IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 December 2023, accepted 25 January 2024, date of publication 5 February 2024, date of current version 14 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3362228

RESEARCH ARTICLE

MOBCSA: Multi-Objective Binary Cuckoo Search Algorithm for Features Selection in Bioinformatics

HUDHAIFA MOHAMMED ABDULWAHAB[1,2], S. AJITHA[2], MUFEED AHMED NAJI SAIF[3], BELAL ABDULLAH HEZAM MURSHED[4,5], AND FAHD A. GHANEM[6,7]

[1]Department of Computer Science and IT, University of Science and Technology, Taizz, Yemen
[2]Department of Computer Application, Ramaiah Institute of Technology (affiliated to VTU), Bengaluru, Karnataka 560054, India
[3]Department of Computer Applications, Sri Jayachamarajendra College of Engineering (affiliated to VTU), JSS TI Campus, Mysore, Karnataka 570006, India
[4]Department of Computer Science, College of Engineering and IT, Amran University, Amran, Yemen
[5]Department of Studies in Computer Science, Mysore University, Mysore, Karnataka 570006, India
[6]Department of Computer Science and Engineering, PES College of Engineering, Mysore University, Mandya 571401, India
[7]Department of Computer Science, College of Education-Zabid, Hodeidah University, Al Hudaydah, Yemen

Corresponding author: Hudhaifa Mohammed Abdulwahab (hudhaifa.alhemyari@gmail.com)

ABSTRACT In bioinformatics, medical diagnosis models might be significantly impacted by high-dimensional data generated by high-throughput technologies. This data includes redundant or irrelevant genes, making it challenging to identify the relevant genes from such high-dimensional data. Therefore, an effective feature selection (FS) technique is crucial to mitigate dimensionality, thereby enhancing the performance and accuracy of medical diagnosis. The Cuckoo Search Algorithm (CSA) has proven effective in gene selection, demonstrating prowess in exploitation, exploration, and convergence. However, most of the current CSA-based FS techniques deal with gene selection problems as a single objective rather than adopting a multi-objective mechanism. This article proposes the Multi-Objective Binary Cuckoo Search Algorithm (MOBCSA) for gene selection. MOBCSA extends the standard CSA by incorporating multiple objectives, including accuracy of classification and number of selected genes. MOBCSA utilizes an S-shaped transfer function for transforming the algorithm's search space from a continuous to a binary search space. MOBCSA integrates two components: an external archive to save the pareto optimal solutions attained during the search process, and an adaptive crowding distance updating mechanism integrated into the archive to maintain diversity and increase the coverage of optimal solutions. To assess MOBCSA's performance, evaluation experiments were conducted on six benchmark biomedical datasets using three different classifiers. Then, the obtained experimental results were compared against four multi-objective-based state-of-the art FS methods. The findings prove that MOBCSA surpasses the other methods in both accuracy of classification and number of selected genes, where it has obtained an average accuracy ranging from 92.79% to 98.42% and an average number of selected genes ranging from 15.67 to 27.88 for different classifiers and datasets.

INDEX TERMS Features selection, multi-objective optimization, cuckoo search algorithm, machine learning, data mining, bioinformatics.

I. INTRODUCTION

Bioinformatics has emerged as a significant research field dedicated to the analysis and interpretation of biological data, particularly in genetics and genomics [1]. It utilizes computer technology to gather, store, and analyze biological information such as DNA and amino acid sequences, as well as annotations related to these sequences. The ultimate goal of bioinformatics is to understand the complex biological processes at the molecular level and discover knowledge that can be harnessed to enhance human healthcare. Cancer remains the leading cause of death globally for both women and men,

The associate editor coordinating the review of this manuscript and approving it for publication was Deepak Mishra.

with an estimated 19.3 million newly diagnosed cases and a total of 10 million deaths recorded in 2020 only. Projections indicate that cancer will persist as a significant contributor to the mortality rate, accounting for approximately one out of every six fatalities worldwide [2]. Nonetheless, early diagnosis and treatment of cancer have the potential to mitigate the mortality rate linked to the ailment. The early detection of such a disease is crucial to increase survival rates. Machine learning (ML) techniques have demonstrated a significant role in the realm of medical diagnosis, particularly in improving cancer prediction by leveraging DNA microarray data analysis, diagnosing diverse diseases, and enabling the extraction of valuable insights from biological data. These techniques empower healthcare professionals to make informed decisions, achieve quick diagnoses, and make precise predictions [3].

Microarray data poses challenges due to its high dimensionality, the presence of high levels of noise and complexity, and the existence of irrelevant or redundant features within the data, making it arduous to extract meaningful insights and draw precise conclusions. ML techniques are not well-suited for handling such high-dimensional biomedical data [4]. To tackle this issue, FS emerges as an effective approach for selecting the most informative genes that play a pivotal role in the cancer prognosis process and mitigating the challenges posed by high-dimensional biomedical data. Features selection, also known as gene selection refers to the process of choosing a concise subset of genes that are highly relevant to a specific disease from a vast pool of genes. Hence, FS methods have a significant influence on enhancing classification accuracy, reducing learning time, and improving the overall performance of medical data analytics models [5]. Therefore, developing efficient gene selection methods is essential for accurate and reliable gene analysis.

Recently, several FS methods have been presented in the literature. These methods are categorized into filter, wrapper, and embedded-based methods [6]. Filter methods operate independently of any learning algorithm in evaluating the importance of features. In this approach, features are assessed and ranked based on statistical techniques and information-theoretic measures, and then the features with the largest scores are selected. Some of the notable recently proposed filter-based FS methods are [7], [8], [9], and [10]. On the other hand, wrapper-based techniques employ learning algorithms and the search method to identify the best solutions; some examples of the recently proposed wrapper-based FS methods are [11], [12], [13], and [14]. However, filter-based methods are fast but less accurate compared to the wrapper method. In contrast, wrapper-based methods are more accurate than filter-based methods but lead to higher computational costs. Whereas embedded methods consider the FS procedure as an integral part of the training model, in comparison to wrapper methods, embedded methods have a lower overhead. Hence, they are more conceptually complex than other methods; some examples of embedded methods are [15], [16], and [17]. The aforementioned approaches often suffer from high computational complexity, overfitting, getting trapped in local optimums, and a lack of interpretability when dealing with high-dimensional datasets.

Meta-heuristic algorithms have emerged as a powerful technique to solve complex optimization problems such as gene selection. These algorithms offer numerous advantages when dealing with the problem of gene selection, such as their effectiveness in exploring large solution spaces of gene expression data, their powerful exploration capabilities, and their ability to prevent falling into local optima. Meta-heuristic algorithms can either consider a single or multiple objective functions. The single-objective function is generally considered for optimizing a single objective, which can typically be a measure of performance or feature size. In contrast, a multi-objective-based function is considered for optimizing multiple objectives simultaneously, where trade-offs between these objectives need to be balanced. Some examples of single objective-based meta-heuristic algorithms are the dynamic salp swarm algorithm [18], monarch butterfly optimization algorithm [19], grasshopper optimization algorithm [20], binary butterfly optimization [21], binary grey wolf optimizer [22], binary whale optimization [23], binary artificial bee colony [24], and binary coyote optimization algorithm [25]. These algorithms are limited to a single objective problem (SOP). The structural nature of gene selection involves at least two contradictory objectives: reduce the size of the gene subset and increase the classification accuracy, which can be considered a multi-objective optimization problem (MOP). However, most current related studies consider only a single objective, while only a limited number of studies consider the problem of gene selection as a multi-objective [26], [27], [28].

A newly introduced nature-inspired optimization algorithm called CSA has gotten more attention due to its ability to deal with various optimization problems [29]. The basic version of CSA has proven effective in addressing several optimization problems, including gene selection. The CSA possesses several appealing features that contribute to its effectiveness as an optimization algorithm. These include its simplicity, ease of implementation, limited number of adjustable parameters, flexibility, robustness, and the ability to find optimal solutions even in high-dimensional spaces. Although the CSA has proven to be successful in handling a single objective for FS. Previous research has not extensively explored the adaptation of this algorithm for binary multi-objectives in wrapper mode for addressing the FS problem in bioinformatics. Consequently, this noticeable gap in the existing literature and the limitations of current studies to investigate such adaptations. Moreover, the inspection of the success, promising results, and attractive features of the single-objective CSA algorithm are the key motivations behind doing this research.

This article aims to propose a binary multi-objective FS method based on the CSA in wrapper mode for gene selection. This method extends the standard CSA by applying a multi-objective fitness function to optimize two conflicting objectives simultaneously: increase the accuracy of

classification and select an optimal number of genes. It utilizes an S-shaped transfer function for transforming the algorithm's search space from a continuous to a binary search space. The proposed method integrates two components: an external archive to save the pareto-optimal solutions attained during the search process. Furthermore, an adaptive crowding distance updating mechanism is integrated into the archive to maintain diversity and increase the coverage of optimal solutions. The proposed method is evaluated on several benchmark gene expression datasets in comparison to various state-of-the-art, multi-objective-based FS methods. The findings show that MOBCSA outperforms other methods in terms of both accuracy of classification and number of selected genes. To the best of our knowledge, this study is the first attempt to investigate the application of the binary multi-objective CSA for gene selection. The proposed method aims to optimize conflicting objectives such as improving classification accuracy and selecting an optimal number of genes in wrapper mode within the context of bioinformatics.

A. CONTRIBUTION

The following are the main contributions of this paper:

- A multi-objective binary CS algorithm (MOBCSA) for optimal gen selection in wrapper mode.
- Optimizing two conflicting objectives: accuracy and number of genes.
- Incorporating a non-dominated ranking and external archive into the CSA to save non-dominated Pareto optimal solutions
- Integrating an adaptive crowding distance to enhance the archive updating mechanism.
- Evaluating the performance of MOBCSA based accuracy and number of genes over three different classifiers, trained on six benchmark microarray datasets, and compared with four recent multi-objective approaches for gene selection from the literature.

B. PAPER ORGANIZATION

The structure of the article is organized as follows: Section II delivers a review of related work covering both single and multi-objective algorithms for FS. In Section III, the preliminary concepts are outlined and the problem is formulated. Section IV provides an in-depth detailed explanation of the proposed method. Section IV presents the experimental outcomes and conducts a performance evaluation. Finally, Section V concludes the article by summarizing the findings and exploring potential future research work.

II. RELATED WORK

Machine learning (ML) methods are experiencing a growing presence in the healthcare sector for the classification and diagnosis of various diseases. However, the high dimensionality of datasets poses a challenge to these methods. Therefore, gene selection becomes a crucial step in reducing data dimensionality, decreasing computational complexity, and enhancing classification accuracy. Over the last decades, several approaches have been introduced to address the challenge of selecting the most suitable subset of genes from high-dimensional microarray datasets. This section provides an overview of the literature on various current FS approaches. FS methods can be broadly categorized into three major groups: filter, wrapper, and embedded techniques [6]. Filter and wrapper approaches differ in how they evaluate a subset of features. Filter methods independently assess feature relevance without involving any learning algorithm. In this method, statistical and information-theoretic measures are used to assess and rank features, and then the highest-ranked ones are selected. Recent studies on filter-based FS methods can be discovered in the references [7], [8], [9], [10]. In contrast, wrapper-based approaches utilize learning algorithms and apply a search method to discover an optimal solution from a set of potential solutions. The wrapper interacts with the predetermined classifier to assess the quality of the features. However, wrapper-based methods offer more accurate outcomes compared to filter-based methods because they incorporate a learning model into the search process. Although wrapper methods come with greater computational costs, some recently proposed wrapper-based FS methods can be found in references [11], [12], [13], and [14]. Furthermore, embedded methods aim to combine the FS phase and classification model into a single process. Embedded techniques boast a lower computation cost compared to wrapper-based methods. However, embedded methods are inherently more intricate in their conceptual framework when compared to alternative methods. Furthermore, they demand modifications to classification models, thereby posing significant challenges to achieving higher performance. It is important to note that the aforementioned methods often face issues such as premature convergence, substantial complexity, high computational costs, and the inherent risk of becoming stuck in local optima.

Meta-heuristic algorithms have proven to be optimal for addressing the aforesaid limitations, as they have exhibited their effectiveness in tackling complex optimization problems such as gene selection. Meta-heuristic algorithms offer several advantages when tackling gene selection problems, such as their ability to efficiently explore vast solution spaces within gene expression data, their robust exploration capabilities, and the ability to avoid becoming trapped in local optima. These optimization techniques simulate the behavior of natural systems like evolution, swarm intelligence, or animal behavior. This article mainly focuses on meta-heuristic algorithms, especially those based on swarm intelligence methods. To this extent, many swarm intelligence FS methods have been proposed in the literature. For instance, the Grasshopper Optimization Algorithm (GOA) was developed by Aljarah et al. [20] to choose features and improve the parameters of the SVM classifier simultaneously. Later, Mafarja et al. [30] integrated the binary GOA with evolutionary population dynamics (EPD), and

Mafarja et al. [20] presented a binary version of GOA based on two binary transfer functions along with mutation operators to improve the exploration capabilities of the BGOA. Similarly, Hichem et al. [31] introduced a new binary GOA utilizing the Hamming distance transfer function, turned continuous variables into a binary vector of grasshoppers, and updated the positions using simple operators. GOA was enhanced to balance its exploitation and exploration abilities, as in [32], [33], and [34].

The Butterfly Optimization Algorithm (BOA) was also adapted to solve the FS problem and has shown promising results. One such adaptation is the binary version of BOA (bBOA) proposed by Arora et al. [21], which focuses on selecting the optimum feature subset for classification tasks. Later, Zhang et al. [35] improved bBOA by employing four strategies, including the differential evolution strategy (DES) and a novel initialization approach, both of which were used to reduce the randomness of bBOAs during the initialization and local search processes. Moreover, Awad et al. [36] integrated BOA with chaotic maps to enhance diversity and prevent BOA from getting stuck in the local optimal solution. Furthermore, Tubishat et al. [37] proposed the dynamic variant of BOA for FS based on the Mutation Operator for local search to mitigate the risk of becoming stuck in local optima and LSAM to enhance the solution diversity of the BOA. Some other authors modified BOA to improve its exploitation ability and prevent premature convergence [38], [39], [40]. The Gray Wolf Optimization (GWO) algorithm mimics the hunting behavior of gray wolves in nature [41]. Recently, several GWO-based methods have been adapted for solving the FS problem, including a binary version of the GWO developed by Emary et al. [42] to identify the best feature subset for classification. Multi-Strategy Ensemble GWO (MEGWO) was proposed by Tu et al. [43], and a wrapper-based GWO approach combined with a mutation operator was suggested by Abdel-Basset et al. [44]. Additionally, Hu et al. [22] presented an updated equation for GWO parameters and new transfer functions to balance exploitation and exploration capabilities for binary GWO.

Another meta-heuristic method within the realm of the principles of bio-inspired optimization is the Whale Optimization Algorithm (WOA), which draws inspiration from the intricate social behavior exhibited by humpback whales during the captivating phenomenon of bubble-net hunting [45]. The WOA has been effectively applied to the FS problem, with various approaches developed by researchers to tackle this problem. For instance, Sharawi et al. [45] proposed a wrapper-based WOA approach to select the most pertinent features for classification tasks, while Mafarja et al. [23] extended WOA by introducing two binary variations of the wrapper-based FS technique. These methods leverage the inherent capabilities of WOA to identify relevant feature subsets for improved classification performance. In a similar vein, Hussien et al. [46] took a different approach by incorporating S and V-shaped transfer functions into the conventional WOA algorithm. By doing so, they aimed to improve the algorithm's ability to solve the FS problem effectively and strike a balance between exploitation and exploration. Furthermore, Sayed et al. [47] proposed a chaotic WOA for FS, which integrated chaotic maps into the search process. This addition introduced randomness and diversity to the algorithm, enabling it to explore the solution space more effectively. Agrawal et al. [48] proposed a quantum-inspired version of WOA. This algorithm utilized quantum bit representation to enhance exploration and exploitation of the classical WOA, enabling it to effectively search for optimal feature subsets [49], [50], [51], [52], [53], [54]. Although these meta-heuristic methods have showcased their efficiency in addressing the FS problem, it is crucial to note that they have primarily focused on single-objective optimization, prioritizing either the accuracy of the classification or the reduction of the selected subset of features. However, FS is a complex problem that involves balancing at least two conflicting objectives: minimizing the number of feature subsets while maximizing the performance of the classification. To tackle this complexity, meta-heuristic algorithms offer substantial potential for tackling the complexity of the FS problem. Their ability to leverage population-based search and generate diverse solutions makes them well-suited for addressing multi-objective FS problems. By simultaneously considering multiple objectives, these algorithms can provide valuable insights into the trade-offs between classification performance and feature subset size.

In recent years, some multi-objective methods for gene selection have been introduced in the literature. One such approach is the forest optimization algorithm (FOA)-based multi-objective FS method introduced by Nouri et al. [26]. This method incorporates the concepts of grid, archive, and region-based selection to enhance gene selection performance. The authors devised two variants of the algorithm: one with a continuous representation called CMOFOA, and the other with a binary representation named BMOFOA. Another method is the binary-based version of the Harris Hawks optimization (HHO) algorithm known as MOHHO, proposed by Dabba et al. [27]. The primary aim of this method is to select the most suitable gene subset by considering multiple objectives simultaneously. Chaudhuri et al. [28] suggested a FS technique named QOMOJaya, which adopts a quasi-oppositional approach to optimize two objectives: the accuracy of the classification and the size of selected genes for microarray datasets. Dashtban et al. [55] introduced a novel binary multi-objective algorithm for gene selection in microarray data classification. Their approach extends the traditional Bat algorithm by incorporating improved formulations, innovative local search strategies, and efficient multi-objective operators. Rostami et al. [56] developed an enhanced multi-objective PSO-based FS method called MPSONC to choose an optimal feature subset. The MPSONC method consists of three primary phases: in the first phase, the original features are modeled as a graph representation. In the subsequent phase, the centralities of features are computed for all nodes in the graph. Finally, an upgraded

PSO-based search process is implemented to finalize the FS. Hancer et al. [57] presented a FS approach that utilizes a multi-objective-based artificial bee colony algorithm (MOABC) combined with a non-dominated sorting procedure and genetic operators. This method aims to achieve a set of non-dominated solutions for enhanced FS performance. Amoozegar et al. [58] proposed a multi-objective PSO algorithm for FS that incorporates a feature prioritization mechanism based on the frequency of features in the archive set. By leveraging this information, the algorithm improves the quality of the archive set and enhances the effectiveness of particle movement during the optimization process. Han et al. [59] developed a multi-objective PSO with adaptive strategies for FS that integrates the penalty boundary interaction (PBI) decomposition approach to choose optimal solutions. Furthermore, the multi-objective optimization approach is employed in various domains, such as [60], with the aim of addressing large-scale MOPs through the introduction of the M2O-CSA algorithm. This algorithm utilizes a multi-orthogonal opposition approach to improve both solution distribution and convergence. Additionally, Rizk-Allah et al. [61] introduced a framework inspired by the behavior of fruit flies to enhance the shapes of tubular linear synchronous motors. Meanwhile, Rizk-Allah et al. [62] introduced a multi-objective algorithm that leverages chaos theory to enhance the optimization process for economic-emission load dispatch. It is worth noting that the number of research efforts on multi-objective FS is comparatively lower than that in the single-objective state, as indicated by previous studies. However, these recent advancements highlight the increasing interest in developing effective multi-objective FS techniques to tackle the challenges associated with FS in complex optimization problems.

Among the various meta-heuristic algorithms, CSA has recently emerged as a promising method for solving FS problems. It was first introduced by Yang and Deb [29] to solve optimization problems, including gene selection. CSA draws inspiration from the behavior of cuckoo birds in the reproductive process. The algorithm mimics the concept of obligatory brood parasitism, wherein cuckoo birds lay their eggs in the nests of other similar species, known as host birds, and rely on the host birds to raise their chicks. Compared to other meta-heuristic algorithms, CSA offers several advantages, including superior exploration and exploitation, rapid convergence, fewer parameters, avoidance of local optima, computational efficiency, and ease of implementation.

Over the years, numerous researchers have proposed approaches to address the FS problem using different variants of CSA. For instance, Rodrigues et al. [63] proposed a binary CSA, which transforms continuous variables into binary form for FS. Gunavathi et al. [64] introduced a CSA specifically for FS in a microarray dataset for cancer classification, using both T and F statistics for feature ranking and a KNN algorithm for the fitness function. Aziz et al. [65] presented a modified CSA that incorporates rough sets to handle high-dimensional data through FS. Their fitness function considers the number of features in the reduced set and the quality of the classification. Similarly, Alia et al. [66] developed an enhanced version of Binary CSA, which introduced a novel objective function based on frequent values and Rough Set Theory (RST). Additionally, they made improvements to the initialization and update mechanisms, resulting in enhanced convergence efficiency. Kumar et al. [67] introduced a binary version of CSA that aims to select an optimum subset of features for online textual content sentiment analysis, while Sudha et al. [68] developed an enhanced CSA to determine the optimal features for the classification of breast cancer. Salesi et al. [69] extended the binary CSA by embedding a pseudo-binary mutation neighborhood search. This enhancement aimed to improve the effectiveness of the binary CSA in handling FS problems within the biomedical domain. Pandey et al. [70] introduced a binary binomial CSA specifically designed to identify the optimal subset of features by leveraging the binomial distribution and CSA principles. Alzaqebah et al. [71] proposed a CSA based on adaptive memory to enhance the FS procedure and keep the history of previous solutions. Wang et al. [72] developed the FS method based on the CSA by applying three approaches: Lévy flight, chaotic maps, and the elite preservation approach with uniform mutation to enhance the FS process. Hamidzadeh et al. [73] specifically created a chaotic cuckoo optimization algorithm with levy flight for the purpose of selecting the best feature subspace for the classification task. Rabia et al. [74] developed a deep-learning model designed for the classification of fish images. To bolster the exploratory prowess of the CS algorithm, the authors integrated it with the genetic algorithm and enhanced the overall performance of their deep-learning model. Khurram et al. [75] proposed a particle-swarm CS optimization algorithm for training deep neural networks for depression detection tasks. Rabia et al. [76] introduced a hybrid method using CS and the artificial bee colony (ABC) algorithm for FS in cancer classification using microarray data. Furthermore, other versions of CSA [77], [78] have been presented within hybrid models as effective solutions for solving FS problems.

Although CSA has demonstrated its effectiveness in solving various optimization problems, including gene selection. Typically, gene selection involves multiple conflicting objectives, such as enhancing classification accuracy and minimizing the number of genes. However, current related works based on CSA for gene selection mainly focus on optimizing a single objective, such as increasing the accuracy of classification or decreasing the number of genes. There is a tradeoff between these two conflicting objectives that needs to be considered simultaneously. To effectively handle this issue, this study extends the standard CSA, considering multiple objectives for gene selection to simultaneously enhance the classification accuracy and minimize the number of genes. The primary objective of this article is to propose a binary multi-objective gene selection algorithm reformed on the original CSA in wrapper mode.

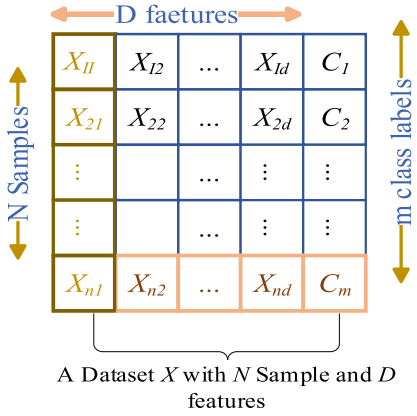


FIGURE 1. Representation of a dataset.

III. PRELIMINARIES AND PROBLEM FORMULATION

This section initially presents the basic concepts and mathematical formulation of FS. Moreover, the basics of the CSA are highlighted. Finally, the basic concepts of multi-objective optimization are presented and the mathematical formulation is introduced.

A. FEATURE SELECTION

Feature selection involves the meticulous process of selecting and identifying the most pertinent features within a high-dimensional dataset. This is accomplished by eliminating redundant, irrelevant, or noisy features without significantly losing information or affecting learning performance. In building an ML model, only a few subsets of features within the dataset are useful for creating an effective model, while the remaining features are often redundant or irrelevant. Neglecting to remove these unnecessary features may adversely affect the model's overall performance and accuracy. As a result, it is essential to recognize and select the most suitable features from the data and remove any irrelevant features. This can be accomplished through FS and machine learning methods.

To get a clear idea on FS problem, let us consider X which represents the dataset as a matrix of dimensions $N * D$, where N signifies the number of samples each having a set of D features $Fet = [f_1, f_2, \ldots, f_{\mathrm{D}}] \in R^{\mathrm{D}}$. X_{ij} represents the value of j^{th} feature in the i^{th} sample, where j ranges from 1 to D and i range from 1 to N. The dataset also includes a subset of classes C which comprises m distinct class labels. Let $C = [c_1, c_2, \ldots, c_{\mathrm{m}}] \in \mathrm{R}^{\mathrm{m}}$, where m represents the number of distinct class labels. The samples are then distributed among a set of classes, where $k = 1, 2 \ldots, m$. So, samples within the same class exhibit similar attributes, while those in different classes possess dissimilar attributes. Figure 1 provides a visual representation of a dataset.

The main task of FS method is to identify a subset of d features, where $(d < D)$ in the global feature space Fet. The subset of selected features from the original features space denoted by $S \in R^d$, which aims to optimize a given criterion by deriving an effective mapping function from the input dataset X to the target class labels C. In the context of data classification, the objective of FS method is to select a subset $S \subseteq Fet$ of features that result in the highest classification accuracy. To represent solutions to the FS problem, binary coding is used with each feature represented by a binary digit indicating its inclusion or exclusion in the subset. The following expression describes the binary coding used to solve the FS problem:

$$X_I = (X_{I1}, X_{I2}, \ldots, X_{Id})\, X_{ij} \in \{0, 1\} \tag{1}$$

where $X_{ij} = 1$ indicates that the j^{th} the feature has been chosen for inclusion in the selected subset; otherwise, if $X_{ij} = 0$ indicates that it is not included.

B. BASIC CUCKOO SEARCH ALGORITHM (CSA)

The CSA is a swarm intelligence-based optimization method that draws its inspiration from nature. First introduced by Yang and Deb [29], it was meant to tackle a wide range of optimization problems in different fields. The algorithm emulates the concept of obligatory brood parasitism in the behavior of certain cuckoo birds. In this behavior, cuckoos lay their eggs in nests of similar species known as host birds and rely on them to care for their offspring. In CSA, the candidate solutions are represented as nests, and their quality is assessed using an objective function. The algorithm maintains a population of candidate solutions and generates new solutions through a process called Lévy flight. It makes random jumps within the search space depending on the information from the current best solution. The Lévy flight is a stochastic walk model that simulates the movement patterns observed in certain animals during their search for food. In the context of the CSA algorithm, this concept is leveraged to generate new candidate solutions by randomly adjusting the current best solution using a step size determined by a heavy-tailed Lévy distribution. By replacing the poorest solution in the population with a modified version of the current best solution.

Additionally, the CSA includes a mechanism for nest abandonment, which simulates the behavior of some cuckoo birds abandoning their nests and creating new ones. In the algorithm, a solution may be replaced with a new random solution with a certain probability. This helps CSA foster diversity within the solution set and mitigates the risk of becoming trapped in local optima. By iteratively applying these mechanisms, the CSA can effectively explore the search space and converge on the best solution for the optimization problem. CSA is effective for solving a diversity of optimization problems, including global optimization, engineering design optimization, and ML optimization. The CSA offers several advantages, such as ease of implementation, minimal parameter tuning requirements, and fast convergence rates. It achieves superior performance when compared to other existing algorithms. The following are the basic components of the CSA:

TABLE 1. Meaning of the symbols used.

Symbol	Meaning
$\boldsymbol{X}$	The dataset with dimensions $N \times D$
$\boldsymbol{N}$	Number of instances in $\boldsymbol{X}$
$\boldsymbol{D}$	Total number of features in $\boldsymbol{X}$
$\boldsymbol{m}$	Number of class labels
$\boldsymbol{Fet}$	Original set of features
$[\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_D]$	D features
$\boldsymbol{S}$	A subset of selected features from Fet
$\boldsymbol{d}$	Number of selected features from Fet
$\boldsymbol{X}_{ij}$	value of j^{th} feature in the i^{th} instances
$\boldsymbol{C}$	Number of classes
$\boldsymbol{i}$	i^{th} instances
$\boldsymbol{j}$	j^{th} feature
$\boldsymbol{F(x)}$	Objective function
$\boldsymbol{k}$	Number of objective functions.
AvgClassAcc	Average classification accuracy
AvgSelFeat	The average number of selected features

1) OBJECTIVE FUNCTION

The superiority of the solution is relational to the objective function's value, which represents both classification accuracy and size of chosen features. It can be formulated as a multi-objective function, as represented by Eq. (2):

$$F(x) = [F_1(x), F_2(x)] \tag{2}$$

where x represents the feature subset, $F_1(x)$ represents the accuracy of the classification, and $F_2(x)$ denotes the size of chosen features.

2) THE NEST

The population size in the FS problem is determined by a specific number of nests that correspond to the size of solution population. Each host nest defines as a sample (solution), where each sample has egg f (feature), or a set of Fet features $Fet = [f_1, f_2, \ldots, f_D]$. The CSA begins with N solutions where each of them represents a nest, by randomly placing the population of nests in the search space.

In FS problem, a nest represents a possible solution and is characterized as a binary vector with a length of N, where N corresponds to the number of features in the dataset. Each element of the vector signifies the selection status of a feature, with a binary value of "1" indicating that the feature is selected, while "0" denotes its non-selection.

3) THE EGG

In context of the FS problem, the eggs in a nest represent a feature f_i and a solution may contain f eggs i.e., $Fet = [f_1, f_2, \ldots, f_D]$.

$$\begin{array}{c} \\ Z_1 \\ Z_2 \\ \\ Z_N \end{array} \begin{array}{c} \begin{array}{cccc} f_1 & f_2 & & f_D \end{array} \\ \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1d} \\ X_{21} & X_{22} & \ldots & X_{2d} \\ \vdots & & \vdots & \vdots \\ X_{N1} & X_{N2} & \cdots & X_{Nd} \end{bmatrix} \end{array}$$

FIGURE 2. Search space.

4) CUCKOO

The cuckoo represents the current solution for the FS problem and also can be described as a nest with a random feature subset.

5) LEVY FLIGHT PROBABILITY

The levy flight probability is employed to control the exploration and exploitation of the search space. Cuckoo birds perform Lévy flights, which are random walks with step sizes that follow a heavy-tailed distribution. This allows the birds to search a large space efficiently. The mathematical formulation of Lévy flights is discussed in Section IV-G.

6) SEARCH SPACE

In CSA, the search space is visualized as a binary array where each element represents the selection status of a feature. When a feature is selected, its corresponding element is assigned a value of 1, while a value of 0 indicates that the feature is not chosen. The search space encompasses a total of 2^n elements, where n denotes the number of features available in the dataset. Figure 2 illustrates a graphical representation of the search space, where Z_i denotes a solution in each row. Each entry in the matrix x_{ij} holds a continuous value, which is then converted into a binary value.

7) STOPPING CRITERIA

The stopping criteria play a pivotal role in establishing the conditions that determine the precise moment when the search process should come to an end. These criteria serve as essential indicators to guide the termination of the search process at the appropriate time.

C. THE MATHEMATICAL FORMULATION OF MULTI-OBJECTIVE OPTIMIZATION PROBLEM (MOP)

Multi-objective optimization problems (MOPs) arise when there is a necessity to strike a balance between two competing objectives that typically contradict one another, necessitating the identification of the optimal solution. MOP involves multiple objective functions that aim to simultaneously either maximize or minimize multiple conflicting objectives.

In general, a multi-objective minimization problem can be mathematically formulated as follows:

If there are a set of k objectives, then the minimizing of the objectives is expressed as:

$$\begin{aligned} &\textit{minimize } F(x) = [F_1(x), F_2(x), \ldots, F_k(x)] \\ &\textit{Subject to} \quad g_i(x) \leq 0, i = 1, 2, \ldots k \\ &\qquad\qquad h_i(x) = 0, i = 1, 2, \ldots l \end{aligned} \tag{3}$$

where $F_1(x), F_2(x), \ldots, F_k(x)$ are K conflicting objective functions associated with vector x, while x represents a set of decision variables, K denotes the number of objective functions to be minimized, and $g_K(x)$ and $h_K(x)$ correspond to constraint functions. As a result, the problem of FS can be formulated as an optimization problem, aiming to discover the subset of features that maximizes a specific criterion while simultaneously minimizing another.

The FS problem is encoded as a MOP that needs to balance two conflicting objectives: maximize the accuracy of the classification and minimize the size of features. To achieve this, we define a multi-objective function $F(x)$ to optimize multiple criteria. The primary criterion is classification accuracy, which assesses the ability of the selected features to predict the class labels of the data accurately. Another important criterion is the size of the feature subset, which quantifies the number of selected features.

The FS problem can be represented mathematically as Eq. (4):

$$maxF\ (x) = criterion_1\ (x) - criterion_2(x) \tag{4}$$

subject to $S \subseteq Fet$, where S is the selected subset of features and Fet represents the set of all features available.

This can be further extended to a MOP in Eq. (5), where multiple criteria need to be simultaneously optimized. The multi-objective formulation:

$$maxF\ (x) = (criterion_1\ (x)\ , criterion_2(x), \ldots) \tag{5}$$

subject to $S \subseteq Fet$, where S is the selected subset of features and Fet is the original features.

In the aforementioned multi-objective formulation, the objective is to identify a trade-off solution that strikes a balance between multiple criteria. These criteria encompass maximizing the accuracy of the classification while simultaneously minimizing size of feature subset.

In MOP, the goodness of a solution is clarified by the trade-offs between the conflicting objectives namely pareto optimal solutions. For instance, let u and v two candidate solutions of the aforesaid K-objective minimization problem. If the following conditions are met, it can be said that u dominates v or u is better than v (denoted by $u \prec v$):

$$\begin{aligned} &\forall i : f_i\ (u) \leq or\ is\ not\ worse\ than\ f_i\ (v)\ \ and \\ &\exists j : f_i\ (u) < or\ is\ strictly\ better\ than\ f_{\ i}\ (v)\ , \\ &where\ i, j \in \{1, 2, \ldots, K\}\ . \end{aligned}$$

where k represent the number of objective functions. The solution u is referred to as the pareto-optimal solution for a given problem if it is not dominated by any other possible solution. In other words, there is no other solution that outperforms u in all objective functions.

IV. PROPOSED METHOD

The problem of FS is often viewed as a MOP with two primary objectives: maximizing accuracy of classification and minimizing size of features. CSA has been applied successfully to address the FS problem; however, the majority of current CSA-based FS techniques treat FS as a single objective. The binary version of the multi-objective CSA for FS in wrapper mode remains a research challenge. To tackle this problem, we have extended the standard CSA, considering binary multi-objectives for FS in bioinformatics. The proposed methodology includes four phases, as illustrated in Figure 4: (a) model representation; (b) binarization and encoding methods; (c) multi-objective binary CSA for FS in wrapper mode; and (d) classification and validation. In the subsequent subsection, a detailed presentation is given for each component.

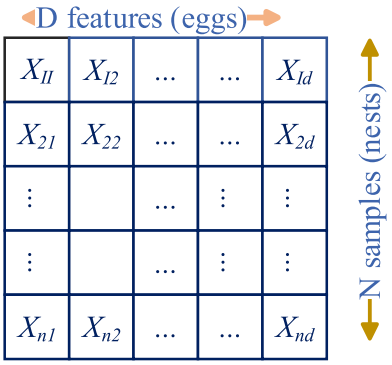


FIGURE 3. Model representation.

A. MODEL REPRESENTATION

The proposed MOBCSA technique represents the dataset a two-dimensional matrix with a size of $N * D$. Here, N represents number of samples or nests (rows), each associated with a set of D features. The primary set of features is represented by $Fet = [f_1, f_2, \ldots, f_D]$. Hence, the problem of FS can be defined as the selection of d features, where d is either less than or equal to D, from the Fet set. The selection process is guided by the quality of objective function $F(x)$ with respect to the k^{th} objectives. A schematic diagram of how the population is represented is illustrated in Figure 3.

B. BINARIZATION METHODS AND ENCODING FUNCTION

The original CSA was initially designed to tackle optimization problems within continuous search spaces where each individual is represented by a floating-point position vector. However, the FS problem is a binary one and requires modeling as a D-dimensional binary representation, where D represents the number of features. Thus, a conversion from the continuous search space to the corresponding binary bit string [1, 0] is necessary, and continuous values must be transformed. This transformation is accomplished by applying the Sigmoidal (S-shaped) transfer function, which maps vectors from the continuous search space into the corresponding binary search space. This research mainly employs the S-shaped function calculated as given in Eq. (6) [79]. The resulting binary vector is then evaluated by the fitness

function, as described in Section IV. The binary encoding mechanism is expected to outperform the continuous method since a search space is limited to only two values (0, 1), and binary operators are easier to handle than continuous operators and enable the algorithm to effectively manage and manipulate the FS process.

$$T\left(x_i^j(t)\right) = \frac{1}{1 + e^{-x_i^j(t)}} \tag{6}$$

where x_i^j represents i^{th} solution in dimension j^{th} at iteration t. Additionally, the notation $T\left(x_i^j(t)\right)$ define the value of the probability obtained by applying the S-shaped transfer function. Then, the probability value result from Eq.2 is then compared with a threshold value to determine the binary value as mentioned in Eq. (7).

$$x_i^j(t+1) = \begin{cases} 1, & if rand \leq T\left(x_i^j(t)\right) \\ 0, , & otherwise \end{cases} \tag{7}$$

The notation $x_i^j(t+1)$ represents new i^{th} solution (nest) in j^{th} dimension (feature) at iteration $(t+1)$. Here, i ranges from 1 to N, representing the index of the solution, and j ranges from 1 to D, denoting the index of the feature. The function $T\left(x_i^j(t)\right)$ represents the corresponding value calculated using Eq. (6). Additionally, the variable *rand* represents a random number between [1, 0], where each 1 denotes a selected feature, while the value of 0 signifies that it is not selected.

To represent the solution for the problem X a binary bit-string is utilized that is computed by using as given in Eq. (8). This bit-string acts as a symbolic representation for sets of feature subsets.

$$X = x_1, x_2, \ldots, x_D, x_j \in 0, 1\} \tag{8}$$

Figure 5 visually depicts a graphical representation of the encoding procedure of the binary CSA, specifically designed to tackle the FS problem. During the initialization of each nest, a distinct binary string is generated where each bit corresponds to a different feature. In this representation, a bit with a value of 1 implies the selection of the corresponding feature, while the value of 0 indicates that the feature is not selected.

C. THE MULTI-OBJECTIVE BINARY CSA FOR FS IN WRAPPER MODE

This section delves into the detailed description of our adapted multi-objective binary CSA for FS problem that considers multiple conflicting objectives.

The standard CSA follows three key rules, which are summarized as follows:

1. Each cuckoo employs random selection process to choose a nest and deposits a single egg into it.
2. The quality of the eggs in each nest is evaluated by objective function $F(x)$, then the nests with the highest quality eggs are chosen to be carried forward to the next generation.
3. The number of available nests remains constant throughout the process. A host bird may discover an alien egg in its nest with a probability $p_a \in [0, 1]$. In this case, the host bird has two options: it can either discard the foreign egg or abandon its current nest and construct a new one.

In this article, the standard CSA is adapted to handle the binary multi-objectives for FS. The basic CSA is reformulated for FS and extended its functionality to deal with different objectives. The following adaptations are added:

1. The first and third rules of the standard CSA were the only ones modified to tackle the multi-objective FS problem. The modifications are outlined as follows:
   - Each cuckoo lays a set of m eggs simultaneously, which are randomly placed in a selected nest. Each egg i, where $i \in (1, 2, 3, \ldots, K)$, corresponds to the solution for the K^{th} objective.
   - The nests with the highest quality eggs are selected to be continued to the next generation.
   - The possibility of a nest being detected by the host is determined by $p_a \in [0, 1]$. If a nest is detected, a fresh nest is constructed with m eggs according to the similarities or differences of the eggs.
2. The S-shaped transfer function was utilized to convert the algorithm's search space from continuous to discrete binary.
3. The fitness function has been modified to assess the quality of solutions based on multiple objectives, such as *Acuraccy* and *Numberoffeatures*.
4. The Pareto dominance concept is utilized to identify non-dominated solutions, ensuring that no other feasible solution dominates them with respect to all objectives.
5. To maintain a record of all non-dominated solution sets obtained so far during the search process, a pareto archive is used.
6. To increase the coverage of optimal solutions across all objectives and preserve diversity in the archive an adaptive crowding distance has been incorporated.

Based on the above rules and integrated components, the basic steps of the MOBCSA are described as follows:

D. DEFINITION OF INPUT PARAMETERS

The MOBCSA algorithm initially starts with defining input parameters. These parameters typically include size of population (N), which specifies number of nests in the algorithm's population. A fraction p_α representing the discovery rate of solutions (i.e., the worst nests that are identified for rejection and replacement). The maximum number of iterations I_{max}, sets an upper limit on the algorithm's execution. Probability of abandonment P_b, controls the likelihood of nests being abandoned and replaced. Finally, the objective functions $F_1(x)$ and $F_2(X)$, and set of original features *Fet*. One advantage of the MOBCSA compared to other population-based

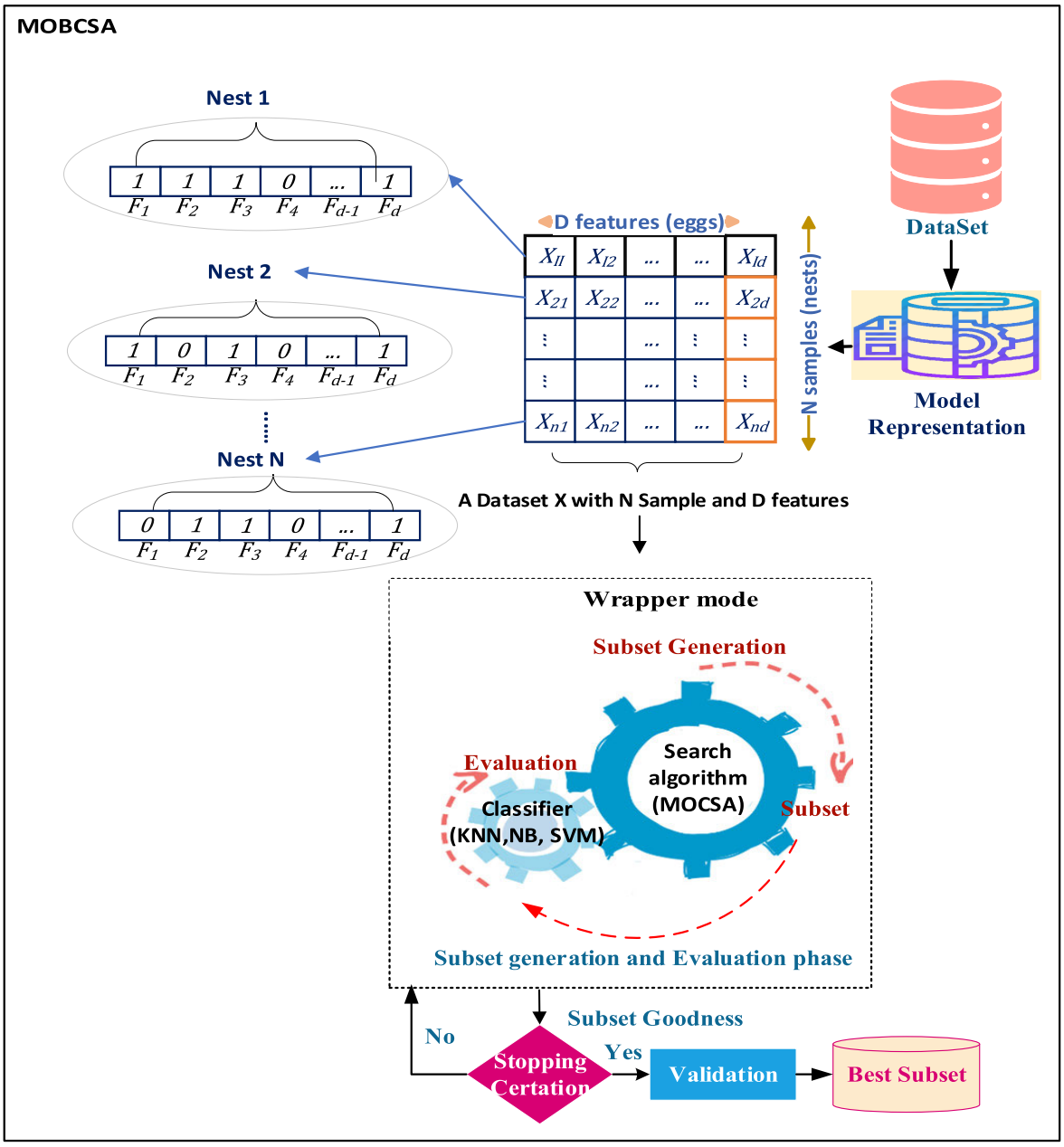


FIGURE 4. Proposed architecture of MOBCSA.

metaheuristics is its requirement for fewer parameters during configuration, making it simpler to implement and operate.

E. INITIALIZATION OF POPULATION

In the initialization phase, the proposed algorithm begins by randomly generating a set of solutions in order to form the initial population. Each solution in a population represents a subset of features. To facilitate the representation of these feature subsets, the positions of individual features are encoded as binary values. This encoding strategy, as outlined in Section IV-B, transforms the feature positions into binary feature vectors, where every element can take on a value of either 0 or 1. The resulting binary feature vectors are organized into a two-dimensional matrix with dimensions $N * D$. Here, N refers to the number of samples or nests, which can be viewed as the rows of the matrix. Each sample or nest is associated with a set of D features, represented by the columns of the matrix. This matrix-based representation allows for efficient handling and manipulation of feature subsets throughout the optimization process. By adopting this initialization scheme, the algorithm establishes an initial population encompassing diverse subsets of features, paving the

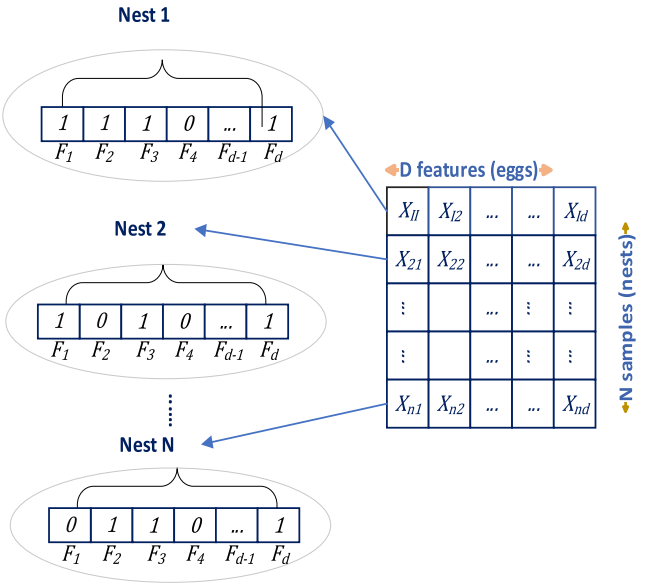


FIGURE 5. Binary representation.

way for subsequent exploration and refinement, enabling the algorithm to find optimum solutions for the multi-objective problem at hand.

F. FITNESS FUNCTION

In the multi-objective FS problem, the primary objective is to identify an optimal set of features that ensures high accuracy of the classification. To achieve this goal, we define two conflicting objectives: maximizing the accuracy of classification represented by the first fitness function $F_1(x)$ and minimizing number of features represented by the second fitness function $F_2(x)$. The classification accuracy is calculated by the following formula:

$$Acuraccy = (\frac{1}{n}\sum_{l=1}^{n}\frac{N_{cor}}{N_{All}}) \times 100 \tag{9}$$

The notation N_{Cor} signifies number of correctly predicted test samples, while N_{All} denotes a total number of instances in datasets and n symbolizes the total number of runs. By summing the accuracy over multiple runs, we obtain a more stable and reliable estimate of the algorithm's true performance.

The second fitness function minimizing number of features. It can be calculated using the following formula:

$$Number\ of\ features = \sum_{i=1}^{D} x_i \tag{10}$$

where x_i represents the value of the i^{th} value in individual X, and D denotes a total number of original features.

G. GENERATE NEW SOLUTIONS WITH LEVY FLIGHT

To further enhance the MOBCSA algorithm's capability to discover optimal solutions in subsequent generations, an improvement called Lévy flights is incorporated. By utilizing Lévy flights, the algorithm gains an increased capacity for exploration and a heightened ability to perform global searches. Given a current solution X_i^t representing the host nest i at iteration t, a new solution X_i^{t+1} can be generated by updating the previous solution using the idea of Levy flights according to Eq. (11). This operation introduces a stochastic component to the algorithm, allowing for long-range jumps within the search space. By incorporating this technique, the algorithm exhibits an enhanced capability to effectively explore uncharted regions within the solution space, thereby increasing the likelihood of uncovering superior solutions that were previously unknown. This mechanism strikes a delicate balance between exploitation and exploration, enhancing the algorithm's ability to navigate the search landscape and find high-quality solutions.

$$X_i^{t+1} = X_i^t + \alpha \oplus Levy\left(\lambda\right) \tag{11}$$

The $\alpha > 0$ represents the step size which is calculated according to the scale size of the problem at hand. The symbol $\oplus$ is a mathematical notation that denotes the element-wise multiplication operation. The Lévy(λ) distribution is employed to introduce random walks via Lévy flights, with the step sizes calculated using Eq. (13). To handle the diversity in solution quality, an additional equation such as Eq. (12) can be incorporated into the algorithm.

$$\alpha = \alpha_0(x_j^t - x_i^t)\left(\lambda\right) \tag{12}$$

The constant α_o is a predetermined value and $(x_j^t - x_i^t)$ represents the difference between two random solutions. The Lévy flights enable the algorithm to perform random walks, with the step sizes being drawn from a Lévy distribution. The calculation of the step size is as follows:

$$Lévy \sim u = t^{-1-\lambda}, (0 < \lambda \leq 2)\left(\lambda\right) \tag{13}$$

The Lévy distribution is employed as a statistical distribution that describes the step lengths in a random walk process. This mechanism plays a pivotal role in augmenting the exploration capabilities of the algorithm. It is renowned for its distinctive characteristics, including an infinite variance and means, as well as a power-law step-length distribution with a heavy tail probability. This heavy-tailed characteristic of the Lévy distribution enables the algorithm to escape local optima and facilitate the discovery of novel and potentially optimal solutions. By incorporating this distribution into the algorithm, it becomes more capable of taking large and infrequent steps, allowing for the exploration of distant and potentially promising regions in the search space. Overall, the utilization of the Lévy distribution increases the algorithm's capabilities to effectively explore the search space and improves its overall search performance.

H. NON-DOMINATED SORTING

Fast Non-Dominated Sorting (FNS) [80] is an efficient technique designed specifically for sorting solutions in MOP

Algorithm 1 MOBCSA

Input dataset (X, C), where X is the dataset and C Set of classes for the dataset and control parameters $p_a \leftarrow$ probability $p_a \in [0, 1]$, $Max_{itr} \leftarrow$ maximum number of iterations and P $\leftarrow$ population size.
Output f_{best} optimal features set
Initialization
    t $\leftarrow$ 0, counter initialization
    For i = 1: $i \leq N$ **do**
        $P_t \leftarrow$ generate population of N candidate solutions x_i^t $(i = 1, 2, \ldots, n)$.
        Calculate $Fitness\_value_i^t \leftarrow f(x_i^t)$ (the objectives functions i.e., accuracy and number of features for each solution x_i based on equations (9) and (10)).
    End for
    Perform non-dominated sorting and crowding distance calculation on the population to identify the best solutions.
    $A_t \leftarrow$ initial pareto archive set, then add the non-dominated solutions to the external archive
    while (t < Max_{itr}) or (stop criterion)
        $P_{t+1} \leftarrow \{\}$
        For each $solution X_t^i i \leq n$ $in P_t$ **do**
            generate a new solution x_i^{t+1} by Lévy flights Eq. (11) and Eq. (13)
            Convert to binary using equations (6) and (7)
            Evaluate fitness function $Fitness\_value_i^t \leftarrow f(x_i^t)$ Equations (9) and (10)
            if (new solutionX_i^{t+1} dominates X_i^t) in the objective space **Then**
                Add the new solution X_i^{t+1} to P_{t+1}
                Else
                previous solution X_i^t is added
            End if
            $A_t \leftarrow$ update the Pareto archive concerning non-dominated solutions and crowding distance
        End for
        For each solution X_i in P_{t+1}
            If new_i dominates X_i
                $X_i = new_i$ // select the worse solution with $p_a \in [0, 1]$ and replaced them for the best solution.
            End if
        End for
        Find the non-dominated solutions
        $A_t \leftarrow$ update the Pareto archive concerning the non-dominated solution
        $t = t + 1$
    End while
return the best-selected features from A_{t-1}

depending on their dominance levels. In a multi-objective optimization scenario where multiple conflicting objectives are considered, the solution is considered non-dominated if no other solution outperforms it. The FNS algorithm is specifically designed to quickly identify these non-dominated solutions within a population without comparing every solution. The algorithm begins by assigning a rank to each solution according to its level of dominance. In this case, solutions that are not dominated by any other solution are assigned to level one. While solutions exclusively dominated by level one solutions are assigned to the second level, this iterative procedure continues until every solution is assigned a level of dominance. The result of this non-dominated sorting is a collection of levels where the solutions within each level are mutually non-dominated, and solutions in higher levels dominate those in lower levels. This collection of levels is commonly known as the Pareto front or Pareto set, representing the best compromise solutions that achieve the optimal balance among the conflicting objectives in the MOP.

Assuming that a total of K objective functions are considered in a given MOP. Let $F_k(x_i)$ denote the objective function value of i^{th} features for K^{th} objectives function. In the context of single-objective optimization or a simple representation of a CSA with only one egg existing per nest, it is easy to compare the superiority of different nests based on their objective function values. However, in the case of MOP, where nests contain multiple eggs, comparing and ranking solutions according to objective function values becomes a complex task and may give a false ranking, especially when objectives are conflicting. In such cases, the solutions are categorized into two groups using the Pareto dominance condition, instead of being ranked to determine the best and

worst solutions. Pareto optimal solutions allow us to compare two solutions in a multi-objective space and show the best equilibrium state connecting the given objectives.

I. EXTERNAL ARCHIVE AND CROWDING DISTANCE UPDATING MECHANISM

The proposed method incorporates an external pareto archive as a key component to store the non-dominated solutions discovered during the search process. An external archive is a repository that stores the best solutions found during the optimization process. It serves as a reference for the optimization algorithm to guide its search toward better solutions. The external archive contains a set of non-dominated solutions, also known as Pareto front or Pareto set, which represents the optimal trade-off between the conflicting objectives in the problem. The solutions contained in the external archive are non-dominated, indicating that no other solutions within the archive can achieve better performance in one objective without sacrificing the performance of at least one other objective.

A crowding distance-based method is adopted for the external archive to update it during the optimization process and ensure that it always has the best solutions found so far. Crowding distance is defined as a measure used in MOP to preserve diversity within the solution population. It serves as a metric to quantify the proximity between neighboring solutions in the objective space, providing valued insights into their proximity and distribution [80]. The crowding distance of a solution is evaluated by calculating the difference between the values of its neighboring solutions in each objective dimension.

The solution exhibiting the highest crowding distance is selected as the next solution to be added to the population. This strategy helps to ensure that the external archive encompasses areas of the objective space that may be underrepresented by the existing solutions, ultimately facilitating a comprehensive exploration of a solution landscape. The crowding distance is computed using the following formula:

$$CD_{ij} = \frac{F_j^{i+1} - F_j^{i-1}}{F_j^{max} - F_j^{min}} \tag{14}$$

where, F_j^{max} and F_j^{min} define maximum and minimum values of the j^{th} objective function. More details can be found in [80]. The general pseudocode of the MOBCSA is given in Algorithm 1.

J. EVALUATION

The proposed algorithm operates as a wrapper approach, where it relies on a learning algorithm in the evaluation stage to evaluate the classification performance of the selected feature subset. So, in the evaluation process, we employed three distinct classifiers, including Naïve Bayes (NB), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), to assess the efficiency of the proposed method.

V. EXPERIMENTAL RESULTS

This section aims to evaluate the efficiency of the proposed MOBCSA for FS. The subsequent subsections present various aspects: the utilized datasets, the employed classifiers, the experimental setup, the comparison methods, the parameter settings, the evaluation metrics, and the experimental findings and discussions.

A. DATASETS

In this section, experiments are carried out on six distinct microarray datasets with diverse characteristics to exhibit the efficiency and robustness of the proposed MOBCSA. These microarray datasets are related to various diseases, including:

1) SRBCT

Small Round Blue Cell Tumors (SRBCT) is a microarray dataset comprises of 83 samples with 2,308 features or genes related to four distinct types of pediatric cancers, namely Burkitt's lymphoma (BL), rhabdomyosarcoma (RMS), neuroblastoma (NB), and Ewing's sarcoma (EWS). The gene expression data is used to classify the tumor type.

2) PROSTATE TUMOR

It is a gene expression dataset consisting of 10,509 gene expression features and 102 samples, including 52 cancer samples and 50 non-cancer samples. The expression levels of genes are analyzed to discover features that can differentiate between tumor and non-tumor samples.

3) LUNG CANCER

Defined as a gene expression dataset including genes related to lung cancer consisting of 203 samples with 12600 features of lung cancer tumors and non-tumors. The data from this dataset is employed to differentiate between five distinct types of lung tumors, namely normal lung (NL), squamous cell carcinoma (SQ), adenocarcinoma (AD), small cell lung cancer (SMCL), and pulmonary carcinoid (COID).

4) LEUKEMIA

It is a gene expression dataset that comprises 72 samples taken from leukemia patients, which includes 47 samples of acute lymphoblastic leukemia (ALL) and 25 samples of acute myeloid leukemia (AML). This dataset is employed to differentiate between ALL and AML based on their gene expression profiles.

5) COLON TUMOR

It is a gene expression dataset consisting of 62 samples of colon tumors and normal colon tissues. This dataset is utilized to categorize the samples as either tumor or normal tissue based on their gene expression profiles.

6) DLBCL

Diffuse Large B-Cell Lymphoma (DLBCL) is a gene expression dataset consisting of 77 samples of DLBCL tumors and

normal tissues. The task is to figure out whether the sample of tissue is from a tumor or normal tissue, according to the gene expression data.

These datasets are available at the Bioinformatics Laboratory, Faculty of Computer and Information Science, University of Ljubljana [81]. Table 2 provides a summary of each dataset's characteristics, such as the number of samples, genes, and classes. As illustrated in Table 2, the prostate tumor dataset is an example of a dataset that has a high dimensionality of features with a small number of samples. The Leukemia, DLBCL, Colon, and Prostate Tumor datasets are designed for the tasks of binary classification that involve detecting cancer types. In contrast, the lung cancer and SRBCT datasets are used for multi-class tumor classification problems. The selected datasets have varying numbers of genes ranging from 2000 to 12600 utilized to demonstrate the proposed technique's ability to handle different levels of dimensionality.

B. COMPARISON ALGORITHMS AND EXPERIMENTAL SETUP

In this section, the performance and effectiveness of the proposed MOBCSA are assessed in comparison to four other state-of-the-art multi-objective optimization-based methods for gene selection from the literature. These methods are MPSONC [56], MOBHHO [27], MOABC [57], and MOPS [58]. In these experiments, the proposed method and other compared methods were implemented on a computer configuration with an Intel Core^{TM} i5 CPU, 8 GB of RAM, and Windows 10 using the Python 3.9 programming language and Sklearn libraries. Each method is executed over ten separate runs in every experiment, and an average of ten runs is calculated to gain more accurate results assessments for the comparison between methods.

Additionally, the dataset is normalized and divided randomly into a training and testing set in every run, where 80% of the initial data were used in the training phase and 20% of the initial data were used for testing purposes. The FS process is carried out on the training dataset, and the proposed method is tested on the test dataset. A ten-fold cross-validation technique is used on each dataset, and all examined methods are evaluated on the same dataset to ensure fairness. The findings are reported with respect to both the average number of features that were selected and the average accuracy of the classification.

C. APPLED CLASSIFIERS

Three different classifiers, including NB, KNN, and SVM, were applied to evaluate the proposed MOBCSA and other methods. The selection of these classifiers was based on their extensive utilization and proven effectiveness across a range of machine-learning applications. The SVM classifier is a common ML algorithm utilized for both regression and classification purposes. It operates by identifying the optimal hyperplane that separates data points into distinct classes. Furthermore, the KNN classifier is a non-parametric classification algorithm commonly employed in ML for regression and classification tasks. It operates on the principle that data points with similar features tend to belong to the same class or exhibit similar values. In KNN, a new data point is classified by considering the class labels of its K nearest neighbors in the training dataset. The value of K is determined by the user and determines the number of neighbors taken into account. The algorithm calculates the distances between the new data point and all other data points in the training set, selecting the K nearest data points. The new data point is then assigned the most common class among these K nearest data points. Lastly, the NB classifier is a probabilistic method that utilizes Bayes' theorem to determine the probability of a data point belonging to a specific class based on its features. It assumes that all features are conditionally independent.

D. PARAMETERS TUNING AND SETTINGS

Table 3 shows the general parameters for all algorithms. Moreover, Table 4 shows the parameter settings of the developed MOBCSA method and all other algorithms used for comparison. The parameters for each method are initialized depending on the recommendations from their respective original papers.

To ensure optimal performance of the proposed algorithm, it is recommended to set the number of iterations Max_{itr} and the size of population P. To tune these two parameters, initial experiments were carried out on several datasets. The different values of the number of iterations are used in experiments, such as 10, 15, 20, 25, 30, 40, 50, 60, and 70. Table 5 indicates that the proposed algorithm attains convergence in approximately 50 iterations for all datasets, with no significant improvement in accuracy or the number of selected genes observed after iteration number 50. Hence, we have set the number of iterations to 50 in this study. Moreover, to tune the value of the size of population, the value of the parameter number of iterations was maintained at 100 while exploring different population sizes ranging from 10 to 100.

The results attained for each population size have been presented in Table 6. As per the observations, the proposed algorithm demonstrates the most favorable outcome at a population size of 30. It is noteworthy that the accuracy and numbers of genes do not register any improvement when the value of P is increased beyond this point. Thus, based on the findings, it has been firmly decided that the population size will be fixed at 30. This decision has been taken after a meticulous analysis of the data and considering all possible scenarios that could present themselves. This step will ensure the optimal performance of the algorithm. In light of this, we conclude that the proposed algorithm can be further improved by setting these two parameters accordingly.

E. EVALUATION METRICS

This section highlights the evaluation metrics utilized in each experiment to assess the efficiency of each optimization-based algorithm. The methods are compared based on two

TABLE 2. A brief description of each dataset's characteristics.

Dataset	Genes number	Samples size	Classes number	Description	Reference
SRBCT	2308	83	4 (multi-class)	{NB, RMS, BL, EWS}	Khan et al. [82]
Prostate tumor	10509	102	2 (binary class)	{Tumor, Normal}	Singh et al. [83]
Lung cancer	12600	203	5 (multi-class)	{NL, AD, SMCL, SQ, COID}	Bhattacharjee et al. [84]
leukemia	7129	72	2 (binary-class)	{ALL, AML}	Golub et al. [85]
Colon tumor	2000	62	2 (binary-class)	{Normal, Tumor}	Alon et al. [86]
DLBCL	5469	77	2 (binary class)	{Normal, Cancer}	Shipp et al. [87]

TABLE 3. General parameters for all algorithms.

Parameters	value
The size of population	30
The iterations number	50
Number of runs	10
The *k*-value in KNN	5
K cross-validation	10

metrics: AvgClassAcc. and AvgSelFeat. The outcomes of these two metrics are averaged across N number of runs. The following equations describe all of these metrics:

1) AVGCLASSACC

This metric evaluates the correctness of classification, measuring how correctly the classifier matches the chosen feature subset to the samples. In this work, the AvgClassAcc is calculated using Eq. (15):

$$AvgClassAcc = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{N}\sum_{j=1}^{N} match(C_j, L_j) \qquad (15)$$

where C_j denotes class label outcome of a single sample j, Lj is the class label corresponding to j, n symbolizes the number of times method has been run, N represents number of all instances in datasets, and *match* denotes the comparator function, which returns 1 if two labels are the same and 0 otherwise.

2) AVGSELFEAT

Reflects the ratio of the overall average of the selected features to the whole number of features, the AvgSelFeat is calculated as given in Eq. (16).

$$AvgSelFeat = \frac{1}{N}\sum_{i=1}^{N}\frac{d_i}{D} \qquad (16)$$

where D signifies the total number of data sets, N represents number of runs and d_i stands for number of selected features.

TABLE 4. Parameter setup of each method.

Algorithms	Parameters	Values
MPSONC	The inertia weight w	0.9
	Social factor c_1	2
	learning factor c_2	2
	Weight of relevance objective w_1	0.4
	Weight size of redundancy objective w_2	0.4
	Objective weight w_2	0.2
	Mutation rate r_{mut}	0.01
MOBHHO	The relevance objective weight w_1	0.7
	Redundancy objective weight size w_2	0.3
	Objective weight w_3	0.6
	The archive size	100
MOABC	The food sources' number	25
	inertial weight	0.72
	Constants: c1 and c2	1.49
	limitation trial	5
MOPSOFS	Inertia weight w	0.729
	Constants $c_1 = c_2$	1.46
	The archive size	50
	The mutation rate	0.01
MOBCSA	Abandon probability $P\alpha$	0.25
	Levy distribution λ	1.5
	Step size α	1
	Archive size	50

F. EXPERIMENTAL FINDINGS AND DISCUSSIONS

This section presented and discussed the experimental findings of the suggested MOBCSA method, in addition to comparisons with four other multi-objective gene selection methods, including MOBHHO [27], MPSONC [56], MOABC [57], and MOPS [58], on six benchmark datasets: SRBCT, prostate tumor, lung cancer, leukemia, colon tumor,

TABLE 5. Tuning the parameter number of iterations.

Iterations	Datasets											
	SRBCT		Colon		Lung cancer		Prostate tumor		leukemia		DLBCL	
	Acc	Genes	Acc	Genes	Acc	Genes	Acc	Genes	Acc	Genes	Acc	Genes
10	93.67	14	94.89	16	89.87	26	90.00	22	92.45	15	93.45	18
15	93.12	14	97.94	17	90.24	26	91.80	24	93.78	19	92.34	17
20	94.21	15	96.91	17	94.12	29	93.64	24	94.23	18	95.38	19
25	96.58	15	98.91	19	93.20	29	94.62	20	94.12	16	94.67	17
30	96.23	19	96.92	19	93.50	28	94.77	21	96.34	18	95.78	17
40	96.50	17	96.02	19	92.87	28	94.94	21	94.34	16	95.78	17
50	97.59	16	98.42	19	93.19	28	94.94	21	94.34	16	95.78	19
60	97.59	16	98.42	19	93.19	28	94.94	21	94.34	16	95.78	19
70	97.59	16	98.42	19	93.19	28	94.94	21	94.34	16	95.78	19

ACC: Accuracy, *Genes*: number of selected genes

TABLE 6. Tuning the parameter size of populations.

Populations	Datasets											
	SRBCT		Colon		Lung cancer		Prostate tumor		leukemia		DLBCL	
	Acc	Genes	Acc	Genes	Acc	Genes	Acc	Genes	Acc	Genes	Acc	Genes
10	93.67	17	95.33	19	92.87	27	91.00	22	92.45	17	91.54	20
20	94.80	15	96.91	19	94.25	28	91.00	24	93.78	19	91.68	24
30	97.56	16	98.09	18	94.25	29	96.44	24	94.65	18	96.30	20
40	97.56	16	98.09	20	94.25	28	96.44	20	94.65	18	96.30	20
50	97.56	17	98.09	19	94.25	30	96.44	21	94.65	18	96.30	20
60	97.56	16	98.09	20	94.25	28	96.44	21	94.65	18	96.30	20
70	97.56	16	98.09	19	94.25	28	96.44	21	94.65	19	96.30	21
80	97.56	14	98.09	21	94.25	30	96.44	21	94.65	20	96.30	21
90	97.56	15	98.09	20	94.25	29	96.44	21	94.65	18	96.30	20
100	97.56	16	98.09	20	94.25	29	96.44	21	94.65	18	96.30	20

ACC: Accuracy, *Genes*: number of selected genes

and DLBCL. The methods were assessed utilizing three different classifiers: Naïve Bayes (NB), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). The results were plotted in three figures and tables, each representing the classifiers' performance with all techniques and datasets. The comparison was based on two primary criteria: AvgClassAcc and AvgSelFeat.

Tables 7, 8, and 9 summarize the AvgClassAcc of the proposed MOBCSA method and other methods over ten separate runs on different microarray datasets employing SVM, KNN, and NB, respectively. The highest value of the AvgClassAcc is highlighted in boldface. Moreover, those tables show the AvgClassAcc of each method across all microarray datasets. Then the AvgClassAcc plots in Figures 6, 7, and 8 for SVM, KNN, and NB classifiers, respectively, across all datasets and methods. Table 7 specifically demonstrates the AvgClassAcc for all methods across all datasets using SVM as the classifier. It can be seen that MOBCSA outperformed all other methods on all datasets. Specifically, MOBCSA achieved an AvgClassAcc of 94.50% across all datasets, whereas the other methods achieved an AvgClassAcc of 85.34% to 88.96%. The next-best method achieved an AvgClassAcc of 88.96%. For instance, in the Leukemia dataset, MOBCSAFS achieved the highest AvgClassAcc of 93.91%, while the next-best method, MOABC, got an AvgClassAcc of 89.43%. Similarly, in the Colon dataset, MOBCSA obtained the highest AvgClassAcc of 96.92%, and so on.

As depicted in Figure 6, in all datasets, the developed MOBCSA method excels when compared to other methods over the SVM classifier. For example, SRBCT, lung cancer,

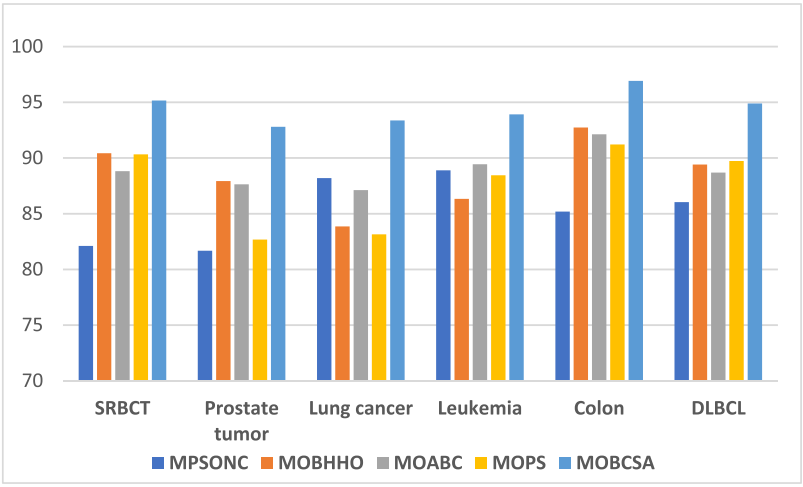


FIGURE 6. Average classification accuracy on SVM classifier.

TABLE 7. Average classification accuracy on SVM classifier.

Dataset	MPSONC	MOBHHO	MOABC	MOPS	MOBCSA
SRBCT	82.10	90.43	88.81	90.32	**95.16**
Prostate tumor	81.67	87.92	87.63	82.67	**92.79**
Lung cancer	88.19	83.86	87.11	83.15	**93.37**
Leukemia	88.89	86.33	89.43	88.44	**93.91**
Colon	85.19	92.73	92.12	91.21	**96.92**
DLBCL	86.03	89.41	88.69	89.72	**94.89**
Average	85.34	88.44	88.96	87.58	**94.50**

TABLE 8. Average classification accuracy on KNN classifier.

Dataset	MPSONC	MOBHHO	MOABC	MOPS	MOBCSA
SRBCT	89.10	88.02	84.78	86.56	**97.56**
Prostate tumor	87.11	83.41	84.81	81.49	**94.81**
Lung cancer	86.87	84.12	82.52	83.19	**92.89**
Leukemia	88.82	86.39	85.21	87.23	**93.14**
Colon	90.23	91.11	90.71	91.01	**98.42**
DLBCL	89.67	87.01	89.53	84.69	**96.33**
Average	88.63	86.67	86.26	85.69	**95.52**

prostate tumor, leukemia, colon, and DLBCL achieved the highest AvgClassAcc of 95.16%, 92.79%, 93.37%, 93.91%, 96.92%, and 94.89%, respectively. In the SRBCT dataset, the MOBCSA method achieved the highest AvgClassAcc rate of 95.16%, whereas the other methods achieved an AvgClassAcc rate between 82.10% and 90.43%, with a difference of 4.73% with respect to the second-best method, MOBHHO, which has an AvgClassAcc rate of 90.43%.

Table 8 demonstrates the AvgClassAcc of all methods on all datasets utilizing KNN as the classifier. Similar to the results with the SVM classifier, MOBCSA achieved the highest AvgClassAcc rate for all datasets, with an AvgClas-

TABLE 9. Average classification accuracy on NB classifier.

Dataset	MPSONC	MOBHHO	MOABC	MOPS	MOBCSA
SRBCT	90.32	92.11	82.51	82.19	**97.10**
Prostate tumor	86.29	80.64	79.43	79.24	**93.34**
Lung cancer	85.24	74.19	89.71	83.31	**92.70**
Leukemia	88.64	88.29	82.19	81.13	**94.31**
Colon	94.11	93.11	90.02	91.12	**97.98**
DLBCL	89.87	90.64	82.51	82.19	**95.72**
Average	89.07	86.49	84.39	83.19	**95.19**

sAcc of 95.52%, which was particularly higher than the accuracy achieved by the other methods (MPSONC: 88.63%, MPSONC: 86.67%, MOABC: 86.26%, MOPS: 85.69%). Moreover, Figure 7 shows that the proposed method scored the highest classification accuracy in all microarray datasets over the KNN classifier, demonstrating the superior performance of the MOBCSA compared to the other methods on all the datasets. For instance, in the Colon dataset, the proposed method scored 96.92%, which is the highest value with a difference of 6.31% compared to the MOBHHO method, which obtained the second-highest value. In the SRBCT dataset, MOBCSA achieved a classification accuracy of 97.56%, while the next-best method, MPSONC, achieved an accuracy of 89.10%. The outcomes in Figure 7 also show that MOBCSA achieved higher AvgClassAcc than other methods in all cases.

Table 9 displays the AvgClassAcc for all methods on all datasets while using NB as the classifier. Once again, MOBCSA achieved the highest AvgClassAcc rate for all datasets, achieving an AvgClassAcc of 95.19% compared to the other methods, which achieved an average accuracy between 83.19% and 89.07%. Furthermore, the reported results in Figure 8 demonstrate that the MOBCSA method is consistently more accurate than the other methods across all datasets. Overall, the obtained findings prove that the MOBCSA surpasses the other methods with respect to classification accuracy across all datasets and classifiers.

Table 10 reports the findings concerning the average number of selected genes by both MOBCSA and the other methods across all datasets, followed by the corresponding outcome plots in Figure 8. From Table 10, it is evident that most methods effectively reduced dimensionality by choosing only a small proportion of the original datasets.

The proposed method MOBCSA demonstrated superior performance by selecting the lowest number of genes, with only an average of 19.28 chosen genes when compared to other methods. The experimental finding of an average number of selected genes by the proposed MOBCSA and the other methods across all datasets is plotted in Figure 8, which proves that the proposed MOBCSA method outperformed the other methods by selecting only 20.87%, 16.47%, 27.66%, 15.67%, 18.52%, and 16.53% of features across the prostate tumor, SRBCT, lung cancer, leukemia, colon, and DLBCL datasets, respectively. For instance, in the leukemia dataset, MOBCSA selected only 15.67%, while the next best method, MPSONC, selected an average of 22.87%. Similarly, in the Colon dataset, MOBCSA identified the smallest number of features. These results demonstrate the proposed method's ability to identify the most useful genes for cancer classification tasks, outperforming all other methods in terms of the average number of selected genes across all datasets.

In conclusion, the proposed MOBCSA method consistently outperformed the other four multi-objective gene selection methods (MOBHHO, MPSONC, MOABC, and MOPS) in terms of the AvgClassAcc across all six datasets (SRBCT, prostate tumor, lung cancer, leukemia, colon tumor, and DLBCL) and three classifiers (KNN, NB, and SVM). MOBCSA achieved the highest AvgClassAcc rates for all datasets, ranging from 94.50% to 95.52%, depending on the classifier used. This performance was noticeably better than the other methods, which achieved average accuracy rates between 85.34% and 88.96%. The proposed MOBCSA method also exhibited exceptional performance in terms of the AvgSelFeat, selecting the lowest number of genes across all datasets with an average of 19.28 chosen genes. This indicates that MOBCSA is more effective at identifying the most useful genes for cancer classification tasks compared to the other methods. Across all three classifiers (SVM, KNN, and NB), MOBCSA consistently demonstrated superior performance by achieving the highest classification accuracy and selecting the fewest number of genes in every scenario. This demonstrates the success of MOBCSA in handling high-dimensional biological data and its potential to be applied practically in big data bioinformatics analytics. Furthermore, the MOBCSA's performance is attributed to its

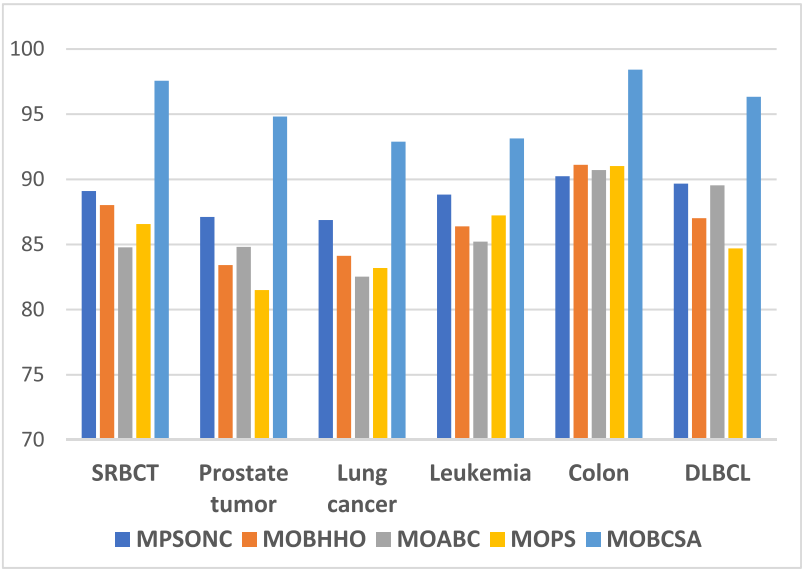


FIGURE 7. Average classification accuracy on KNN classifier.

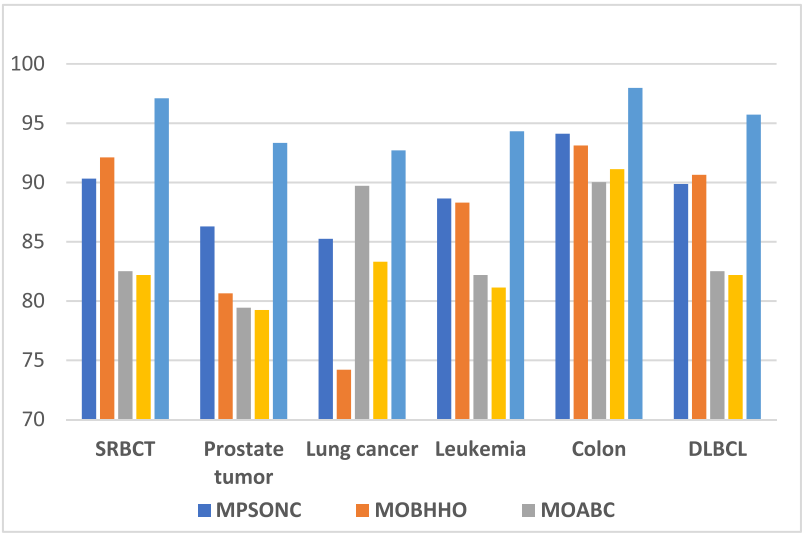


FIGURE 8. Average classification accuracy on NB classifier.

ability to balance the trade-off between classification accuracy and the number of chosen genes, efficiently discover the search space, and choose the optimum solution.

G. STATISTICAL VALIDATION OF THE RESULTS

This subsection examined the statistical validation of the reported results using various classifiers such as SVM, NB, and KNN for both the proposed method and other methods. These validations were conducted on different datasets, namely SRBCT, prostate tumor, lung cancer, leukemia, colon, and DLBCL. Our analysis focused on several criteria, including standard deviation (Std), maximum value (Max), minimum value (Min), and median (Med), for each method and dataset. Table 11 and Figure 10a showcase a statistical analysis of the accuracy using the SVM classifier for both the proposed method and other methods across the different datasets. The criteria considered for this analysis were Std, Max, Min, and Med for each method and dataset.

For the SRBCT dataset, it was observed that the MOBHHO method exhibited a lower standard deviation (0.85) compared to the other methods. Additionally, the proposed method demonstrated the highest maximum value (96.87) for the SRBCT dataset. Furthermore, the proposed method displayed competitive performance by achieving high maximum values (93.87, 94.98, 94.89, 97.93, and 96.80) for the prostate tumor, lung cancer, leukemia, colon, and DLBCL datasets, respec-

TABLE 10. Average number of selected genes.

Dataset	Methods				
	MPSONC	MOBHHO	MOABC	MOPS	MOBCSA
SRBCT	19.58	23.54	21.54	27.01	**16.47**
Prostate Tumor	28.90	34.19	28.38	25.06	**20.87**
Lung Cancer	30.33	33.77	31.45	37.60	**27.66**
Leukemia	22.87	27.09	24.75	29.19	**15.67**
Colon	19.40	23.28	25.97	26.25	**18.52**
DLBCL	22.43	20.76	24.21	23.00	**16.53**
Average	23.91	27.10	26.05	28.01	**19.28**

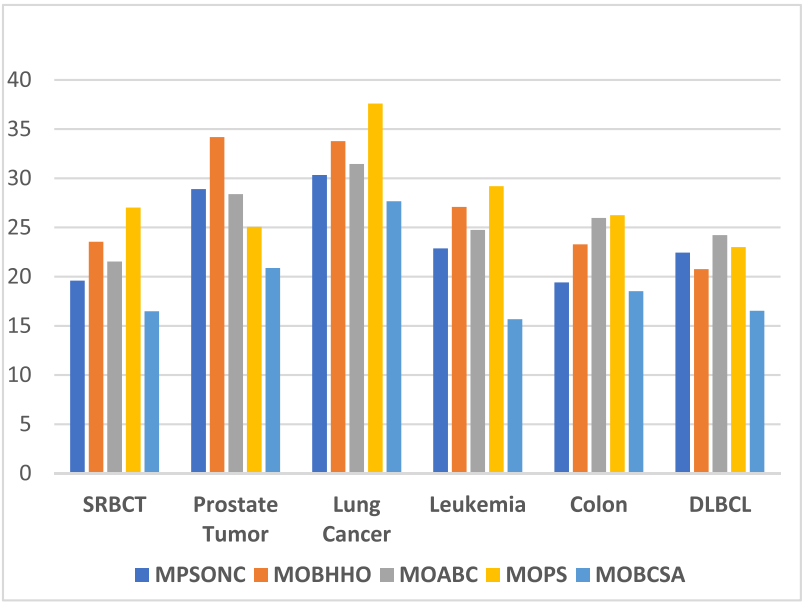


FIGURE 9. The average number of selected genes.

tively. Moreover, the proposed method consistently achieved competitive results by attaining lower standard deviations across multiple datasets, including prostate tumors, lung cancer, leukemia, and colon cancer. In conclusion, the MOBCSA method consistently performed well across various datasets, exhibiting a lower standard deviation and higher maximum values, minimum values, and medians compared to other methods.

Table 12 and Figure 10b present a statistical result of accuracy using the KNN classifier for all methods across different datasets. The analysis involved assessing the criteria of Std, Max, Min, and Med for each method and dataset. It is evident that the proposed method demonstrated competitive performance by attaining high maximum values (98.89, 96.007, 93.89, 94.21, 98.99, and 97.84) and minimum values (95.91, 93.01, 91.97, 92.18, 96.89, and 95.232) for the SRBCT, prostate tumor, lung cancer, leukemia, colon, and DLBCL datasets, respectively. For the prostate tumor dataset, it was observed that the MOBHHO and MOBCSA methods exhibited a lower standard deviation (0.64 and 0.85), respectively, compared to the other methods. Moreover, the MOBHHO method achieved lower standard deviations across multiple datasets, including SRBCT, lung cancer, and DLBCL. Whereas, the proposed method has the second-best lower standard deviations across multiple datasets. In conclusion, the MOBCSA method consistently performs well across various datasets, exhibiting lower standard deviations, minimum values, medians, and higher maximum values compared to other methods.

Table 13 and Figure 11a report a statistical result of accuracy using the NB classifier for all methods across different datasets. The MOBCSA method achieves the highest

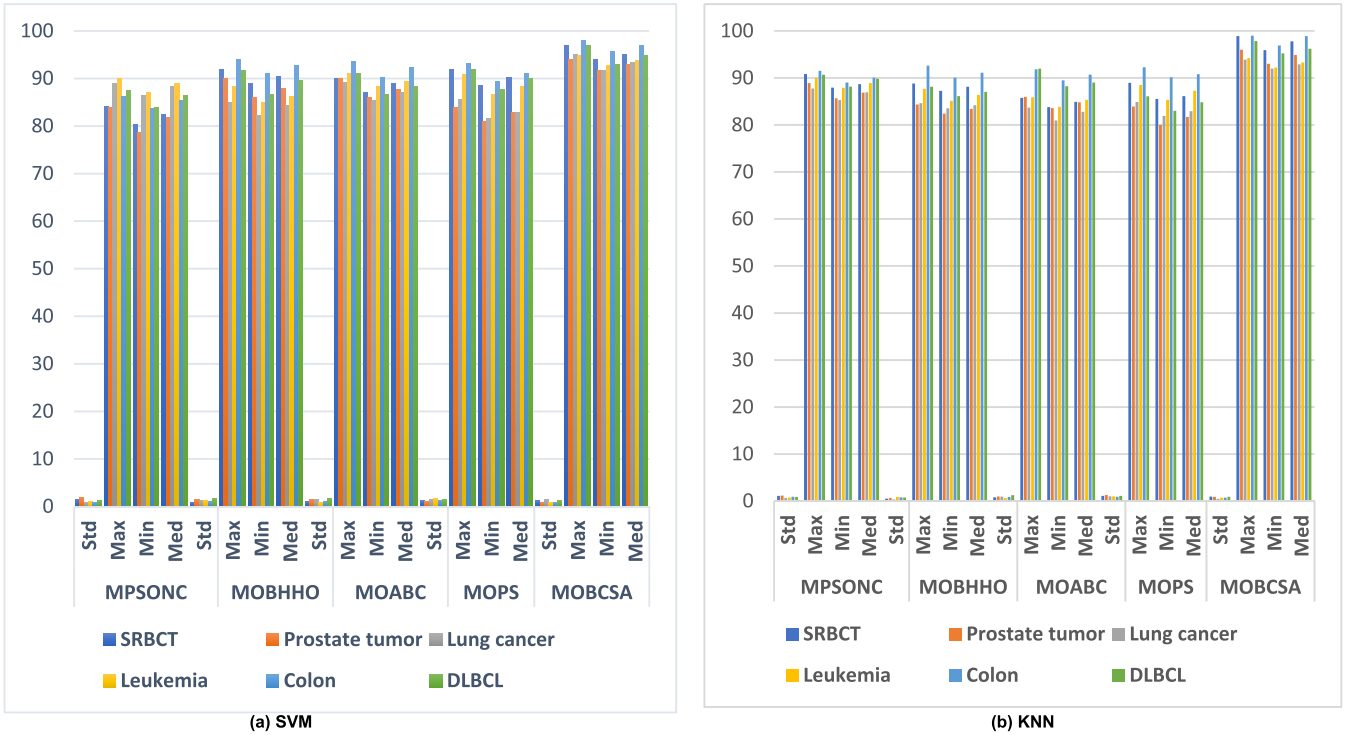


FIGURE 10. Statistical analysis for all methods over all data sets on different classifiers.

maximum value of 98.65 for the SRBCT dataset. In terms of the prostate tumor dataset, the MOBCSA method has the highest maximum value of 94.68, the minimum value of 95.87, and the median value of 97.07. The MOBCSA method achieves the highest maximum value of 93.95 for the lung cancer dataset. For the Leukemia, Colon, and DLBCL datasets, the MOBCSA method has the highest maximum, median, and minimum values. Therefore, the MOBCSA method shows competitive performance, achieving high maximum, median, and minimum values for all datasets. For the SRBCT dataset, it can be observed that the MOBHHO method has a lower standard deviation of 0.64 compared to the other methods. Whereas the proposed method has a lower standard deviation across lung cancer, leukemia, and colon data sets (0.66, 0.74, and 0.73), respectively. In conclusion, the MOBCSA method consistently performs well across various datasets and classifiers, exhibiting a lower standard deviation, minimum value, median, and higher maximum values compared to other methods.

Table 14 and Figure 11b present a statistical analysis of the number of selected genes for different methods across various datasets. The table provides Std, Max, Min, and Med values for each method and dataset. These results offer valuable insights into the performance and variability of different methods for selecting genes across diverse datasets.

The MOBCSA method demonstrates exceptional consistency, as evidenced by its lower standard deviation, ranging from 0.16 to 0.29 across different datasets, surpassing the second-best method, MOABC, which exhibits a wider range from 0.23 to 0.32. This indicates that the MOBCSA method consistently maintains a more stable gene selection pattern across varied datasets. Furthermore, the MOBCSA method excels in achieving the best maximum values, which range from 23.03 to 34.9 when compared to the other methods. This underscores its capability to identify a higher number of relevant genes in different dataset contexts.

H. CONVERGENCE RATE ANALYSIS

This subsection analyzes the convergence rate of the proposed method by conducting additional experiments to validate the convergence rate of the proposed MOBCSA method in comparison to other existing methods. The convergence rates of MOBCSA and the other methods across all datasets spanning a total of 100 iterations are depicted in Figures 12a, b, c, d, e, and f. The experimentation process is carried out at 10 intervals (a total of 100 iterations). In each interval, there are 10 iterations, and then the average convergence rate of each interval (10 iterations) was calculated. As can be seen in Figure 12(a), in the case of the SRBCT dataset, the MOBCSA method reaches its optimum value in the 30th iteration. Similarly, for the colon dataset, depicted in Figure 12(b), the MOBCSA method achieves an optimal value of 98.91 at iteration 30. Moving on to the leukemia dataset in Figure 12(c), the MOBCSA method exhibited exceptionally fast convergence, achieving the optimal value in just 20 iterations. In the case

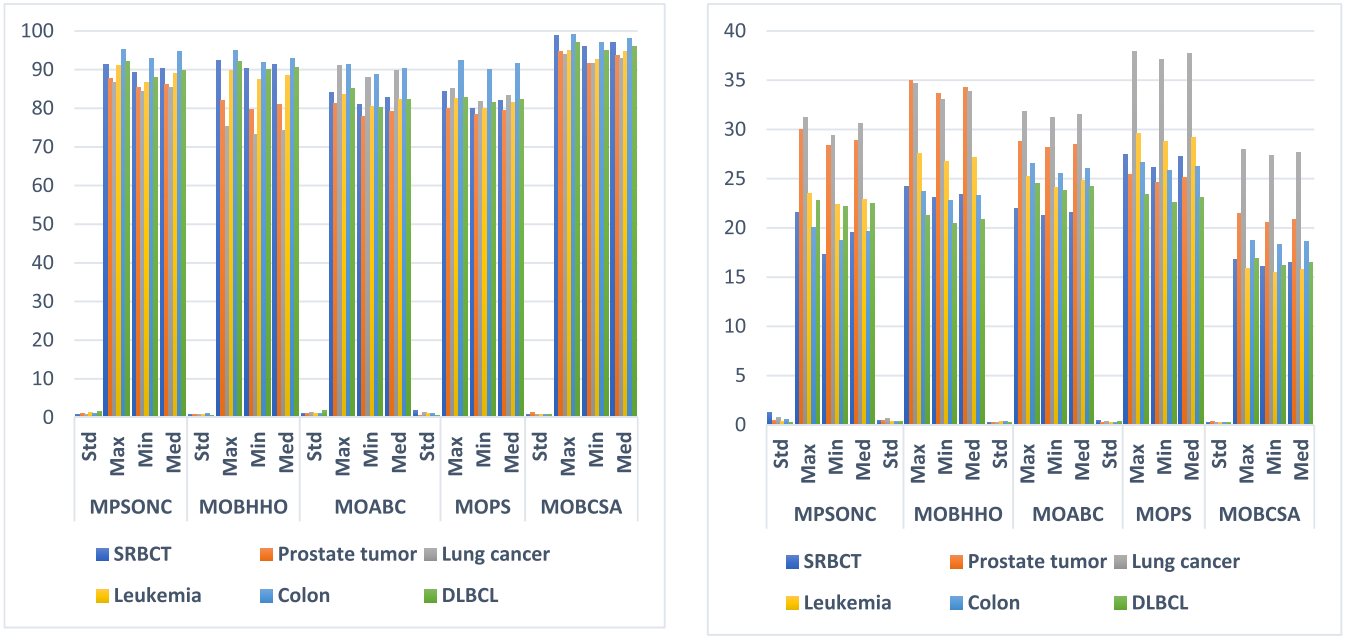
TABLE 11. Statistical analysis using SVM classifier.

Methods		Datasets					
		SRBCT	Prostate tumor	Lung cancer	Leukemia	Colon	DLBCL
MPSONC	Std	1.42	1.79	0.79	0.93	0.89	1.13
	Max	83.97	83.91	88.98	89.98	86.27	87.39
	Min	80.25	78.65	86.32	86.97	83.64	83.91
	Med	82.32	81.82	88.33	88.98	85.33	86.28
MOBHHO	Std	0.85	1.34	1.14	1.15	0.96	1.61
	Max	91.87	89.90	84.97	88.22	93.86	91.72
	Min	88.94	85.89	82.24	84.90	90.92	86.53
	Med	90.44	87.93	84.24	86.24	92.75	89.63
MOABC	Std	1.02	1.36	1.38	0.76	0.98	1.55
	Max	89.95	89.99	89.12	90.93	93.44	90.92
	Min	86.97	85.88	85.27	88.34	90.13	86.66
	Med	88.84	87.53	86.95	89.41	92.25	88.35
MOPS	Std	1.156	1.02	1.36	1.56	1.21	1.31
	Max	91.82	83.92	85.47	90.87	93.00	91.81
	Min	88.47	80.94	81.53	86.54	89.29	87.58
	Med	90.21	82.72	82.77	88.24	91.05	89.90
MOBCSA	Std	1.13	0.75	1.30	0.78	0.75	1.182
	Max	96.87	93.87	94.98	94.89	97.93	96.80
	Min	93.87	91.53	91.55	92.62	95.71	92.88
	Med	94.94	92.84	93.36	93.73	96.85	94.76

TABLE 12. Statistical analysis using KNN classifier.

Methods		Datasets					
		SRBCT	Prostate tumor	Lung cancer	Leukemia	Colon	DLBCL
MPSONC	Std	1.07	1.15	0.65	0.74	0.88	0.87
	Max	90.81	88.89	87.73	89.98	91.52	90.67
	Min	87.91	85.64	85.28	87.89	89.01	88.15
	Med	88.66	86.88	86.92	88.89	90.05	89.82
MOBHHO	Std	0.48	0.64	0.32	0.85	0.76	0.75
	Max	88.81	84.35	84.62	87.69	92.61	88.11
	Min	87.25	82.39	83.54	85.11	90.05	86.11
	Med	88.12	83.40	84.20	86.37	91.11	87.01
MOABC	Std	0.76	0.94	0.92	0.61	0.86	1.22
	Max	85.76	85.97	83.69	85.87	91.83	91.99
	Min	83.79	83.57	80.94	83.84	89.47	88.25
	Med	84.89	84.80	82.78	85.34	90.71	89.02
MOPS	Std	1.08	1.25	0.97	0.99	0.90	1.07
	Max	88.95	83.893	84.88	88.50	92.24	86.09
	Min	85.53	79.99	81.92	85.35	90.13	83.01
	Med	86.12	81.71	82.91	87.27	90.78	84.80
MOBCSA	Std	0.91	0.85	0.45	0.70	0.71	0.90
	Max	98.89	96.00	93.89	94.21	98.99	97.84
	Min	95.91	93.01	91.97	92.18	96.89	95.23
	Med	97.78	94.87	92.88	93.27	98.88	96.20

of the lung cancer dataset, as shown in Figure 12(d), the MOBCSA method achieved rapid convergence, reaching the optimal value much earlier in just 30 iterations compared to other methods. In the prostate tumor dataset, the MOBCSA method attains its optimum value by the 40th iteration. Furthermore, in the DLBCL dataset, the MOBCSA method



(a) NB

(b) Number of selected genes

FIGURE 11. Statistical analysis for all methods over all data sets.

TABLE 13. Statistical analysis using NB classifier.

Methods		Datasets					
		SRBCT	Prostate tumor	Lung cancer	Leukemia	Colon	DLBCL
MPSONC	Std	0.64	0.84	0.81	1.33	0.92	1.38
	Max	91.19	87.68	86.71	90.99	95.08	91.95
	Min	89.22	85.22	84.17	86.62	92.79	87.88
	Med	90.31	86.09	85.21	88.86	94.63	89.70
MOBHHO	Std	0.64	0.80	0.55	0.76	1.02	0.55
	Max	92.18	81.89	75.22	89.81	94.99	91.98
	Min	90.12	79.52	73.18	87.28	91.73	90.01
	Med	91.25	80.81	74.26	88.33	92.80	90.52
MOABC	Std	1.08	1.07	1.19	0.94	0.95	1.77
	Max	83.99	81.29	90.99	83.58	91.29	85.02
	Min	80.78	77.83	87.78	80.47	88.54	80.25
	Med	82.60	79.16	89.78	82.06	90.28	82.18
MOPS	Std	1.62	0.52	1.25	0.96	0.87	0.47
	Max	84.32	79.81	84.98	82.37	92.28	82.84
	Min	79.90	78.37	81.68	79.78	89.87	81.38
	Med	82.03	79.41	83.34	81.32	91.43	82.29
MOBCSA	Std	0.79	1.10	0.66	0.74	0.73	0.61
	Max	98.65	94.68	93.95	94.86	98.92	96.95
	Min	95.87	91.54	91.54	92.47	96.84	94.81
	Med	97.07	93.62	92.84	94.62	97.89	95.83

demonstrates swift convergence, reaching the optimal value notably early, in just 20 iterations, outperforming other methods.

The results indicate that MOBCSA can rapidly reach optimal or near-optimal solutions by the 30th iteration while consistently maintaining the highest accuracy values

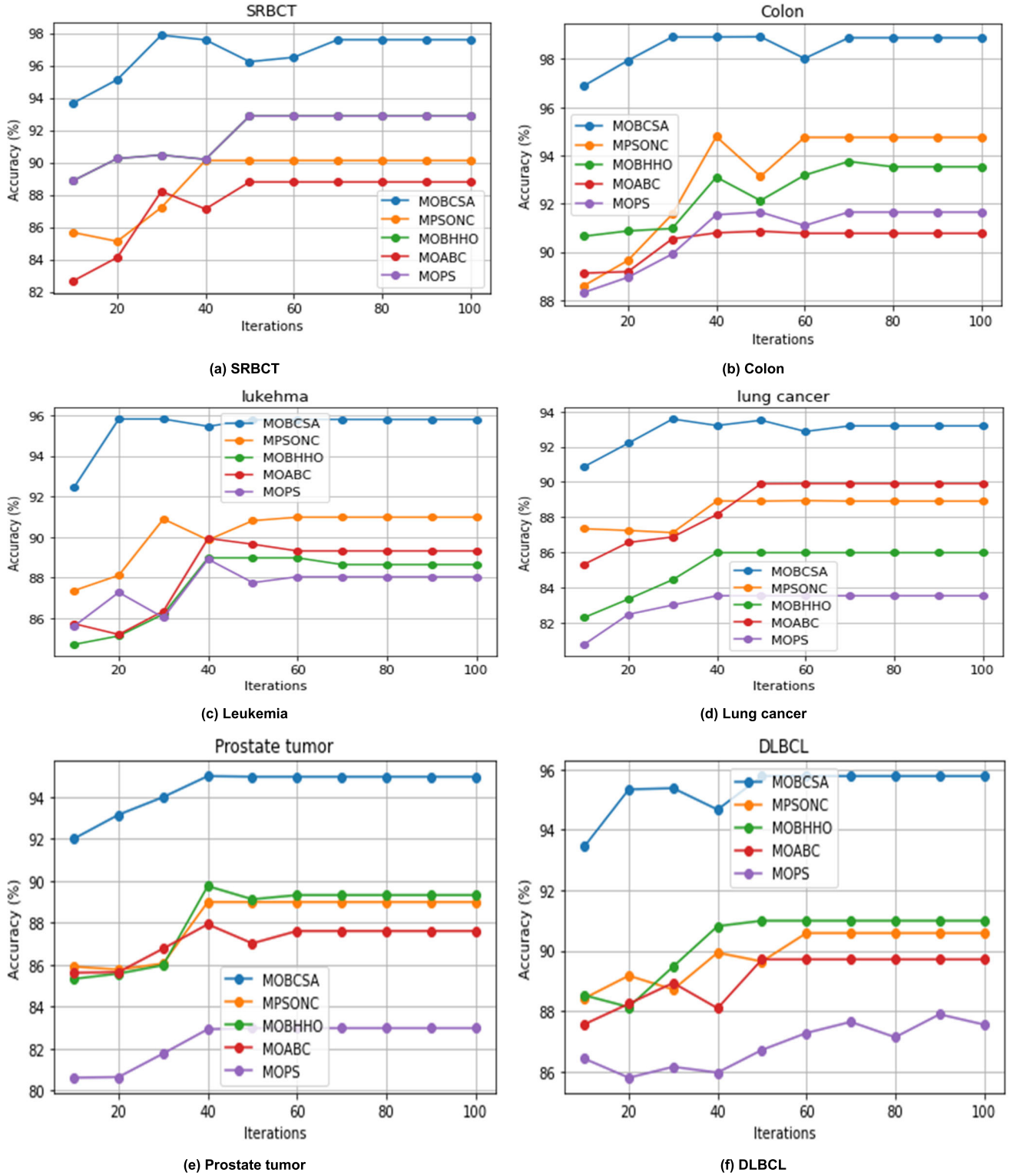


FIGURE 12. Convergence rate of MOBCSA and other methods across all datasets.

compared to other methods in most cases, except in the prostate tumor dataset. It can be concluded that MOBCSA excels at finding excellent solutions within a reasonable number of iterations, reflecting the rapid convergence speed of the proposed method.

Based on the findings and our observations, the proposed method has proven its efficiency in handling multiple conflicting objectives and achieving good results. Researchers can apply this method to solve some multi-objective optimization problems and also investigate its applications in

TABLE 14. Statistical analysis of number of selected genes for all methods over all data sets.

Methods		Datasets					
		SRBCT	Prostate tumor	Lung cancer	Leukemia	Colon	DLBCL
MPSONC	Std	1.25	0.41	0.68	0.35	0.53	0.18
	Max	21.58	29.99	31.2	23.51	19.99	22.73
	Min	17.25	28.38	29.32	22.38	18.72	22.15
	Med	19.5	28.86	30.55	22.84	19.63	22.42
MOBHHO	Std	0.42	0.42	0.57	0.27	0.32	0.29
	Max	24.2	34.9	34.65	27.5	23.7	21.2
	Min	23.03	33.6	33	26.7	22.8	20.4
	Med	23.35	34.25	33.84	27.14	23.3	20.8
MOABC	Std	0.23	0.21	0.22	0.31	0.32	0.23
	Max	21.9	28.7	31.8	25.2	26.5	24.5
	Min	21.2	28.1	31.15	24.1	25.5	23.8
	Med	21.55	28.4	31.45	24.75	26	24.2
MOPS	Std	0.46	0.25	0.27	0.25	0.26	0.32
	Max	27.4	25.4	37.9	29.6	26.6	23.4
	Min	26.1	24.6	37.1	28.8	25.8	22.5
	Med	27.2	25.1	37.65	29.2	26.25	23.1
MOBCSA	Std	0.21	0.29	0.19	0.16	0.16	0.26
	Max	16.8	21.4	27.9	15.9	18.7	16.9
	Min	16.1	20.5	27.3	15.4	18.3	16.2
	Med	16.45	20.85	27.65	15.7	18.55	16.5

some other domains. It can also be utilized with other FS methods to extend its potential impact in various domains.

VI. CONCLUSION AND FUTURE WORK

This article presents a Multi-Objective Binary Cuckoo Search Algorithm (MOBCSA) for gene selection in bioinformatics. MOBCSA aims to choose the relevant genes from a high-dimensional microarray dataset to enhance the effectiveness of medical diagnosis models. MOBCSA extends the standard CSA, considering multiple conflicting objectives, specifically increasing accuracy of classification and reducing number of genes in a wrapper mode. MOBCSA employs an S-shaped transfer function to convert the algorithm's search space from continuous search space to binary search space. Additionally, it integrates an external archive to store non-dominated optimal solutions obtained during the search process and an adaptive crowding distance updating mechanism to preserve diversity and expand the coverage of optimal solutions. To assess MOBCSA's performance, we conducted experiments on six benchmark biomedical datasets using three distinct classifiers. Then, we compared the results with four state-of-the-art multi-objective gene selection methods from the existing literature. The final findings indicate that MOBCSA surpasses the other methods in terms of accuracy and number of selected genes, where it has obtained an average accuracy ranging from 92.79% to 98.42% and an average number of selected features ranging from 15.67 to 27.88 across all classifiers and datasets. Overall, the proposed MOBCSA algorithm reported good results and was found to be efficient in gene selection from high-dimensional biomedical datasets for effective and accurate classification and medical diagnosis. It proved to be robust and efficient in handling multiple conflicting objectives simultaneously, leading to improved accuracy of classification and a minimum number of selected genes. The potential limitations associated with our method are limited to two fitness functions and the importance of domain-specific knowledge for effective feature interpretation. We have provided insights into how these limitations can be addressed in future enhancements. Future research could explore the applicability of MOBCSA to real-world bioinformatics problems and collaborate with domain experts. Moreover, we plan to incorporate additional fitness functions, such as computational complexity, into our analysis. By doing so, we aim to further enhance the algorithm's performance and extend its potential impact.

REFERENCES

[1] S. Osama, H. Shaban, and A. A. Ali, "Gene reduction and machine learning algorithms for cancer classification based on microarray gene expression data: A comprehensive review," *Expert Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 118946, doi: 10.1016/j.eswa.2022.118946.

[2] S. Azadifar, M. Rostami, K. Berahmand, P. Moradi, and M. Oussalah, "Graph-based relevancy-redundancy gene selection method for cancer diagnosis," *Comput. Biol. Med.*, vol. 147, Aug. 2022, Art. no. 105766, doi: 10.1016/j.compbiomed.2022.105766.

[3] N. Mahendran, P. M. D. R. Vincent, K. Srinivasan, and C.-Y. Chang, "Machine learning based computational gene selection models: A survey, performance evaluation, open issues, and future research directions," *Frontiers Genet.*, vol. 11, Dec. 2020, Art. no. 603808, doi: 10.3389/fgene.2020.603808.

[4] V. Bolón-Canedo and A. Alonso-Betanzos, *Microarray Bioinformatics*. Louisville, KY, USA: Humana, 2019.

[5] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognit.*, vol. 43, no. 1, pp. 5–13, Jan. 2010, doi: 10.1016/j.patcog.2009.06.009.

[6] H. M. Abdulwahab, S. Ajitha, and M. A. N. Saif, "Feature selection techniques in the context of big data: Taxonomy and analysis," *Int. J. Speech Technol.*, vol. 52, no. 12, pp. 13568–13613, Jan. 2022, doi: 10.1007/s10489-021-03118-3.

[7] W. Ke, C. Wu, Y. Wu, and N. N. Xiong, "A new filter feature selection based on criteria fusion for gene microarray data," *IEEE Access*, vol. 6, pp. 61065–61076, 2018, doi: 10.1109/ACCESS.2018.2873634.

[8] J. Lee, I. Y. Choi, and C.-H. Jun, "An efficient multivariate feature ranking method for gene selection in high-dimensional microarray data," *Expert Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 113971, doi: 10.1016/j.eswa.2020.113971.

[9] K. K. Ghosh, S. Begum, A. Sardar, S. Adhikary, M. Ghosh, M. Kumar, and R. Sarkar, "Theoretical and empirical analysis of filter ranking methods: Experimental study on benchmark DNA microarray data," *Expert Syst. Appl.*, vol. 169, May 2021, Art. no. 114485, doi: 10.1016/j.eswa.2020.114485.

[10] Z. Hua, J. Zhou, Y. Hua, and W. Zhang, "Strong approximate Markov blanket and its application on filter-based feature selection," *Appl. Soft Comput.*, vol. 87, Feb. 2020, Art. no. 105957, doi: 10.1016/j.asoc.2019.105957.

[11] V. Kalaimani and R. Umagandhi, "WITHDRAWN: A novel wrapper FS based on binary swallow swarm optimization with score-based criteria fusion for gene expression microarray data," *Mater. Today, Proc.*, Dec. 2020, doi: 10.1016/j.matpr.2020.11.064.

[12] T. Ragunthar and S. Selvakumar, "A wrapper based feature selection in bone marrow plasma cell gene expression data," *Cluster Comput.*, vol. 22, no. 6, pp. 13785–13796, Feb. 2018, doi: 10.1007/s10586-018-2094-2.

[13] M. Ghosh, S. Begum, R. Sarkar, D. Chakraborty, and U. Maulik, "Recursive memetic algorithm for gene selection in microarray data," *Expert Syst. Appl.*, vol. 116, pp. 172–185, Feb. 2019, doi: 10.1016/j.eswa.2018.06.057.

[14] M. A. Tawhid and A. M. Ibrahim, "Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm," *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 3, pp. 573–602, Aug. 2019, doi: 10.1007/s13042-019-00996-5.

[15] C. Kang, Y. Huo, L. Xin, B. Tian, and B. Yu, "Feature selection and tumor classification for microarray data using relaxed lasso and generalized multi-class support vector machine," *J. Theor. Biol.*, vol. 463, pp. 77–91, Feb. 2019, doi: 10.1016/j.jtbi.2018.12.010.

[16] H. Zhu, N. Bi, J. Tan, and D. Fan, "An embedded method for feature selection using kernel parameter descent support vector machine," in *Proc. Chin. Conf. Pattern Recognit. Comput. Vis. (PRCV)*, in Lecture Notes in Computer Science, vol. 11258, Jan. 2018, pp. 351–362, doi: 10.1007/978-3-030-03338-5_30.

[17] S. Maldonado and J. López, "Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification," *Appl. Soft Comput.*, vol. 67, pp. 94–105, Jun. 2018, doi: 10.1016/j.asoc.2018.02.051.

[18] M. Tubishat, S. Ja'afar, M. Alswaitti, S. Mirjalili, N. Idris, M. A. Ismail, and M. S. Omar, "Dynamic salp swarm algorithm for feature selection," *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113873, doi: 10.1016/j.eswa.2020.113873.

[19] M. Alweshah, S. Al Khalaileh, B. B. Gupta, A. Almomani, A. I. Hammouri, and M. A. Al-Betar, "The monarch butterfly optimization algorithm for solving feature selection problems," *Neural Comput. Appl.*, vol. 34, pp. 11267–11281, Jul. 2020, doi: 10.1007/s00521-020-05210-0.

[20] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi, and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," *Expert Syst. Appl.*, vol. 117, pp. 267–286, Mar. 2019, doi: 10.1016/j.eswa.2018.09.015.

[21] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Exp. Syst. Appl.*, vol. 116, pp. 147–160, Feb. 2019, doi: 10.1016/j.eswa.2018.08.051.

[22] P. Hu, J.-S. Pan, and S.-C. Chu, "Improved binary grey wolf optimizer and its application for feature selection," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105746, doi: 10.1016/j.knosys.2020.105746.

[23] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018, doi: 10.1016/j.asoc.2017.11.006.

[24] H. Rao, X. Shi, A. K. Rodrigue, J. Feng, Y. Xia, M. Elhoseny, X. Yuan, and L. Gu, "Feature selection based on artificial bee colony and gradient boosting decision tree," *Appl. Soft Comput.*, vol. 74, pp. 634–642, Jan. 2019, doi: 10.1016/j.asoc.2018.10.036.

[25] R. C. T. de Souza, C. A. de Macedo, L. dos Santos Coelho, J. Pierezan, and V. C. Mariani, "Binary coyote optimization algorithm for feature selection," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107470, doi: 10.1016/j.patcog.2020.107470.

[26] B. Nouri-Moghaddam, M. Ghazanfari, and M. Fathian, "A novel multi-objective forest optimization algorithm for wrapper feature selection," *Expert Syst. Appl.*, vol. 175, Aug. 2021, Art. no. 114737, doi: 10.1016/j.eswa.2021.114737.

[27] A. Dabba, A. Tari, and S. Meftali, "A new multi-objective binary Harris hawks optimization for gene selection in microarray data," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 4, pp. 3157–3176, Aug. 2021, doi: 10.1007/s12652-021-03441-0.

[28] A. Chaudhuri and T. P. Sahu, "Multi-objective feature selection based on quasi-oppositional based Jaya algorithm for microarray data," *Knowl.-Based Syst.*, vol. 236, Jan. 2022, Art. no. 107804, doi: 10.1016/j.knosys.2021.107804.

[29] X.-S. Yang and S. Deb. (Dec. 1, 2009). *Cuckoo Search via Lévy Flights*. IEEE Xplore. Accessed: Mar. 14, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/5393690

[30] M. Mafarja, I. Aljarah, A. A. Heidari, A. I. Hammouri, H. Faris, A. M. Al-Zoubi, and S. Mirjalili, "Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems," *Knowl.-Based Syst.*, vol. 145, pp. 25–45, Apr. 2018, doi: 10.1016/j.knosys.2017.12.037.

[31] H. Hichem, M. Elkamel, M. Rafik, M. T. Mesaaoud, and C. Ouahiba, "A new binary grasshopper optimization algorithm for feature selection problem," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 2, pp. 316–328, Feb. 2022, doi: 10.1016/j.jksuci.2019.11.007.

[32] A. Zakeri and A. Hokmabadi, "Efficient feature selection method using real-valued grasshopper optimization algorithm," *Exp. Syst. Appl.*, vol. 119, pp. 61–72, Apr. 2019, doi: 10.1016/j.eswa.2018.10.021.

[33] Z. Y. Algamal, M. K. Qasim, M. H. Lee, and H. T. M. Ali, "Improving grasshopper optimization algorithm for hyperparameters estimation and feature selection in support vector regression," *Chemometric Intell. Lab. Syst.*, vol. 208, Jan. 2021, Art. no. 104196, doi: 10.1016/j.chemolab.2020.104196.

[34] D. Wang, H. Chen, T. Li, J. Wan, and Y. Huang, "A novel quantum grasshopper optimization algorithm for feature selection," *Int. J. Approx. Reasoning*, vol. 127, pp. 33–53, Dec. 2020, doi: 10.1016/j.ijar.2020.08.010.

[35] B. Zhang, X. Yang, B. Hu, Z. Liu, and Z. Li, "OEbBOA: A novel improved binary butterfly optimization approaches with various strategies for feature selection," *IEEE Access*, vol. 8, pp. 67799–67812, 2020, doi: 10.1109/ACCESS.2020.2985986.

[36] A. A. Awad, A. F. Ali, and T. Gaber, "Feature selection method based on chaotic maps and butterfly optimization algorithm," *Adv. Intell. Syst. Comput.*, vol. 1153, pp. 159–169, Jan. 2020, doi: 10.1007/978-3-030-44289-7_16.

[37] M. Tubishat, M. Alswaitti, S. Mirjalili, M. A. Al-Garadi, M. T. Alrashdan, and T. A. Rana, "Dynamic butterfly optimization algorithm for feature selection," *IEEE Access*, vol. 8, pp. 194303–194314, 2020, doi: 10.1109/ACCESS.2020.3033757.

[38] W. Long, J. Jiao, X. Liang, T. Wu, M. Xu, and S. Cai, "Pinhole-imaging-based learning butterfly optimization algorithm for global optimization and feature selection," *Appl. Soft Comput.*, vol. 103, May 2021, Art. no. 107146, doi: 10.1016/j.asoc.2021.107146.

[39] Z. Sadeghian, E. Akbari, and H. Nematzadeh, "A hybrid feature selection method based on information theory and binary butterfly optimization algorithm," *Eng. Appl. Artif. Intell.*, vol. 97, Jan. 2021, Art. no. 104079, doi: 10.1016/j.engappai.2020.104079.

[40] A. Tiwari and A. Chaturvedi, "A hybrid feature selection approach based on information theory and dynamic butterfly optimization algorithm for data classification," *Expert Syst. Appl.*, vol. 196, Jun. 2022, Art. no. 116621, doi: 10.1016/j.eswa.2022.116621.

[41] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
[42] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016, doi: 10.1016/j.neucom.2015.06.083.
[43] Q. Tu, X. Chen, and X. Liu, "Multi-strategy ensemble grey wolf optimizer and its application to feature selection," *Appl. Soft Comput.*, vol. 76, pp. 16–30, Mar. 2019, doi: 10.1016/j.asoc.2018.11.047.
[44] M. Abdel-Basset, D. El-Shahat, I. El-henawy, V. H. C. de Albuquerque, and S. Mirjalili, "A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection," *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112824, doi: 10.1016/j.eswa.2019.112824.
[45] M. Sharawi, H. M. Zawbaa, E. Emary, H. M. Zawbaa, and E. Emary, "Feature selection approach based on whale optimization algorithm," in *Proc. 9th Int. Conf. Adv. Comput. Intell. (ICACI)*, Feb. 2017, pp. 163–168, doi: 10.1109/ICACI.2017.7974502.
[46] A. G. Hussien, A. E. Hassanien, E. H. Houssein, S. Bhattacharyya, and M. Amin, "S-shaped binary whale optimization algorithm for feature selection," in *Recent Trends in Signal and Image Processing*. Singapore: Springer, 2018, pp. 79–87, doi: 10.1007/978-981-10-8863-6_9.
[47] G. I. Sayed, A. Darwish, and A. E. Hassanien, "A new chaotic whale optimization algorithm for features selection," *J. Classification*, vol. 35, no. 2, pp. 300–344, Jul. 2018, doi: 10.1007/s00357-018-9261-2.
[48] R. K. Agrawal, B. Kaur, and S. Sharma, "Quantum based whale optimization algorithm for wrapper feature selection," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106092, doi: 10.1016/j.asoc.2020.106092.
[49] B. A. H. Murshed, S. Mallappa, J. Abawajy, M. A. N. Saif, H. D. E. Al-ariki, and H. M. Abdulwahab, "Short text topic modelling approaches in the context of big data: Taxonomy, survey, and analysis," *Artif. Intell. Rev.*, vol. 56, no. 6, pp. 5133–5260, Oct. 2022, doi: 10.1007/s10462-022-10254-w.
[50] B. A. H. Murshed, J. Abawajy, S. Mallappa, M. A. N. Saif, S. M. Al-Ghuribi, and F. A. Ghanem, "Enhancing big social media data quality for use in short-text topic modeling," *IEEE Access*, vol. 10, pp. 105328–105351, 2022, doi: 10.1109/ACCESS.2022.3211396.
[51] B. A. H. Murshed, J. Abawajy, S. Mallappa, M. A. N. Saif, and H. D. E. Al-Ariki, "DEA-RNN: A hybrid deep learning approach for cyberbullying detection in Twitter social media platform," *IEEE Access*, vol. 10, pp. 25857–25871, 2022, doi: 10.1109/ACCESS.2022.3153675.
[52] B. A. H. Murshed, S. Mallappa, O. A. M. Ghaleb, and H. D. E. Al-ariki, "Efficient Twitter data cleansing model for data analysis of the pandemic tweets," in *Emerging Technologies During the Era of COVID-19 Pandemic* (Studies in Systems, Decision and Control), vol. 348. Cham, Switzerland: Springer, 2021, pp. 93–114, doi: 10.1007/978-3-030-67716-9_7.
[53] B. A. H. Murshed, H. D. E. Al-ariki, and S. Mallappa, "Semantic analysis techniques using Twitter datasets on big data: Comparative analysis study," *Comput. Syst. Sci. Eng.*, vol. 35, no. 6, pp. 495–512, 2020, doi: 10.32604/csse.2020.35.495.
[54] B. A. H. Murshed, N. Suresha, J. Abawajy, M. A. N. Saif, H. M. Abdulwahab, and F. A. Ghanem, "FAEO-ECNN: Cyberbullying detection in social media platforms using topic modelling and deep learning," *Multimedia Tools Appl.*, vol. 82, no. 30, pp. 46611–46650, May 2023, doi: 10.1007/s11042-023-15372-3.
[55] M. Dashtban, M. Balafar, and P. Suravajhala, "Gene selection for tumor classification using a novel bio-inspired multi-objective approach," *Genomics*, vol. 110, no. 1, pp. 10–17, Jan. 2018, doi: 10.1016/j.ygeno.2017.07.010.
[56] M. Rostami, S. Forouzandeh, K. Berahmand, and M. Soltani, "Integration of multi-objective PSO based feature selection and node centrality for medical datasets," *Genomics*, vol. 112, no. 6, pp. 4370–4384, 2020, doi: 10.1016/j.ygeno.2020.07.027.
[57] E. Hancer, B. Xue, M. Zhang, D. Karaboga, and B. Akay, "Pareto front feature selection based on artificial bee colony optimization," *Inf. Sci.*, vol. 422, pp. 462–479, Jan. 2018, doi: 10.1016/j.ins.2017.09.028.
[58] M. Amoozegar and B. Minaei-Bidgoli, "Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism," *Expert Syst. Appl.*, vol. 113, pp. 499–514, Dec. 2018, doi: 10.1016/j.eswa.2018.07.013.
[59] F. Han, W.-T. Chen, Q.-H. Ling, and H. Han, "Multi-objective particle swarm optimization with adaptive strategies for feature selection," *Swarm Evol. Comput.*, vol. 62, Apr. 2021, Art. no. 100847, doi: 10.1016/j.swevo.2021.100847.
[60] R. M. Rizk-Allah, A. E. Hassanien, and A. Slowik, "Multi-objective orthogonal opposition-based crow search algorithm for large-scale multi-objective optimization," *Neural Comput. Appl.*, vol. 32, no. 17, pp. 13715–13746, Mar. 2020, doi: 10.1007/s00521-020-04779-w.
[61] R. M. Rizk-Allah, R. A. El-Sehiemy, S. Deb, and G.-G. Wang, "A novel fruit fly framework for multi-objective shape design of tubular linear synchronous motor," *J. Supercomput.*, vol. 73, no. 3, pp. 1235–1256, Jul. 2016, doi: 10.1007/s11227-016-1806-8.
[62] R. M. Rizk-Allah, E. A. Hagag, and A. A. El-Fergany, "Chaos-enhanced multi-objective tunicate swarm algorithm for economic-emission load dispatch problem," *Soft Comput.*, vol. 27, no. 9, pp. 5721–5739, Dec. 2022, doi: 10.1007/s00500-022-07794-2.
[63] D. Rodrigues, L. A. M. Pereira, T. N. S. Almeida, J. P. Papa, A. N. Souza, C. C. O. Ramos, and X.-S. Yang, "BCS: A binary cuckoo search algorithm for feature selection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 465–468, doi: 10.1109/ISCAS.2013.6571881.
[64] C. Gunavathi and K. Premalatha, "Cuckoo search optimisation for feature selection in cancer classification: A new approach," *Int. J. Data Mining Bioinf.*, vol. 13, no. 3, p. 248, 2015, doi: 10.1504/ijdmb.2015.072092.
[65] M. A. E. Aziz and A. E. Hassanien, "Modified cuckoo search algorithm with rough sets for feature selection," *Neural Comput. Appl.*, vol. 29, no. 4, pp. 925–934, Jul. 2016, doi: 10.1007/s00521-016-2473-7.
[66] A. Alia and A. Taweel, "Enhanced binary cuckoo search with frequent values and rough set theory for feature selection," *IEEE Access*, vol. 9, pp. 119430–119453, 2021, doi: 10.1109/ACCESS.2021.3107901.
[67] A. Kumar, A. Jaiswal, S. Garg, S. Verma, and S. Kumar. (2022). *Sentiment Analysis Using Cuckoo Search for Optimized Feature Selection on Kaggle Tweets*. Accessed: Feb. 1, 2024. [Online]. Available: https://www.igi-global.com/chapter/sentiment-analysis-using-cuckoo-search-for-optimized-feature-selection-on-kaggle-tweets/308540
[68] M. N. Sudha and S. Selvarajan, "Feature selection based on enhanced cuckoo search for breast cancer classification in mammogram image," *Circuits Syst.*, vol. 7, no. 4, pp. 327–338, 2016, doi: 10.4236/cs.2016.74028.
[69] S. Salesi and G. Cosma, "A novel extended binary cuckoo search algorithm for feature selection," in *Proc. 2nd Int. Conf. Knowl. Eng. Appl. (ICKEA)*, Oct. 2017, pp. 6–12, doi: 10.1109/ICKEA.2017.8169893.
[70] A. C. Pandey, D. S. Rajpoot, and M. Saraswat, "Feature selection method based on hybrid data transformation and binary binomial cuckoo search," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 2, pp. 719–738, May 2019, doi: 10.1007/s12652-019-01330-1.
[71] M. Alzaqebah, K. Briki, N. Alrefai, S. Brini, S. Jawarneh, M. K. Alsmadi, R. M. A. Mohammad, I. ALmarashdeh, F. A. Alghamdi, N. Aldhafferi, and A. Alqahtani, "Memory based cuckoo search algorithm for feature selection of gene expression dataset," *Informat. Med. Unlocked*, vol. 24, Apr. 2021, Art. no. 100572, doi: 10.1016/j.imu.2021.100572.
[72] L. Wang, Y. Gao, J. Li, and X. Wang, "A feature selection method by using chaotic cuckoo search optimization algorithm with elitist preservation and uniform mutation for data classification," *Discrete Dyn. Nature Soc.*, vol. 2021, Jun. 2021, Art. no. e7796696, doi: 10.1155/2021/7796696.
[73] M. Kelidari and J. Hamidzadeh, "Feature selection by using chaotic cuckoo optimization algorithm with levy flight, opposition-based learning and disruption operator," *Soft Comput.*, vol. 25, no. 4, pp. 2911–2933, Oct. 2020, doi: 10.1007/s00500-020-05349-x.
[74] R. M. Aziz, N. P. Desai, and M. F. Baluch, "Computer vision model with novel cuckoo search based deep learning approach for classification of fish image," *Multimedia Tools Appl.*, vol. 82, no. 3, pp. 3677–3696, Jul. 2022, doi: 10.1007/s11042-022-13437-3.
[75] R. M. Aziz, "Cuckoo search-based optimization for cancer classification: A new hybrid approach," *J. Comput. Biol.*, vol. 29, no. 6, pp. 565–584, Jun. 2022, doi: 10.1089/cmb.2021.0410.
[76] K. Jawad, R. Mahto, A. Das, S. U. Ahmed, R. M. Aziz, and P. Kumar, "Novel cuckoo search-based metaheuristic approach for deep learning prediction of depression," *Appl. Sci.*, vol. 13, no. 9, p. 5322, Apr. 2023, doi: 10.3390/app13095322.
[77] V. Elyasigomari, D. A. Lee, H. R. C. Screen, and M. H. Shaheed, "Development of a two-stage gene selection method that incorporates a novel hybrid approach using the cuckoo optimization algorithm and harmony search for cancer classification," *J. Biomed. Informat.*, vol. 67, pp. 11–20, Mar. 2017, doi: 10.1016/j.jbi.2017.01.016.
[78] V. Jayaraman and H. P. Sultana, "Artificial gravitational cuckoo search algorithm along with particle bee optimized associative memory neural network for feature selection in heart disease classification," *J. Ambient Intell. Humanized Comput.*, Jan. 2019, doi: 10.1007/s12652-019-01193-6.

[79] J. Kennedy and R. C. Eberhart. (Oct. 1, 1997). *A Discrete Binary Version of the Particle Swarm Algorithm*. IEEE Xplore. Accessed: Dec. 5, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/637339/

[80] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[81] (2024). *Bioinformatics Laboratory*. Accessed: Feb. 1, 2024. [Online]. Available: https://fri.uni-lj.si/en/laboratory/biolab-27

[82] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Med.*, vol. 7, no. 6, pp. 673–679, Jun. 2001, doi: 10.1038/89044.

[83] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers, "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, no. 2, pp. 203–209, Mar. 2002, doi: 10.1016/s1535-6108(02)00030-2.

[84] A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerson, "Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 24, pp. 13790–13795, Nov. 2001, doi: 10.1073/pnas.191502998.

[85] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, Oct. 1999, doi: 10.1126/science.286.5439.531.

[86] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Nat. Acad. Sci. USA*, vol. 96, no. 12, pp. 6745–6750, Jun. 1999, doi: 10.1073/pnas.96.12.6745.

[87] M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. T. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub, "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning," *Nature Med.*, vol. 8, no. 1, pp. 68–74, Jan. 2002, doi: 10.1038/nm0102-68.

HUDHAIFA MOHAMMED ABDULWAHAB received the B.Sc. degree in information technology from the University of Science and Technology, Taizz, Yemen, in 2011, and the M.Sc. degree in computer applications from the University of Mysore, Mysore, India, in 2017. He is currently a Ph.D. Research Scholar with the Department of Computer Application, Ramaiah Institute of Technology (affiliated to VTU), Bengaluru, India. He is also working in artificial intelligence and data science, under the guidance of Dr. S. Ajitha. His area of interests includes machine learning, big data analytics, data mining, and the Internet of Things.

S. AJITHA received the B.Sc. degree in computer science from Madurai Kamaraj University, India, in 1991, the M.Sc. degree in computer applications and the M.Phil. degree in computer science from Manonmaniam Sundaranar University, India, in 1997 and 2003, respectively, and the Ph.D. degree in computer application from Visvesvaraya Technological University, India, in 2015. She is currently an Associate Professor with the Department of Computer Applications, Ramaiah Institute of Technology, India. She has more than 25 years of experience and teaching. She has published more than 30 technical papers in books, journals, and conference proceedings. Her research interests include multi agent systems, data analytics, data mining, and software performance engineering.

MUFEED AHMED NAJI SAIF received the B.Sc. degree in computer applications from Osmania University, Hyderabad, India, in 2010, the M.Sc. degree in information technology from Bharathiar University, India, in 2012, and the Ph.D. degree in computer applications (cloud computing) from Visvesvaraya Technological University, India, in 2023. He has more than six years of experience in research and academia. His area of interests include software engineering, distributed systems, cloud computing, networking, big data, the IoT.

BELAL ABDULLAH HEZAM MURSHED received the B.Sc. degree in computer science and information system from the University of Taiz, Yemen, in 2008, the M.Sc. degree in computer science from the University of Mysore, Mysore, India, in 2016, and the Ph.D. degree in computer science from the University of Mysore. He is currently an Assistant Professor with the Faculty of Engineering and Information Technology, Amran University, University of Science and Technology, Yemen. He has more than 12 years of experience in industry and teaching. His research interests include data mining, machine learning, NLP, topic modeling, deep learning, big data, and the Internet of Things.

FAHD A. GHANEM received the B.Sc. degree in computer science from Hodeidah University, Al Hudaydah, Yemen, in 2011, and the M.Sc. degree in computer science from the University of Mysore, India, in 2019. He is a Faculty Member with Hodeidah University. He is currently a full-time Research Scholar toward the Ph.D. degree with the Department of Computer Science and Engineering, PES College of Engineering (affiliated to the University of Mysore), Mandya, India, under the guidance of Dr. M. C. Padma. His area of interests include data mining, big data analytics, and machine learning.