate, g-mean, recall, specificity, and the selected number of features. As well as reports the values of standard deviation. In terms of average error rate, MODSSA-lbest accomplished superior results in 46.2% of the datasets. Where it minimized the error rate optimally at Zoo dataset by having 0.126, While MOPSO obtained 0.279, NSGA-II had the same error rate of 0.126, and MOEA/D had 0.186. Secondly, is at Glioma, it had optimally 0.140, while MOSPO, NSGA-II, and MOEA/D had 0.394, 0.187, 0.362, respectively. At Lymphography, MODSSA-lbest achieved slightly better than MOPSO and MOEA/D. Also, at WaveformEW, it had superior yet close performance results to MOPSO and NSGA-II. Nonetheless, it outperformed the other algorithms in minimizing the error rate at SonarEW and SpectEW by having 0.153, and 0.198, respectively.

Regarding the g-mean, MODSSA-lbest achieved better than MOPSO, NSGA-II, and MOEA/D in 77% of the datasets with feasible standard deviation (STD) values. It had the best g-mean at the PenglungEW dataset, where it had 0.968 and MOPSO, NSGA-II, and MOEA/D obtained 0.763, 0.964, 0.958, respectively. Also, at the Zoo dataset, it achieved merits by having 0.940 while MOPSO, NSGA-II, and MOEA/D had (0.923, 0.751, 0.860), respectively. Even that at Lymphography, WaveformEW, and HeartEW, MODSSA-lbest achieved superior results, but it slightly performed close to the other algorithms. To illustrate, at WaveformEW, MODSSA-lbest had 0.776, while MOPSO, NSGA-II, and MOEA/D obtained 0.773, 0.761, and 0.736, respectively. On the contrast, at Nci9, Glioma, and Leukemia, MOEA/D performed the best by having a g-mean equals to 0.800, 0.873, and 0.854, respectively.

However, with respect to recall, MODSSA-lbest achieved better than the other algorithms in 31% of the datasets, including Lymphography, HeartEW, SonarEW, and SpectEW. At Lymphography, MOPSO, NSGA-II, and MOEA/D accomplished 0.813, 0.802, and 0.808, respectively. Whereas, MODSSA-lbest obtained 0.840. Also, at HeartEW, MODSSA-

Table 4
Comparison between MODSSA-lbest and classical multi-objective evolutionary algorithms over all datasets.

Datasets	Algorithm	Error rate	G-mean	Recall	Specificity	#Features
		AVG ± ;STD	AVG ± ;STD	AVG ± ;STD	AVG ± ;STD	AVG ± ;STD
Nci9	MODSSA_lbest	0.733 ± ;0.009	0.785 ± ;0.031	0.994 ± ;0.300	0.624 ± ;0.200	5753.193 ± ;366.379
	MOPSO	**0.559** ± ;0.021	0.617 ± ;0.059	0.666 ± ;0.134	0.589 ± ;0.016	4751.374 ± ;26.441
	NSGA-II	0.584 ± ;0.036	0.505 ± ;0.069	0.330 ± ;0.080	**0.792** ± ;0.044	4627.570 ± ;39.205
	MOEA/D	0.722 ± ;0.003	**0.800** ± 0.021	**1.000** ± ;0.000	0.64 ± ;0.0340	**4614.118** ± ;69.889
Glioma	MODSSA_lbest	**0.140** ± ;0.007	0.866 ± ;0.000	0.750 ± ;0.000	**1.000** ± ;0.000	2565.064 ± ;167.484
	MOPSO	0.394 ± ;0.009	0.817 ± ;0.000	0.667 ± ;0.000	**1.000** ± ;0.000	2149.792 ± ;19.961
	NSGA-II	0.187 ± ;0.031	0.508 ± ;0.091	0.268 ± ;0.096	**1.000** ± ;0.000	2058.221 ± ;23.447
	MOEA/D	0.362 ± ;0.063	**0.873** ± ;0.069	**0.873** ± ;0.126	0.878 ± ;0.041	**2026.800** ± ;42.196
Lymphography	MODSSA_lbest	**0.226** ± ;0.050	**0.769** ± ;0.048	**0.840** ± ;0.078	**0.713** ± ;0.054	7.400 ± ;1.189
	MOPSO	0.228 ± ;0.035	0.741 ± ;0.042	0.813 ± ;0.052	0.689 ± ;0.055	5.132 ± ;0.904
	NSGA-II	0.300 ± ;0.047	0.676 ± ;0.042	0.802 ± ;0.056	0.581 ± ;0.047	**4.227** ± ;0.925
	MOEA/D	0.287 ± ;0.055	0.687 ± ;0.071	0.808 ± ;0.102	0.619 ± ;0.089	4.332 ± ;1.321
PenglungEW	MODSSA_lbest	0.230 ± ;0.016	**0.968** ± ;0.032	0.955 ± ;0.057	0.989 ± ;0.009	180.791 ± ;16.156
	MOPSO	0.320 ± ;0.024	0.763 ± ;0.064	0.596 ± ;0.109	**0.999** ± ;0.004	142.209 ± ;4.934
	NSGA-II	0.277 ± ;0.030	0.964 ± ;0.062	0.961 ± ;0.108	0.973 ± ;0.032	**116.146** ± ;7.011
	MOEA/D	**0.148** ± ;0.032	0.958 ± ;0.020	**1.000** ± ;0.000	0.918 ± ;0.039	125.129 ± ;10.277
WaveformEW	MODSSA_lbest	**0.248** ± ;0.018	**0.776** ± ;0.018	0.691 ± ;0.023	0.873 ± ;0.013	21.474 ± ;1.712
	MOPSO	0.263 ± ;0.016	0.773 ± ;0.015	**0.701** ± 0.021	0.853 ± ;0.011	14.821 ± ;1.071
	NSGA-II	0.260 ± ;0.015	0.761 ± ;0.025	0.667 ± ;0.032	**0.887** ± ;0.009	**7.319** ± ;0.594
	MOEA/D	0.307 ± ;0.031	0.736 ± ;0.034	0.658 ± ;0.043	0.825 ± ;0.025	13.074 ± ;2.266
Zoo	MODSSA_lbest	**0.126** ± ;0.044	**0.940** ± ;0.038	0.972 ± ;0.028	0.917 ± ;0.054	6.6845 ± ;0.867

Table 4 (*Continued*)

Datasets	Algorithm	Error rate	G-mean	Recall	Specificity	#Features
		AVG ± ;STD	AVG ± ;STD	AVG ± ;STD	AVG ± ;STD	AVG ± ;STD
	MOPSO	0.279 ± ;0.052	0.923 ± ;0.087	0.904 ± ;0.086	**0.982** ± ;0.021	4.402 ± ;0.969
	NSGA-II	**0.126** ± ;0.034	0.751 ± ;0.048	**0.999** ± ;0.004	0.679 ± ;0.058	**3.143** ± ;0.323
	MOEA/D	0.186 ± ;0.051	0.860 ± ;0.085	0.865 ± ;0.090	0.908 ± ;0.051	4.545 ± ;0.846
Exactly	MODSSA_lbest	0.272 ± ;0.030	**0.547** ± ;0.070	0.865 ± ;0.035	0.431 ± ;0.066	4.983 ± ;0.476
	MOPSO	**0.270** ± ;0.039	0.501 ± ;0.059	0.859 ± ;0.055	**0.451** ± ;0.059	4.424 ± ;0.610
	NSGA-II	0.258 ± ;0.014	0.405 ± ;0.068	0.945 ± ;0.023	0.320 ± ;0.072	3.837 ± ;0.360
	MOEA/D	0.289 ± ;0.009	0.068 ± ;0.118	**0.980** ± ;0.037	0.043 ± ;0.078	**1.767** ± ;1.216
Exactly2	MODSSA_lbest	0.290 ± ;0.023	**0.298** ± ;0.096	0.902 ± ;0.043	**0.133** ± ;0.048	5.887 ± ;1.581
	MOPSO	0.286 ± ;0.039	0.179 ± ;0.109	0.905 ± ;0.072	0.099 ± ;0.075	3.412 ± ;1.548
	NSGA-II	**0.238** ± ;0.032	0.059 ± ;0.084	0.979 ± ;0.031	0.021 ± ;0.033	2.272 ± ;1.591
	MOEA/D	0.243 ± ;0.000	0.000 ± ;0.000	**1.000** ± ;0.000	0.000 ± ;0.000	**1.400** ± ;0.675
HeartEW	MODSSA_lbest	**0.205** ± ;0.029	**0.786** ± ;0.033	**0.824** ± ;0.033	0.758 ± ;0.046	5.567 ± ;1.155
	MOPSO	0.223 ± ;0.041	0.768 ± ;0.064	0.757 ± ;0.060	**0.802** ± ;0.040	3.698 ± ;0.835
	NSGA-II	0.215 ± ;0.007	0.783 ± ;0.013	0.783 ± ;0.021	0.787 ± ;0.030	**2.723** ± ;0.341
	MOEA/D	0.223 ± ;0.036	0.727 ± ;0.087	0.821 ± ;0.027	0.716 ± ;0.090	3.982 ± ;0.857
SonarEW	MODSSA_lbest	**0.153** ± ;0.018	**0.827** ± ;0.020	**0.755** ± ;0.033	**0.908** ± ;0.019	30.834 ± ;2.241
	MOPSO	0.234 ± ;0.023	0.754 ± ;0.027	0.667 ± ;0.047	0.856 ± ;0.019	21.257 ± ;1.891
	NSGA-II	0.232 ± ;0.035	0.764 ± ;0.037	0.692 ± ;0.052	0.847 ± ;0.040	**9.178** ± ;2.283
	MOEA/D	0.232 ± ;0.036	0.762 ± ;0.039	0.694 ± ;0.065	0.840 ± ;0.032	17.934 ± ;4.597
SpectEW	MODSSA_lbest	**0.198** ± ;0.014	**0.670** ± ;0.061	**0.522** ± ;0.061	0.895 ± ;0.016	9.142 ± ;1.426
	MOPSO	0.254 ± ;0.015	0.122 ± ;0.108	0.079 ± ;0.089	**0.964** ± ;0.022	6.619 ± ;0.824

Table 4 (*Continued*)

Datasets	Algorithm	Error rate	G-mean	Recall	Specificity	#Features
		AVG ± ;STD	AVG ± ;STD	AVG ± ;STD	AVG ± ;STD	AVG ± ;STD
	NSGA-II	0.206 ± ;0.009	0.360 ± ;0.108	0.236 ± ;0.098	0.932 ± ;0.028	**3.907** ± ;0.557
	MOEA/D	0.205 ± ;0.013	0.380 ± ;0.270	0.335 ± ;0.241	0.909 ± ;0.068	4.878 ± ;2.829
Colon	MODSSA_lbest	0.225 ± ;0.024	**0.784** ± ;0.029	0.753 ± ;0.027	**0.823** ± ;0.053	1181.788 ± ;90.767
	MOPSO	0.286 ± ;0.020	0.530 ± ;0.039	**0.955** ± ;0.023	0.301 ± ;0.045	948.436 ± ;12.868
	NSGA-II	**0.210** ± ;0.045	0.740 ± ;0.072	0.833 ± ;0.048	0.667 ± ;0.120	**894.621** ± ;17.106
	MOEA/D	0.311 ± ;0.057	0.662 ± ;0.066	0.773 ± ;0.070	0.574 ± ;0.105	894.987 ± ;31.488
Leukemia	MODSSA_lbest	0.185 ± ;0.006	0.794 ± ;0.008	**1.000** ± ;0.000	0.630 ± ;0.012	3862.202 ± ;246.339
	MOPSO	0.135 ± ;0.007	0.818 ± ;0.010	**1.000** ± ;0.000	0.671 ± ;0.017	3475.702 ± ;25.214
	NSGA-II	0.141 ± ;0.030	0.847 ± ;0.035	0.999 ± ;0.006	0.719 ± ;0.058	3365.504 ± ;30.451
	MOEA/D	**0.110** ± ;0.022	**0.854** ± ;0.032	**1.000** ± ;0.000	**0.730** ± ;0.053	**3344.983** ± ;58.627

lbest had 0.824, while the other algorithms in the same order attained 0.757, 0.783, and 0.821. Additionally, at SonarEW, it performed 0.755, whilst the other algorithms reached an average equals to 0.667 for MOPSO, 0.692 for NSGA-II, and 0.694 for MOEA/D. In essence, at SpectEW, MODSSA-lbest achieved approximately the double performance of the other algorithm. Where MOPSO, NSGA-II, and MOEA/D had 0.079, 0.236, and 0.335, respectively. At Leukemia, MODSSA-lbest, MOPSO, and MOEA/D achieved the same by having 1.000, while NSGA-II performed very closely by obtaining 0.999. Remarkably, at Nci9, Glioma, PenglungEW, Exactly, and Exactly2, MOEA/D accomplished the best by having 1.000, 0.873, 1.000, 0.980, and 1.000, respectively. Nonetheless, MOPSO did the best at Colon (0.955) and WaveformEW (0.701), whereas NSGA-II performed the best at the Zoo dataset (0.999).

Regarding the true negative rate (specificity), MODSSA-lbest obtained the best in 31% of the datasets. At Lymphography (0.840), Exactly2 (0.133), SonarEW (0.908), and Colon (0.823). At Glioma, MODSSA-lbest, MOPSO, and NSGA-II performed the optimally the same average of 1.000, while MOEA/D had a declined performance of average specificity of 0.878. However, NSGA-II accomplished the best at Nci9 and WaveformEW (0.792, 0.887), respectively. While MOPSO obtained superior results at PenglungEW (0.999), at Zoo (0.982), at HeartEW (0.802), and at SpectEW (0.964).

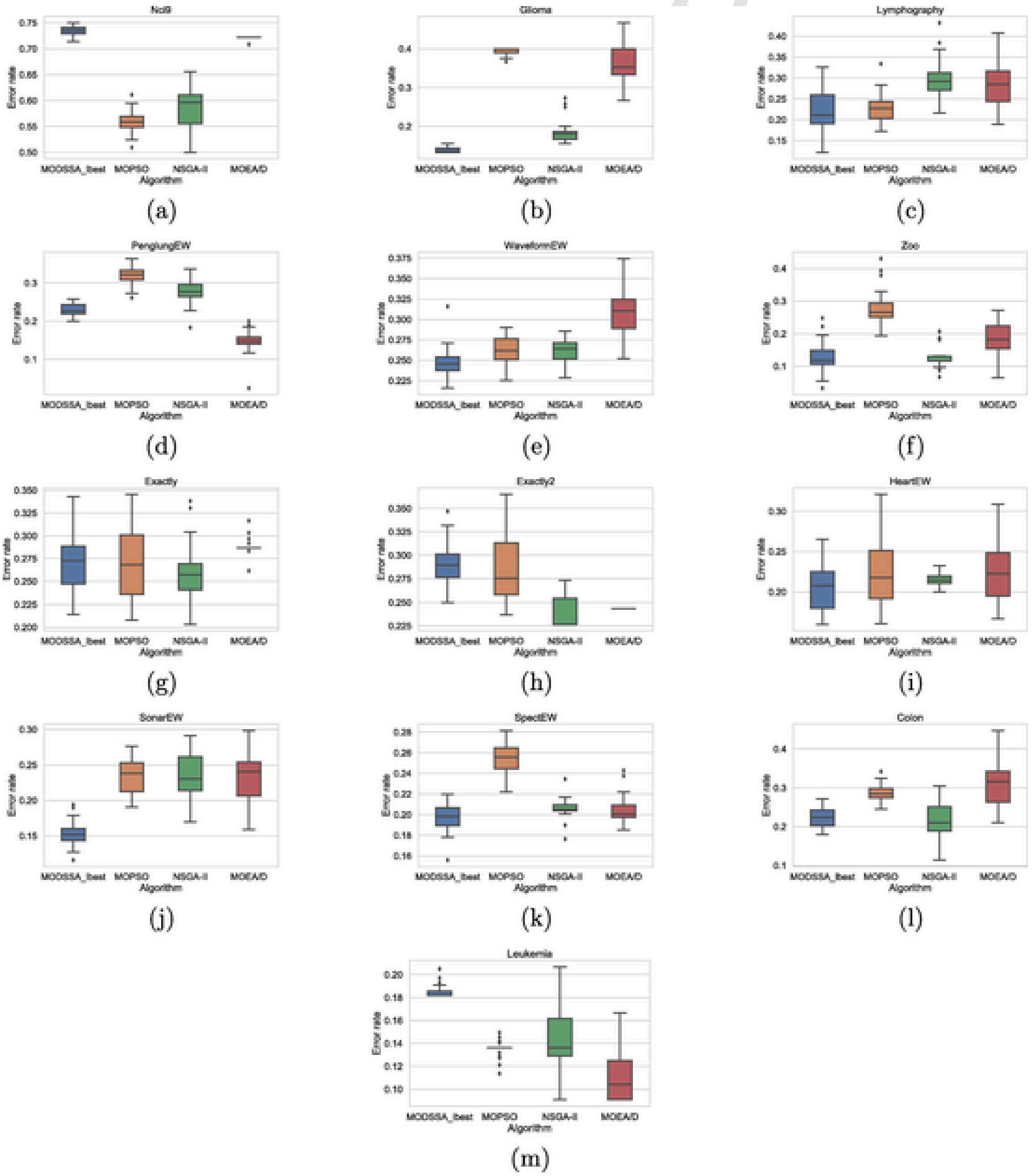
Minimizing the number of redundant features is crucial. Assessing the algorithms' ability to diminishing the irrelevant features is presented in the last column in Table 4. Where the average numbers of selected features over the 30 runs were reported besides the standard deviation. Even that MODSSA-lbest significantly reduced the original number of features, yet the other algorithms could obtain a fewer number of features. For instance, at Nci9 dataset, MODSSA-lbest has reduced the number of features from 9712 to 5753.2 with a reduction rate of 41%, while the MOEA/D reduced the number of features by 52.1%. Whereas at Glioma, MODSSA-lbest could minimize the features nearly to one third, however, MOEA/D reduced them approximately to the half with a RR of 54.3%. At Lymophography, MODSSA-lbest decreased the features by 58.9%, but the NSGA-II has decreased them by 76.5%. Noticeably, NSGA-II had eminent reduction rate over the other algorithms at PenglungEW, WaveformEW, Zoo, HeartEW, SonarEW, SpectEW, and Colon. Where MODSSA-lbest and NSGA-II had RR (44.4%, 64.3%), (46.3%, 81.7%), (58.2%, 80.4%), (57.2%, 79.1%), (48.6%, 84.7%), (58.4%, 82.2%), (40.9%, 55.3%) for the aforementioned datasets, respectively. Also, MOEA/D has merits in decreasing the number of features at Exactly, Exactly2, and Leukemia. While in comparison with MODSSA-lbest, they minimized the features with a reduction rate of (61.6%, 86.4%), (54.7%, 89.2%), (45.8%, 53.1%) for MODSSA-lbest and MOEA/D, respectively, in order with the three datasets. Even the obtained results for MODSSA-lbest regarding the number of selected features; in most of the cases, it still demonstrates a superb ability in reducing the number of features to approximately 50% at most of the datasets with reasonable standard deviation values.

Measuring the significance of MODSSA-lbest results can be achieved by statistical tests. Friedman test is a non-parametric statistical test, which tests the differences between the four used algorithms whether they are considerable or not. The average Friedman ranking is utilized, the smaller the rank, the more significant is the difference. From Table 4, MODSSA-lbest achieved the best results in terms of g-mean. where it had 1.385, while MOPSO, NSGA-II, and MOEA/D had 2.923, 2.923, and 2.769, respectively.

Fig. 6 compares the error presented by the box-plots between the proposed MODSSA-lbest and MOPSO, NSGA-II, and MOEA/D. It is clear from the figure that MODSSA-lbest performed competitively better at Glioma, WaveformEW, Zoo, HeartEW, SonarEW, and SpectEW datasets at sub-figures (b, c, e, f, i, j, and k, respectively). While at Nci9 (a) and Leukemia (m), MODSSA-lbest did not achieve better than other MOEAs, even that it could obtain satisfying results particularly at Leukemia (0.180). Noticeably, at sub-figures (l and h), NSGA-II outperformed all algorithms. Also, at Exactly dataset (g), all of MODSSA-lbest, MOPSO, and NSGA-II performed slightly the same.

Whereas, Fig. 7 presents the average Pareto front plots of MODSSA-lbest, MOPSO, NSGA-II, and MOEA/D which depict the obtained Pareto front (non-dominated) solutions. In which, the red line is the MODSSA-lbest. At Glioma (b), Lymphography (c), Zoo (f), SonarEW (j), SpectEW (k), and Colon (l), MODSSA-lbest can achieve superior solutions most of the time. Yet at some other points, the other MOEAs can accomplish better results. For example, at Glioma, MODSSA-lbest performed better than MOPSO, NSGA-II, and MOEA/D, however, it produced a solution with a higher number of features. As well as, at SpectEW, obviously MODSSA-lbest achieved two non-dominated solutions with the optimal number of features and minimal error rate. At Nci9 (a), Colon (l), SonarEW (j), and WaveformEW (e), MOPSO performed better than the used MOEAs. At PenglungEW (d) and Leukemia (m), MOEA/D obtained promising solutions among all.

Table 5 shows the P-values based on Wilcoxon test Gehan (1965), with a significance level ($\alpha = 5\%$). The test illustrates whether there is a significant difference between the proposed MODSSA-lbest and MOPSO, NSGA-II, and MOEA/D. The table shows P-values for the three algorithms overall used datasets. Remarkably, the P-values indicate significant results with $P \leqslant 0.05$. However, at Glioma and MOEA/D the P-value was greater than 0.05. Also, at PenglungEW, HeartEW, and SpectEW with NSGA-II they were above 0.05. While for MOPSO, the P-values were more than 0.05 at WaveformEW and HeartEW.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m)

Fig. 6. Comparison of average error rate for MODSSA-lbest versus MOPSO, NSGA-II, and MOEA/D over all datasets using box-plots.

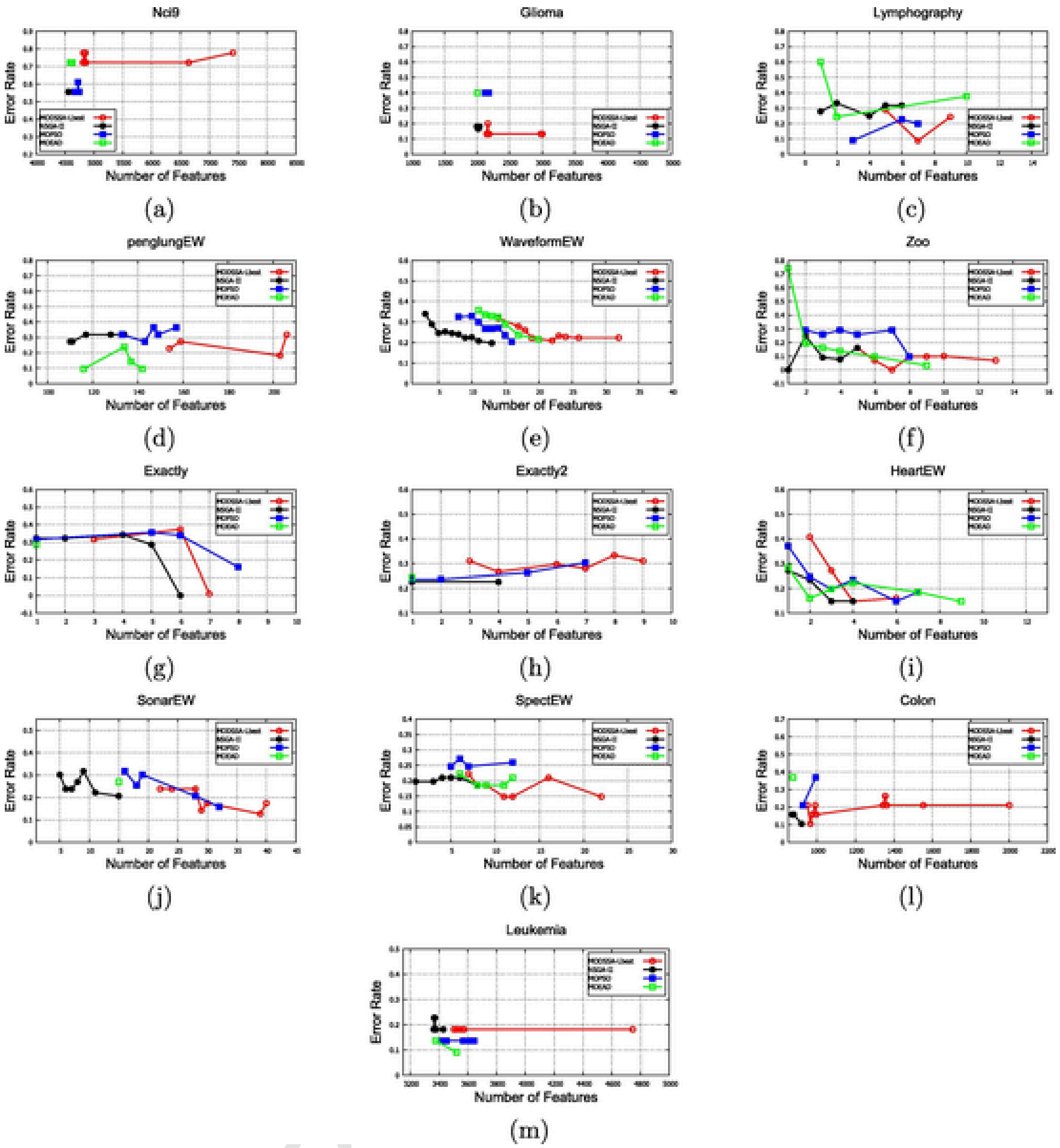


Fig. 7. The Pareto front plots that compare MODSSA-Lbest versus NSGA-II, MOPSO, and MOEAD over all datasets.

To sum up, the addition of a dynamic population (multiple leaders) with a local best updating mechanism brought an eminent capability of MODSSA-lbest in converging toward optimal solutions. Where it is justified by the outperforming results of g-mean and error rate. However, even that MODSSA-lbest maintained efficient performance results but it minimized the number of features to some extent where NSGA-II had the minimal reduction rate. Deciding the best solution in terms of efficiency and number of features depends highly on the decision-maker which relates to the respective problem. Yet, MODSSA-lbest proved highly competitive results to be considered for various problems.

Table 6 presents the results of comparing filter-based methods with the proposed MODSSA_lbest regarding the g-mean. The used filter-based methods are the ReleifF, Correlation, Information gain, and the Symmetrical filtering method. The same experimental settings were applied, which includes using the simple split approach of (70%,30%) for training, and testing, respectively. Now that a KNN classifier was also used with k = 5. The experiment was conducted on the Weka platform Hall et al. (2009). The results show that the proposed MODSSA_lbest can perform better than the traditional filter-based methods in 70% of the datasets. For instance, at the Nci9, the Lymphography, and the Zoo datasets, all filter-based methods failed to achieve any correct classification, while the MODSSA_lbest achieved a g-mean of (0.785) at the Nci9, (0.769) at the Lymphography, and (0.940) at the Zoo. Also, it achieved better results at Glioma, PenglungEW, Exactly, SonarEW, SpectEW, and Colon, by having (0.866, 0.968, 0.547, 0.827, 0.670, and 0.784), respectively. However, it did not perform better than other filter-based methods at the WaveformEW, Exactly2, HeartEW, and the Leukemia dataset.

7. Conclusion

Feature selection is a decisive preprocessing step, which highly affects the performance of machine learning algorithms. This article has formulated the problem of feature selection as a multi-objective optimization problem. In this work, the SSA algorithm is utilized for addressing the problem. The MOSSA algorithm is modified to handle binary problems using the S1 transfer function. Moreover, it adopted two operators to enhance its convergence ability toward optimal solutions. First, the use of multiple leader salps and multiple sub-swarms in-

Table 5
The P-values based on G-mean, where $P \geqslant 0.05$ are underlined.

Dataset	MOPSO	NSGA-II	MOEA/D
Nci9	1.04E-10	2.87E-11	3.09E-02
Glioma	2.87E-11	2.87E-11	underline1.00E + 00
Lymphography	1.47E-02	1.63E-08	6.97E-06
PenglungEW	4.40E-10	underline3.79E-01	3.85E-02
WaveformEW	underline1.98E-01	1.60E-02	1.93E-06
Zoo	5.44E-01	4.73E-11	1.29E-04
Exactly	2.03E-02	1.93E-08	2.87E-11
Exactly2	1.48E-05	1.77E-09	5.32E-10
HeartEW	underline4.51E-01	underline5.54E-01	1.25E-02
SonarEW	5.77E-11	4.00E-09	6.26E-08
SpectEW	2.87E-11	underline1.35E-01	3.39E-07
Colon	2.87E-11	7.13E-03	7.04E-10
Leukemia	2.49E-10	1.93E-08	5.32E-10

stead of relying on one leader to guide the whole population. Second is modifying the salps positions in regard to their best past solutions. The proposed MODSSA-lbest is compared with the standard MOSSA and other MOEAs, including MOPSO, NSGA-II, and MOEA/D. The MODSSA-lbest achieved significant results in terms of average error rate and g-mean over the 13 benchmark datasets.

CRediT authorship contribution statement

Ibrahim Aljarah: Conceptualization, Methodology, Formal analysis, Writing - review & editing, Supervision. **Maria Habib:** Conceptualization, Formal analysis, Data curation, Writing - review & editing. **Hossam Faris:** Conceptualization, Visualization, Investigation, Formal analysis, Writing - review & editing, Supervision. **Nailah Al-Madi:** Methodology, Software, Writing - review & editing. **Ali Asghar Heidari:** Methodology, Visualization, Writing - review & editing. **Majdi Mafarja:** Methodology, Writing - review & editing. **Mohamed Abd Elaziz:** Conceptualization, Writing - review & editing. **Seyedali Mirjalili:** Conceptualization, Visualization, Writing - review & editing, Supervision, Project administration.

Table 6
The g-mean values of the filter-based methods versus the proposed MODSSA_lbest.

Datasets	ReleifF	Correlation	Information gain	Symmetrical	MODSSA_lbest
Nci9	NAN	NAN	NAN	NAN	**0.785**
Glioma	0.707	0.707	0.707	0.707	**0.866**
Lymphography	NAN	NAN	NAN	NAN	**0.769**
PenglungEW	0.944	0.944	0.944	0.944	**0.968**
WaveformEW	**0.812**	0.803	0.809	0.809	0.776
Zoo	NAN	NAN	NAN	NAN	**0.940**
Exactly	0.527	0.099	0.195	0.195	**0.547**
Exactly2	**0.394**	0.219	0.000	0.000	0.298
HeartEW	0.712	**0.788**	0.764	0.764	0.786
SonarEW	0.784	0.791	0.784	0.784	**0.827**
SpectEW	0.603	0.609	0.609	0.609	**0.670**
Colon	0.666	0.577	0.666	0.666	**0.784**
Leukemia	**1.000**	**1.000**	**1.000**	**1.000**	0.794

*NAN: The algorithm failed to classify correctly any instance for some classes.

References

Abd Elaziz, M, Heidari, A A, Fujita, H, & Moayedi, H (2020). A competitive chain-based Harris Hawks Optimizer for global optimization and multi-level image thresholding problems. Applied Soft Computing, 106347.

Aljarah, I, Ala'M, Faris, H, Hassonah, M A, Mirjalili, S, & Saadeh, H (2018). Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. Cognitive Computation, 10(3), 478–495.

Aljarah, I, Mafarja, M, Heidari, A A, Faris, H, Zhang, Y, & Mirjalili, S (2018). Asynchronous accelerating multi-leader salp chains for feature selection. Applied Soft Computing, 71, 964–979.

Alresheedi, S S, Lu, S, Elaziz, M A, & Ewees, A A (2019). Improved multiobjective salp swarm optimization for virtual machine placement in cloud computing. Human-centric Computing and Information Sciences, 9, 15.

Amoozegar, M, & Minaei-Bidgoli, B (2018). Optimizing multi-objective pso based feature selection method using a feature elitism mechanism. Expert Systems with Applications, 113, 499–514.

Antonio, L M, & Coello, C A C (2017). Coevolutionary multi-objective evolutionary algorithms: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation.

Auer, P., Chiang, C.-K., Ortner, R., & Drugan, M. (2016). Pareto front identification from stochastic bandit feedback. In *Artificial Intelligence and Statistics* (pp. 939–947).

Bandaru, S, Ng, A H, & Deb, K (2017). Data mining methods for knowledge discovery in multi-objective optimization: Part a-survey. Expert Systems with Applications, 70, 139–159.

Banks, A, Vincent, J, & Anyakoha, C (2008). A review of particle swarm optimization. part ii: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. Natural Computing, 7, 109–124.

Behravan, I., Dehghantanha, O., & Zahiri, S. H. (2016). An optimal svm with feature selection using multi-objective pso. In *2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)* (pp. 76–81). IEEE.

Bermejo, P., Gámez, J. A., & Puerta, J. M. (2009). Incremental wrapper-based subset selection with replacement: An advantageous alternative to sequential forward selection. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on* (pp. 367–374). IEEE.

Chandrashekar, G, & Sahin, F (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40, 16–28.

Chen, B, Chen, L, & Chen, Y (2013). Efficient ant colony optimization for image feature selection. Signal Processing, 93, 1566–1576.

ChenH. ZhangQ. LuoJ. XuY. ZhangX. 2020 An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machineApplied Soft Computing86105884

Chuang, L-Y, Yang, C-H, Wu, K-C, & Yang, C-H (2011). A hybrid feature selection method for dna microarray data. Computers in Biology and Medicine, 41, 228–237.

Coello, C. C., & Lechuga, M. S. (2002). Mopso: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)* (Vol. 2, pp. 1051–1056). IEEE.

Coello, C A C, Lamont, G B, & Van Veldhuizen, D A (2007). Evolutionary algorithms for solving multi-objective problems: Vol. 5. Springer.

Dash, M, & Liu, H (1997). Feature selection for classification. Intelligent Data Analysis, 1, 131–156.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *International conference on parallel problem solving from nature* (pp. 849–858). Springer.

De Souza, R. C. T., dos Santos Coelho, L., De Macedo, C. A., & Pierezan, J. (2018). A v-shaped binary crow search algorithm for feature selection. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE (pp. 1–8).

Dickson, B B, Wang, S, Dong, R, & Wen, C (2015). A feature selection method based on multi-objective optimisation with gravitational search algorithm. Geo-Informatics in Resource Management and Sustainable Ecosystem (pp. 549–558). Springer.

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on* (pp. 39–43). IEEE.

Emary, E, Yamany, W, Hassanien, A E, & Snasel, V (2015). Multi-objective gray-wolf optimization for attribute reduction. Procedia Computer Science, 65, 623–632.

Emmanouilidis, C., Hunter, A., & MacIntyre, J., (2000). A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)* (Vol. 1, pp. 309–316). IEEE.

Faris, H, Aljarah, I, & Mirjalili, S (2016). Training feedforward neural networks using multi-verse optimizer for binary classification problems. Applied Intelligence, 45, 322–332.

Faris, H, Mafarja, M M, Heidari, A A, Aljarah, I, Ala'M, A-Z, Mirjalili, S, & Fujita, H (2018). An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. Knowledge-Based Systems, 154, 43–67.

Ferranti, A, Marcelloni, F, Segatori, A, Antonelli, M, & Ducange, P (2017). A distributed approach to multi-objective evolutionary generation of fuzzy rule-based classifiers from big data. Information Sciences, 415, 319–340.

Gehan, E. A. (1965). A generalized wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika* 52, 203–224.

Gheyas, I A, & Smith, L S (2010). Feature subset selection in large dimensionality domains. Pattern Recognition, 43, 5–13.

Ghosh, A, Datta, A, & Ghosh, S (2013). Self-adaptive differential evolution for feature selection in hyperspectral image data. Applied Soft Computing, 13, 1969–1977.

Guyon, I, & Elisseeff, A (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157–1182.

Hall, M, Frank, E, Holmes, G, Pfahringer, B, Reutemann, P, & Witten, I H (2009). The weka data mining software: an update. ACM SIGKDD Explorations Newsletter, 11, 10–18.

Hancer, E, Xue, B, Zhang, M, Karaboga, D, & Akay, B (2018). Pareto front feature selection based on artificial bee colony optimization. Information Sciences, 422, 462–479.

Han, K-H, & Kim, J-H (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Transactions on Evolutionary Computation, 6, 580–593.

Hussien, A. G., Hassanien, A. E., & Houssein, E. H. (2017). Swarming behaviour of salps algorithm for predicting chemical compound activities. In *Intelligent Computing and Information Systems (ICICIS), 2017 Eighth International Conference on* (pp. 315–320). IEEE.

Heidari, A A, Mirjalili, S, Faris, H, Aljarah, I, Mafarja, M, & Chen, H (2019). Harris hawks optimization: Algorithm and applications. Future Generation Computer Systems, 97, 849–872.

Ibrahim, R A, Elaziz, M A, Ewees, A A, Selim, I M, & Lu, S (2018). Galaxy images classification using hybrid brain storm optimization with moth flame optimization. Journal of Astronomical Telescopes, Instruments, and Systems, 4, 38001.

Ibrahim, R A, Elaziz, M A, & Lu, S (2018). Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. Expert Systems with Applications, 108, 1–27.

Ibrahim, R A, Ewees, A A, Oliva, D, Elaziz, M A, & Lu, S (2018). Improved salp swarm algorithm based on particle swarm optimization for feature selection. Journal of Ambient Intelligence and Humanized Computing, 1–15.

Ibrahim, R A, Ewees, A A, Oliva, D, Elaziz, M A, & Lu, S (2019). Improved salp swarm algorithm based on particle swarm optimization for feature selection. Journal of Ambient Intelligence and Humanized Computing, 10, 3155–3169.

Karaboga, D, & Akay, B (2009). A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation, 214, 108–132.

Khan, A, & Baig, A R (2015). Multi-objective feature subset selection using non-dominated sorting genetic algorithm. Journal of Applied Research and Technology, 13, 145–159.

Kim, Y, Street, W. N., & Menczer, F. (2000). Feature selection in unsupervised learning via evolutionary search. In *KDD* (pp. 365–369).

Kiziloz, H E, Deniz, A, Dokeroglu, T, & Cosar, A (2018). Novel multiobjective tlbo algorithms for the feature subset selection problem. Neurocomputing, 306, 94–107.

Langley, P. (1994). Selection of relevant features in machine learning. In Proceedings of the AAAI Fall symposium on relevance (Vol. 184 pp. 245–271).

Lichman, M. (2013). UCI machine learning repository.

Mafarja, M, Aljarah, I, Heidari, A A, Faris, H, Fournier-Viger, P, Li, X, & Mirjalili, S (2018). Binary dragonfly optimization for feature selection using time-varying transfer functions. Knowledge-Based Systems, 161, 185–204.

Mafarja, M, Aljarah, I, Heidari, A A, Hammouri, A I, Faris, H, Ala'M, A-Z, & Mirjalili, S (2018). Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. Knowledge-Based Systems, 145, 25–45.

Mafarja, M M, & Mirjalili, S (2018). Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection. Soft Computing, 1–17.

Ma, L, Li, M, Gao, Y, Chen, T, Ma, X, & Qu, L (2017). A novel wrapper approach for feature selection in object-based image classification using polygon-based cross-validation. IEEE Geoscience and Remote Sensing Letters, 14, 409–413.

Li, S, Chen, H, Wang, M, Heidari, A A, & Mirjalili, S (2020). Slime mould algorithm: A new method for stochastic optimization. Future Generation Computer Systems, 111, 300–323.

Ma, A, Zhong, Y, He, D, & Zhang, L (2018). Multiobjective subpixel land-cover mapping. IEEE Transactions on Geoscience and Remote Sensing, 56, 422–435.

Mirjalili, S (2015). The ant lion optimizer. Advances in Engineering Software, 83, 80–98.

Mirjalili, S, Gandomi, A H, Mirjalili, S Z, Saremi, S, Faris, H, & Mirjalili, S M (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software.

Mirjalili, S, Jangir, P, & Saremi, S (2017). Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. Applied Intelligence, 46, 79–95.

Mirjalili, S, & Lewis, A (2013). S-shaped versus v-shaped transfer functions for binary particle swarm optimization. Swarm and Evolutionary Computation, 9, 1–14.

Mirjalili, S, & Lewis, A (2016). The whale optimization algorithm. Advances in Engineering Software, 95, 51–67.

Mirjalili, S, Mirjalili, S M, & Lewis, A (2014). Grey wolf optimizer. Advances in Engineering Software, 69, 46–61.

Mohemmed, A. W., & Zhang, M. (2008). Evaluation of particle swarm optimization based centroid classifier with different distance metrics. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)* (pp. 2929–2932). IEEE.

Pappa, G. L., Freitas, A. A., & Kaestner, C. A. (2002). Attribute selection with a multi-objective genetic algorithm. In: *Brazilian Symposium on Artificial Intelligence* (pp. 280–290). Springer.

Paul, S, & Das, S (2015). Simultaneous feature selection and weighting–an evolutionary multi-objective optimization approach. Pattern Recognition Letters, 65, 51–59.

Quinlan, J R (1986). Induction of decision trees. Machine Learning, 1, 81–106.

Reddy, M. J., & Kumar, D. N. (2015). Elitist-mutated multi-objective particle swarm optimization for engineering design. In *Encyclopedia of Information Science and Technology*, 3rd ed. (pp. 3534–3545). IGI Global.

Rizk-Allah, R M, Hassanien, A E, Elhoseny, M, & Gunasekaran, M (2019). A new binary salp swarm algorithm: development and application for optimization tasks. Neural Computing and Applications, 1–23.

Sarkar, S, Das, S, & Chaudhuri, S S (2017). Multi-level thresholding with a decomposition-based multi-objective evolutionary algorithm for segmenting natural and medical images. Applied Soft Computing, 50, 142–157.

Sayed, G I, Hassanien, A E, & Azar, A T (2019). Feature selection via a novel chaotic crow search algorithm. Neural Computing and Applications, 31, 171–188.

Sayed, G I, Khoriba, G, & Haggag, M H (2018). A novel chaotic salp swarm algorithm for global optimization and feature selection. Applied Intelligence, 1–20.

Schiezaro, M, & Pedrini, H (2013). Data feature selection based on artificial bee colony algorithm. EURASIP Journal on Image and Video Processing, 2013, 47.

Setiono, R, & Liu, H (1997). Neural-network feature selector. IEEE Transactions on Neural Networks, 8, 654–662.

ShenL. ChenH. YuZ. KangW. ZhangB. LiH. … LiuD. 2016 Evolving support vector machines using fruit fly optimization for medical data classificationKnowledge-Based Systems966175

Soliman, G M, Abou-El-Enien, T H, Emary, E, & Khorshid, M M (2018). A novel multi-objective moth-flame optimization algorithm for feature selection. Indian Journal of Science and Technology, 11, 1–13.

Spolaôr, N, Lorena, A C, & Diana Lee, H (2017). Feature selection via pareto multi-objective genetic algorithms. Applied Artificial Intelligence, 31, 764–791.

Tan, C J, Lim, C P, & Cheah, Y-N (2014). A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models. Neurocomputing, 125, 217–228.

Tan, K C, Teoh, E J, Yu, Q, & Goh, K (2009). A hybrid evolutionary algorithm for attribute selection in data mining. Expert Systems with Applications, 36, 8616–8630.

Tian, Y, Cheng, R, Zhang, X, Li, M, & Jin, Y (2019). Diversity assessment of multi-objective evolutionary algorithms: Performance metric and benchmark problems [research frontier]. IEEE Computational Intelligence Magazine, 14, 61–74.

Wang, J, Gao, Y, & Chen, X (2018). A novel hybrid interval prediction approach based on modified lower upper bound estimation in combination with multi-objective salp swarm algorithm for short-term load forecasting. Energies, 11, 1561.

Wang, C-M, & Huang, Y-F (2009). Evolutionary-based feature selection approaches with new criteria for data mining: A case study of credit approval data. Expert Systems with Applications, 36, 5900–5908.

WangM. ChenH. 2020 Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosisApplied Soft Computing88105946

WangM. ChenH. YangB. ZhaoX. HuL. CaiZ. … TongC. 2017 Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnosesNeurocomputing2676984

Wang, Z, Li, M, & Li, J (2015). A multi-objective evolutionary algorithm for feature selection based on mutual information with a new redundancy measure. Information Sciences, 307, 73–88.

Wang, H, Wang, W, Cui, L, Sun, H, Zhao, J, Wang, Y, & Xue, Y (2018). A hybrid multi-objective firefly algorithm for big data optimization. Applied Soft Computing, 69, 806–815.

Witten, I H, Frank, E, Hall, M A, & Pal, C J (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Wolpert, D H, & Macready, W G (1997). No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1, 67–82.

XuX. ChenH.-L. 2014 Adaptive computational chemotaxis based on field in bacterial foraging optimizationSoft Computing184797807

XuY. ChenH. LuoJ. ZhangQ. JiaoS. ZhangX. 2019 Enhanced Moth-flame optimizer with mutation strategy for global optimizationInformation Sciences492181203

Xue, B, Cervante, L, Shang, L, Browne, W N, & Zhang, M (2014). Binary pso and rough set theory for feature selection: A multi-objective filter based approach. International Journal of Computational Intelligence and Applications, 13, 1450009.

Xue, B., Fu, W., & Zhang, M. (2014). Multi-objective feature selection in classification: a differential evolution approach. In *Asia-Pacific Conference on Simulated Evolution and Learning* (pp. 516–528). Springer.

Xue, B, Zhang, M, & Browne, W N (2012). Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE Transactions on Cybernetics, 43, 1656–1671.

Xue, B, Zhang, M, & Browne, W N (2013). Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE Transactions on Cybernetics, 43, 1656–1671.

Xue, B, Zhang, M, Browne, W N, & Yao, X (2015). A survey on evolutionary computation approaches to feature selection. IEEE Transactions on Evolutionary Computation, 20, 606–626.

Yang, J., & Honavar, V. (1998). Feature subset selection using a genetic algorithm. In *Feature extraction, construction and selection* (pp. 117–136). Springer.

Yao, G, Ding, Y, Jin, Y, & Hao, K (2017). Endocrine-based coevolutionary multi-swarm for multi-objective workflow scheduling in a cloud system. Soft Computing, 21, 4309–4322.

Yu, L. & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)* (pp. 856–863).

Yu, X, Lu, Y, & Yu, X (2018). Evaluating multiobjective evolutionary algorithms using mcdm methods. Mathematical Problems in Engineering, 2018.

Zhang, X, Xu, Y, Yu, C, Heidari, A A, Li, S, Chen, H, & Li, C (2020). Gaussian mutational chaotic fruit fly-built optimization and feature selection. Expert Systems with Applications, 141, 112976.

Zhang, Y, Gong, D-W, & Cheng, J (2015). Multi-objective particle swarm optimization approach for cost-based feature selection in classification. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 14, 64–75.

Zhang, Y, Gong, D-W, Sun, X-Y, & Guo, Y-N (2017). A pso-based multi-objective multi-label feature selection method in classification. Scientific Reports, 7, 1–12.

Zhang, Q, & Li, H (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 11, 712–731.

Zhao, L, Chen, Z, Hu, Y, Min, G, & Jiang, Z (2018). Distributed feature selection for efficient economic big data analysis. IEEE Transactions on Big Data, 164–176.

ZhaoX. LiD. YangB. MaC. ZhuY. ChenH. 2014 Feature selection based on improved ant colony optimization for online detection of foreign fiber in cottonApplied Soft Computing24585596

ZhaoX. ZhangX. CaiZ. TianX. WangX. HuangY. … HuL. 2019 Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patientsComputational biology and chemistry78481490

Zhu, Y, Liang, J, Chen, J, & Ming, Z (2017). An improved nsga-iii algorithm for feature selection used in intrusion detection. Knowledge-Based Systems, 116, 74–85.

Zhu, Z, Ong, Y-S, & Dash, M (2007). Wrapper–filter feature selection algorithm using a memetic framework. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 37, 70–76.