Proceedings of the Pakistan Academy of Sciences: A Physical and Computational Sciences 61(3): 257-271 (2024)

ISSN (Print): 2518-4245; ISSN (Online): 2518-4253
http://doi.org/10.53560/PPASA(61-3)848

Pakistan Academy of Sciences

Research Article

Selecting the Optimal Classifier for Wrapper-Based Feature Selection Methods

Farzad Zandi[1], Parvaneh Mansouri[1], and Reza Sheibani[2*]

[1]Department of Mathematics and Computer Science, Arak Branch,
Islamic Azad University, Arak, Iran
[2]Department of Computer Engineering, Mashhad Branch,
Islamic Azad University, Mashhad, Iran

Abstract: Dimensionality reduction, the elimination of irrelevant features, and the selection of an optimal subset of features are critical components in the construction of an efficacious machine learning model. Among the various feature selection methodologies, wrapper-based methods yield superior results due to their evaluation of candidate subsets. Numerous meta-heuristic methods have been employed in this feature selection process. A significant and complex issue in feature selection utilizing these methods is the selection of the most suitable classifier. In this study, we propose a novel method for selecting the optimal classifier during the feature selection process. We employ ten distinct classifiers for two swarm intelligence methods, namely Bat and Gray Wolf, and compute their results on four cancer datasets: Leukemia, SRBCT, Prostate, and Colon. Our findings demonstrate that the proposed method identifies the optimal classifiers for all four datasets. Consequently, when employing wrapper-based methods to select features for each dataset, the optimal classifier is identified.

Keywords: Feature Selection, Wrapper-based Methods, Metaheuristic Algorithms, Roulette Wheel, Optimal Classifier.

1. INTRODUCTION

Feature selection is a pivotal process in machine learning that entails the selection of the most pertinent features from the dataset that contribute significantly to the prediction variable or outcome. The objective is to eliminate irrelevant or redundant features. This can lead to a decrease in model accuracy and performance. Feature selection is instrumental in constructing an effective machine learning model for several reasons:

- **Enhances Accuracy**: Irrelevant or redundant features can adversely affect the model's performance. By utilizing only, the most pertinent features, we can construct more accurate models.
- **Mitigates Overfitting**: A model trained with irrelevant features is more prone to overfitting, where it performs well on the training data but poorly on unseen data.
- **Accelerates Training**: Less data equates to faster training times. By reducing the number of features, we can expedite the training process.
- **Boosts Interpretability**: Models with fewer features are simpler to understand and interpret.

There are several techniques for feature selection [1], each with its unique strengths and weaknesses. Here are a few commonly employed methods:

- **Filter Methods**: These methods are often univariate and consider each feature independently or in relation to the dependent variable. Examples include the Chi-Squared Test [2], Information Gain [3], and Correlation Coefficient Scores [4].
- **Wrapper Methods**: These methods perceive the selection of a set of features as a search problem. Examples include Recursive Feature Elimination [5], Forward Selection, and Backward Elimination [6].
- **Embedded Methods**: These methods ascertain which features best contribute to the accuracy of the model during the model creation process. An example includes LASSO [7].

Received: March 2024; Revised: August 2024; Accepted: September 2024
* Corresponding Author: Reza Sheibani <reza.shni@gmail.com>

Among the various methodologies for feature selection, wrapper-based methods yield superior results. This can be attributed to the employment of a fitness function, which is utilized to evaluate each selected subset. For a considerable duration, meta-heuristic algorithms have been instrumental in resolving numerous optimization problems that are either exceedingly complex or possess an extensive problem space, rendering them unsolvable [8]. Novel meta-heuristic algorithms, encompassing population-based, evolutionary-based, and nature-inspired based methods, are continually being developed. With minor modifications, these methods can be transformed into a robust wrapper-based feature selection method [9]. These methods typically commence with a subset of features and, after evaluating them with a fitting function, they converge towards the optimal results based on the proposed algorithm. The fitting function serves as a crucial criterion that could be a supervised machine learning algorithm, calculating the suitability of candidate features. In machine learning, there exists a multitude of supervised algorithms, each with its unique advantages and disadvantages [10]. A particular classifier may yield satisfactory results with a dataset, while another classifier may produce unacceptable or even subpar results on the same dataset. This challenge is also applicable to the evaluation function of feature selection problems. Consequently, the selection of an appropriate classifier can enhance the evaluation performance. Given that it is not feasible to test individual classifiers on all datasets due to the time-consuming nature of this process, we propose a solution to this challenge, enabling the utilization of the optimal classifier for quality assessment.

Numerous studies have been conducted to predict various types of cancers, employing a range of machine learning algorithms in the process. Common algorithms such as K-Nearest Neighborhood, Support Vector Machine, Logistic Regression, Decision Tree, and Bayes have been widely utilized. More complex methods, including Random Forest, Ensemble, Boosting, Neural Network, and deep learning, have also been explored. However, these methods often encounter high time complexity due to the large data dimensions, particularly the number of genes. To address this issue, researchers have considered the use of different dimension reduction methods. Among the feature selection methods, filtering-based methods have been favored over wrapper-based and embedding-based methods. For instance, Purbolaksono *et al.* [11] employed mutual information to reduce dimensions and identify informative genes. Aydadenta and Adivijaya [12] utilized k-means and Information Gain for feature selection. Initially, k-means was used to group similar features into a cluster, thereby eliminating redundancy. Subsequently, the Relief algorithm was used to rank the elements of the clusters, and the top-ranking features of each cluster were combined for Random Forest training. Cilia *et al.* [13] concentrated on feature ranking techniques that evaluate each feature individually. In the work of Bol'on *et al.* [14], various feature selection techniques such as ReliefF, SVM-RFE, mRMR, IG, and FCBF were used for gene selection. Al-Batah *et al.* [15] employed the filter method, CFS, to eliminate redundant genes and identify informative ones. Baliarsingh *et al.* [16] used the Wilcoxon rank sum test to select relevant genes. Su *et al.* [17] introduced a gene selection method based on the Kolmogorov-Smirnov (K-S) test and CFS. Initially, the K-S test removed redundant genes and noise by comparing the distribution of two types of samples. The filtered subset was then evaluated by CFS, leaving only genes with high correlation with the class and low redundancy. Lastly, Ahmad *et al.* [18] used different filter feature selection techniques, namely SNR, FC, IG, and t-Test, to select informative genes. This comprehensive review of methods and techniques provides a solid foundation for future research in cancer prediction and gene selection.

In addition to the aforementioned methods, some researchers have employed wrapper-based methods to identify the most effective genes. For instance, Wu *et al.* [19] proposed a hybrid approach that leverages an enhanced HI-BQPSO binary quantum particle swarm optimization algorithm for feature selection. This innovative method amalgamates the benefits of filtering and random heuristic search. The process begins with the utilization of the Maximum Information Coefficient (MIC) to compute the correlation between features and class, thereby obtaining an initial feature subset. Subsequently, the enhanced BQPSO is employed to derive the optimal feature subset. This methodological approach underscores the potential of hybrid models in gene selection for cancer prediction.

Several studies have also explored the combination of filtering and wrapper-based methods. For instance, Medjahed *et al.* [20] utilized Support Vector Machines based on Recursive Feature Elimination (SVM-RFE) to eliminate 40% of features. The remaining subset was subsequently processed through Binary Dragonfly (BDF) to retain only the informative genes. In another study, Jain *et al.* [21] proposed a hybrid feature selection method that amalgamated Correlation-based Feature Selection (CFS) and Improved Binary Particle Swarm Optimization (IBPSO). The use of IBPSO enhanced the initial convergence to the local optimum of Binary Particle Swarm Optimization (BPSO). Furthermore, Shahbeig *et al.* [22] introduced a hybrid Teaching-Learning-Based Optimization - Particle Swarm Optimization (TLBO-PSO) method. This approach combines the principles of Teaching-Learning-Based Optimization (TLBO) and a mutated version of Fuzzy Adaptive Particle Swarm Optimization (PSO) algorithms. These studies highlight the potential of hybrid models in gene selection for cancer prediction.

A thorough examination of feature selection techniques and cancer prediction methodologies is presented in the study conducted by Abd-Elnaby *et al.* [23]. This comprehensive review provides valuable insights into the current state of research in this field. An important observation to note is that all the introduced wrapper-based methods inherently require a classifier. This observation prompts an intriguing question: would modifying the classifiers used in these methods influence the efficiency of the problem, either in a beneficial or detrimental manner? This potential variability could introduce complexities and pose challenges in the application of these methods.

In this study, we introduce a novel feature selection model that, in addition to selecting the optimal features, enables the determination of the most suitable classifier for use with each dataset. Initially, we selected two swarm-intelligence-based methods, Bat and Gray Wolf, from among meta-heuristic methods and adapted them into two distinct feature selection approaches. Subsequently, from the realm of machine learning algorithms, we selected ten classifiers - KNN [24], SVM [25], Bayes [26], Ridge [27], DT [28], RF [29], Bagging [30], LightGBM [31], Perceptron [32], and LDA [33] - to serve as the fitness function for both algorithms and executed the feature selection process with them. This process was repeated ten times to enhance confidence, owing to the nature of wrapper-based methods. The results were noted, and the optimal classifier was identified among them for comparison with our proposed model. In our proposed model, we incorporated a heuristic component into both algorithms to select the optimal classifier concurrently with the feature selection. This process was also executed ten times for assurance. Furthermore, to demonstrate the efficiency of the proposed method, the test was conducted on four datasets of gene expressions for Leukemia, Prostate, SRBCT, and Colon cancers. Utilizing the results obtained from the Bat and Gray Wolf, using individual classifiers, and comparing them with the proposed method, we demonstrate that the proposed method identifies the optimal classifier on the evaluated datasets. Therefore, for the feature selection process using swarm intelligence algorithms, the optimal classifier can be selected for each dataset.

Figure 1 presents our proposed model steps for selecting the optimal classifier for wrapper-based feature selection methods. To evaluate

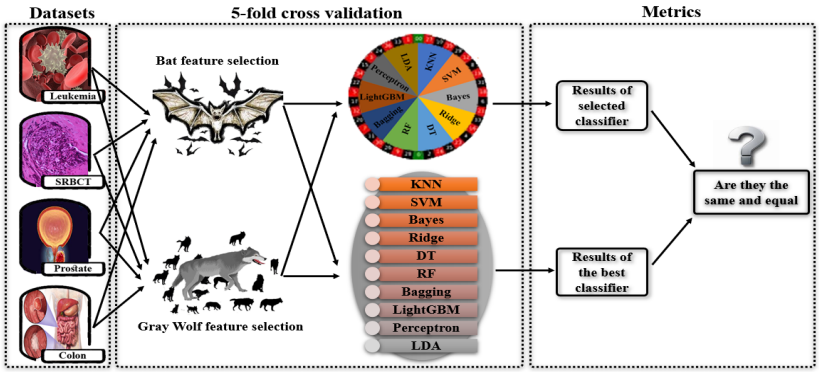

Fig. 1. Our proposed model steps for selecting the optimal classifier for wrapper-based feature selection methods.

the performance of the proposed model, four microarray datasets were employed. The Bat and Gray Wolf optimization algorithms were utilized as dimensionality reduction methods. These datasets were processed using these two meta-heuristic techniques to select the optimal subset of features. The accuracy of the selected features was then evaluated using ten distinct machine learning methods, and the results were documented for comparison. Subsequently, a roulette wheel mechanism was integrated into both the Bat and Gray Wolf algorithms. During the feature selection process, the probability of selecting each of the ten machine learning methods was determined. In the following iteration, one of these ten methods was employed to compute the fitness function via the roulette wheel mechanism. The accuracy of the selected features was recalculated, leveraging this mechanism alongside the top-performing machine learning method (highest probability), and the outcomes were recorded. A 5-fold cross-validation procedure was conducted during the feature selection process by both the Bat and Gray Wolf algorithms, with and without the roulette wheel step, ensuring that all samples were included in the training and testing phases. Finally, a comparative analysis was performed on the results obtained from the Bat and Gray Wolf feature selection methods, both with and without the roulette wheel step, alongside the selected machine learning method.

The main purpose of our study is to identify the optimal classifier to be used as a fitness function within the wrapper-based feature selection process. Selecting the optimal machine learning method tailored to the specific dataset, such as cancer diagnosis, can enhance the accuracy of computations and identify the best subset of features. Our method concentrates on determining the optimal classifier from several candidate machine learning methods. This has been implemented on four cancer datasets using two meta-heuristic algorithms, incorporating the roulette wheel mechanism.

2. MATERIALS AND METHODS

Feature selection constitutes a crucial phase in the machine learning pipeline. It serves to enhance the performance of a model, mitigate overfitting, augment interpretability, and expedite training. By comprehending and implementing suitable feature selection methods, we can construct more efficient and efficacious machine learning models. In this section, we present a novel pipeline capable of identifying the optimal classifier for utilization in wrapper-based feature selection methods.

2.1. Datasets

In this study, we utilized four datasets, as detailed in Table 1, to evaluate the proposed model. These datasets are a type of high-dimensional data structure known as microarrays, which carry genetic information from a given sample. The first dataset employed in this study carries information about Leukemia cancer, distinguishing between two classes: Acute Myeloid Leukemia (AML) and Acute Lymphoblastic Leukemia (ALL). This dataset comprises 7129 identified and quantified genes for 72 patients. We partitioned the dataset into a training set and a test set [34]. The second dataset is the Small Round Blue Cell Tumors (SRBCT), a gene expression dataset from a childhood cancer study [35]. It contains 83 samples with 2308 genes across four classes, with the following distribution: 29 cases of Ewing Sarcoma (EWS), 11 cases of Burkitt Lymphoma (BL), 18 cases of Neuroblastoma (NB), and 25 cases of Rhabdomyosarcoma (RMS). The third dataset pertains to Prostate cancer and consists of 12600 genes across two classes: 77 wild type and 52 Prostate cancer samples [36]. The final dataset is the Colon dataset, composed of 2000 genes and 62 samples taken from Colon cancer patients. Among the samples, there are 40 tumor biopsies (marked as abnormal) and 22 normal [37]. We employed the same strategy of splitting the dataset into a training set and a test set to validate the constructed model.

Table 1. The dataset utilized for evaluating our proposed model is characterized by the number of samples, genes, and classes.

Dataset	Number of samples	Number of genes	Number of classes
Leukemia	72	7129	2
SRBCT	83	2308	4
Prostate	136	12600	2
Colon	62	2000	2

2.2. Feature Selection

In order to implement the wrapper-based feature selection method, we opted for two meta-heuristic algorithms, namely Bat and Gray Wolf, which will be succinctly elucidated. The Bat and Gray Wolf algorithms necessitate an estimator, such as classifiers, to compute the performance of the optimal subset of features. In this study, we selected ten classifiers and executed each of the Bat and Gray Wolf algorithms with them on the four datasets delineated in the preceding section. Owing to the stochastic nature of swarm intelligence methods, the evaluation process is reiterated ten times. Subsequently, the method of selecting the optimal classifier, which is expounded upon in the ensuing sections using these ten classifiers, is implemented. In addition to selecting the optimal subset of data, the optimal classifier is identified on the target dataset without incurring any additional time overhead.

2.3. Bat Algorithm

Bats are intriguing creatures whose advanced echolocation capabilities have garnered the attention of researchers across various fields. Echolocation operates akin to sonar: primarily micro-bats emit a loud, brief pulse of sound, which, upon striking an object, returns as an echo after a fraction of time [38]. Consequently, bats can calculate their distance from an object [39]. Moreover, this remarkable orientation mechanism enables bats to discern the difference between an obstacle and prey, facilitating hunting even in complete darkness [40]. All bats utilize echolocation to gauge distance, and they also possess the ability to distinguish between food/prey and background barriers; A bat 'b' flies randomly with velocity 'v' at position 'x' with a fixed frequency 'f_{min}', varying wavelength 'λ', and loudness 'A_0' to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and modulate the rate of pulse emission 'R ϵ [0, 1]', contingent on the proximity of their target. Although the loudness can vary in numerous ways, Yang [41] posits that the loudness varies from a large (positive) 'A_0' to a minimum constant value 'A_{min}'.

Algorithm1. Feature selection by Bat Algorithm.

1. Initialize the bat population X_b, random number in {0, 1}, b = 1, 2, ..., nBats.
2. Initialize the loudness A_b, random number in range [1, 2], b = 1, 2, ..., nBats.
3. Initialize the pulse emission rate R_b, random number in range [0, 1], b = 1, 2, ..., nBats.
4. Initialize the velocity $V_b = 0$, b = 1, 2, ..., nBats.
5. Initialize the fitness $fitness_b = -\infty$, b = 1, 2, ..., nBats.
6. Initialize the globalFitness = -∞.
7. Initialize the maximum and minimum frequency, $f_{min} = 0$, $f_{max} = 1$.
8. For (t in maxIter):
9. For (b in nBats):
10. Find ($X_b == 1$) and make train and test subsets.
11. Calculate accuracy.
12. accuracy = (theta × accuracy + (1-theta) × (1-$\frac{sum(X_b==1)}{nFeatures}$))
13. If (rand < A_b & accuracy > $fitness_b$):
14. $fitness_b$ = accuracy
15. $A_b = \alpha A_b$
16. $R_b = R_b^0[1-\exp(-\gamma t)]$
17. [maxFitness, maxIndex] = max(fitness)
18. If (maxFitness > globalFitness):
19. globalFitness = maxFitness
20. xHat = $x_{maxIndex}$
21. For (b in nBats):
22. If (rand > R_b):
23. $X_b = X_b + \epsilon \bar{A}$
24. Convert X_b to {0, 1} by Sigmoid function
25. If (rand < A_b & $fitness_b$ > globalFitness):
26. $F_b = f_{min} + (f_{max} - f_{min})$ rand
27. $V_b = V_b + (xHat - X_b) F_b$
28. $X_b = X_b + V_b$
29. Convert X_b to {0, 1} by Sigmoid function
30. Return xHat.

Algorithm 1, shows the process of feature selection by Bat Algorithm. Initially, a population with random positions is established as candidate genes for each bat. Subsequently, the loudness, pulse emission rate, velocity, minimum and maximum frequency, and fitness are initialized. In each iteration, candidate genes are specified for each bat, and the accuracy of the selected subset is calculated. Based on the number of candidate genes, the fitness of each bat is then evaluated. Lines 13 to 20 indicate the best position and global fitness if the fitness is improved. Lines 21 to 29 update the positions of the bats. Finally, Line 30 returns the best subset of selected genes (best position) as the final solution.

2.4. Gray Wolf Algorithm

The Gray Wolf algorithm is a representative swarm-intelligence algorithm, inspired by the leadership hierarchy and hunting mechanism of gray wolves in nature. Gray wolves, recognized as apex predators, typically have an average group size of 5–12. Within the hierarchy of the Gray Wolf algorithm, the alpha (α) is considered the most dominant member of the group. The remaining subordinates, beta (β) and delta (δ), assist in controlling the majority of wolves in the hierarchy, which are considered as omega (ω). The ω wolves hold the lowest ranking within the hierarchy [42]. The mathematical model of the hunting mechanism of gray wolves comprises the following stages:

- Tracking, chasing, and approaching the prey.

- Pursuing, encircling, and harassing the prey until it ceases movement.
- Attacking the prey.

During the hunt, gray wolves encircle the prey. The hunting of prey is typically guided by α, β, and δ, who participate occasionally. The best candidate solutions, namely α, β, and δ, possess superior knowledge about the potential location of the prey. The other search agents (ω) update their positions according to the positions of the three best search agents [43].

Algorithm 2, illustrates the process of feature selection by the Gray Wolf Algorithm. Initially, a population is generated with random positions representing candidate genes for each wolf. Subsequently, the values of the parameters a, A, and C are initialized. Candidate genes are identified for each wolf, and the accuracy of the selected subset is calculated. The fitness of each wolf is then evaluated based on the number of candidate genes. Line 6 specifies the first, second, and third best wolves. During each iteration of the algorithm (lines 7 to 13), the positions and parameters of a, A, and C are updated, and both accuracy and fitness are recalculated. The positions of the top three wolves are updated accordingly, reflecting any improvements in global fitness. Finally, line 14 returns the best subset of the selected genes (the position of the best wolf) as the solution.

Algorithm 2. Feature selection by Gray Wolf Algorithm.
1. Initialize the gray wolf population X_w, random number in {0, 1}, w = 1, 2, ..., nWolves.
2. Initialize a, A and C.
3. Find ($X_w == 1$) and make train and test subsets.
4. Calculate accuracy.
5. fitness = (theta × accuracy + (1-theta) × (1- $\frac{\text{sum}(X_w == 1)}{\text{nFeatures}}$))
6. X_α, X_β, X_δ = The first, second and third best wolves.
7. For (t in maxIter):
8. For (w in nWolves):
9. Update the position of the current wolf.
10. Update a, A, C.
11. Calculate accuracy.
12. fitness = (theta × accuracy + (1-theta) × (1- $\frac{\text{sum}(X_w == 1)}{\text{nFeatures}}$))
13. Update X_α, X_β, X_δ.
14. Return X_α

2.5. Classifiers

In order to assess the subset of candidate features selected through the feature selection process, we employed two algorithms as described in the preceding section. Among the machine learning classification algorithms, we utilized ten classifiers: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes, Ridge Regression, Decision Tree (DT), Random Forest (RF), Bagging Classifier, Light Gradient Boosting Machine (LightGBM), Perceptron, and Linear Discriminant Analysis (LDA). Each of these classifiers was independently applied in the feature selection processes of both the Bat Algorithm and the Gray Wolf Optimization Algorithm. For the KNN classifier, the number of neighbors was set to 7. For ensemble methods such as RF and LightGBM, the number of weak learners was set to 100. The parameters for all other classifiers were kept at their default settings. It should be noted that the choice of classifiers is not limited to the ones used in this study, and any other classifier can be incorporated as needed.

2.6. Selecting the Optimal Classifier

In order to identify the optimal classifier from the ten methods selected during the feature selection process, we introduced modifications and incorporated a heuristic component into the Bat and Gray Wolf algorithms. This approach ensures that while the best features are being selected, the most suitable classifier for each dataset is also identified. In both the Bat and Gray Wolf feature selection methods, we initially generate a random population. For both methods, ten agents are considered. Each agent possesses a position represented by a binary vector, with candidate features assigned a specific value within this vector. To evaluate the fitness of candidate features, we employ the ten classifiers mentioned earlier in the cost function. Initially, each agent is randomly assigned one of the classifiers, and the fitness of candidate features is evaluated. In addition to assessing the fitness of each agent, we also retain the classifier with which that agent was evaluated. Up to this point, no additional computational burden has been imposed on the algorithms, meaning the performance of the ten agents with the ten classifiers has been estimated. Meta-heuristic methods, particularly swarm intelligence algorithms, commence with an initial population, and the algorithm strives to converge towards the most favorable positions. In both of our feature selection methods (Bat and Gray Wolf), changes in agents' positions are introduced in each round to guide them towards the optimal solutions. In addition to evaluating the fitness of each agent in the current position, the fitness in new positions must also be assessed with a classifier. If these positions are superior, they are updated; otherwise, the new position is disregarded.

Prior to initiating the second iteration, we implement a Roulette Wheel using the calculated score for each classifier. The Roulette Wheel Selection algorithm, also known as Fitness Proportionate Selection, is a method employed in genetic algorithms [44] to select potentially beneficial solutions for recombination. Each potential solution (in this case, features selected by agents) in the population is assigned a fitness score based on its quality or suitability. Each classifier's fitness score is then utilized to assign a proportion of the roulette wheel. The higher the fitness score, the larger the classifier's portion of the wheel. The proportion of the Roulette Wheel assigned to each classifier is calculated as follows:

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \qquad (1)$$

Where p_i represents the proportion of the i-th classifier, f_i denotes the fitness of the i-th agent as determined by the i-th classifier, f_j signifies the fitness of the j-th classifier, and n is the total number of classifiers.

In the second iteration, the implemented Roulette Wheel is utilized to assign each classifier. A random number is generated within the range of 0 to 1. The classifier whose segment encompasses this number is selected. This process is iteratively performed until all agents are selected for the second generation. The fundamental concept is that classifiers with superior fitness have a higher probability of being selected, yet there remains a possibility for less fit classifiers to be chosen. This strategy maintains diversity within the population and prevents premature convergence. From this iteration forward, each time there is a requirement to calculate the fitness of an agent, the Roulette Wheel is updated. Consequently, classifiers that have yielded superior results will likely have more selection opportunities.

Algorithm 3 illustrates the process of identifying the optimal classifier in conjunction with selecting the best features. Initially, a classifier is assigned to each agent (bat or wolf). The fitness of each agent is then evaluated using the assigned classifier. Based on the obtained fitness, the probability of selecting each classifier is initialized. A roulette wheel is constructed using these probabilities. Between lines 5 and 10, the position of each agent is updated using Algorithms 1 and 2. Subsequently, a classifier is selected for each agent using the roulette wheel, and the fitness is recalculated with the assigned classifier. The probabilities of the classifiers and the roulette wheel are then updated accordingly. Finally, in line 11, the best subset of genes (optimal positions) identified by the best classifier is returned as the final solution.

Algorithm 3. Rating the classifiers by Roulette Wheel probabilities.
1. Assign a classifier to each agent randomly.
2. Calculate the fitness of each agent by assigned classifier.
3. Calculate the classifiers selecting probabilities by obtained fitness.
4. Make a Roulette Wheel.
5. For (t in maxIter):
6. Update the Bats and Wolves positions by algorithms 1 and 2.
7. Select a classifier by Roulette Wheel for each agent.
8. Calculate the fitness of each agent by assigned classifier.
9. Update the classifiers selecting probabilities by obtained fitness.
10. Update the Roulette Wheel.
11. Return best classifier and best position.

The number of features selected by the Bat and Gray Wolf methods varies in each iteration. Therefore, to evaluate the candidate features in proportion to their numbers, we utilized the following equation:

$$\text{Fitness} = (\theta \times \text{accuracy} + (1-\theta) \times (1 - \frac{\Sigma(x_i = 1)}{\text{nFeatures}})) \qquad (2)$$

The number of Bats and Wolves was set to 10, and the termination condition was established at 50. To ensure the participation of all samples in both the training and testing processes, we employed a 5-fold cross-validation technique and computed the mean value.

3. RESULTS AND DISCUSSION

In this section, we examine the proposed method. We employed two dimensionality reduction methods, namely the Bat and Gray Wolf algorithms, to select the optimal genes from four cancer datasets. Ten classifiers were used to evaluate candidate data subsets by both the Bat and Gray Wolf methods. For the Bat method, we considered 10 bats and 50 repetitions, and for the Gray Wolf method, we considered 10 wolves and 50 repetitions. To ensure all data were included in both the training and testing processes, we utilized a 5-fold cross-validation method. For increased certainty, the feature selection process was repeated 10 times with both methods. The results are presented in Tables 2 to 9 (given in supplementary data).

As can be observed in Table 2, the best values obtained from 10 executions on the leukemia cancer dataset show that the KNN classifier achieved an accuracy of 84.76% with 3767 features out of 7129.

The SVM method achieved an accuracy of 90.29% with 3984 features. The Bayes method yielded better results, achieving an accuracy of 98.57% with 3031 features. The Ridge method achieved the same accuracy as the Bayes method, but with fewer selected features (2545 genes). The DT method achieved an accuracy of 95.81% with 5224 features. The RF method achieved an accuracy of 98.57% with 3031 features. The Bagging method, in its best performance, obtained an accuracy of 97.24% with 4536 features. The best result, 100% accuracy, was achieved by LightGBM with 4828 features. The Perceptron method achieved an accuracy of 96.00% with 3300 features, and finally, the LDA method achieved an accuracy of 91.90% with 4834 features in its best iteration.

After noting the above results, the Bat feature selection method was implemented using the same 10 classifiers, but with the proposed method. In each repetition of the Bat algorithm, after recording the obtained accuracy, the roulette wheels are updated with new values for the classifiers. Therefore, in the next iteration, the probability of choosing a classifier that has yielded better results is higher. The results of executing the above process 10 times on the leukemia cancer dataset are shown in the last part of Table 2. As can be seen, LightGBM, Bayes, Bagging, and Ridge have been selected as the best classifiers in different iterations. Among these, LightGBM, with 5260 features, achieved the best result that was discussed in the previous section, 100 percent. As a result, the best classifier has been found among the 10 used classifiers. In Figure 2(a), the change in the values of the selection probabilities of the classifiers in iterations 1, 10, 20, 30, 40, and 50 is shown. Classifiers such as KNN and SVM, due to poor results, had their values remain constant on the Roulette Wheel. In contrast, classifiers such as LightGBM, with more repetitions, saw an increase in the probability of their selection.

With respect to the SRBCT cancer dataset (as shown in Table 3), the Bat feature selection method, in its optimal performance, achieved an accuracy of 100%. Classifiers such as Bayes, Ridge, Random Forest (RF), Light Gradient Boosting Machine (LightGBM), and Perceptron yielded similar results. As can be observed in the final section of Table 3, our proposed method identified the Ridge classifier as the most effective for the feature extraction process using the Bat method, achieving the highest accuracy of 100%. Figure 2(b) illustrates the probabilities associated with selecting classifiers for the feature selection process, utilizing the Bat method on the Small Round Blue Cell Tumors (SRBCT) cancer dataset. Table 4 presents the outcomes of the aforementioned method on the prostate cancer dataset. It is evident that due to the substantial number of features (12,600 genes), the most favorable result is an accuracy of 93.44% with 9,889 features, which is associated with the Ridge classifier. The Light Gradient Boosting Machine (LightGBM) also achieved an accuracy of 93.41% with 6,968 features. Upon examining the final section of Table 4, the accuracy and efficacy of our proposed method become apparent. Among the classifiers, LightGBM has achieved the highest accuracy of 94.12% with 7,527 features, thereby establishing itself as the optimal classifier in this context. The Bayes, Random Forest (RF), and Perceptron methods maintained a constant selection probability across different iterations due to their relatively weaker performance, as illustrated in Figure 2(c). Notably, the selection probability for the Light Gradient Boosting Machine (LightGBM) increased consistently across different iterations due to its superior performance.

In our analysis of the final dataset, Colon cancer (presented in Table 5), two classifiers demonstrated superior results compared to others across various executions. The Ridge classifier, with 923 features, and the Linear Discriminant Analysis (LDA) classifier, with 853 features out of 2000 genes, both achieved an accuracy of 91.92%. In the final section of Table 5, our method identified LDA as the best classifier, selecting 914 features with an accuracy of 87.18%. Figure 2(d), similar to other Figures, displays the selection probabilities of classifiers across different iterations. To validate the effectiveness of our proposed method, we also implemented the entire process using the Gray Wolf feature selection method. The results are presented in Tables 6, 7, 8, and 9. As shown in Table 6, our Gray Wolf-based method selected the LightGBM classifier for the Leukemia dataset, achieving 100% accuracy, which matches the best result obtained from the implementation of Greg's method with individual classifiers. For the SRBCT dataset, our method selected the Bayes classifier, achieving 100% accuracy, as can be verified by referring to Table 7. This suggests that it may be the most suitable classifier for this data. Upon examining the

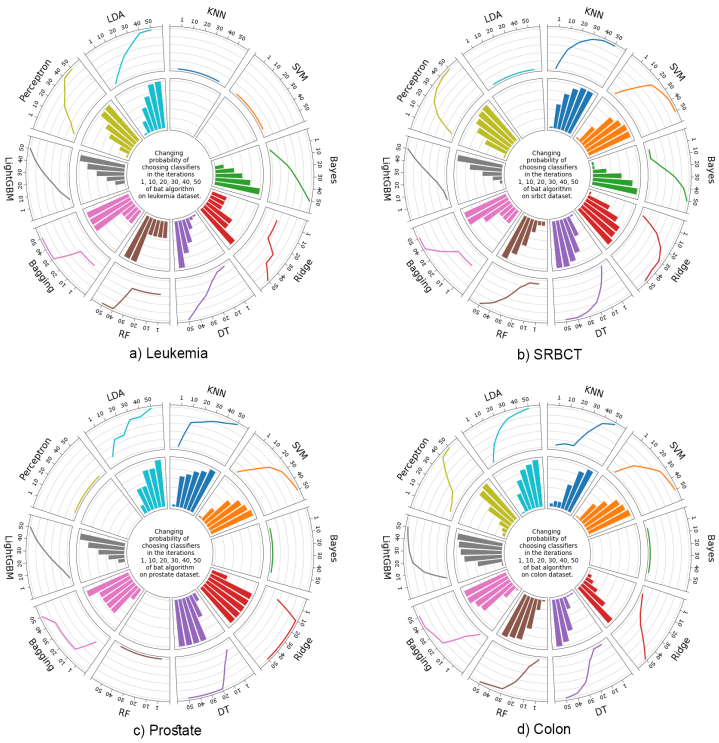

Fig. 2. Probabilities of selecting optimal classifiers for the feature selection process using the Bat algorithm on various cancer datasets.

Prostate cancer dataset with our proposed method, the LightGBM classifier selected 7747 features and achieved an accuracy of 94.13%, as can be verified by referring to Table 8. This indicates that it is the optimal classifier. As can be seen in Table 9, the proposed Gray Wolf feature selection method also identified the best classifier for the Colon cancer dataset, LDA, which achieved an accuracy of 91.92%.

Figure 3, illustrate the probabilities associated with selecting different classifiers across various iterations, utilizing the Gray Wolf feature selection method on a range of cancer datasets. Figure 3(a) presents the probabilities obtained at iterations 1, 10, 20, 30, 40, and 50 using our proposed method for leukemia cancer. As can be seen, the probability of choosing KNN, LDA and Perceptron methods is less than other methods. Similarly, Figures 3(b), 3(c), and 3(d) illustrate the probabilities obtained for SRBCT, Prostate, and Colon cancers, respectively.The experimental procedures were conducted on a computer system equipped with a 2.40 GHz processor, 8.0 GB of RAM, and operating on the Windows 10 platform. Feature selection is crucial in machine learning methodologies, as it identifies irrelevant features, thereby enhancing accuracy and reducing computational time. Due to the superior results of wrapper-based methods compared to filter-based and embedding-based methods, meta-heuristic approaches have been employed in various feature selection problems. The fitness function that estimates the merit of the selected features by these methods is of significant importance and can greatly influence both accuracy and efficiency. The challenge lies in determining the most suitable algorithm for estimating the merit of a specific dataset. The proposed method was evaluated against a number of existing methods, with the comparative results presented in Table 10. This comparison provides a comprehensive understanding of the performance of the proposed method relative to established methodologies.

Table 10. The accuracy of our proposed method versus several existing methods for comparative analysis.

Method	Leukemia	SRBCT	Prostate	Colon
Aydadenta [12]	N/A	N/A	88.97	85.87
Cilia (NN) [13]	99.44	N/A	N/A	91.94
Wu [19]	97.81	N/A	N/A	88.36
Jian [21]	100	100	N/A	95.00
Our proposed model	100	100	94.13	91.92

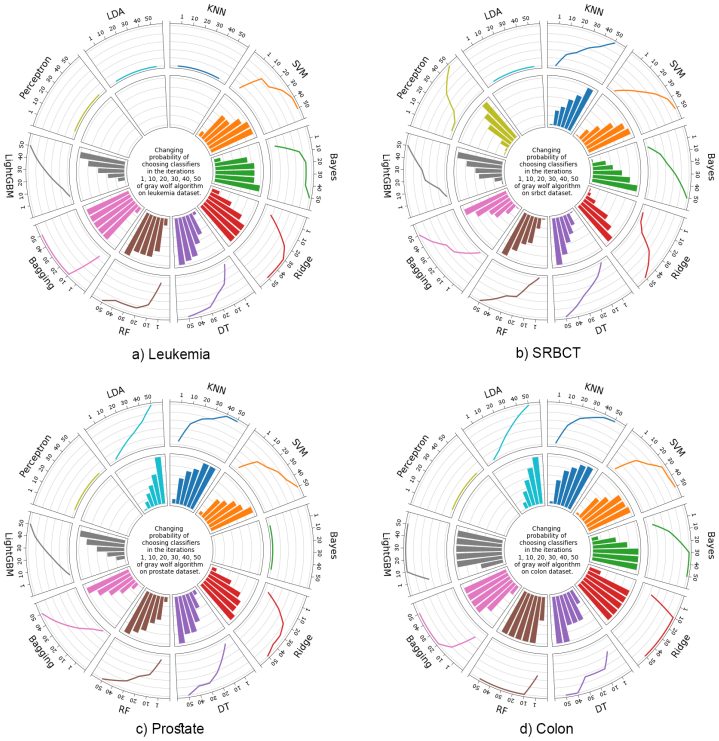

Fig. 3. Probabilities of selecting optimal classifiers for the feature selection process using the Gray Wolf algorithm on various cancer datasets.

4. CONCLUSIONS

In the present study, we introduced a simple yet practical method to select the optimal classifier in the fitness function. Our proposed method demonstrates which classifier is best suited for the studied collection. To implement the proposed model, we utilized two metaheuristic algorithms based on the swarm intelligence of the bat and the gray wolf. We then conducted the feature selection process using 10 various classifiers separately and recorded the results.

Subsequently, we integrated our proposed method into both the bat and gray wolf algorithms, performed the feature selection process with them, and noted the selected classifiers apart from the results. We analyzed Leukemia, SRBCT, Prostate, and Colon cancer datasets to evaluate our proposed method. Our findings indicate that our model identifies the best classifier for the desired dataset without incurring computational overhead while selecting the best features. Consequently, when using wrapper-based methods for feature selection, the optimal classifier can be selected for the investigated dataset.

5. DATA AVAILABILITY

The datasets examined in this study, along with the corresponding Python codes and results, are accessible to editors, reviewers, and readers via our GitHub page at the following URL: https://github.com/Farzad-Zandi/Selecting-Optimal-Classifier.

6. CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

7. REFERENCES

1. A. Jović, K. Brkić, and N. Bogunović. A review of feature selection methods with applications. *38th international convention on information and communication technology, electronics and microelectronics (MIPRO), Ieee, Opatija, Croatia, pp.* 1200-1205 (2015).
2. P.E. Greenwood and M.S. Nikulin (Eds.). A guide to chi-squared testing. *John Wiley & Sons, New York, USA* (1996).
3. B. Azhagusundari and A.S. Thanamani. Feature selection based on information gain. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* 2(2): 18-21 (2013).

4. A.G. Asuero, A. Sayago, and A. González. The correlation coefficient: An overview. *Critical Reviews in Analytical Chemistry* 36(1): 41-59 (2006).
5. X.W. Chen and J.C. Jeong. Enhanced recursive feature elimination. *6th International Conference on Machine Learning and Applications (ICMLA), IEEE, Cincinnati, OH, USA, pp.* 429-435 (2007).
6. V. Kumar and S. Minz. Feature selection. *SmartCR* 4(3): 211-229 (2014).
7. V. Fonti and E. Belitser. Feature selection using lasso. *VU Amsterdam Research Paper in Business Analytics* 30: 1-25 (2017).
8. X.S. Yang. Metaheuristic optimization. *Scholarpedia* 6(8): 11472 (2011).
9. P. Agrawal, H.F. Abutarboush, T. Ganesh, and A.W. Mohamed. Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *Ieee Access* 9: 26766-26791 (2021).
10. I. Muhammad and Z. Yan. Supervised machine learning approaches: a survey. *ICTACT Journal on Soft Computing* 5(3): 946-952 (2015).
11. M.D. Purbolaksono, K.C. Widiastuti, M.S. Mubarok, and F.A. Ma'ruf. Implementation of mutual information and bayes theorem for classification microarray data. *Journal of Physics: Conference Series* 971(1): 012011 (2018).
12. H. Aydadenta and A. Adiwijaya. A clustering approach for feature selection in microarray data classification using random forest. *Journal of Information Processing Systems* 14(5): 1167-1175 (2018).
13. N.D. Cilia, C. De Stefano, F. Fontanella, S. Raimondo, and A. Scotto di Freca. An experimental comparison of feature-selection and classification methods for microarray datasets. *Information* 10(3): 109 (2019).
14. V. Bolón-Canedo, N. Sánchez-Marono, A. Alonso-Betanzos, J.M. Benítez, and F. Herrera. A review of microarray datasets and applied feature selection methods. *Information sciences* 282: 111-135 (2014).
15. M. Al-Batah, B. Zaqaibeh, S.A. Alomari, and M.S. Alzboon. Gene microarray cancer classification using correlation based feature selection algorithm and rules classifiers. *International Journal of Online & Biomedical Engineering* 15(8): 62 (2019).
16. S.K. Baliarsingh, C. Dora, and S. Vipsita. Jaya optimized extreme learning machine for breast cancer data classification. *Intelligent and Cloud Computing: Proceedings of ICICC 2019* 2: *Springer, Singapore, pp.* 459-467 (2021).
17. Q. Su, Y. Wang, X. Jiang, F. Chen, and W.C. Lu. A cancer gene selection algorithm based on the KS test and CFS. *BioMed Research International* 1: 1645619 (2017).
18. F.K. Ahmad. A comparative study on gene selection methods for tissues classification on large scale gene expression data. *Jurnal Teknologi* 78(5-10): 116-125 (2016).
19. Q. Wu, Z. Ma, J. Fan, G. Xu, and Y. Shen. A feature selection method based on hybrid improved binary quantum particle swarm optimization. *Ieee Access* 7: 80588-80601 (2019).
20. S.A. Medjahed, T.A. Saadi, A. Benyettou, and M. Ouali. Kernel-based learning and feature selection analysis for cancer diagnosis. *Applied Soft Computing* 51: 39-48 (2017).
21. I. Jain, V.K. Jain, and R. Jain. Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification. *Applied Soft Computing* 62: 203-215 (2018).
22. S. Shahbeig, M.S. Helfroush, and A. Rahideh. A fuzzy multi-objective hybrid TLBO–PSO approach to select the associated genes with breast cancer. *Signal Processing* 131: 58-65 (2017).
23. M. Abd-Elnaby, M. Alfonse, and M. Roushdy. Classification of breast cancer using microarray gene expression data: A survey. *Journal of Biomedical Informatics* 117: 103764 (2021).
24. O. Kramer (Ed.). Dimensionality reduction with unsupervised nearest neighbors. *Springer, Berlin, Heidelberg, Germany* (2013).
25. D.A. Pisner and D.M. Schnyer. Support vector machine. *Machine learning, Elsevier, Amesterdam, Netherlands, pp.* 101-121 (2020).
26. M.E. Glickman and D.A. Van Dyk. Basic bayesian methods. *Topics in Biostatistics* 404: 319-338 (2007).
27. C. Peng and Q. Cheng. Discriminative ridge machine: A classifier for high-dimensional data or imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems* 32(6): 2595-2609 (2020).
28. Y.Y. Song and L.U. Ying. Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry* 27(2): 130 (2015).
29. L. Breiman. Random forests. *Machine learning* 45: 5-32 (2001).
30. L. Breiman. Bagging predictors. *Machine Learning* 24: 123-140 (1996).
31. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *31st Conference*

on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA (2017).
32. M.C. Popescu, V.E. Balas, L. Perescu-Popescu, and N. Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems* 8(7): 579-588 (2009).
33. P. Xanthopoulos, P.M. Pardalos, T.B. Trafalis, P. Xanthopoulos, P.M. Pardalos, and T.B. Trafalis (Eds.). Linear discriminant analysis. In: Robust Data Mining. *SpringerBriefs in Optimization. Springer, New York* pp. 27-33 (2013).
34. T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, and *et al*. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286(5439): 531-537 (1999).
35. J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, ... and P.S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* 7(6): 673-679 (2001).
36. K. Nirmalakumari, H. Rajaguru, and P. Rajkumar. Microarray prostate cancer classification using eminent genes. In: *2021 Smart Technologies, Communication and Robotics (STCR). Sathyamangalam, India, pp.* 1-5. (2021).
37. 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. *Proceedings of the National Academy of Sciences* 96(12): 6745-6750 (1999).
38. D.R. Griffin, F.A. Webster, and C.R. Michael. The echolocation of flying insects by bats. *Animal Behavior* 8(3-4): 141-154 (1960).
39. W. Metzner. Echolocation behaviour in bats. *Science Progress* 75: 453-465 (1991).
40. H.U. Schnitzler and E.K. Kalko. Echolocation by insect-eating bats: we define four distinct functional groups of bats and find differences in signal structure that correlate with the typical echolocation tasks faced by each group. *Bioscience* 51(7): 557-569 (2001).
41. X.S. Yang. Bat algorithm for multi-objective optimization. *International Journal of Bio-Inspired Computation* 3(5): 267-274 (2011).
42. S. Mirjalili, S.M. Mirjalili, and A. Lewis. Grey wolf optimizer. *Advances in Engineering Software* 69: 46-61 (2014).
43. N. Mittal, U. Singh, and B.S. Sohi. Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing* 1: 7950348 (2016).
44. S. Anand, N. Afreen, and S. Yazdani. A novel and efficient selection method in genetic algorithm. *International Journal of Computer Applications* 129(15): 7-12 (2015).

SUPPLEMENTARY DATA

Table 2. The accuracy values obtained from the Bat feature selection algorithm were evaluated using ten different classifiers on the Leukemia dataset.

KNN		SVM		Bayes		Ridge		DT		RF		Bagging		LightGBM		Perceptron		LDA		Proposed		
nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	Selected Classifier
5311	83.43	5978	87.62	3535	98.57	4835	98.57	3960	94.57	3033	98.57	4536	97.24	5501	100	4879	90.38	5420	90.38	5258	98.57	Ridge
5087	84.76	3984	90.29	4542	98.57	5700	98.57	4606	93.05	3573	98.57	4491	97.12	4828	100	3496	95.90	5191	90.48	4481	98.67	LightGBM
5649	84.76	5071	87.62	3831	98.57	4274	98.57	4295	93.05	4843	98.57	4451	96.89	5980	98.57	5330	95.90	5068	90.48	5260	100	LightGBM
3611	83.43	5201	86.19	3331	98.57	4320	98.57	5745	93.05	5791	98.57	4429	97.01	5402	100	4220	94.57	5777	90.38	5271	98.57	Bayes
5065	83.43	5345	87.62	4523	98.57	5717	98.57	4059	94.48	3031	98.57	4578	97.24	5736	98.57	3300	96.00	3390	91.81	3347	98.57	Bagging
3767	84.76	3024	86.19	3965	98.57	5681	98.57	3449	93.05	5577	98.57	4459	96.76	5483	98.57	3375	94.57	3343	90.48	2971	9857	Bagging
4554	84.76	5519	86.19	3031	98.57	5880	98.57	3758	93.05	3954	98.57	4821	97.01	5237	98.57	4563	92.00	4834	91.90	3977	98.67	LightGBM
5354	84.76	2386	86.19	3460	98.57	2545	98.57	4138	93.05	3171	98.57	4675	97.12	4971	100	4257	90.67	5713	90.38	6082	98.57	Bayes
5754	84.76	5919	87.62	3052	98.57	3778	98.57	3501	93.14	4876	98.57	4875	96.93	4949	100	5244	90.57	4617	89.05	4054	98.57	Ridge
5567	83.43	4325	86.19	5887	98.57	4064	98.57	5224	95.81	3180	98.57	4931	97.03	5780	98.57	4967	95.90	3145	89.05	5707	98.57	Bagging

Table 3. The accuracy values obtained from the Bat feature selection algorithm were evaluated using ten different classifiers on the SRBCT dataset.

KNN		SVM		Bayes		Ridge		DT		RF		Bagging		LightGBM		Perceptron		LDA		Proposed		
nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	Selected Classifier
1456	90.15	899	97.57	847	100	1387	100	1879	92.72	1498	100	1270	98.82	1621	100	1032	100	1803	78.31	983	100	Ridge
1339	87.72	1194	97.57	1892	100	1129	100	1828	93.97	1704	100	1356	98.75	1649	100	1303	98.82	1053	74.93	1244	100	Ridge
1166	87.72	1774	97.57	1473	100	1731	100	1353	91.47	1596	100	1321	98.46	1670	100	1774	98.82	1581	72.28	1130	100	Ridge
1765	90.15	1893	97.57	1539	100	976	100	1459	93.97	1678	100	1293	98.78	1691	100	1300	98.82	1183	69.85	1584	100	Ridge
1058	87.79	1109	97.57	1932	100	1416	100	1064	92.79	1579	100	1311	98.64	1673	100	1774	98.82	1290	77.13	1614	100	Ridge
1413	87.79	1425	97.57	1898	100	1762	100	1107	92.72	1564	100	1342	98.26	1752	100	1950	98.82	1618	72.43	1897	100	Ridge
1052	87.72	1567	97.57	988	100	1581	100	1746	93.97	1626	100	1281	98.01	1638	100	1389	98.82	1795	71.10	1880	100	Ridge
1778	91.32	1130	97.57	1811	100	1354	100	1860	92.79	1653	100	1330	97.76	1652	100	1378	98.82	1851	71.18	1956	100	Ridge
1866	87.65	1791	97.57	1173	100	1275	100	1564	93.97	1527	100	1262	98.37	1679	100	1877	100	1577	75.00	1468	100	Ridge
1062	87.87	1820	97.57	1224	100	1438	100	1514	92.79	1546	100	1257	97.89	1713	100	1831	100	1457	72.35	1449	100	Ridge

Table 4. The accuracy values obtained from the Bat feature selection algorithm were evaluated using ten different classifiers on the Prostate dataset.

KNN		SVM		Bayes		Ridge		DT		RF		Bagging		LightGBM		Perceptron		LDA		Proposed		
nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	Selected Classifier
9813	80.93	7538	80.19	5135	56.67	9889	93.44	10222	85.34	7606	92.72	9872	91.93	6968	93.41	8080	81.75	9530	86.11	9081	92.70	LightGBM
6722	80.19	8170	81.64	7026	56.67	9628	92.70	8138	88.20	6561	91.75	9991	91.93	7002	90.91	5341	81.75	9229	84.66	8447	93.39	LightGBM
6054	79.47	8390	80.19	8445	56.67	9477	92.72	10645	86.75	8012	91.26	9615	91.47	6898	92.49	7102	80.26	9177	86.08	8910	92.01	Bagging
5772	80.19	7116	79.47	10422	56.67	8910	91.27	9828	87.76	7316	92.41	9646	91.78	6416	92.17	9230	85.24	7512	86.11	9984	92.67	LightGBM
10252	78.73	8780	80.19	8258	56.67	6571	92.70	9159	87.92	7189	91.43	9719	90.99	7137	91.46	7068	81.64	7029	87.59	7378	92.72	RF
7986	80.90	5266	79.44	9750	56.67	8987	92.72	10573	88.09	7145	92.66	9486	90.41	6459	91.73	10058	81.69	9102	84.60	7527	94.12	LightGBM
9315	80.21	8374	79.47	4581	56.67	10182	92.72	8845	86.26	7416	91.83	9629	91.38	6743	90.78	9160	84.68	8994	85.34	9827	91.93	LightGBM
7346	80.19	8284	80.16	8415	56.67	7353	91.27	8291	85.47	6985	92.58	9813	90.12	6972	91.49	6743	82.43	9347	86.85	5867	92.70	RF
6453	80.19	10165	80.93	5003	56.67	8407	92.72	8697	86.59	7373	91.22	9687	91.28	6842	92.83	10194	81.00	7067	84.63	6079	93.44	LightGBM
5013	80.19	6380	81.67	10520	56.67	9540	91.98	10357	86.38	7842	91.68	9649	91.09	6794	93.30	8382	83.17	8204	85.34	7043	91.24	RF

Table 5. The accuracy values obtained from the Bat feature selection algorithm were evaluated using ten different classifiers on the Colon dataset.

KNN		SVM		Bayes		Ridge		DT		RF		Bagging		LightGBM		Perceptron		LDA		Proposed		
nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	Selected Classifier
838	80.64	1109	87.31	1071	70.64	1238	90.26	916	88.97	1030	85.51	1161	88.97	1178	87.18	1331	87.05	1392	91.92	1013	87.05	LDA
1431	77.56	1386	84.23	1150	67.44	872	90.26	1680	84.10	1127	83.19	1073	90.13	1150	85.64	1505	88.59	732	91.92	1263	85.51	LDA
1126	77.56	1620	84.23	1346	69.23	1350	91.92	1139	85.64	1132	84.61	1692	87.18	912	88.85	1249	86.92	1519	91.92	1598	87.18	LDA
1069	77.44	1254	84.23	1117	69.10	1615	91.92	1510	84.10	1039	85.12	1529	88.42	1415	88.72	1571	87.05	853	91.92	1661	87.18	LDA
1242	79.10	1103	84.23	762	69.10	1534	90.38	1574	87.18	1143	83.49	1743	88.19	1107	85.64	1408	86.92	1520	91.92	914	87.18	LDA
924	77.56	722	85.77	1024	72.31	1108	91.92	1454	87.05	1087	84.67	1746	88.84	1197	87.31	1189	86.92	1450	90.38	807	85.51	LDA
1331	79.10	1650	84.23	1052	70.77	1495	90.26	923	85.64	1053	85.20	1673	87.76	1041	87.18	1136	86.92	1094	91.92	1701	85.51	LDA
1518	77.56	1567	85.77	1747	69.10	1370	91.92	1161	87.44	969	84.89	1694	88.39	1512	85.77	1699	86.92	967	90.38	1285	85.51	LDA
995	80.64	1104	84.23	818	70.64	1343	91.92	1566	85.64	993	84.51	1358	87.94	1477	88.72	1418	88.59	1051	90.38	1503	85.51	LDA
1406	79.10	1388	85.77	1341	70.64	923	91.92	1383	90.26	1057	83.24	1361	88.09	1250	85.64	1378	87.05	1476	91.92	1327	85.51	LDA

Table 6. The accuracy values obtained from the Gray Wolf feature selection algorithm were evaluated using ten different classifiers on the Leukemia dataset.

KNN		SVM		Bayes		Ridge		DT		RF		Bagging		LightGBM		Perceptron		LDA		Proposed		
nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	Selected Classifier
4191	83.43	4269	87.62	4145	98.57	3585	98.57	4311	93.05	4158	98.57	4356	97.24	4265	100	4577	94.48	4472	90.38	4177	98.57	Ridge
4263	84.76	3590	86.19	3550	98.57	3542	98.57	4480	91.62	3484	98.57	4174	97.24	4572	100	4529	95.90	4299	90.48	3504	98.57	Ridge
3565	83.43	3532	86.19	4183	98.57	3588	98.57	4554	94.57	3504	98.57	4528	97.24	4493	100	4532	93.24	4448	90.48	4070	98.57	LightGBM
4517	84.76	3641	86.19	3586	98.57	3605	98.57	4523	93.05	4221	98.57	4496	95.90	4592	100	4395	94.57	4434	90.48	4206	98.57	Ridge
4439	84.76	3553	86.19	3571	98.57	3625	98.57	4284	93.05	4320	98.57	4344	97.24	4468	98.67	4276	90.57	4564	91.81	4066	98.57	Ridge
4463	84.76	4262	87.62	3591	98.57	3538	98.57	4436	93.05	4078	98.57	4470	97.24	4231	98.67	4646	95.71	4363	90.48	4196	98.57	RF
4112	83.43	4405	87.62	3548	98.57	3518	98.57	4290	91.71	4205	98.57	4346	98.67	4334	100	4240	95.90	4233	90.38	4079	98.57	Ridge
4356	84.76	3756	86.19	3919	98.57	3585	98.57	4554	91.62	4290	98.57	4491	97.24	4250	98.67	4289	94.57	4461	91.81	4088	98.57	LightGBM
4436	84.76	4022	86.19	4205	98.57	4194	98.57	4348	93.05	4171	98.57	4263	97.14	4579	100	4396	95.90	4483	90.48	4128	98.57	Ridge
4558	84.76	3743	86.19	4037	98.57	3526	98.57	4451	91.62	4238	98.57	4618	97.24	4140	100	4502	95.90	4578	91.90	4502	100	LightGBM

Table 7. The accuracy values obtained from the Gray Wolf feature selection algorithm were evaluated using ten different classifiers on the SRBCT dataset.

KNN		SVM		Bayes		Ridge		DT		RF		Bagging		LightGBM		Perceptron		LDA		Proposed		
nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	Selected Classifier
1510	87.72	1151	97.57	1337	100	1151	100	1407	93.97	1115	100	1388	100	1335	100	1411	100	1421	72.43	1206	100	Ridge
1427	87.72	1349	97.57	1397	100	1152	100	1358	92.79	1155	100	1359	100	1176	100	1449	100	1447	73.46	1126	100	Bayes
1359	88.97	1132	97.57	1308	100	1130	100	1421	93.97	1131	100	1385	100	1127	100	1382	100	1427	71.25	1159	100	Ridge
1482	90.15	1180	97.57	1321	100	1131	100	1485	92.72	1180	100	1403	98.82	1344	100	1369	100	1505	72.35	1170	100	Ridge
1356	88.90	1146	97.57	1420	100	1186	100	1416	93.97	1152	100	1376	98.82	1340	100	1341	100	1506	72.35	1182	100	Ridge
1363	87.72	1346	97.57	1364	100	1168	100	1481	93.97	1174	100	1215	100	1334	100	1409	100	1506	71.10	1171	100	Ridge
1389	88.90	1155	97.57	1384	100	1159	100	1416	92.72	1155	100	1465	100	1439	100	1412	98.82	1504	72.35	1141	100	Ridge
1450	87.72	1135	97.57	1366	100	1158	100	1352	92.79	1139	100	1152	100	1155	100	1459	100	1382	73.53	1170	100	Ridge
1476	87.79	1155	97.57	1403	100	1160	100	1459	93.97	1160	100	1509	100	1398	100	1367	100	1327	73.46	1153	100	Ridge
1408	87.72	1165	97.57	1398	100	1140	100	1505	92.79	1119	100	1520	100	1147	100	1146	100	1391	72.28	1158	100	Ridge

Table 8. The accuracy values obtained from the Gray Wolf feature selection algorithm were evaluated using ten different classifiers on the Prostate dataset.

KNN		SVM		Bayes		Ridge		DT		RF		Bagging		LightGBM		Perceptron		LDA		Proposed		
nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	Selected Classifier
7497	80.19	6686	67.62	6258	56.67	7554	92.01	7878	87.49	7925	93.41	7640	91.19	7474	94.13	7531	83.23	7841	86.11	7329	92.67	LightGBM
7886	79.47	7324	67.62	7013	56.67	7677	92.72	7875	86.03	7991	92.70	7631	90.73	7361	92.39	7778	86.19	8166	85.34	8002	92.90	Bagging
7644	80.19	6320	67.62	6345	56.67	7599	92.72	8235	87.51	7873	94.18	7631	90.16	7118	93.19	7836	86.83	7689	84.66	7510	91.98	RF
8071	80.19	7552	67.62	6328	56.67	7981	92.72	8163	85.34	7921	92.72	7199	90.72	7243	92.67	7710	81.01	8140	85.37	7837	91.96	RF
8000	80.93	7417	67.62	6393	56.67	7506	92.70	7672	86.75	7476	92.72	7618	90.49	7521	92.21	8022	84.71	6336	86.11	7747	94.13	LightGBM
7928	80.19	7325	67.62	7316	56.67	8146	92.70	7786	87.51	7824	92.72	7475	90.49	7419	92.17	8036	83.23	8061	85.40	7817	92.72	RF
7974	80.93	7411	67.62	7718	56.67	7518	92.72	7812	86.43	7639	94.17	7194	90.17	7589	92.55	7992	83.17	7976	85.37	7850	91.98	RF
7787	80.21	6336	67.62	7390	56.67	8061	92.72	7749	86.18	7782	93.78	7643	90.41	7364	93.73	6398	83.02	7866	86.08	7829	91.16	Bagging
7736	79.44	6317	67.62	6231	56.67	7876	92.72	7718	85.76	7716	93.91	7215	90.32	7008	93.58	8168	83.23	7984	84.66	7946	91.98	LightGBM
8051	80.19	6575	67.62	6240	56.67	7781	92.72	7728	86.74	7549	93.19	7821	90.94	6952	93.91	7738	86.14	7732	85.37	7797	92.65	LightGBM

Table 9. The accuracy values obtained from the Gray Wolf feature selection algorithm were evaluated using ten different classifiers on the Colon dataset.

KNN		SVM		Bayes		Ridge		DT		RF		Bagging		LightGBM		Perceptron		LDA		Proposed		
nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	nFeatures	Accuracy	Selected Classifier
1260	77.56	1003	84.23	1241	70.77	1250	91.92	1269	87.31	1277	87.18	1318	87.31	1285	87.18	1260	87.05	1194	91.92	1168	90.26	LDA
1242	77.44	1027	84.23	1221	70.64	1301	91.92	1215	84.10	1320	85.64	1257	88.85	1198	87.18	1313	86.92	1169	91.92	994	88.72	LDA
1211	77.56	1223	85.77	1268	70.64	1157	90.38	1268	85.64	1241	87.18	1286	88.72	1227	87.18	1312	86.92	1223	91.92	1201	88.72	LDA
1285	78.97	1169	85.77	1251	70.77	1209	91.92	1221	85.64	1283	88.72	1261	90.38	1257	85.64	1243	86.92	1276	91.92	1167	90.26	LDA
1014	77.56	989	84.23	1221	70.64	1284	91.92	1257	85.64	1288	88.59	1288	88.72	1206	86.92	1297	88.59	1272	91.92	1169	88.72	LDA
1237	80.64	1180	85.77	1200	70.64	1183	90.38	1268	85.51	1186	88.07	1133	88.85	1212	85.64	1017	86.92	1195	91.92	1254	90.26	LDA
1254	77.56	1028	84.23	1215	70.77	1312	90.38	1260	87.31	1147	87.92	1252	88.72	1233	87.18	1214	88.59	1277	91.92	1314	91.92	LDA
1254	79.10	1015	84.23	1201	70.64	1007	90.38	1198	85.64	1223	87.49	1255	88.59	1249	87.18	1184	87.05	1199	91.92	1266	87.31	Bagging
1009	77.56	1012	85.77	1178	70.64	1244	91.92	1305	85.64	1279	87.41	1245	87.31	1242	85.64	1226	87.05	1182	91.92	1193	88.72	LDA
1308	78.97	1012	84.23	1172	70.64	1164	91.92	1226	87.18	1257	87.19	1221	88.59	1215	87.18	1186	88.59	1178	91.92	1254	90.38	LDA