

Binary Puma Optimizer: A Metaheuristic Approach for Gene Selection in Bioinformatics with Machine Learning Classification

Mohammad Wedyan

Jordan University of Science and Technology

Ryan Alturki

Umm Al-Qura University

Hasan Alyamani

King Abdulaziz University

Bandar Alshaw

Umm Al-Qura University

Ahmad Nasayreh

`ahmadnasayreh@correo.ugr.es`

University of Granada

Article

Keywords: Feature Selection, Puma Optimizer, gene Selection, Machine Learning, meta-heuristic optimization, and Support Vector Machine

Posted Date: May 26th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9534937/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Binary Puma Optimizer: A Metaheuristic Approach for Gene Selection in Bioinformatics with Machine Learning Classification

Mohammad Wedyan¹, Ryan Alturki², Hasan J. Alyamani³, Bandar Alshawhi⁴, and Ahmad Nasayreh^{5*}

¹Department of Cybersecurity, Faculty of Computer & Information Technology, Jordan University of Science and Technology, Irbid, Jordan, Email: mwedyan@just.edu.jo

²Department of Software Engineering, College of Computing, Umm Al-Qura University, Makkah 24382, Saudi Arabia, Email: rmturki@uqu.edu.sa

³Department of Information Systems, Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University, Rabigh 21911, Saudi Arabia, Email: hjalyamani@kau.edu.sa

⁴Department of Computer and Network Engineering, College of Computing, Umm Al-Qura University, Makkah 24382, Saudi Arabia, Email: bmhshawhi@uqu.edu.sa

⁵Department of Computer Engineering, Automation and Robotics, School of Computer and Telecommunication Engineering, University of Granada, Granada, Spain, Email: ahmadnasayreh@correo.ugr.es

*Corresponding author: ahmadnasayreh@correo.ugr.es

Abstract

Gene expression data is a matrix derived from DNA microarray analysis; a single chip can contain a thousand genetic instructions via DNA microarray technology. Each microarray experiment can simultaneously evaluate thousands of genes. Algorithms for pattern recognition are commonly employed to differentiate between samples of healthy and malignant patients using gene expression data. Gene expression data is categorized as high-dimensional data and typically includes irrelevant, noisy, and redundant genes. Such datasets confront the challenge of the application of machine learning algorithms. This is due to their requirement for extensive computational resources, which negatively impacts classification performance and obstructs the training and testing processes. Gene selection is necessary to avoid these risks. This work proposes a unique binary puma optimizer (BPo) to choose the top-ranked genes. The modified Bpo uses a U-shaped transfer function to transform the algorithm from continuous to discrete. Nine widely utilized microarray datasets are employed to evaluate the performance of the proposed technique. The Support Vector Machine (SVM) is employed to evaluate the efficacy of the selected gene subset for the classification task and as a fitness function. The effectiveness of BPo is evaluated by achieving the least number of genes and the least time. Furthermore, utilizing the identical datasets, the results of BPo are compared with ten state-of-the-art gene selection methods based on classification accuracy and the number of genes selected. The results indicate that the proposed method provides the optimal results for the nine datasets. The experiment's results demonstrated the efficacy of the proposed method in exploring the gene search space while determining the optimal gene pairs. To statistically support the data, we performed a statistical analysis employing the Friedman and Wilcoxon tests, highlighting the significance of the proposed technique in comparison to existing algorithms based on accuracy. This analysis indicates that the proposed BPo substantially improves gene selection performance.

Keywords Feature Selection, Puma Optimizer, gene Selection, Machine Learning, meta-heuristic optimization, and Support Vector Machine

Introduction

Feature selection is selecting an attribute or combination of characteristics that optimize data processing efficiency is vital [1]. In genetic research, the DNA microarray method lets one examine and evaluate more than a thousand genes from many cells and tissues in one experiment. Among other applications, the development of DNA microarrays has produced high-dimensional data in domains including clinical diagnosis and drug development [2]. Data generated by DNA microarray studies in high dimensions is known as gene expression data. It is used frequently to diagnose and

classify neoplastic disorders [3], [4]. Nevertheless, the data includes many insignificant, meaningless, and redundant genes that complicate the machine learning algorithm's capacity to create a predictive model depending on them. This leads to a decrease in performance in classification. Gene selection is the most often used method for resolving this difficulty. By removing unrelated and duplicated genes, gene selection keeps the most fundamental and relevant ones [5]. Further insights come from gene selection: (1) helping researchers and biologists clarify the molecular mechanisms connected with cancer gene expression; (2) possibly enable the creation of new treatments by thorough study of particular gene patterns; and (3) lowering of clinical costs. Gene selection techniques find great difficulty in the exponential increase of possible solutions in relation to the number of genes. Many researchers thus adapt current metaheuristic algorithms to find and choose near-optimal candidate gene subsets. A metaheuristic method is an algorithm framework independent of problems that combines several tactics to create optimization search algorithms [6].

Data has become so vital in the past ten years that it is considered finances. In conjunction with developments in machine learning and data mining, the amount of data has made data processing increasingly time-consuming. Companies are constantly trying to create more effective algorithms and enhance data processing methods to handle these difficulties. As both academic fields and commercial activities depend on data needing processing and analysis, the exponential growth in such areas is driving the improvement of machine learning and data mining.

Knowledge discovery from data depends on the conversion process, which consists of cyclical tasks including data translation, reduction, cleansing, and integration. These tasks are necessary to turn processable data into insightful analysis and useful knowledge [7]. The procedures described involve pre-processing; so, the outcomes directly affect the effectiveness of the next machine learning and data mining techniques. An increase in dimensions requires more time and effort for training machine learning and data mining algorithms, therefore increasing the computing cost of these operations. Therefore, data management must be carried out correctly. There are several ways researchers have developed to deal with the dimensionality challenge. Eliminating unnecessary data that may delay the categorization process [8], [9], and [10] is the aim of feature selection.

To enhance the accuracy of classifiers, feature selection a pre-processing process finds the most essential ones and removes the others. This method removes unnecessary components [11] thereby lowering computational complexity. The main approaches for feature selection are filter and wrapper strategies [12], [13]. Features subsets in the wrapper method are found using one or more learning methods. Although more effort is required, this method produces better results than the filter approach. Usually, wrapper-based feature selection is identified as an optimization issue [14]. Independent of a learning algorithm, the filter-based approach chooses significant features using information gain, mutual information, and other strategies [15], [16]. Although computationally cheap, this method shows worse performance than wrapper-based methods. selecting the most relevant subset of features could be trying given the objective of using the fewest features to achieve the maximum accuracy. The large time needed to find the best set of features makes feature selection regarded as an NP-hard task [17]. Choosing the best feature set requires analysis of $[2N - 1]$ permutations when N features exist [18]. As such, access to a high-performance meta-heuristic is crucial to solving this type of problem and saving computing resources.

Metaheuristic search methods seek to balance exploration (diversification), which investigates possible solutions outside the immediate vicinity, with exploitation (intensification), which conducts a thorough search inside the neighborhood to identify the best feasible alternatives. These two factors shape the approach to solve problems. Because of their better performance than exact methods [19], [20], [21], [22], and [23] metaheuristic algorithms are currently used to address the feature selection optimization problem. The no-free lunch (NFL) theorem shows that no one algorithm can solve all optimization problems, so novel or enhanced ways of generating high-quality solutions for the particular problem at hand are necessary.

The algorithm's efficiency is a crucial consideration while relying on the binary optimization approach. The proposed method performed remarkably well compared to other advanced optimization strategies created from related fields. Given the algorithm's success in tackling very challenging continuous optimization problems, we want to investigate whether its operators and optimization process could produce optimal solutions for binary optimization problems.

This work investigates the effects, advantages, and consequences of developing and applying a binary Po method through thorough testing on different high-dimensional datasets. The effectiveness of the method motivated this binarization even though it has had no relevance for the feature selection issue since its introduction. We provide a binary form of the Po that tackles the limited-continuity element of the feature selection problem. We validate the efficiency of the selected feature set using SVM as a classifier. The list of the work's main contributions is as follows:

- We propose an innovative binary approach utilizing the Puma Optimizer (BPO) to handle the feature-selection challenge.
- Employing a set of computational analytical metrics to evaluate the feature selection efficiency of the proposed BPO algorithm.
- Extensive testing was performed on nine gene datasets of varying sizes and dimensionalities to demonstrate the efficacy of the proposed feature selection algorithm.
- The relative superiority of the BPO is evaluated by comparing it with many existing metaheuristic algorithms approaches on same datasets.
- Statistical analysis to demonstrate the proposed approach's superiority and effectiveness.

The remaining part of this paper consists of the following sections: Section 2 presents the related work; Section 3 elaborates on the research background of Po concepts and the proposed methodology for gene selection. The evaluation of the proposed method and statistical analysis is conducted in the experiment and results in Section 5. The paper is concluded, and future work is recommended in Section 6.

Related Work

Several gene selection optimization methods have been suggested to enhance the accuracy of cancer classification. This section presents the latest research on selecting the most relevant genes for cancer diagnosis and categorization. The authors in [24] proposed integrating support vector machines with an optimization method inspired by bee colonies to assess microarray gene expression datasets. The categorization accuracy of the selected genes has been evaluated using the proposed model. The proposed model exceeded previous studies employing this strategy; however, the high complexity of microarray data resulted in significant processing costs for all strategies. Furthermore, in [25], two techniques for evaluating the categorization of high-dimensional microarray datasets are introduced: Particle Swarm Optimization (PSO) and a Genetic Algorithm (GA), both integrated with SVM as a classifier. The proposed model GA improved with SVM demonstrates superior performance compared to the proposed model GA added with SVM. Nonetheless, the extensive dimensionality of cancer gene expression datasets led both models to develop a classifier utilizing all genes at the same time, which is considered time intensive.

In this framework, the authors of [26] integrated SVM with Information Gain to achieve optimal classification performance for cancer. The most relevant and informative genes from the initial datasets were selected via IG, while redundant genes were eliminated through SVM. Finally, the extracted informative genes are evaluated using the LIBSVM classifier [27]. The authors of this study focused on selecting the minimal number of genes, although they ignored the crucial significance of achieving the maximum accuracy. Additionally, in [28], the authors employed correlation-based feature selection (CFS), as well as filter and wrapper algorithms, on microarray datasets for diffuse large B-cell lymphoma and acute leukaemia. These methods were integrated with several machine learning techniques, including SVM, decision trees, and naive Bayes. To reduce redundant information and choose informative genes, the authors applied the Singular Value Decomposition (SVD) method [29]. Furthermore, to choose the significant data items that enhance cancer classification efficacy, the scientists used the information gain (IG) algorithm. Nonetheless, the primary limitation of the work is the computationally expensive and slow application of SVD [30].

Although the outstanding outcomes of previous investigations, we contend that enhanced accuracy and improved performance can still be achieved with a reduced number of informative genes. The authors of [31] have examined the optimization problem of image contrast enhancement. They have employed the Barnacles Mating Optimiser (BMO) algorithm to solve this issue. Their findings indicate that this method outperforms others, including adaptively adjusted histogram equalization, genetic algorithms, artificial bee colony algorithms, modified histogram equalization,

and cuckoo searches. The authors of [32] presented an application based on a BMO algorithm to address the established economic dispatch (ED) problem in power system operation. The optimal power generation combination is determined through BMO to reduce expenses while complying with any regulations. The evaluation results indicate that, compared to other contemporary algorithms, the BMO technique achieves the highest efficiency. The authors of [33] proposed a novel Laplacian barnacles mating algorithm (NLBMA) utilizing a conscious neighborhood technique to address solar cell models. The NLBMA effectively showcased its reliability and efficiency in extracting parameters from diode models in comparison to other methods.

3. Methodology

This section provides a detailed description of the binarization technique we propose to utilize with the Po algorithm. After establishing the search space, the methodology for generating its binary representation will be explained. The binary variant of Po is afterwards generated and incorporated into the binary search matrix. The variant can generate a discrete map from the continuous space due to the provided transformation functions.

3.1 Background of Puma Optimization

3.1.1 Inspiration

The Puma, a felid species belonging to the subfamily Felinae, inhabits the American continent. As depicted in Figure 1, this creature is also referred to by alternative names, including mountain lion and cougar. The natural habitats of this species encompass the Andes Mountains in South America and the Yukon region in Canada. The mountain lion is the largest feline species in the Americas, while the jaguar holds the second position in terms of size. The puma is a versatile species that consumes other animals and can thrive in several settings. While it is mostly nocturnal, it can also be observed during daylight hours. The puma is a stealthy predator that mostly preys on deer [34]. However, it feeds on small prey, such as insects and rodents, as well as domesticated animals. The creature has a strong preference for rocky and densely vegetated areas, which provide ideal conditions for ambushing its prey. Additionally, it dwells in expansive grasslands. Puma, like other cats, has a wide territory that she guards. Its population density is limited, and in some locations, it competes with other predators like wolves, alligators, jaguars, and American black bears. Mountain lions were threatened in many locations because of widespread hunting and the establishment of human settlements following European arrivals in America. Apart from the small population in Florida, the eastern puma subspecies has disappeared from eastern North America. The puma's range has gradually extended from west to east, with a rebound along the Atlantic coast in Connecticut. Puma is a petite and swift cat. This is the fourth-largest cat. An adult puma's shoulder height ranges between 60 and 90 cm. The male puma's tail, measured from tip to tip, reaches roughly 2.4 meters. A female puma normally grows to be between 1.50 and 2 meters long. The tail's length ranges from 63 to 95 cm. Male pumas often weigh between 55 and 100 kg. Females weigh an average of 55 kg, ranging from 29 to 64 kg. The puma is larger in polar zones and smaller in tropical habitats. The largest puma, weighing 105.2 kg, was hunted in 1901. Pumas are normally smaller, weaker, and have less muscle mass than jaguars; yet their size can fluctuate depending on the environment. Pumas are not classified as big cats, despite their huge size, because cougars lack a special larynx and laminar bone that permit them to roar like leopards. However, Puma can sprint. They are searching for food search. However, they frequently ambush while hunting. When hunting, they move through shrubs, trees, and other dense vegetation before lunging behind their victim to suffocate it with their fangs. The puma then uses its strong jaws to grab any small prey by the neck and smash it to the ground. Pumas often catch a hefty animal every two weeks. After relocating its meal to a specified spot and covering it with plants, the puma returns to feast in a few days. Cougars, like most cats, are solitary creatures. Mature animals rarely interact with each other; only moms and puppies live in packs. Despite being primarily alone, they share prey and appear to form little communities around the dominant male's realm. In these fields, cougars interact more than foreign cougars do. A puma's range varies from 25 square kilometers (9.7 square miles) to 1,300 square kilometers (500 square miles) for males. Male cougars define a permanent territory that overlaps with that of female cougars. The amount of foliage and prey in the area, as well as the overall number of pumas, dictate the boundaries of a puma's territory. Large males' territories extend from 150 to 1000 square kilometers (58 to 386 square miles), whereas female territories are typically half the size. One study revealed that the number of pumas varied by region, ranging from 0.5 to 7 animals per 100 km² (39 square miles).

3.1.2 Mathematical model

This subdivision provides a mathematical description of the PO method that is based on actual puma hunting operations. Meta-heuristic optimization methods modify existing exploration settings and propose random solutions based on natural occurrences. The PO optimizer method is the first of its kind since it includes a revolutionary phase change mechanism for exploration and exploitation. Two unique approaches were used to carry out optimization efforts during the exploitation and exploration phases. The PO method portrays the optimal solution as a male puma and considers the entire optimization space to be its territory. Furthermore, other solutions (X_i) have been discovered as pumas in female form. Each iteration of this method uses a phase transition mechanism to assign all solutions to either the exploration or exploitation phases. The stages have been carefully and deliberately determined. At each experimental level, several optimization algorithms and processes were applied, drawing inspiration from pumas' actual behavior. Then, to help readers appreciate the optimization processes done by PO, mathematical concepts and formulas are introduced. Fig 3 presents an overview of the optimization technique and PO approaches.

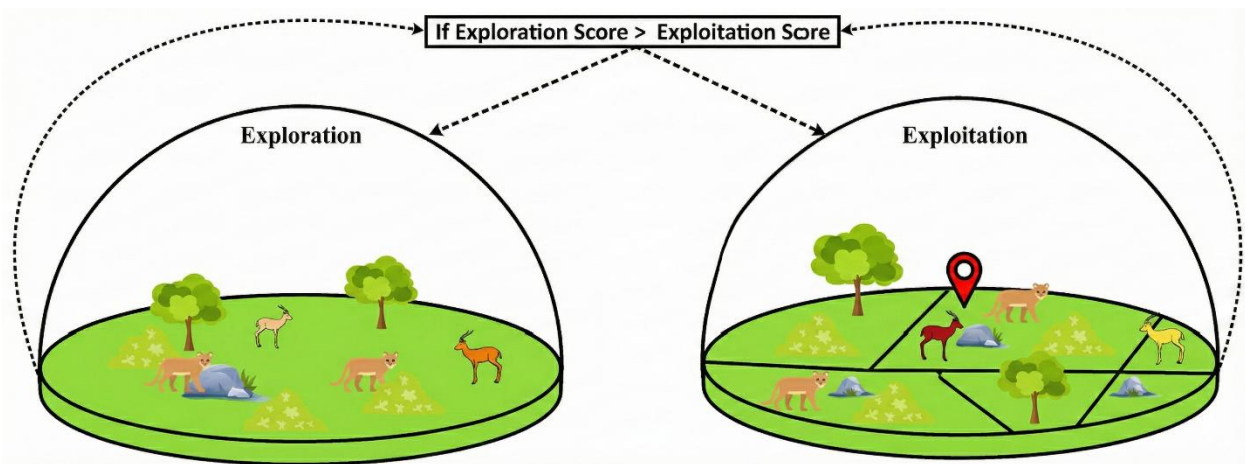


Fig 1. PO optimization strategy [34]

3.1.3 Puma intelligence (phase change mechanism)

Pumas have a strong memory and are incredibly intelligent. Depending on their previous experience, they typically hunt in places where hunting is more likely. Hunting missions with a specified purpose may necessitate revisiting previously studied and concealed regions. Alternatively, go to a more current site where he has not gone hunting in earlier levels. In this essay, we looked at the exploration phase for traveling to new places and the exploitation phase for puma journeys to previously promising destinations. We take pumas' intellect and memory as inspiration to design an innovative hyper-heuristic method (explained in full below) that allows for phase shifts. The recommended strategy incorporates a two-component heuristic selection mechanism: intensification and diversity. To score, it uses reward and punishment procedures. The phase transition component is modeled after cougars' intelligence. It takes two approaches. The first is that cougars hunt while exploring new territory because they lack experience and energy. As previously indicated in the part of the first generation, they ambush prospective regions due to their inexperience.

3.1.3.1 Unexperienced phase

Puma had limited experience in its early life because of its atypical living habitat and lack of knowledge about the locations of hunts inside its region, and it regularly engaged in concurrent exploratory activities. In contrast, it seeks ideal hunting areas. In the first three iterations of the Puma algorithm, exploration and exploitation are carried out concurrently until initialization is completed in the phase transition phase. This component requires only two functions ($f1$ and $f2$), which are determined using equations because the exploitation and exploration phases are selected in each iteration. (1-4).

$$f1_{Explr} = PF_1 \cdot \left(\frac{Seq_{CostExplr}^1}{Seq_T} \right) \quad 1$$

$$f1_{Explt} = PF_1 \cdot \left(\frac{Seq_{CostExplt}^1}{Seq_T} \right) \quad 2$$

$$f2_{Explr} = PF_2 \cdot \left(\frac{Seq_{CostExplr}^1 + Seq_{CostExplr}^2 + Seq_{CostExplr}^3}{Seq_T^1 + Seq_T^2 + Seq_T^3} \right) \quad 3$$

$$f2_{Explt} = PF_2 \cdot \left(\frac{Seq_{CostExplt}^1 + Seq_{CostExplt}^2 + Seq_{CostExplt}^3}{Seq_T^1 + Seq_T^2 + Seq_T^3} \right) \quad 4$$

The variable SeqCost values for each stage of exploration and exploitation are defined by equations (1-4). Another variable with a constant value is SeqTime, which we have set to one. Before beginning the optimization process, the fixed-value parameters PF1 and PF2 must be set. They are used to rank all f1 and f2 functions.

$$Seq_{CostExplr}^1 = |Cost_{Best}^1 - C_{Explr}^1| \quad 5$$

$$Seq_{CostExplr}^2 = |Cost_{Best}^2 - C_{Explr}^1| \quad 6$$

$$Seq_{CostExplr}^3 = |Cost_{Best}^3 - C_{Explr}^2| \quad 7$$

$$Seq_{CostExplt}^1 = |Cost_{Best}^1 - C_{Explt}^1| \quad 8$$

$$Seq_{CostExplt}^2 = |Cost_{Best}^2 - C_{Explt}^1| \quad 9$$

$$Seq_{CostExplt}^3 = |Cost_{Best}^3 - C_{Explt}^2| \quad 10$$

Equations (5 and 8) use six variables to compute the cost of the ideal starting solution: $Cost_{Explr}^1$, $Cost_{Explr}^2$, $Cost_{Explr}^3$, $Cost_{Explt}^1$, $Cost_{Explt}^2$, and $Cost_{Explt}^3$. In phases 1, 2, and 3, the cost of the optimum solution that results from each exploitation and exploration phase is repeated. Following the third iteration, only one of the investigation and exploitation phases is chosen for further action based on the findings of the functions f1 and f2. Equations 11 and 12 are applied to define the two stages of exploitation and exploration based on prior Pumas' useful experiences.

$$Score_{Explr} = (PF_1 \cdot f1_{Explr}) + (PF_2 \cdot f2_{Explr}) \quad 11$$

$$Score_{Explt} = (PF_1 \cdot f1_{Explt}) + (PF_2 \cdot f2_{Explt}) \quad 12$$

After obtaining the values of $Score_{Explr}$ and $Score_{Explt}$ (through equations 11 and 12), move on to the exploitation and exploration phases. If $Score_{Explt} \geq Score_{Explr}$, move on to the exploitation phase. Otherwise, move on to the exploration phase. Note that, as of the third iteration, every step will have been capable of developing solutions that are greater than the population. This problem will be resolved as, at the third iteration, the total cost of solutions developed within the last two phases is calculated, and only solutions from the best pool are kept as equivalent to the population that will replace the existing solutions.

3.1.3.2 Experienced phase

Pumas continue with the next generation of its population after the first pass of the optimization process. In the next generation of Pumas, the phase to be altered and the single phase to be used for the optimization process are selected. In the phase of the experiment, functions f_1 , f_2 , and f_3 are used for the generation of scores. The first function defines the escalation element and hence prioritizes the importance of either of the two selected exploration and exploitation phases that have outperformed the others as shown in equation (1).

$$f_{1t}^{exploit} = PF_1 \cdot \left| \frac{Cost_{old}^{exploit} - Cost_{new}^{exploit}}{T_t^{exploit}} \right| \quad 13$$

$$f_{1t}^{explore} = PF_1 \cdot \left| \frac{Cost_{old}^{explore} - Cost_{new}^{explore}}{T_t^{explore}} \right| \quad 14$$

The amounts of the initial function relating to the exploration or exploitation phases are denoted by $f_{1t}^{exploit}$ and $f_{1t}^{explore}$ in Eqs. (13 and 14), where t denotes the current iteration number. The costs of the optimal choice before an improvement in the current option are $Cost_{old}^{exploit}$ and $Cost_{old}^{explore}$. However, the costs associated with the optimal option identified after evaluating the existing options are $Cost_{new}^{exploit}$ and $Cost_{new}^{explore}$. The counts of unselected iterations between the previous selection and the current selection are $T_t^{exploit}$ and $T_t^{explore}$. Before the optimization procedure, the user-adjustable parameter PF_1 must be assigned a value between 0 and 1. This parameter dictates the significance of the initial function, which is prioritized as its value increases and decreases as its priority reduces.

The values of initial functions related to exploration and exploitation phases are represented by $f_{1t}^{exploit}$ and $f_{1t}^{explore}$ through Equations (13) and (14). Here, t is used to represent the current number of iterations. The cost values of optimality before improvement in terms of the current option are expressed as $Cost_{old}^{exploit}$ and $Cost_{old}^{explore}$. At the same time, the values of optimality considering the optimality of available options are given as $Cost_{new}^{exploit}$ and $Cost_{new}^{explore}$. Here, $T_t^{exploit}$ and $T_t^{explore}$ describe the number of unselected iterations between last and present selection. Prior to commence of the optimization algorithm, a user-controlled variable PF_1 is to be defined with a value within [0, 1]. This variable defines the priority of initial function values.

The phase that exceeds the other priority phase is the outcome of the second function, which also highlights the resonance component. Satisfactory performance is systematically evaluated and assessed. Consequently, this feature could help the selection process throughout the exploitation phase. Equations 15 and 16 are employed to compute the second function.

$$f_{2t}^{exploit} = PF_2 \cdot \left| \frac{(Cost_{old,1}^{exploit} - Cost_{new,1}^{exploit}) + (Cost_{old,2}^{exploit} - Cost_{new,2}^{exploit}) + (Cost_{old,3}^{exploit} - Cost_{new,3}^{exploit})}{T_{t,1}^{exploit} + T_{t,2}^{exploit} + T_{t,3}^{exploit}} \right| \quad 15$$

$$f_{2t}^{explore} = PF_2 \cdot \left| \frac{(Cost_{old,1}^{explore} - Cost_{new,1}^{explore}) + (Cost_{old,2}^{explore} - Cost_{new,2}^{explore}) + (Cost_{old,3}^{explore} - Cost_{new,3}^{explore})}{T_{t,1}^{explore} + T_{t,2}^{explore} + T_{t,3}^{explore}} \right| \quad 16$$

The secondary function linked to the exploration process, or the exploitation step, is represented by $f_{2t}^{exploit}$ and $f_{2t}^{explore}$, as given in equations (15) and (16), with t being the current number of iterations. Costs linked to the optimal solution prior to any improvements within the existing selection process for the exploration and exploitation phases are given as $(Cost_{old,1}^{explore})$ and $(Cost_{old,1}^{exploit})$, while the optimal solution prior to any improvements within an earlier selection process for the exploration and exploitation phases is given as $Cost_{old,2}^{exploit}$ and $Cost_{old,2}^{explore}$. Furthermore, the costs of the optimal Puma prior to improvements within the two earlier selection processes for the exploration and

exploitation phases are given as $Cost_{old,3}^{exploit}$ and $Cost_{old,3}^{explore}$. These variables, $Cost_{new,1}^{exploit}$ and $Cost_{new,1}^{explore}$, are costs associated with the optimal solution obtained after refinement of current selection made in both exploration and exploitation phases, respectively. $Cost_{new,2}^{exploit}$ and $Cost_{new,2}^{explore}$ are costs associated with most effective search agent obtained after refinement of a previous selection in exploration and exploitation phases, respectively. $Cost_{new,3}^{exploit}$ and $Cost_{new,3}^{explore}$ are costs associated with best solution obtained after refinement of first two selection made within exploration and exploitation phases, respectively. In both exploration and exploitation phases, variables $T_{t,1}^{exploit}$ and $T_{t,1}^{explore}$ show number of unselected iterations from prior to current selection. Similarly, variables $T_{t,2}^{exploit}$ and $T_{t,2}^{explore}$ show number of unselected iterations from two selections prior to one selection prior. Variables $T_{t,3}^{exploit}$ and $T_{t,3}^{explore}$ show number of unselected iterations from three selections before two selections prior in exploration and exploitation phases, respectively. Before beginning with optimization algorithm, parameter PF_2 is to be assigned a value in range [0,1] since this parameter defines range for second function. This is also given a higher priority since it keeps on varying according to its assigned priority.

The selection of a single-phase leads to confinement in a local optimum; therefore, the third function in the selection process emphasizes the importance of diversity, indicating that phases not frequently chosen yet possess the potential for selection over multiple iterations. The function is explained in Equations 17 and 18.

$$f_{3t}^{exploit} = \{if \text{ selected}, f_{3t}^{exploit} = 0 \text{ otherwise}, f_{3t}^{exploit} + PF_3 \quad 17$$

$$f_{3t}^{explore} = \{if \text{ selected}, f_{3t}^{explore} = 0 \text{ otherwise}, f_{3t}^{explore} + PF_3 \quad 18$$

The third function pertaining to the exploitation or exploration process is represented as f_3 (exploit t and f_3) (explore t); these are given by Equations (17) and (18); the variable t represents the current iteration number. From Equations (17) and (18), if either exploration or exploitation is not chosen, then the value of the third function corresponding to each phase increases by a certain value given by the parameter PF_3 in every iteration, whereas its value is set to zero otherwise. Prior to the optimization process, it is necessary to initialize the value of the user-controllable parameter PF_3 within the range [0, 1].

By Equations (19) and (20), the cost related to the phase change function is evaluated.

$$F_t^{exploit} = (\alpha_t^{exploit} \cdot (f_{1t}^{exploit} + f_{2t}^{exploit})) + (\gamma_t^{exploit} \cdot (lc \cdot f_{3t}^{exploit})) \quad 19$$

$$F_t^{explore} = (\alpha_t^{explore} \cdot (f_{1t}^{explore} + f_{2t}^{explore})) + (\gamma_t^{explore} \cdot (lc \cdot f_{3t}^{explore})) \quad 20$$

$$lc = [(|Cost_{old} - Cost_{new}|)^{exploitation}, (|Cost_{old} - Cost_{new}|)^{exploration}], 0 \notin lc \quad 21$$

$$\alpha_t^{exploit,explore} = \{if F^{exploit} > F^{explore}, \alpha^{exploit} = 0.99, \alpha^{explore} = [\alpha^{explore} - 0.01, 0.01] \text{ otherwise}, \alpha^{exploit} = 0.99, \alpha^{explore} = [\alpha^{explore} - 0.01, 0.01] \quad 22$$

$$\gamma_t^{exploit} = 1 - \alpha_t^{exploit} \quad 23$$

$$\gamma_t^{explore} = 1 - \alpha_t^{explore} \quad 24$$

Total cost for each phase of exploration and exploitation is calculated using Equations 19 and 20 respectively. The values of a and d , which represent the parameters for phases of exploration and exploitation, are variable during the search process, as they change based on the results obtained from each phase. Diversity plays a key role, and if it is not sufficient, then the role of diversity decreases and the value of resonance increases, as given by Equation 21, given that the value of the parameter α approaches unity. From Equation (22), the value of the exploitation function's parameter α is associated with a linear cost, which is 0.01, if the cost related to the exploration phase function is greater than the cost related to the exploitation function. The value of the parameter α corresponding to the exploitation phase approaches its highest value, which is close to unity. The above process is reversed if the cost related to the exploitation phase function is greater than the cost related to the exploration phase function. The value lc is defined as a group of values not containing zero, i.e., $0 \notin lc$, and these values represent a compilation of computational cost differences, which result from benefits received during the phases of exploitation and exploration, respectively.

3.1.3.3 Exploration

Pumas often go large distances within their region to hunt and gather food. This search could involve traveling regions of his territory where he has previously hunted, or it may necessitate a search effort in unknown places within his territory where hunting and food have yet to be discovered. The actions of these pumas have prompted our exploration department to seek food. organized at this point, pumas either randomly approach other pumas to utilize their prey or search their region for nutrition. Consequently, the puma either enters the search region randomly or hunts for food within nearby areas. The population is first organized in ascending order, and Puma employs Equation (24) to enhance its solutions throughout the exploration phase.

$$\begin{aligned} \{ \text{if } rand_1 > 0.5, Z_{i,G} = R_{Dim} * (Ub - Lb) + Lb \text{ otherwise, } Z_{i,G} \\ = X_{a,G} + G * (X_{a,G} - X_{b,G}) + G * (((X_{a,G} - X_{b,G}) - (X_{c,G} - X_{d,G})) + (X_{c,G} - X_{d,G}) - (X_{e,G} \\ - X_{f,G})) \} \end{aligned} \quad 25$$

$$G = 2 * rand_2 - 1 \quad 26$$

Ub and Lb indicate the bottom and upper limits of the problem, respectively, whereas R_{Dim} consists of randomly generated numbers within the problem's dimensions, ranging from 0 to 1. A different randomly generated integer between 0 and 1 is $rand_1$. The responses ($X_{a,G}$, $X_{b,G}$, $X_{c,G}$, $X_{d,G}$, $X_{e,G}$, and $X_{f,G}$) are selected randomly from the total population. Equation (26) is utilized to compute G , where $rand_2$ represents a uniformly distributed random variable ranging from 0 to 1. Equation (25) indicates that one of the two equations is selected to produce an alternative solution depending on the prevailing conditions, thereafter, utilized to refine the existing respond.

$$X_{new} = \{ \text{if } Z_{i,G,ifj} = j_{rand} \text{ or } rand_3 \leq U \text{ otherwise, } X_{a,G} \} \quad 27$$

$$NC = 1 - U \quad 28$$

$$P = \frac{NC}{N_{pop}} \quad 29$$

$$\text{if } CostX_{new} < CostX_i, U = U + p \quad 30$$

Eq. 27 shows a solution from Eq. (25), $Z_{i,G}$. The problem parameters generate j_{rand} , a random integer. The integer $rand_3$ is also uniformly distributed and randomly produced from 0 to 1. NC is calculated using Equation (28). Pre-optimization parameter value U ranges from 0 to 1. As required by Equation (30), Equations (28–30) increase the number of dimensions substituted by unique solutions each cycle. Equation (29) provides Pumas' aggregate count, N_{pop} . Eq. (30)'s criterion that adjusts solution dimensions only if it's met determines how to improve the solution. This method avoids local optima and increases product variety. At the start of each exploration iteration, search agents

are ordered by cost in an increasing order to achieve high-quality results. Due to the low U parameter, high-quality solutions change little, as shown in Eqs. (28–30). However, increasing this parameter changes to higher-cost solutions numerous times. This strategy helps identify important ideal areas by examining suboptimal solutions in the issue space. Equation (30) will remain insufficient until Pumas exceeds competitive options, as improving it eliminates the need for unnecessary innovations. However, high-quality solutions have mainly remained unchanged and aim to prevent local optimality. Finally, Equation (31) replaces the current solution with the created ones.

$$X_{a,G} = X_{new}, \text{ if } X_{i,new} < X_{a,G} \quad 31$$

Equation (31) indicates that if the new production solution is less costly than the current one, it will overtake it.

3.1.3.4 Exploitation

The PO approach employs two separate operators to improve solutions during the exploitation phase. These two processes depend on the dual behaviors of pumas: running and ambush hunting. Pumas in the wild endeavor to ambush their prey by hiding themselves in plants or by positioning themselves on rocks and trees. It sporadically hunts its prey, a behavior that can be duplicated using Eq. (32).

$$\begin{aligned} X_{new} = \{ \text{if } rand_4 \geq 0.5, X_{new} &= \frac{\left(\frac{\text{mean}(Sol_{total})}{\mu_2} \right) \cdot x_1^r - (-1)^\beta \cdot x_i}{1 + (\alpha \cdot rand_5)} \text{ otherwise, if } rand_6 \geq L, X_{new} \\ &= Puma_{male} + (2 \cdot rand_7) \cdot \exp \exp(randn_1) \cdot x_2^r - x_i \text{ otherwise, } X_{new} \\ &= (2 \cdot rand_8) \cdot \frac{(F_1 \cdot R \cdot X(i)) + F_2 \cdot (1 - R) \cdot Puma_{male}}{(2 \cdot rand_9 - 1 + rand_2)} - Puma_{male} \end{aligned} \quad 32$$

The two strategies used in the Population Optimization (PO) approach are described in Equation (32). The purpose of adopting a division operation in Equation (32) is to describe Case 1 in relation to hunting in the first mode, during exploitation, while sprinting, and for ambush tactics. Equation (32) shows that based on $rand_5$, which is a random number generated in the interval $[0, 1]$, when it is greater than 0.5, quick running is selected, or else ambush is used. The strategy involves two different operations: (i) simulating short jump operations towards hunting performed by other pumas, and (ii) long jump operations towards hunting performed by the best pumas. Equation (32) shows that mean functions are described in terms of mean, Sol_{total} is the sum of total solutions, and $Npop$ is the total number of population size that is essential in optimized strategies. The expression x_1^r is any random solution from total population size. Value b is any binary number that is produced at random with 0 or 1. The parameters α and L are fixed parameters that must be adjusted before jumping into optimized strategies. The expression x_i is any present solution in this iterative stage. $Puma_{male}$ is any optimal solution towards total population size. The random numbers generated in intervals $[0, 1]$ are $rand_4$, $rand_5$, $rand_6$, $rand_7$, $rand_8$, and $rand_9$. The expression $\exp$ describes any exponential function. The expression x_2^r is any random solution that is generated in Equation (33), whereas $randn_1$ and $randn_2$ are random numbers generated in terms of normally distributed dimension with any number of problems.

$$round(1 + (Npop - 1) \cdot rand_{10}) \quad 33$$

Each element of X is approximated to the nearest integer in Eq. (33), where $rand_{10}$, represents a randomly generated number between 0 and 1, and $Npop$ is the total population of Pumas.

Equations (34–36) are employed to compute (R , $F1$, and $F2$) accordingly.

$$R = 2 \cdot rand_{11} - 1 \quad 34$$

$$F = randn_3 \cdot \exp(2 - Iter. (\frac{2}{MaxIter})) \quad 35$$

A random number within the problem's dimensions and the normal distribution is represented by $randn_2$ in Eq. (35). The current iteration is denoted as $Iter$, while the total iterations necessary for the completion of the optimization process are represented by $MaxIter$. The exponential function is denoted by $\exp$.

$$F2 = w \cdot (v)^2 \cdot \cos((2 \cdot rand_{12}) \cdot w) \quad 36$$

$$w = randn_4 \quad 37$$

$$v = randn_5 \quad 38$$

Both $randn_4$ and $randn_5$, which represent randomly produced numbers from the normal distribution inside the specified issue dimensions, are in Eqs. (36–38). The cosine function is represented as $\cos$, and the randomly generated number ($rand_{12}$) varies from 0 to 1.

Finally, if the newly developed solutions are more cost-effective than the current solution at the conclusion of this phase, they will be replaced. approach 2 presents the pseudocode for the proposed approach. The Puma optimizer is comparably more complex to deploy than other optimizers due to its multiple phases and further processes required for the development of intelligent systems. The cost function shows suitable computational complexity, as detailed in the relevant section, and is utilized only once per search agent in each iteration. Flowchart 2 belongs to the overarching procedure of the puma optimizer, whilst Algorithm 3 presents the pseudo-code of the optimizer.

3.2 Binarization of Po

An improved version of the algorithm has been developed to improve solutions inside a discrete solution space. By combining the ability of the original Po with several additional operators, the first phase defines transformation functions, thus allowing the transfer of the optimization process and solution representation from a continuous to a discrete format. This is crucial for the new approach to handle problems specific to feature selection. The other modeled technique for obtaining the new BPo variation is changing the fitness function. Choosing the best possible solution overall depends on evaluating the effectiveness of every prospective one. We offer a specification of the fitness function to handle the details of the present state. We indicate and examine the BPo's algorithmic structure. With S_{best} being the optimum continuous solution provided by the continuous Po, the sigmoid function below is used to convert the continuous solution produced by the Po algorithm into binary form.

The two control parameters, α and β , have been used to derive the U-shaped function [35]. α is used to decide the slope value of the function, and β decides the basin of attraction of the U-shaped function. Table 1 shows the mathematical equations for the functions, where the mechanism of the function operation is represented by 1, which symbolizes determinacy and 0, which symbolizes indeterminacy. The choice of the function is attributed to the low computational complexity introduced by the function. In this analysis, an experimental method is used to determine the best function among the eight functions with the following value assignments for the function $U(x) = (\alpha = 1, \beta = 2, 4, 6, 8)$. From the experimental function equations, the value of $U(x) = (\alpha = 1, \beta = 8)$.

$$U(x) = \alpha |x|^\beta \quad 39$$

Fig 2 illustrates many U-shaped transfer functions characterized by different slopes and maximum points. Two control parameters α and β , have been employed in the equation of Equation 8. The previous parameter allows for the modification of the optimum point of the U-shaped curve. The probability of flipping the bit increases with the velocity at which the U-shaped function achieves the saturation point. The unexpected changes in the variables encourage exploration. The slope increases about the parameter α , as illustrated in Fig 2.

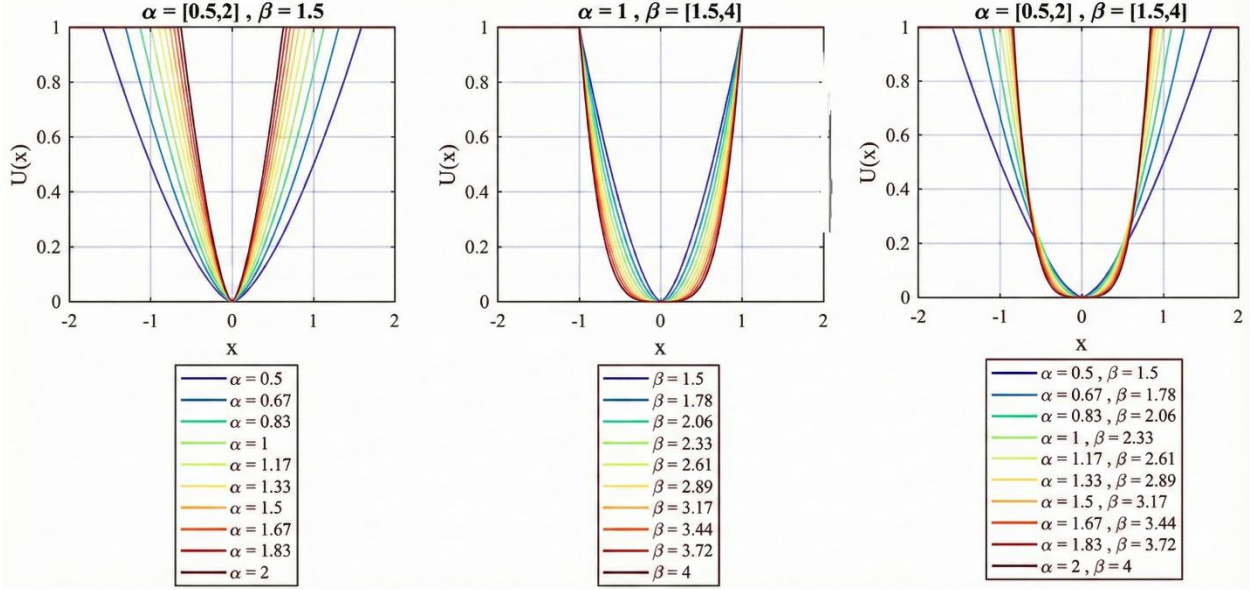


Fig 2. U-shaped transfer functions [35]

Table 1. S-Shaped Transfer Functions

S-Shaped	Function
S1	$\frac{1}{1 + e^{-2x}}$
S2	$\frac{1}{1 + e^{-x}}$
S3	$\frac{1}{1 + e^{\frac{-x}{2}}}$
S4	$\frac{1}{1 + e^{\frac{-x}{3}}}$

3.3 Feature Selection

The proportion of the dataset used to compute the fitness and cost functions depends on how well the classifier performs on a subset of the dataset. The primary classifier in this system is the SVM algorithm. Our findings are presented in this article, which looks at how different well-known classifiers affect the feature selection problem. The number of selected features for an individual can be found using the Eq. 40 provided. It requires selecting the variable $1^{ind_j^k}$, which indicates the number of feature sites with a value of 1, and D , which indicates the dataset's feature size dimension.

$$fs_i = \sum_{k=0}^D \frac{1^{ind_j^k}}{D} \quad 40$$

The SVM technique finds groups of items with comparable attributes to solve the classification problem. A more comprehensive overview of the computer system and experimental setup used to assess this methodology will be given in the next section.

3.4 Fitness Function

The primary goal of gene selection techniques is to decrease the gene number while enhancing classification accuracy. To do this, many studies combined the gene subset size and classification accuracy into a singular weighted function, which is used as a fitness function to assess each potential gene subset, as seen in Eq. 41 below:

$$F = h1 \frac{|s|}{|f|} + h2 \text{Error}(D) \quad 41$$

where s represents the size of the suggested gene subset and f denotes the number of genes in the experimental dataset. The term $\text{Error}(D)$ corresponds to the classification error, which is evaluated using the SVM classifier. The two weighting variables, $h1$ and $h2$, are associated with the size of the candidate gene subset and the classification accuracy, respectively. In this study, $h1=\beta=1$ and $h2=\alpha=0.001$, as suggested in previous works [45, 51].

3.5 Data Preprocessing

Data preprocessing includes two very important phases, namely handling missing values and normalization. The missing values play a very important role in determining the efficiency of models. Missing values were treated using estimation methods, where missing values were replaced with feature values such as feature means and medians. This ensures that missing values do not affect the machine learning algorithm. Normalization is a phase that uses Min-max Scaling to scale values within a fixed interval of $[0, 1]$ [36]. This is very appropriate and demonstrates scaled features, thereby increasing its efficiency. The selection of Min-max Scaling is due to its ability to scale features to a fixed range without having any effect on relationships among features. This is further supported as shown in Equation 42.

$$x_{scale} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad 42$$

Normalization data is represented by x_{scale} , whereas the minimum and maximum values of the dataset are shown by x_{min} and x_{max} , respectively.

3.6 The Proposed Method

To enhance feature selection and classification accuracy, we integrated BPo with a Support Vector Machine (SVM). The SVM classifier evaluates each chimpanzee in the Po algorithm as a potential feature subset. SVM was picked as the classifier due to its capacity to adapt to overfitting, effectiveness in both linear and non-linear classification using kernel functions, and ability to manage high-dimensional data. This makes it particularly suitable for evaluating the relevance of feature subsets in optimization processes. This approach iteratively enhances the feature set to optimize detection performance. Furthermore, it constitutes a substantial enhancement to machine learning-based optimization algorithms that select optimal genes. Furthermore, it proposed converting the Po algorithm by the application of two-step transfer functions to convert it from continuous to discrete and then to binary form. Algorithm 1 provides the pseudocode for BPo.

Algorithm 1. Pseudo-Code of Binary Puma Optimizer (BPo)

Algorithm 1: Pseudo-Code of Binary Puma Optimizer (BPo)

#Po sitting

Inputs: The maximum iterations, parameter configurations, and population size N .

Outputs: The location and fitness potential of Puma

#Initilization

Apply the formula x_i ($i = 1, 2, \dots, N$) to generate a random population. Evaluate the fitness levels of Puma.

for iter = 1:3

```

#Explorartion phase
Sort ascending population
Calculate p
for (each Puma ( $x_i$ )) do
    Select four solutions randomly
    Calculate the new vector by Eq. (24).
    Check the boundary of the new vector position
    Update the current solution by Eq. (26).
    Calculate the cost of the updated solution
    if New  $x_i$  Cost <  $x_i$  Cost
         $x_i$  = New  $x_i$ 
    else
         $U = U + P$ ;
    end
end

#Exploitation phase
for (each Puma ( $x_i$ )) do
    Calculate R, F1, and F2 by Eqs. (34), (35), and (36) respectively
    Produce New  $x_i$  by Eq. (32)
    Calculate the cost of New  $x_i$ 
    if New  $x_i$  Cost <  $x_i$  Cost
         $x_i$  = New  $x_i$ 
    end
end
end

#Apply Unexperienced Phase
for iter = 4: Max iteration
    Apply experienced Phase
    If  $score_{explore} > score_{exploit}$ 
        Apply Exploration phase
        If Exploration NewBest  $x_i$  Cost < Pumamale Cost
            Pumamale = NewBest  $x_i$ 
        end
    else
        Apply Exploitation phase
        If Exploitation NewBest  $x_i$  Cost < Pumamale Cost
            Pumamale = NewBest  $x_i$ 
        end
    end
    Update T, f1, f2, and f3
    Update  $score_{explore}$  and  $score_{exploit}$  by Eq. (17)
    Convert best solution to binary using Eq. (39)
End

```

4. EXPERIMENTAL SETUP

We provide an extensive description of the experimental setup. First, it is important to mention that the PC used in the study had the following specifications: Operating system: Windows 10; Processor: Intel® Core i5-10300H CPU 2.50 GHz and 2.50 GHz; Memory: 16 GB.

4.1 Evaluation metrics

Performance analysis of proposed methodology is carried out based on criteria expressed in the table below. The criteria cover average accuracy, average fitness, best fitness, worst fitness, standard deviation, and average running time. Table 2 shows equations used to calculate these parameters and comparisons among proposed optimizers with existing algorithms based on parameters. The best result is indicated by j th run number, which is represented as S_j^* , size S_j^* describes size of superior solution vector. The average run time is indicated as actual time during execution of optimizer o in its i th run, in which M is number of iterations for optimizer o.

Table 2. The measures for evaluation are utilized to evaluate the proposed feature selection methodology.

Metric	Value
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Avg fitness	$\frac{1}{M} \sum_{i=1}^M S_i^*$
Best fitness	$\min_{i=1}^M S_i^*$
Worst fitness	$\max_{i=1}^M S_i^*$
Standard deviation	$\sqrt{\frac{1}{M-1} \sum_{i=1}^M (S_i^* - Mean)^2}$
Avg Run time	$\frac{1}{M} \sum_{i=1}^M RunTime_{o,i}$
Average Rank	$\underline{R}_j = \frac{1}{N} \sum_{i=1}^N R_{i,j}$
Friedman Test	$x_F^2 = \frac{12N}{k(K+1)} \sum_{j=1}^k \left(\underline{R}_j - \frac{k+1}{2} \right)^2$
Wilcoxon Signed-Rank	$T = \min(T^+, T^-)$
z-value	$z = \frac{T - \frac{n'(n'+1)}{4}}{\frac{\sqrt{n'(n'+1)(2n'+1)}}{24}}$

In the Friedman test, the test statistic x_F^2 is a function of the difference between the average ranks and the expected rank $\frac{k+1}{2}$. In the above context, N is the number of datasets, k is the cumulative number of algorithms, and $R_{i,j}$ is the rank of the j th algorithm in the i th dataset. The overline denotes the average, and $\underline{R}_j$ is its value. In the Wilcoxon Signed-Rank test, the test statistic is given as $T = \min(T^+, T^-)$, where T^+ and T^- are the sums of the ranks of the positive and negative differences of the pair-wise observations. In other words, T is a measure of how the algorithms' performances are similar. It is noted that smaller T provides evidence against the null hypothesis of equal performances. The variable n' is the number of non-zero pair-wise differences. Test statistic T is transformed into a z -statistic to determine the significance of T in the standard normal distribution. The z -statistics are given as z .

Table 3 represents the gene dataset used in this research. Binary identification algorithms were created using Python 3.10 and its complementary libraries, including Numpy and others. This is a description of the computing environment.

The following sections include the input types, parameters, and experimental results. This section discusses the evaluation criteria applied to compare the results and explains the rationale behind the selection of these specific metrics.

4.2 Dataset Description

This study evaluates the performance of the proposed technique utilizing nine benchmark biological datasets [34–44]. The data sets employed in this study are described in Table 3.

Table 3. Description of Dataset

Dataset	No. of instances	No. of features	No. of classes	Type
Leukemia [37]	72	7070	2	Discrete, binary
Lymphoma [38]	96	4026	9	Discrete, multiclass
Lung [39]	203	3312	5	Continuous, multiclass
Lung discrete [40]	73	325	7	Discrete, multiclass
Colon [41]	62	2000	2	Discrete, binary
CLL SUB 111 [42]	111	11340	3	Continuous, multiclass
Prostate GE [43]	102	5966	2	Discrete, binary
nci9 [44]	60	9712	9	Discrete, multiclass
SMK CAN 187 [45]	187	19993	2	Discrete, binary

Nine different metaheuristic optimization algorithms were evaluated compared to the (BPo) algorithm in a series of experiments to assess the proposed model: Binary Political Optimizer V-shaped Function (BPO-V), Binary Bat Algorithm (BBA) [47], Binary Genetic Algorithm (BGA) [46], and Binary Particle Swarm Optimization (BPSO) [45] Binary Tree Growth Algorithm (BTGA) [52], Binary Differential Evolution (BDE) [48], Binary Grey Wolf Optimizer (BGWO) [49], Binary Atom Search Algorithm (BASO) [50], and Binary Harris Hawks Optimizer (BHHO) [51]. Focusing on the Puma Optimizer (Po) and its binary relative, the Binary Puma Optimizer (BPo), Table 4 summarizes parameters for multiple optimization algorithms. Po seeks constant search areas, thereby balancing exploration and exploitation by using pumas' natural hunting techniques. Its properties have been modified to deal with a dynamic, sloped landscape where workable answers are always changing. By contrast, BPo uses these biologically inspired techniques inside a binary search framework. This adaptation converts continuous movements into discrete, binary processes to preserve the efficiency of the optimizer, therefore requiring a new calibration. Po succeeds in continuous domains; BPo guarantees that the Puma Optimizer's basic principles are correctly applied in binary environments, thus allowing strong performance over many optimization obstacles. Po and BPo both show the value of parameter optimization specific for a given scenario.

Table 4. Configuration of compared algorithms with 100 iterations and ten runs for each.

Algorithm	Parameter (s)	Value (s)
BPO	Parties	5
	Lambda	1
BPSO	c1	2
	c2	2
	Vmax	6
	Wmax	0.9
	Wmin	0.4
BGWO	a	2
BBA	Gama	1
	Alpha	1
	Zigma	1
	Beta	1
	Freq_Min	20
	Freq_Max	50
BDE	CrossRate	0.9

BTGA	N1	3
	N2	5
	N4	3
	Tree reduction rate	0.8
	Parameter controls nearest tree	0.5
BHHO	Beta	1.5
BASO	Alpha	50
	Beta	0.2
	Vmax	6
BGA	Crossover Rate	0.9
	Mutation Rate	0.1
Po Bpo	Population size	100
	LB	-1
	UB	1
	Beta	0.01
	Alpha	0.99
	Transfer Function (U3_F(x))	$ x ^8$
All	SearchAgent(Bats, wolfs, particles, . . .)	30
	Maximum iterations	100

Table 5 shows that, using U function transfer learning especially with the optimal transfer function, U3, the Binary Puma Optimizer (BPo) exceeds eleven other metaheuristic algorithms in gene selection. Over many datasets, including CLL_SUB_111, Colon, Leukaemia, Lung, Lung_discrete, Lymphoma, nci9, Prostate_GE, and SMK_CAN_187, BPo routinely produced significantly lower fitness function values. With a better fitness rate of 0.001 and an average of 0.038, BPo greatly outperformed other approaches, which often showed fitness values over 1.0 on the CLL_SUB_111 dataset. Similar findings were observed on the Colon and Leukaemia datasets, whereby competitors had considerably higher scores while BPo had the best fitness of 0.000 or values near zero. BPo's smaller standard deviation indicates that it was more dependable and consistent even in cases when the original Puma Optimizer (Po) produced similar findings, such as in the Lung dataset. These results highlight the remarkable efficiency of BPo for gene selection when used with the U3 transfer function, producing frequently better solutions with lower variability than other advanced metaheuristics.

Table 5. Experimental result of the fitness function of the proposed algorithms compared to ten metaheuristics that used in this paper [46].

Dataset		BPo	Po	BPO-V	BGA	BGWO	BBA	BHHO	BASO	BDE	BPSO	BTGA
CLL_SUB_111	Best	0.001	0.044	1.421	1.206	1.176	1.205	1.254	1.423	1.125	1.205	1.208
	Avg	0.038	0.087	1.325	1.128	1.092	1.123	1.143	1.292	1.049	1.125	1.119
	Worst	0.044	0.130	1.252	1.070	1.036	1.070	1.089	1.216	0.975	1.068	1.068
	std	0.012	0.027	0.034	0.026	0.028	0.026	0.031	0.037	0.026	0.028	0.029
Colon	Best	0.000	0.076	1.499	1.284	1.263	1.271	1.380	1.488	1.263	1.275	1.271
	Avg	0.068	0.084	1.492	1.273	1.255	1.263	1.343	1.488	1.242	1.269	1.263
	Worst	0.076	0.153	1.433	1.263	1.237	1.257	1.309	1.45	1.204	1.263	1.258
	std	0.022	0.022	0.016	0.003	0.005	0.002	0.016	0.008	0.014	0.002	0.002
Leukemia	Best	0.000	0.000	1.499	1.276	1.261	1.262	1.384	1.491	1.257	1.267	1.263
	Avg	0.000	0.014	1.497	1.270	1.254	1.259	1.361	1.476	1.252	1.262	1.259
	Worst	0.000	0.066	1.479	1.264	1.249	1.256	1.339	1.462	1.247	1.259	1.256
	std	0.000	0.026	0.003	0.002	0.002	0.001	0.010	0.006	0.002	0.001	0.001
Lung	Best	0.024	0.024	1.495	1.287	1.263	1.267	1.381	1.485	1.265	1.273	1.271
	Avg	0.024	0.024	1.475	1.279	1.256	1.263	1.361	1.471	1.254	1.268	1.263
	Worst	0.024	0.024	1.450	1.271	1.246	1.259	1.341	1.457	1.244	1.265	1.260
	std	0.000	0.000	0.010	0.003	0.003	0.001	0.009	0.006	0.003	0.001	0.002
Lung_discrete	Best	0.000	0.000	1.493	1.329	1.295	1.306	1.381	1.48	1.286	1.323	1.306
	Avg	0.000	0.001	1.425	1.312	1.269	1.286	1.354	1.461	1.256	1.301	1.288
	Worst	0.000	0.002	1.340	1.295	1.230	1.275	1.32	1.429	1.210	1.287	1.275
	std	0.000	0.000	0.028	0.007	0.013	0.006	0.012	0.009	0.013	0.006	0.006

Lymphoma	Best	0.198	0.248	1.438	1.208	1.161	1.161	1.282	1.426	1.157	1.169	1.164
	Avg	0.232	0.258	1.388	1.199	1.154	1.158	1.267	1.378	1.151	1.165	1.159
	Worst	0.247	0.297	1.348	1.190	1.150	1.156	1.253	1.363	1.146	1.162	1.156
	std	0.022	0.019	0.035	0.003	0.002	0.001	0.006	0.010	0.002	0.001	0.001
nci9	Best	0.330	0.661	1.326	1.173	1.162	1.172	1.209	1.312	1.094	1.172	1.166
	Avg	0.478	0.751	1.185	1.070	1.043	1.068	1.080	1.217	0.987	1.067	1.053
	Worst	0.660	0.825	1.073	1.006	0.975	1.002	0.993	1.122	0.916	1.002	1.000
	std	0.103	0.044	0.050	0.043	0.040	0.040	0.042	0.039	0.037	0.039	0.042
Prostate_GE	Best	0.047	0.094	1.499	1.267	1.258	1.261	1.382	1.482	1.257	1.264	1.262
	Avg	0.084	0.095	1.475	1.262	1.245	1.258	1.345	1.464	1.241	1.260	1.258
	Worst	0.094	0.096	1.437	1.258	1.218	1.255	1.309	1.446	1.205	1.257	1.255
	std	0.018	0.000	0.020	0.001	0.011	0.001	0.015	0.008	0.013	0.001	0.001
SMK_CAN_187	Best	0.052	0.131	1.391	1.173	1.166	1.195	1.261	1.398	1.142	1.196	1.197
	Avg	0.083	0.136	1.329	1.135	1.107	1.134	1.169	1.305	0.072	1.127	1.130
	Worst	0.104	0.157	1.276	1.111	1.061	1.089	0.026	1.264	1.029	1.089	1.088
	std	0.015	0.010	0.023	0.017	0.022	0.020	0.026	0.026	0.022	0.019	0.024

A large and outstanding outcome is shown in Table 6 and Fig 3: the Binary Puma Optimiser (BPo) selects much less features than its competitors. While other algorithms keep hundreds or even thousands of features, BPo averages just 6.3 and 6.7 features in the Colon and Leukaemia datasets respectively. This notable decrease shows BPo's capacity to determine the most relevant genes, hence producing models more understandable and require less computational resources, a great benefit in high-dimensional gene expression study.

Table 6. Average of selected features

Dataset	BPo	Po	BPO-V	BGA	BGWO	BBA	BHHO	BASO	BDE	BPSO	BTGA
CLL SUB 111	175.2	1200	187.6	5585.4	6384.1	5623.5	4399.4	1103.7	6901.9	5630.8	5662.6
Colon	6.3	219.1	31.0	907.2	979.4	944.5	626.9	125.5	1032.0	924.1	946.8
Leukemia	6.7	676.4	42.1	3240.7	3470.5	3404.0	1961.3	339.5	3498.4	3356.9	3400.4
Lung	46.8	234.9	109.2	1463.4	1615.3	1569.2	915.6	192.2	1629.4	1533.2	1566.4
Lung discrete	15.2	41.7	31.5	121.7	149.94	138.7	94.9	25.3	158.4	129.2	137.7
Lymphoma	11.9	350.8	71.5	1574.2	1931.3	1903.4	1022.3	155.8	1959.5	1844.8	1890.7
nci9	10.6	809.9	785.5	4720.8	5232.8	4811.2	3932.7	1062.2	6128.7	4799.8	4879.4
Prostate GE	13.9	532.9	171.2	2833.8	3032.7	2841.6	1841.6	424.9	3086.7	2853.4	2883.2
SMK_CAN_187	28.7	2575.4	371.6	9922.0	11069.3	9945.3	7199.6	1945.6	11519.6	9945.6	9977.4

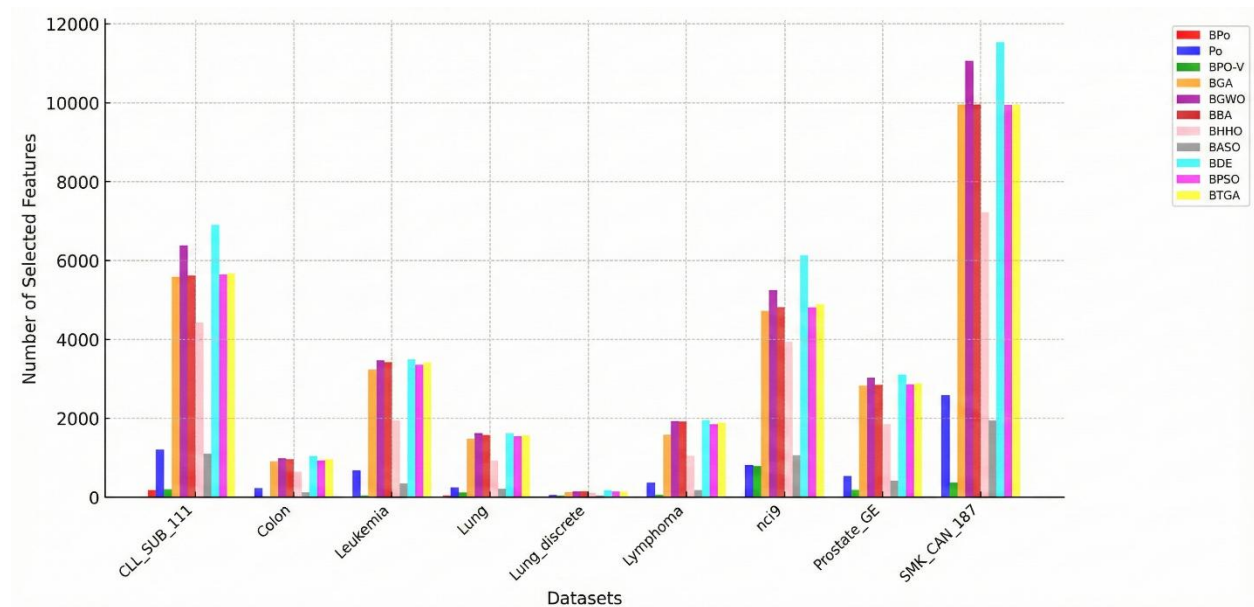


Fig 3. Comparison of Selected Features Across Algorithms

Tables 7 and 8 show that the Binary Puma Optimizer (BPo) is an incredible method for gene selection using the U3 transfer function that demonstrates both high accuracy and execution speed. Based on Fig 4, while continuously reaching remarkable classification accuracy across several datasets, including a perfect score of 1.00 on Leukaemia and Lung_discrete, Table 7 shows that BPo maintains competitive performance on progressively tougher datasets like CLL_SUB_111 (0.960) and Lung (0.965). These results highlight how well BPo detects relevant gene features, especially in high-dimensional settings where accuracy is crucial. Table 8 shows that BPo is scalable and efficient, so significantly less execution time is needed than its alternatives. For the SMK_CAN_187 dataset, for example, BPo performed in just 61.498 seconds when other alternative techniques ran more than 400 seconds. Managing large gene expression data requires a significant decrease in computational expenses to support real-time decision-making and faster iterations. BPo is ultimately an extremely useful tool for gene selection because of its superior accuracy and cost-effective processing cost as shown in Fig 5.

Table 7. The average accuracy of the proposed algorithms compared to ten metaheuristics

Dataset	BPo	Po	BPO-V	BGA	BGWO	BBA	BHHO	BASO	BDE	BPSO	BTGA
CLL SUB 111	0.960	0.913	0.686	0.681	0.590	0.454	0.545	0.500	0.681	0.500	0.5909
Colon	0.930	0.900	0.866	0.833	0.833	0.750	0.750	0.816	0.750	0.750	0.750
Leukemia	1.00	0.986	0.771	0.928	0.857	0.857	0.857	0.714	0.785	0.857	0.928
Lung	0.965	0.975	0.920	0.917	0.915	0.900	0.900	0.875	0.925	0.915	0.920
Lung_discrete	1.00	0.980	0.821	0.785	0.785	0.714	0.714	0.642	0.785	0.785	0.714
Lymphoma	0.765	0.740	0.826	0.894	0.789	0.842	0.894	0.789	0.842	0.894	0.894
nci9	0.516	0.241	0.441	0.333	0.333	0.500	0.500	0.250	0.500	0.416	0.500
Prostate GE	0.914	0.895	0.865	0.900	0.950	0.850	0.900	0.850	0.750	0.950	0.900
SMK CAN 187	0.915	0.863	0.894	0.621	0.810	0.864	0.729	0.702	0.594	0.675	0.756

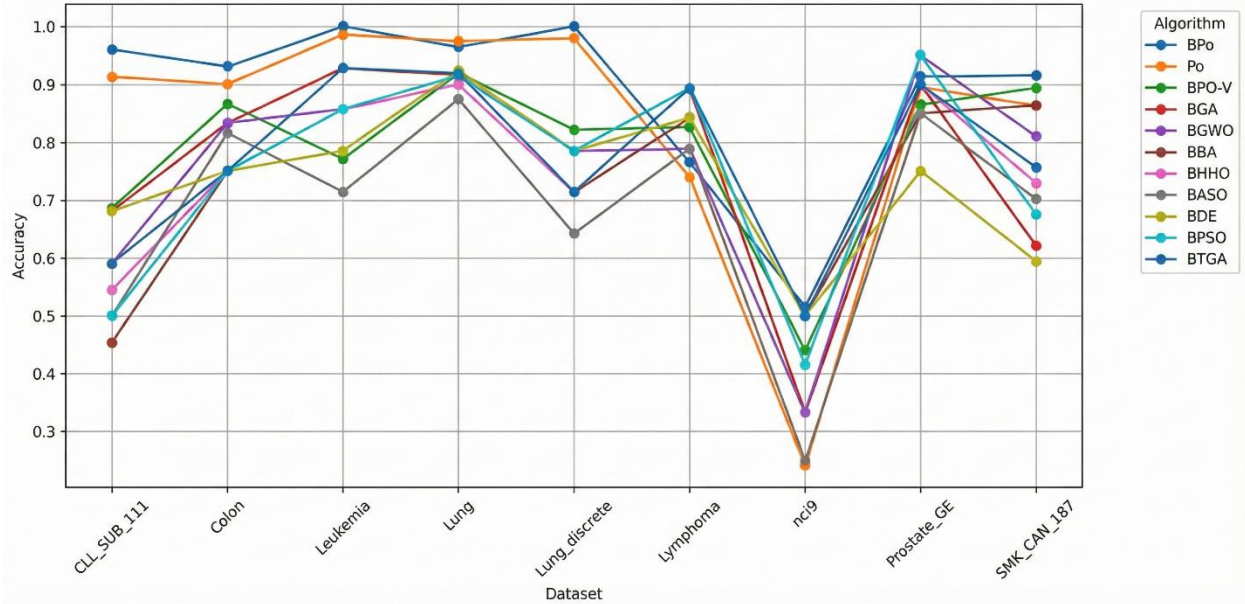


Fig 4. Performance Comparison Across Datasets

Table 8. The average execution time of the proposed algorithms compared to ten metaheuristics.

Dataset	BPo	Po	BPO-V	BGA	BGWO	BBA	BHHO	BASO	BDE	BPSO	BTGA
CLL SUB 111	56.109	127.95	166.09	156.32	156.14	166.73	269.35	245.72	172.91	235.55	160.38
Colon	14.372	15.563	110.32	94.106	96.418	94.625	164.37	105.77	89.931	123.42	93.963
Leukemia	20.003	30.942	117.01	113.64	127.05	127.41	195.20	170.51	115.36	137.63	110.36
Lung	52.519	70.662	135.10	130.19	132.89	125.63	227.70	157.05	139.63	126.61	116.73

Lung_discrete	17.895	18.851	108.97	94.692	90.66	82.415	187.53	101.86	93.448	80.048	78.451
Lymphoma	36.080	51.756	117.57	135.31	109.74	114.09	202.25	147.47	116.19	103.03	100.12
nci9	25.341	43.465	127.46	141.51	138.79	137.83	226.38	236.84	130.96	128.75	116.39
Prostate_GE	32.527	47.896	131.65	125.09	130.51	130.15	211.76	188.38	134.16	119.47	114.90
SMK_CAN_187	61.498	425.26	351.32	253.59	342.11	332.16	564.87	423.72	368.13	342.08	289.68

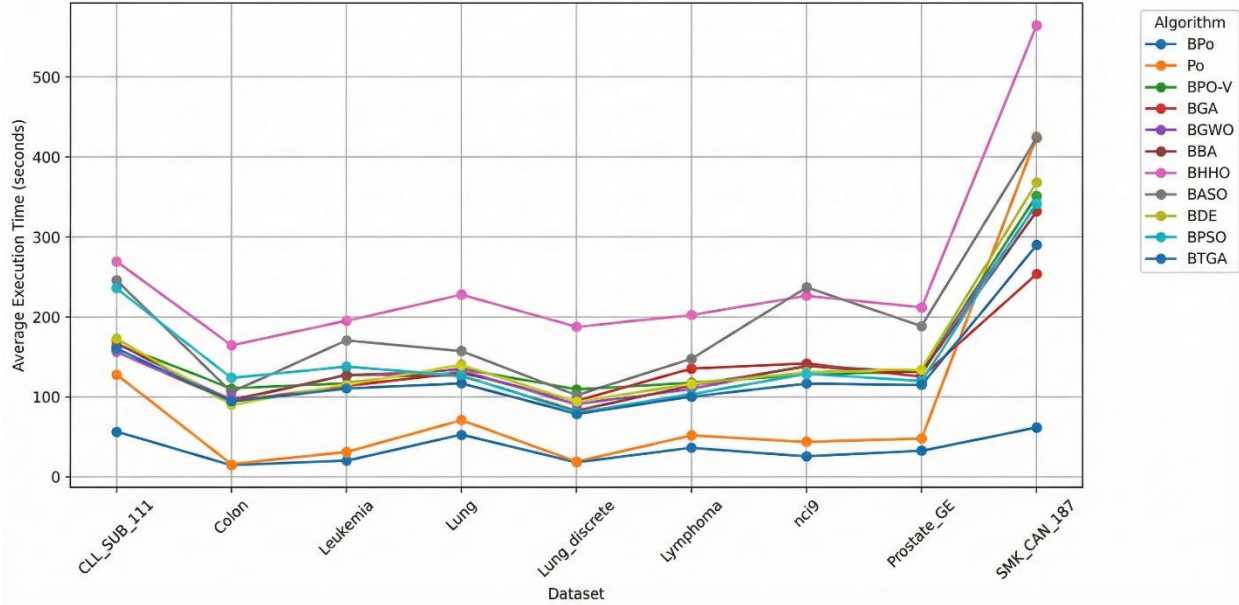


Fig 5. Average execution time (in seconds) of the proposed algorithms (BPo and Po) compared to ten metaheuristics across nine datasets. BPo demonstrates the lowest execution time consistently.

Convergence analysis of all datasets shows that the BPo outperforms the standard Po in stability and performance as shown in Fig 6. The results show that BPo often achieves a lower fitness value over all datasets, indicating its improved capacity to efficiently maximize the objective function. Since it lets BPo effectively discuss the solution space and avoid early convergence, the optimal trade-off between exploration and exploitation supports this enhancement. Transfer functions especially U-shaped transfer functions such U3, which allows smoother transitions between various search strategies and accelerate the rate of convergence help improve this equilibrium.

On datasets including Colon, Leukaemia, and nci9, BPo rapidly converges to an optimal solution in its initial iterations, so demonstrating its efficiency of achieving high-quality solutions faster than Po. Po, on the other hand, shows a more slow decline in fitness and usually stabilises at high levels, indicating a reduced refining capacity. The Lung_discrete and Prostate_GE datasets also support this trend; Po differs before stability but BPo rapidly reaches a minimum fitness value. In gene expression investigations, this fast yet constant convergence ensures a more efficient and dependable feature selection method and lowers computational cost.

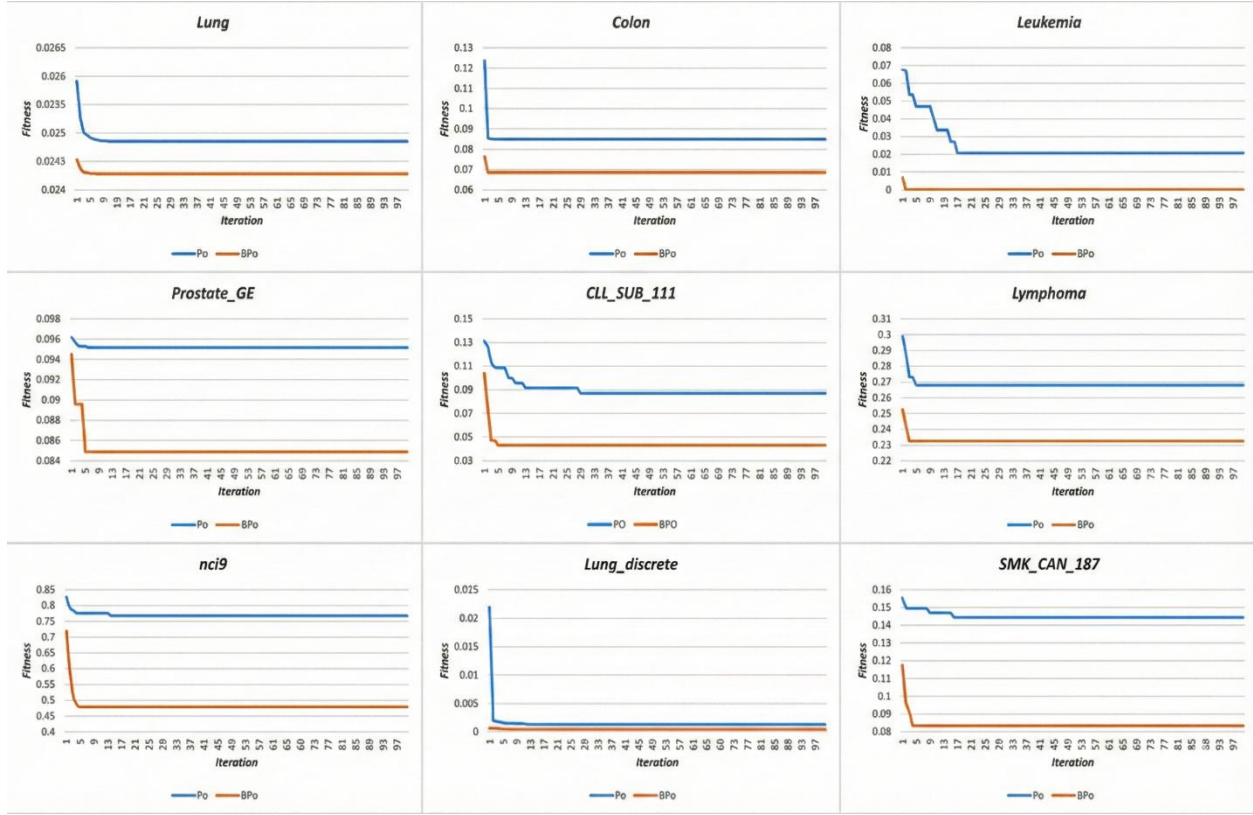


Fig 6. Convergence Performance of the proposed model across all datasets

4.3 Statistical Analysis

Table 9 compares the classification accuracy of BPo concerning ten alternative algorithms over nine datasets using Friedman's test with an average score of 2.33, BPo regularly exceeds all other competing algorithms to achieve the best elevation. Po ranks second with an average rank of 4.67; BPO-V comes in at 5.17 and BGA comes in at 5.56. Ranked highest in six of the nine datasets, BPo shows exceptional stability and efficiency. The much lower ranks of models including BASO (9.39), BDE (6.89), and BVA (7.39) verify BPo's competitive edge in classification tasks even further as shown in Fig 7.

Table 10 improves these results by using the Wilcoxon Signed-Rank test to evaluate the statistical relevance of BPo's dominance over every alternative technique. P-values below 0.05 for every comparison show that the data show BPo much surpasses all other algorithms. Minimal p-values for comparisons between Po (0.0039) and BASO (0.0039) demonstrate significant performance differences. These results show that BPo frequently outperforms its competition in classification accuracy; the observed variations are statistically significant rather than resulting from chance.

Table 9. A comparison of BPo with other algorithms according to Friedman's test when applied to classification accuracy

Dataset	BPo	Po	BPO-V	BGA	BGWO	BBA	BHHO	BASO	BDE	BPSO	BTGA
CLL SUB 111	1	2	3	4.5	7	11	8	9.5	4.5	9.5	6
Colon	1	2	3	4.5	4.5	9	9	6	9	9	9
Leukemia	1	2	10	3.5	6.5	6.5	6.5	11	9	6.5	3.5
Lung	2	1	4.5	6	7.5	9.5	9.5	11	3	7.5	4.5
Lung discrete	1	2	3	5.5	5.5	9	9	11	5.5	5.5	9
Lymphoma	10	11	7	2.5	8.5	5.5	2.5	8.5	5.5	2.5	2.5
nci9	1	11	6	8.5	8.5	3.5	3.5	10	3.5	7	3.5
Prostate GE	3	7	8	5	1.5	9.5	5	9.5	11	1.5	5

SMK CAN 187	1	4	2	10	5	3	7	8	11	9	6
Average Rank	2.33	4.67	5.17	5.56	6.06	7.39	6.67	9.39	6.89	6.44	5.44

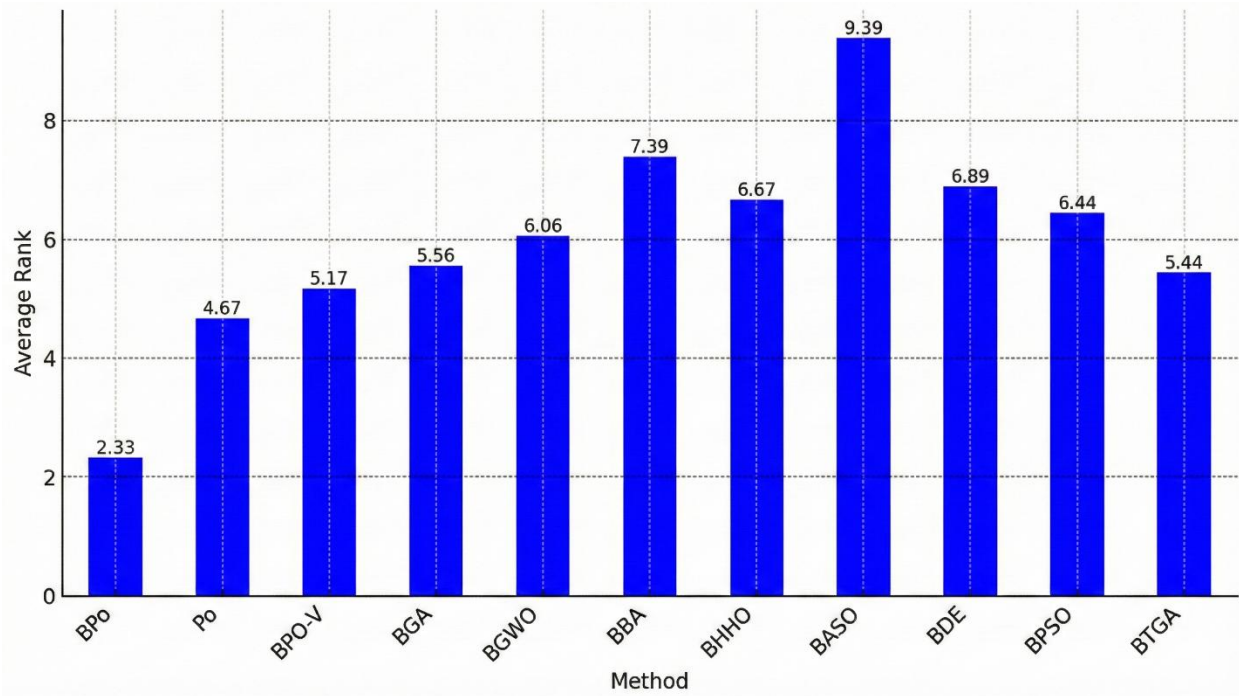


Fig 7. Comparative analysis of BPo and other algorithms in terms of classification accuracy using Friedman's test

Table 10. Wilcoxon test results indicate effectively BPo classification accuracy of Table 7 in comparison to other algorithms.

Model	P-value
BPo vs Po	0.0039
BPo vs BPO-V	0.0078
BPo vs BGA	0.0117
BPo vs BGWO	0.0078
BPo vs BBA	0.0117
BPo vs BHHO	0.0078
BPo vs BASO	0.0039
BPo vs BDE	0.0078
BPo vs BPSO	0.0117
BPo vs BTGA	0.0117

4.4 Limitation

Despite the promising performance of the proposed Binary Puma Optimizer (BPo) in gene selection, several limitations should be acknowledged. First, like most metaheuristic algorithms, the BPo is stochastic in nature, which may lead to slight variations in the selected feature subsets across different independent runs. Second, while the algorithm demonstrated superior efficiency on nine diverse gene datasets, its computational complexity might increase when scaled to extremely large-scale multi-omics datasets with hundreds of thousands of features, a common challenge in the "curse of dimensionality." Third, the current study focused primarily on binary classification tasks within bioinformatics; therefore, its generalizability to multi-class high-

dimensional problems or other domains remains to be fully explored. Finally, although the BPo effectively identifies a minimal set of informative genes, this study did not delve into the biological functional analysis (e.g., pathway enrichment) of the selected genes, focusing instead on the computational and statistical validation of the feature selection process.

5. Conclusion

In conclusion, this study introduced the Binary Puma Optimizer (BPo) for gene selection in microarray datasets, effectively addressing challenges associated with high-dimensional, noisy, and redundant gene expression data. By utilizing a U-shaped transfer function, BPo efficiently transformed the optimization process from continuous to discrete, enabling the selection of the most relevant genes while minimizing computational costs. The Support Vector Machine (SVM) was employed as a fitness function to evaluate the quality of the selected gene subsets. Experimental results on nine benchmark microarray datasets demonstrated that BPo significantly outperformed ten state-of-the-art gene selection techniques in terms of classification accuracy and the number of selected genes. Statistical validation using the Friedman and Wilcoxon tests further confirmed the significance of the proposed approach. For future work, we aim to enhance the optimization process to achieve more optimal gene selection while reducing computational time, ensuring a balance between accuracy and efficiency for large-scale genomic datasets.

Conflict of Interest

The authors declare that they have no conflict of interest.

Ethics Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Funding

Not applicable.

Data Availability

The data supporting the findings of this study are derived from previously published research and public datasets, specifically those cited in references [37–45]. No new datasets were generated during the current study.

Authors' Contributions

Mohammad Wedyan, Ryan Alturki, Hasan J. Alyamani, Bandar Alshawi, and Ahmad Nasayreh contributed equally to all aspects of this research. All authors were collectively responsible for the conceptualization and development of the research methodology, software implementation, and technical validation protocols. Furthermore, they jointly performed the formal analysis, investigation, and data curation, and were fully involved in drafting the original manuscript, as well as conducting the final review and intellectual editing of the paper.

Acknowledgements

Not applicable.

Human Participants and/or Animals

This study does not involve human participants or animals.

References

- [1] Venkatesh, B., & Anuradha, J. (2019). A review of feature selection and its methods. *Cybernetics and information technologies*, 19(1), 3-26.
- [2] S. Wang, Q. Cheng, Microarray analysis in drug discovery and clinical applications, in: *Bioinformatics and Drug Discovery*, Springer, 2006, pp.49–65.
- [3] V. Bolón-Canedo, N. Sánchez-Marono, A. Alonso-Betanzos, J.M. Benítez, F. Herrera, A review of microarray datasets and applied feature selection methods, *Inform. Sci.* 282 (2014) 111–135.
- [4] M.A. Al-Betar, O.A. Alomari, S.M. Abu-Romman, A TRIZ-inspired bat algorithm for gene selection in cancer classification, *Genomics* 112 (1) (2020) 114–126.
- [5] A. Jain, D. Zongker, Feature selection: Evaluation, application, and small sample performance, *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (2) (1997) 153–158.
- [6] M. Nasr, O. Farouk, A. Mohamedeen, A. Elrafie, M. Bedeir, A. Khaled, Benchmarking meta-heuristic optimization, 2020, arXiv preprint arXiv: 2007.13476.
- [7] A. Hatamlou, “Black hole: A new heuristic optimization approach for data clustering,” *Inf. Sci.*, vol. 222, pp. 175–184, Feb. 2013.
- [8] E.-S.-M. El-Kenawy, S. Mirjalili, A. A. Abdelhamid, A. Ibrahim, N. Khodadadi, and M. M. Eid, “Meta-heuristic optimization and keystroke dynamics for authentication of smartphone users,” *Mathematics*, vol. 10, no. 16, p. 2912, Aug. 2022.
- [9] E.-S.-M. El-kenawy, F. Albalawi, S. A. Ward, S. S. M. Ghoneim, M. M. Eid, A. A. Abdelhamid, N. Bailek, and A. Ibrahim, “Feature selection and classification of transformer faults based on novel meta-heuristic algorithm,” *Mathematics*, vol. 10, no. 17, p. 3144, Sep. 2022.
- [10] H. A. Alsayadi, N. Khodadadi, and S. Kumar, “Improving the regression of communities and crime using ensemble of machine learning models,” *J. Artif. Intell. Metaheuristics*, vol. 1, no. 1, pp. 27–34, 2022.
- [11] O. A. Akinola, J. O. Agushaka, and A. E. Ezugwu, “Binary dwarf mongoose optimizer for solving high-dimensional feature selection problems,” *PLoS One*, vol. 17, no. 10, Oct. 2022, Art. no. e0274850.
- [12] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Cham, Switzerland: Springer, Dec. 2012.
- [13] A. A. Abdelhamid et al., “Innovative feature selection method based on hybrid sine cosine and dipper throated optimization algorithms,” *IEEE Access*, vol. 11, pp. 79750–79776, 2023, doi: 10.1109/ACCESS.2023.3298955.
- [14] Y. Li, T. Li, and H. Liu, “Recent advances in feature selection and its applications,” *Knowl. Inf. Syst.*, vol. 53, no. 3, pp. 551–577, Dec. 2017.
- [15] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.
- [16] I. J. Mohammed, B. T. Al-Nuaimi, and T. I. Baker, “Weather forecasting over Iraq using machine learning,” *J. Artif. Intell. Metaheuristics*, vol. 2, no. 2, pp. 39–45, 2022.
- [17] J. Žerovnik, “Heuristics for NP-hard optimization problems—Simpler is better!” *Logistics Sustain. Transp.*, vol. 6, no. 1, pp. 1–10, Nov. 2015.

- [18] A. I. Hammouri, M. Mafarja, M. A. Al-Betar, M. A. Awadallah, and I. Abu-Doush, "An improved dragonfly algorithm for feature selection," *Knowl.-Based Syst.*, vol. 203, Sep. 2020, Art. no. 106131.
- [19] S. Ahmed, K. H. Sheikh, S. Mirjalili, and R. Sarkar, "Binary simulated normal distribution optimizer for feature selection: Theory and application in COVID-19 datasets," *Expert Syst. Appl.*, vol. 200, Aug. 2022, Art. no. 116834.
- [20] H. Banka and S. Dara, "A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation," *Pattern Recognit. Lett.*, vol. 52, pp. 94–100, Jan. 2015.
- [21] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54–65, Nov. 2016.
- [22] E. Emary and H. M. Zawbaa, "Feature selection via Lévy antlion optimization," *Pattern Anal. Appl.*, vol. 22, no. 3, pp. 857–876, Aug. 2019.
- [23] B. Ji, X. Lu, G. Sun, W. Zhang, J. Li, and Y. Xiao, "Bio-inspired feature selection: An improved binary particle swarm optimization approach," *IEEE Access*, vol. 8, pp. 85989–86002, 2020.
- [24] H. M. Alshamlan, G. H. Badr, and Y. A. Alohal, "ABC-SVM: Artificial bee colony and SVM method for microarray gene selection and multi class cancer classification," *Int. J. Mach. Learn. Comput.*, vol. 6, no. 3, p. 184, 2016.
- [25] E. Alba, J. Garcia-Nieto, L. Jourdan, and E.-G. Talbi, "Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 284290.
- [26] L. Gao, M. Ye, X. Lu, and D. Huang, "Hybrid method based on information gain and support vector machine for gene selection in cancer classification," *Genomics, Proteomics Bioinf.*, vol. 15, no. 6, pp. 389395, Dec. 2017.
- [27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 127, Apr. 2011.
- [28] Y. Wang, I. V. Tetko, M. A. Hall, E. Frank, A. Facius, K. F. Mayer, and A. H. W. Mewes, "Gene selection from microarray data for cancer classification: A machine learning approach," *Comput. Biol. Chem.*, vol. 29, no. 1, pp. 3746, 2005.
- [29] H. Abdi, "Singular value decomposition (SVD) and generalized singular value decomposition," in *Encyclopedia of Measurement and Statistics*. Richardson, TX, USA: Univ. of Texas at Dallas, 2007, pp. 907912.
- [30] W. Ahmed, "Fast orthogonal search for training radial basis function neural networks," Ph.D. dissertation, Univ. Maine, Orono, ME, USA, 1994.
- [31] S. Ahmed, K. K. Ghosh, S. K. Bera, F. Schwenker, and R. Sarkar, "Gray level image contrast enhancement using barnacles mating optimizer," *IEEE Access*, vol. 8, pp. 169196169214, 2020.
- [32] M. H. Sulaiman, Z. Mustaffa, and O. Aliman, "An application of barnacles mating optimizer for solving economic dispatch problems," in *Proc. IEEE Jordan Int. Joint Conf. Electr. Eng. Inf. Technol. (JEEIT)*, Apr. 2019, pp. 835839.
- [33] R. M. Rizk-Allah and A. A. El-Fergany, "Conscious neighborhood scheme-based Laplacian barnacles mating algorithm for parameters optimization of photovoltaic single- and double-diode models," *Energy Convers. Manage.*, vol. 226, Dec. 2020, Art. no. 113522.
- [34] Abdollahzadeh, B., Khodadadi, N., Barshandeh, S., Trojovský, P., Gharehchopogh, F. S., El-kenawy, E. S. M., ... & Mirjalili, S. (2024). Puma optimizer (PO): A novel metaheuristic optimization algorithm and its application in machine learning. *Cluster Computing*, 1-49.

- [35] Mirjalili, S., Zhang, H., Mirjalili, S., Chalup, S., & Noman, N. (2020). A novel U-shaped transfer function for binary particle swarm optimisation. In *Soft Computing for Problem Solving 2019: Proceedings of SocProS 2019, Volume 1* (pp. 241-259). Springer Singapore.
- [36] Henderi, H., Wahyuningsih, T., & Rahwanto, E. (2021). Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. *International Journal of Informatics and Information Systems*, 4(1), 13-20.
- [37] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., ... & Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439), 531-537.
- [38] Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., ... & Staudt, L. M. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769), 503-511.
- [39] Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., ... & Meyerson, M. (2001). Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, 98(24), 13790-13795.
- [40] Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8), 1226-1238.
- [41] Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12), 6745-6750.
- [42] Haslinger, C., Schweifer, N., Stilgenbauer, S., Döhner, H., Lichter, P., Kraut, N., ... & Abseher, R. (2004). Microarray gene expression profiling of B-cell chronic lymphocytic leukemia subgroups defined by genomic aberrations and VH mutation status. *Journal of Clinical Oncology*, 22(19), 3937-3949.
- [43] Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., ... & Sellers, W. R. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer cell*, 1(2), 203-209.
- [44] Ross, D. T., Scherf, U., Eisen, M. B., Perou, C. M., Rees, C., Spellman, P., ... & Brown, P. O. (2000). Systematic variation in gene expression patterns in human cancer cell lines. *Nature genetics*, 24(3), 227-235.
- [45] Spira, A., Beane, J. E., Shah, V., Steiling, K., Liu, G., Schembri, F., ... & Brody, J. S. (2007). Airway epithelial gene expression in the diagnostic evaluation of smokers with suspect lung cancer. *Nature medicine*, 13(3), 361-366.
- [46] Manita, G., & Korbaa, O. (2020). Binary political optimizer for feature selection using gene expression data. *Computational Intelligence and Neuroscience*, 2020(1), 8896570.