



Cancer data classification using binary bat optimization and extreme learning machine with a novel fitness function

Kaveri Chatra¹ · Venkatanareshbabu Kuppili¹ · Damodar Reddy Edla¹ · Ajeet Kumar Verma¹

Received: 31 May 2018 / Accepted: 24 August 2019 / Published online: 12 November 2019
© International Federation for Medical and Biological Engineering 2019

Abstract

Cancer classification is one of the crucial tasks in medical field. The gene expression of cells helps in identifying the cancer. The high dimensionality of gene expression data hinders the classification performance of any machine learning models. Therefore, we propose, in this paper a methodology to classify cancer using gene expression data. We employ a bio-inspired algorithm called binary bat algorithm for feature selection and extreme learning machine for classification purpose. We also propose a novel fitness function for optimizing the feature selection process by binary bat algorithm. Our proposed methodology has been compared with original fitness function that has been found in the literature. The experiments conducted show that the former outperforms the latter.

Keywords Gene · Cancer · DNA

1 Introduction

A gene is a functional unit of a cell, i.e., each gene provides instruction which contributes to the functionality of the cell. Gene expression contains thousands of genes of a cell which determines the functional characteristics of that particular cell in the form of protein products, also known as polypeptides. Gene expression helps us understand the genetic behavior of a cell/ tissue. The rapid growth of research in DNA microarray technology has made researchers possible to assess the expression levels of large number of genes. Studying gene expression can help a lot in cancer diagnosis [9]. Cancer tissues from normal cells can be distinguished by studying the differences in their

gene expressions. From just the morphological appearance of the tumors, identifying cancers has limitations. Therefore researchers make use of gene expression information to identify cancer [19]; also, gene expression profiling identifies the cancer more accurately. Machine learning allows us to build classification models that learns from the data or experience and make decisions [15, 22]. Therefore, using DNA microarray data, cancer cells can be classified. As the number of genes in DNA microarray can be in tens of thousands, the problem of “curse of dimensionality” occurs. Therefore, to address the high-dimensionality problem, feature selection is employed [4, 17]. One of the challenges involved in classification of cancer from DNA microarray data is to select relevant genes which distinguish normal cells from cancer cells. In [22], feature selection methods are systematically studied for classification of cancer data. Feature selector based on correlation and classification using naive Bayes, decision trees, and support vector machines have been studied. Feature selection comprises of choosing a subset of significant features that correctly express a given problem by removing the redundant and correlated features as redundant features can act as noise. Feature selection is mainly of three types—wrapper methods, filter-based methods, and embedded approach. In filter-based feature selection [12], based on some statistical measures, a subset of features is selected that gives maximum predictive power, without requirement of any learning algorithm. In filter methods, a feature subset of

✉ Venkatanareshbabu Kuppili
venkatanaresh@nitgoa.ac.in

Kaveri Chatra
chatra.kaveri@gmail.com

Damodar Reddy Edla
dr.reddy@nitgoa.ac.in

Ajeet Kumar Verma
ajeet@nitgoa.ac.in

¹ Department of Computer Science and Engineering, National Institute of Technology Goa, Ponda, India

cardinality m , denoted by f , selected from a set F is by maximizing a criterion J .

$$f^* = \arg \max_{f \in F} J(f), s.t. |f| = m \quad (1)$$

Wrapper approaches train a model using a subset of features, and based on the performance, features are added or removed from the subset. These methods are reliant on a classifier that is being used and are also time consuming but gives better performance. Embedded methods of feature selection incorporate both wrapper and filter methods. Obtaining a global optimum for Eq. 1 is an NP-hard problem. Nonetheless, a lot of heuristic approaches are being found in the literature which is known to provide suboptimal results. Metaheuristics, especially nature-based computation approaches, are extensively made use in feature subset selection task. They are implemented by beginning with randomly initializing a population in first generation. For every individual in the population, a fitness function or an objective function is evaluated and the solutions are improved iteratively according to the fitness measure. The best solution is obtained at the end of all iterations. These optimization algorithms, when employed for feature selection, yield efficient feature subsets and are used extensively in gene selection for cancer classification [1, 23]. These methods are genetic algorithms [10, 20], particle swarm optimization algorithms [5, 6], ant colony optimization [8], bacterial foraging optimization [21], and bee colony optimization [2]. In these approaches, candidates evolve iteratively evaluating and optimizing an objective function. Since feature selection is an optimization problem with objectives of maximizing the classification accuracies and minimizing the dimension of the feature set, bio-inspired algorithms are employed for feature selection. The rest of the paper is ordered as follows. We discuss about the related work to our proposed methodology in Section 2, Our proposed work is explained in Section 3. Results obtained from the proposed work are in Section 4. We conclude in Section 5.

2 Related work

2.1 Bat algorithm

It is a metaheuristic, bio-inspired algorithm for global optimization problem, which is inspired from echolocation of the microbats, developed by Yang et al. [24]. Echolocation is the way by which microbats hunt for prey in the dark. They are able to tell apart obstacle from prey using echolocation technique. Bats emit a series of loud and short pulses and wait for them to come back. When the pulses that hit an object return, they calculate how far the object is from

the time taken by the pulse to travel to and fro. Some of the preliminaries for the bat algorithm are as follows:

- 1 It is assumed the bats “know” the difference between obstacle and prey.
- 2 Each bat is denoted by b_i fly with velocity v_i , at a position p_i , with a frequency fr_{min} which is fixed, changing wavelength λ , and a loudness L_0 . Based on the closeness of the target object, they can adjust the wavelength and rate of emission of pulse $r \in [0, 1]$
- 3 The loudness is assumed to be varying between L_0 and L_{min} .

The update of position p_i ($p = p^1, \dots, p^n$), and velocity v_i for each bat b_i ($i = 1, \dots, m$) at time step s are given as follows:

$$fr_i = fr_{min} + (fr_{min} - fr_{max}) \quad (2)$$

$$v_i^j(s) = v_i^j(s-1) + [\hat{p}^j - [\hat{p}_i^j(s-1)]] fr_i \quad (3)$$

$$p_i^j = p(s-1) + v_i^j(s) \quad (4)$$

where \hat{p}^j denotes the current global best for the decision variable j . Random walks are done in order to introduce the variability to the solutions. For this, one solution among all the current best solution is selected and random walk is applied to it if the condition $rand > r_i$ is satisfied.

$$p_{new} = p_{old} + \epsilon \bar{L}(s) \quad (5)$$

where \bar{L} is the loudness averaged among the whole bat population and ϵ is in the range $[-1, 1]$ controls the direction of the random walk. The loudness and the pulse rate for each iteration are updated as follows:

$$L_i(s+1) = \alpha L_i(s) \quad (6)$$

$$r_i(s+1) = r_i[1 - \exp(-\gamma s)] \quad (7)$$

where α and γ are constants. The binary version of the bat algorithm can be obtained by using a transfer function:

$$S(v_i^j) = \frac{1}{1 + \exp^{-v_i^j}} \quad (8)$$

Then Eq. 4 can be replaced by

$$p_i^j = \begin{cases} 1, & \text{if } S(v_i^j) > \text{rand.} \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

2.2 Extreme learning machine

Traditionally, all the parameters of the feedforward networks needed tuning which creates the dependency between parameters (weights and biases) of completely different layers.

For past decades, gradient descent algorithm-based methods have been utilized in various feedforward neural network learning. However, it is clear that gradient descent-based learning strategies are generally slow and can lead to improper learning steps or could easily converge to native minima. In order to get higher learning performance, these gradient methods also gives plenty of unvarying learning steps needed in learning algorithm.

Extreme learning machine (ELM) is a learning algorithm for fully connected feedforward neural networks that can be used for tasks like classification, feature learning, regression, compression, clustering, and sparse approximation. In ELM, the weights and bias between hidden nodes and input nodes are not tuned and are randomly initialized once. And the output parameters of the hidden nodes are learned in a single pass. ELMs have good generalization performance and are much faster than back propagation [11]. In Fig. 1, an ELM with a single hidden layer is shown.

Let X be the input, and T be the target. Let W and b be the weights and bias, respectively, between the input layer and the hidden layer. The output of the i th hidden node given by

$$h_i(x) = A(W, b, x) \tag{10}$$

where A is a transfer activation function, such as Fourier, Sigmoid, Gaussian, hardlimit, and so on. The output of the ELM is given by

$$O = \sum_{i=1}^L \beta_i h_i(x) \tag{11}$$

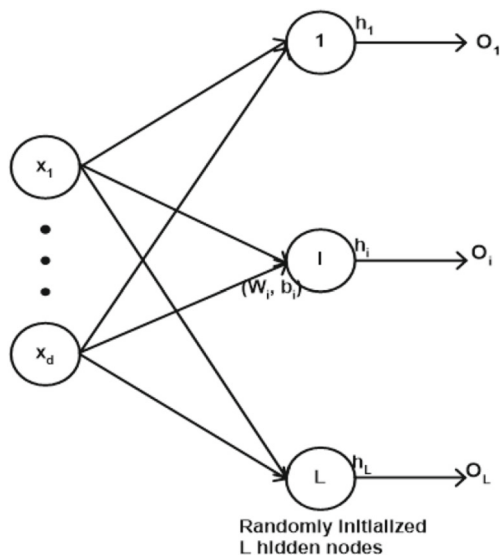


Fig. 1 An ELM with a single hidden layer

The number of neurons present in the hidden layer is denoted by L . If there are N instances, the hidden layer output matrix is given by

$$H = \begin{bmatrix} h_1(x_1) & \dots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \dots & h_L(x_N) \end{bmatrix}, T = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}$$

The algorithm proceeds as

- 1 Assign W with random values.

$$\beta = H^\dagger T \tag{12}$$

- 2 Estimate β by computing pseudo-inverse of the matrix H .
- 3 Compute the output using Eq. 11.

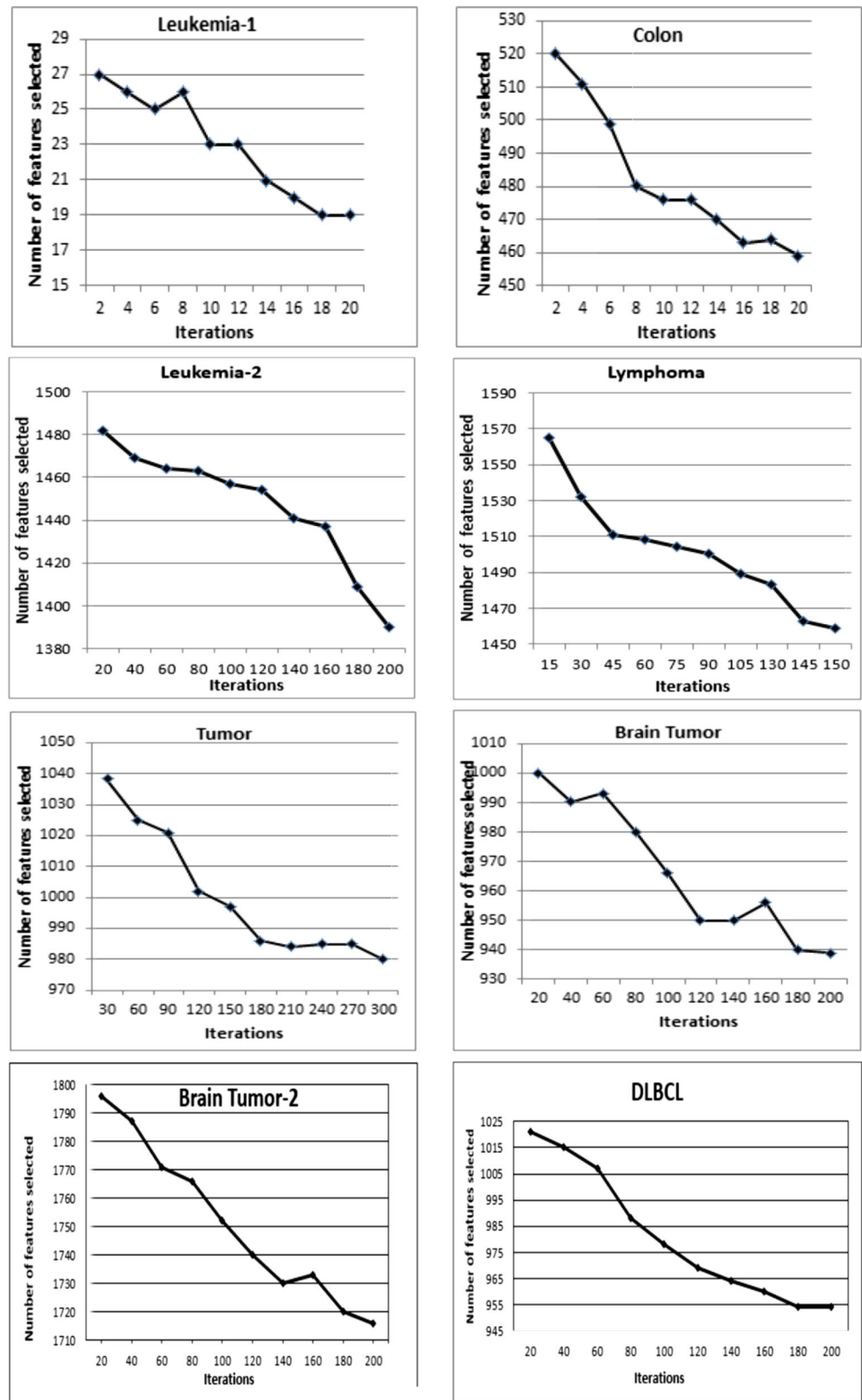
3 Proposed work

One of the challenging areas of research in machine learning is feature selection. The methods existing for feature selection result in suboptimal solutions. Without doing an exhaustive search on features, optimal solutions can't be guaranteed. For datasets with high dimensionality like DNA microarray, exhaustive search is infeasible. In such cases, fairly efficient solution can be obtained to the optimization problem by metaheuristic approaches. In our proposed work, we are employing binary bat algorithm to choose the best combination of features. Number of feature selected in every iteration for each dataset are shown in Fig. 2. The obtained feature subsets from the binary bat algorithm is evaluated at each iteration, using the novel fitness function which is proposed in this work.

3.1 Feature selection by BBA using proposed fitness function

The following is how the feature selection works using binary bat algorithm (BBA). Each bat in the population is initialized randomly with binary array of length equal to the cardinality of feature set in the input data. In an array, zero and one represents absence and presence of the feature, respectively. For each bat, input data is constructed corresponding to the binary array and its fitness is evaluated in each iteration and updated if its value is greater than the previous iteration. Loudness and pulse rates of the bat are modified according to Eqs. 6 and 7 if the solutions are accepted. This process is continued till the user defined highest number of iteration is reached. The best solution

Fig. 2 Graphs showing number of features selected by the binary bat algorithm through the iterations for each dataset



obtained is then evaluated for performance. The proposed fitness function is given as

$$\text{Fitness} = \text{Accuracy} + (1 - \frac{f}{F}) + \sum_{j \in |V|} \frac{\sum_{k=1}^C n_k (\mu_k^j - \mu^j)^2}{(\sigma^j)^2} \tag{13}$$

where, Accuracy is the classification accuracy obtained from the classification algorithm, corresponding to the best feature subset obtained from the binary bat algorithm. F is the length of the original feature set and V is the feature subset obtained from the bat algorithm, $f = |V|$; the total number of classes is denoted by C ; n_k denotes the size of class k ; μ_k^j is the mean of the j th feature of the k th class and $(\sigma^j)^2 = \sum_{k=1}^C n_k (\sigma^j)_k^2$, where $(\sigma^j)_k$ is the standard deviation of k th class. The term $\frac{\sum_{k=1}^C n_k (\mu_k^j - \mu^j)^2}{(\sigma^j)^2}$ computes a score for each feature such that interclass distance of the data points are maximized and the intraclass distance of the data points are minimized. For all the features in the subset, the scores are computed and added together. The binary bat algorithm maximizes the fitness function given in Eq. 13 throughout the iterations. The first term ensures that the accuracy is maximized. The second term ensures that minimum number of features are selected out of a given number of features. Finally, the third term quantifies the relevance of the feature subset.

3.2 Classification

For classification purpose, we employ an ELM classifier. we have used K -fold protocol ($K = 10$) as cross-validation techniques in learning and testing procedure. K -fold means we have to divide the dataset into K parts, $K = 10$ means that the dataset is divided into ten parts that implies 90% data is used in training and 10% is for testing. $K = 10$ or 10-fold: the dataset is divided in 10 parts, and for ten times, a part is considered a test set and the remaining 9 as learning set, and this procedure is repeated for all 10 parts. And the average (or some other combination) classification of the 10 testing sets represents a CM (classification measure) of the entire classifier. This should be one complete iteration. Then, this can be repeated many times: the data set is shuffled and a new complete iteration is performed. The advantages of using an ELM are its good generalization power and fast learning speed. The fast learning speed is a result from the fact that the weights present between hidden and output layers are learned in one pass as opposed to gradient descent with backpropagation. In this work, ELM has been used in binary bat algorithm for computing the accuracy term of the fitness function.

Algorithm 1 Feature selection using BBA.

Result: Feature subset

- 1 Initialize the number of bats in the population.
- 2 Set the maximum number of iterations.
- 3 Initialize the number of variables as equal to the total number of features.
- 4 **for each bat do**
- 5 **for each variable do**
- 6 Randomly initialize each variable to zero or one. (here variable is corresponding feature.)
- 7 **end**
- 8 **end**
- 9 **while maximum iteration is reached do**
- 10 **for all the bats do**
- 11 Obtain the new solutions using the Eqs. 2, 3 and 4.
- 12 **if rand > r_i (the pulse rate) then**
- 13 Select one solution from the current best solutions and apply the random walk as given in the Eq. 5.
- 14 **end**
- 15 Calculate the fitness function according to the formula,
- 16
$$\text{Fitness} = A + (1 - \frac{f}{F}) + \sum_{j \in |V|} \frac{\sum_{k=1}^C n_k (\mu_k^j - \mu^j)^2}{(\sigma^j)^2}$$
- 17 **if rand < A_i and $\text{Fitness}_i > \text{Fitness}_{i-1}$ then**
- 18 Update the solutions and fitness values.
- 19 Increase the pulse rate and decrease the loudness.
- 20 **end**
- 21 **end**
- 22 Obtain the best fitness value and solution which corresponds to best feature subset.

4 Results

In this section, effectiveness of the proposed methodology is analyzed using various performance measurements such as classification accuracy, recall, precision, specificity, and F score on various datasets. The effect of feature selection on the classification accuracy is also analyzed. The nature of the dataset is also discussed which plays an important role in this work.

The gene expression datasets have the following characteristics:

- 1 Fewer data items [7, 14]: The data samples corresponds to the expression levels of genes of tissues of different patients. The classes are different subcategories of cancer. With the typical characteristics of the datasets, small samples and high dimensionality, they can be called as degenerate datasets. The number of samples

Table 1 Description of the datasets used for the experiments

Dataset	Instances	Attributes	Classes	Class–instance (C–I) pair
Leukemia-1	72	50	3	(0–38), (1–9), (2–25)
Colon	62	2000	2	(0–40), (1–22)
Leukemia-2	72	3859	3	(0–28), (1–24), (2–20)
Lymphoma	96	4026	2	(1–42), (2–54)
9_Tumors	61	6167	9	(0–9), (1–7), (2–8), (3–6), (4–6), (5–8), (6–8), (7–2), (8–6)
Brain_Tumor1	90	5920	5	(0–60), (1–10), (2–9), (3–4), (4–6)
Brain_Tumor2	50	10368	4	(0–14), (1–7), (2–14), (3–15)
DLBCL	77	5469	2	(0–58), (1–49)

generally available are less than 100. Therefore it hinders the generalizing capability of the models.

- Class imbalance [14]: The proportion of the classes are not equal. One class can dominate other(s). The minor classes can get misclassified.

The datasets that are used for experiments are publicly available and obtained from the gene expression model analyzer (gems-system) and KanGAL portal of IIT-Kanpur, (<https://www.iitk.ac.in/kangal/bioinfo.shtml>). For the experimentation purpose, we have used Intel core i3-3240, 3.40 GHz processor. OS of 64 bit and memory of 3.8 GB. Experiments are performed in MATLAB.

Table 1 presents the description of the dataset used for evaluation of our methodology. We have used eight gene expression datasets, whose description about number of features, instances, and classes with individual class with corresponding instances is given. The dataset Leukemia-1 has 50 genes and 72 samples, dataset colon has 2000 genes

and 62 samples and so on. Also the number of instances in each class of eight datasets is given by class–instance (C–I) pair, e.g., in the given table, C–I pair (0–38) represents class 0 have 38 instances. It can be observed that the number of attributes is way more than the number of instances, which indicates the need of feature selection.

The next table, i.e., Table 2 presents the name of the datasets, number of attributes before and after the feature selection, and also percentage of feature selected. The lowest percentage of feature selected is 15.89 for 9_Tumors dataset and the highest is 39.07 for Leukemia-2 dataset. The average percentage of features selected is 25.80.

The comparative performance evaluation of the proposed method with novel fitness function and the existing fitness function are shown in Table 3. ELM classifier with a sigmoid activation function has been used. The sigmoid activation function is found to be performing better than other activation functions such as sine, hard limit, and radial basis. It can be observed that the proposed method's results are better than the existing methods with regard to evaluation measures such as classification accuracy, precision, recall, specificity, and F score. Here, the existing method refers to the fitness function which comprises of just the classification accuracy and the dimensionality of the feature subset selected, which is found in literature. The proposed methodology obtained highest accuracies for all the datasets when compared with the method with existing fitness function. The classification accuracies for Leukemia-1, Leukemia-2, 9_Tumors, Brain_Tumor1, Brain_Tumor2 and DLBCL dataset are 100% and for Colon and Lymphoma are 99.29 and 89.50, respectively.

Accuracy, precision, recall, specificity, and F score are computed using the following formulas:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \quad (14)$$

Table 2 Number and percentage of features selected by BBA

Dataset	Attributes before feature selection	Attributes after feature selection	Percentage selected (%)
Leukemia-1	50	19	38
Colon	2000	546	27.3
Leukemia-2	3859	1508	39.07
Lymphoma	4026	1459	36.23
9_Tumors	6167	980	15.89
Brain_Tumor1	5920	964	16
Brain_Tumor2	10368	1716	16.5
DLBCL	5469	954	17.44
Average			25.80

Table 3 Performance evaluation of the proposed and the existing method

Dataset	Accuracy		Precision		Recall		Specificity		F score	
	Proposed	Existing*	Proposed	Existing*	Proposed	Existing*	Proposed	Existing*	Proposed	Existing*
Leukemia-1	100	87.5	100	100	100	83.33	100	100	100	90.01
Colon	99.29	86.43	99	84.17	100	52.50	98.33	100	99.50	91.40
Leukemia-2	100	74.38	100	84.64	100	77.30	100	74.58	100	80.80
Lymphoma	89.50	71.50	76.25	68.33	98.00	52.50	86.43	84.17	85.77	59.38
9_Tumors	98.36	76.67	100	59.37	100	65.48	100	96.92	100	62.27
Brain_Tumor1	100	96.67	100	98.95	100	93.75	100	96.42	100	96.27
Brain_Tumor2	100	88.89	100	93.75	100	93.75	100	95.83	100	93.75
DLBCL	100	98.07	100	95.45	100	98.81	100	98.98	100	97.10

*Existing method refers to the paper presented by Parvithara et al. [17]. Authors used an existing framework of ELM learning

Table 4 Effect of feature selection on classification accuracies

Dataset	Accuracy before feature selection (%)	Accuracy after feature selection (%)
Leukemia-1	94.44	100
Colon	77.42	99.29
Leukemia-2	97.22	100
Lymphoma	87.50	89.50
9_Tumors	64.52	100
Brain_Tumor1	80	100
Brain_Tumor2	60	100
DLBCL	93.75	100

$$\text{Precision} = \frac{tp}{tp + tf} \tag{15}$$

$$\text{Recall} = \frac{tp}{tp + fn} \tag{16}$$

$$\text{Specificity} = \frac{tn}{tn + fp} \tag{17}$$

$$F \text{ score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

where, tp denotes true positive, i.e., number of observations that are actually positive and also predicted positive.

tn denotes true negative, i.e. number of observations that are actually negative and also predicted negative.

fp denotes false positive, i.e., number of observations that are actually negative but are predicted positive.

fn denotes false negative, i.e., number of observations that are actually positive but are predicted negative.

Table 4 shows the effect of feature selection on the classification accuracies for all the datasets that are used. From the table, we can notice that there is a significant change in four datasets, they are Colon, 9_Tumors, Brain_Tumor1, and Brain_Tumor2. For the remaining datasets, the difference in accuracies is less than 6%. It can be realized that feature selection is necessary for such high-dimensional datasets like microarray gene expression datasets. Some authors highlighted few critical issues about cross-validation error estimates for small-sample microarray classification [25, 26]. Small samples of one class may also be useful and enough for a classifier to learn.

Table 5 presents the comparative performance of different methods of gene expression data classification and our proposed method. Performance of the proposed methodology is slightly less in terms of accuracy for Colon and Lymphoma datasets. For the rest of the datasets, the classification accuracies of the proposed are greater than or equal to the accuracies of the methods being compared.

Table 5 Comparative testing accuracies of BBA-ELM with other methods

Methods	Accuracies (%)*
Leukemia-1 Dataset	
Proposed	100
Wang et al. [21]	100
Lv et al. [16]	100
Chuang et al. [5]	100
Colon Dataset	
Proposed	99.29
Tan et al. [20]	100
Hasnat et al. [10]	82.3
Banka et al. [3]	83.9
Pavithra et al. [17]	88.7
Leukemia_2 Dataset	
Proposed	100
Chuang et al. [5]	100
Lymphoma Dataset	
Proposed	89.50
Andaru et al. [2]	87.9
Hasnat et al. [10]	88
Banka et al. [3]	95.8
9_Tumors Dataset	
Proposed	98.36
Chuang et al. [5]	78.33
Wang et al. [21]	92.22
Brain_Tumor1 Dataset	
Proposed	100
Chuang et al. [5]	94.44
Wang et al. [21]	96.3
Brain_Tumor2 Dataset	
Proposed	100
Chuang et al. [5]	94.0
Wang et al. [21]	100
DLBCL dataset	
Proposed	100
Chuang et al. [5]	100
Wang et al. [21]	100
Liang et al. [13]	94.8
Sahu et al. [18]	100

*Testing accuracy

5 Conclusions

This paper presents a methodology to classify cancer using gene expression data. First, we perform a feature extraction task to overcome the high-dimensionality problem. We perform a feature selection task, for which we make use

of a bio-inspired algorithm called binary bat algorithm with a novel fitness function that is proposed in this paper. The proposed fitness function involves minimizing the intraclass and maximizing the interclass distances of the data points, along with maximizing the accuracy and minimizing the dimension of the data. For the classification task, we make use of the extreme learning machine which is found to be fast and has good generalization capability. We have conducted the experiments on eight gene expression datasets. We have compared our methodology with the proposed fitness function and the existing fitness function that is used mostly in the literature for feature selection. It has been observed that our proposed method performs better than the original method with regard to the classification accuracy, precision, recall, specificity, and F score.

References

1. Abdullah AS, Ramya C, Priyadharsini V, Reshma C, Selvakumar S (2017) A survey on evolutionary techniques for feature selection. In: 2017 conference on emerging devices and smart systems (ICEDSS). IEEE, pp 58–62
2. Andaru W, Syarif I, Barakbah AR (2017) Feature selection software development using artificial bee colony on dna microarray data. In: 2017 international electronics symposium on knowledge creation and intelligent computing (IES-KCIC). IEEE, pp 6–11
3. Banka H, Dara S (2012) Feature selection and classification for gene expression data using evolutionary computation. In: 2012 23rd international workshop on database and expert systems applications (DEXA). IEEE, pp 185–189
4. Chao S, Lihui C (2004) High dimensional gene expression data dimension reduction. In: 2004 IEEE conference on cybernetics and intelligent systems. IEEE, vol 1, pp 451–455
5. Chuang LY, Chang HW, Tu CJ, Yang CH (2008) Improved binary PSO for feature selection using gene expression data. *Comput Biol Chem* 32(1):29–38
6. Dara S, Banka H (2014) A binary PSO feature selection algorithm for gene expression data. In: 2014 international conference on advances in communication and computing technologies (ICACACT). IEEE, pp 1–6
7. Dougherty ER (2001) Small sample issues for microarray-based classification. *Compar Funct Genom* 2(1):28–34
8. Fahrudin TM, Syarif I, Barakbah AR (2016) Ant colony algorithm for feature selection on microarray datasets. In: 2016 international electronics symposium (IES). IEEE, pp 351–356
9. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA et al (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286(5439):531–537
10. Hasnat A, Molla AU (2016) Feature selection in cancer microarray data using multi-objective genetic algorithm combined with correlation coefficient. In: International conference on emerging technological trends (ICETT). IEEE, pp 1–6
11. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004.

- proceedings. 2004 IEEE international joint conference on neural networks. IEEE, vol 2, pp 985–990
12. Lazar C, Taminau J, Meganck S, Steenhoff D, Coletta A, Molter C, de Schaetzen V, Duque R, Bersini H, Nowe A (2012) A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Trans Comput Biol Bioinform (TCBB)* 9(4):1106–1119
 13. Liang Y, Liu C, Luan XZ, Leung KS, Chan TM, Xu ZB, Zhang H (2013) Sparse logistic regression with a $l_{1/2}$ penalty for gene selection in cancer classification. *BMC Bioinform* 14(1):198
 14. Lorena AC, Costa IG, de Souto MC (2008) On the complexity of gene expression classification data sets. In: 2008. HIS'08. Eighth international conference on hybrid intelligent systems. IEEE, pp 825–830
 15. Lu Y, Han J (2003) Cancer classification using gene expression data. *Inf Syst* 28(4):243–268
 16. Lv J, Peng Q, Chen X, Sun Z (2016) A multi-objective heuristic algorithm for gene expression microarray data classification. *Expert Syst Appl* 59:13–19
 17. Pavithra D, Lakshmanan B (2017) Feature selection and classification in gene expression cancer data. In: 2017 international conference on computational intelligence in data science (ICCIDS). IEEE, pp 1–6
 18. Sahu B, Mishra D (2012) A novel feature selection algorithm using particle swarm optimization for cancer microarray data. *Procedia Eng* 38:27–31
 19. Sserwadda A, Saraç ÖS (2017) Gene selection and classification of pancreatic microarray datasets. In: Signal processing and communications applications conference (SIU), 2017 25th. IEEE, pp 1–4
 20. Tan F, Fu X, Zhang Y, Bourgeois AG (2006) Improving feature subset selection using a genetic algorithm for microarray gene expression data. In: 2006. CEC 2006. IEEE congress on evolutionary computation. IEEE, pp 2529–2534
 21. Wang H, Jing X, Niu B (2017) A discrete bacterial algorithm for feature selection in classification of microarray gene expression cancer data. *Knowl-Based Syst* 126:8–19
 22. Wang Y, Tetko IV, Hall MA, Frank E, Facius A, Mayer KF, Mewes HW (2005) Gene selection from microarray data for cancer classification—a machine learning approach. *Comput Biol Chem* 29(1):37–46
 23. Xue B, Zhang M, Browne WN, Yao X (2016) A survey on evolutionary computation approaches to feature selection. *IEEE Trans Evol Comput* 20(4):606–626
 24. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, pp 65–74
 25. Braga-Neto UM, Dougherty ER (2004) Is cross-validation valid for small-sample microarray classification? *Bioinformatics* 20(3):374–380
 26. Isaksson A, Wallman M, Göransson H, Gustafsson MG (2008) Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recogn Lett* 29(14):1960–1965

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Kaveri Chatra completed M. Tech. degree with the Department of Computer Science and Engineering, National Institute of Technology Goa, India. She is a recipient of MHRD Gate Scholarship for the post-graduation program. She has been working with several sponsored research projects.



Venkatanaresbhabu Kuppili received the Ph.D. degree from IIT-Delhi. He is a senior member of IEEE and currently with the Machine Learning Group, Department of Computer Science and Engineering, National Institute of Technology (NIT) Goa, Farmagudi, India, as an Assistant Professor. He was with Evaluate-serve Pvt. Ltd, as a Senior Research Associate. He is also actively involved in teaching and research development for the Graduate Program in the

Department of Computer Science and Engineering, NIT Goa. He has authored a number of research papers published in reputed international journals.



Damodar Reddy Edla received the M.Sc. Degree from the University of Hyderabad in 2006, and the M.Tech. and Ph.D. degrees in CSE from the Indian School of Mines Dhanbad, in 2009 and 2013, respectively. He is currently an Assistant Professor and the Head of the Computer Science and Engineering Department, National Institute of Technology Goa. He has published more than 20 research papers articles in reputed journal and International conferences. He is a Senior Member of the IACSIT. His research interest are in data mining and wireless sensor networks.



Ajeet Kumar Verma received M. Tech. degree from BBAU, Lucknow. He is currently with the Department of Computer Science and Engineering, National Institute of Technology Goa, India as Junior Research Fellow. He was with CMC Limited, as FMS engineer project at IIM-Lucknow. Research interests include Machine Learning, Artificial Neural Networks, Spiking Neural Networks and soft computing.