Skip to main content

Advertisement

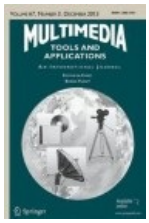
404 Not Found

nginx/1.18.0 (Ubuntu)

Search

Go to cart

- Log in



Multimedia Tools and Applications

An International Journal

Volume 81, issue 25, October 2022

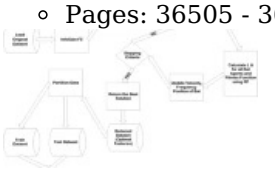
66 articles in this issue

1. **Feature selection and computational optimization in high-dimensional microarray cancer datasets via InfoGain-modified bat algorithm**

   Authors (first, second and last of 5)

   - Moshood A. Hambali
   - Tinuke O. Oladele
   - Wei Gao
   - Content type: 1213: Computational Optimization and Applications for Heterogeneous Multimedia Data

- Published: 11 August 2022
- Pages: 36505 - 36549



2. **Study of cross-domain person re-identification based on DCGAN**

   Authors (first, second and last of 4)

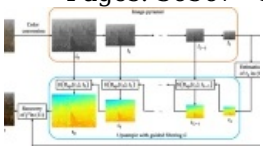
   - Wei Fang
   - Weinan Yi
   - Victor S. Sheng
   - Content type: 1213: Computational Optimization and Applications for Heterogeneous Multimedia Data
   - Published: 11 August 2022
   - Pages: 36551 - 36565



3. **Real-time image and video dehazing based on multiscale guided filtering**

   Authors

   - Thuong Van Nguyen
   - An Gia Vien
   - Chul Lee
   - Content type: 1213: Computational Optimization and Applications for Heterogeneous Multimedia Data
   - Published: 13 August 2022
   - Pages: 36567 - 36584



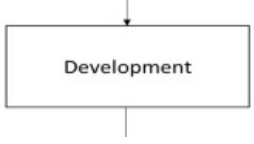
4. **Multimedia security based on quantum cryptography and blockchain**

   - Content type: Editorial Note
   - Published: 16 September 2022
   - Pages: 36585 - 36585

5. **Blockchain for record-keeping and data verifying: proof of concept**

   Authors (first, second and last of 7)

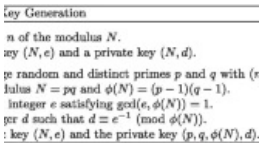
   - Razatulshima Ghazali
   - Faizura Haneem Mohamed Ali
   - Ali Ahmadian
   - Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain
   - Published: 11 August 2021
   - Pages: 36587 - 36605



6. **Exponential increment of RSA attack range via lattice based cryptanalysis**

   Authors (first, second and last of 5)

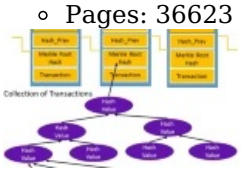
   - Abderahmanne Nitaj
   - Muhammad Rezal Kamel Ariffin
   - Ali Ahmadian
   - Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain
   - Published: 15 August 2021
   - Pages: 36607 - 36622



7. **[Blockchain-based synergistic solution to current cybersecurity frameworks]**

   Authors (first, second and last of 4)

   - Sumit Kumar Yadav
   - Kavita Sharma
   - Arushi Arora
   - Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain
   - Published: 16 September 2021
   - Pages: 36623 - 36644



8. **[Multiagent-based secure energy management for multimedia grid communication using Q-learning]**

   Authors

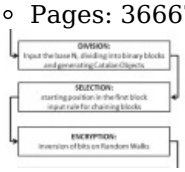
   - Aparna Kumari
   - Sudeep Tanwar
   - Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain
   - Published: 16 September 2021
   - Pages: 36645 - 36665



9. **[A novel block encryption method based on Catalan random walks]**

   Authors

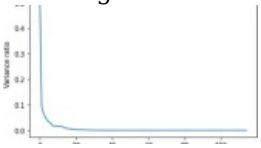
   - Muzafer Saračević
   - Sudhir Kumar Sharma
   - Khaleel Ahmad
   - Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain
   - Published: 17 September 2021
   - Pages: 36667 - 36684



10. **[Using Machine Learning to Quantify the Multimedia Risk Due to Fuzzing]**

    Authors (first, second and last of 4)

    - Gautam Siddharth Kashyap
    - Karan Malik
    - Rijwan Khan
    - Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain
    - Published: 06 October 2021
    - Pages: 36685 - 36698



11. **[Design of high-speed blockchain-based sidechaining peer to peer communication protocol over 5G networks]**

Authors

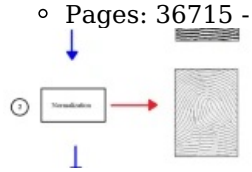
- Nisha Balani
- Pallavi Chavan
- Mangesh Ghonghe
- Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain
- Published: 12 July 2022
- Pages: 36699 - 36713



12. **New approach for fingerprint recognition based on stylometric features with blockchain and cancellable biometric aspects**

Authors (first, second and last of 5)

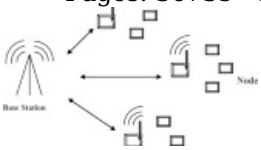
- Ali Elsadai
- Saša Adamović
- Sudhir Kumar Sharma
- Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain
- Published: 12 July 2022
- Pages: 36715 - 36733



13. **An efficient key distribution Scheme for WSN with Mutual Healing Capability**

Authors

- Neeraj Patel
- Vinod Kumar
- Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain
- Published: 02 August 2022
- Pages: 36735 - 36749



14. **A novel blockchain federated safety-as-a-service scheme for industrial IoT using machine learning**

Authors

- Nabeela Hasan
- Kiran Chaudhary
- Mansaf Alam
- Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain
- Published: 02 August 2022
- Pages: 36751 - 36780

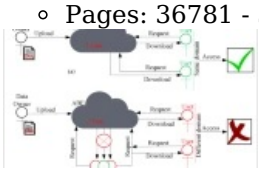


15. **QuSigS: A quantum Signcryption scheme to recover key escrow problem and key revocation problem in cloud computing**

Authors

- Shamsher Ullah
- Muhammad Tanveer Hussain
- Mahwish Yousaf
- Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain

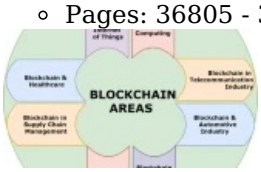
- Published: 13 August 2022
- Pages: 36781 - 36803



16. **Latent DIRICHLET allocation (LDA) based information modelling on BLOCKCHAIN technology: a review of trends and research patterns used in integration**

    Authors

    - Chetan Sharma
    - Shamneesh Sharma
    - Sakshi
    - Content type: 1219: Multimedia Security Based on Quantum Cryptography and Blockchain
    - Published: 20 August 2022
    - Pages: 36805 - 36831



←

- Previous page
- 1
- 2

→

- Next page

For authors

Submission guidelines
Manuscript editing services
Ethics & disclosures
Open Access fees and funding
Contact the journal
Submit manuscript

Working on a manuscript?

Avoid the most common mistakes and prepare your manuscript for journal editors.

Learn more

Explore

Online first articles
Volumes and issues
Collections
Sign up for alerts

Advertisement

404 Not Found

---

nginx/1.18.0 (Ubuntu)

Over 10 million scientific documents at your fingertips

Switch Edition

- Academic Edition
- Corporate Edition

- Home
- Impressum
- Legal information
- Privacy statement
- California Privacy Statement
- How we use cookies
- Manage cookies/Do not sell my data
- Accessibility
- FAQ
- Contact us
- Affiliate program

Not logged in - 102.89.41.221

Not affiliated

Springer Nature

© 2023 Springer Nature Switzerland AG. Part of Springer Nature.

Multimedia Tools and Applications
https://doi.org/10.1007/s11042-022-13532-5

1213: COMPUTATIONAL OPTIMIZATION AND APPLICATIONS FOR HETEROGENEOUS MULTIMEDIA DATA



Feature selection and computational optimization in high-dimensional microarray cancer datasets via InfoGain-modified bat algorithm

Moshood A. Hambali[1] · Tinuke O. Oladele[2] · Kayode S. Adewole[2] · Arun Kumar Sangaiah[3] · Wei Gao[4]

Received: 5 February 2021 / Revised: 23 August 2021 / Accepted: 13 July 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Achieving a satisfactory cancer classification accuracy with the complete set of genes remains a great challenge, due to the high dimensions, small sample size, and presence of noise in gene expression data. Feature reduction is critical and sensitive in the classification task, most importantly in heterogeneous multimedia data. One of the major drawbacks in cancer study is recognizing informative genes from thousands of available genes in microarray data. Traditional feature selection algorithms have failed to scale on large space data like microarray data. Therefore, an effective feature selection algorithm is required to explore the most significant subset of genes by removing non-predictive genes from the dataset without compromising the accuracy of the classification algorithm. The study proposed an information Gain – Modified Bat Algorithm (InfoGain-MBA) features

✉ Arun Kumar Sangaiah
aksangaiah@ieee.org

Moshood A. Hambali
hambali@fuwukari.edu.ng

Tinuke O. Oladele
oladele.to@unilorin.edu.ng

Kayode S. Adewole
adewole.ks@unilorin.edu.ng

Wei Gao
gaowei@ynnu.edu.cn

[1] Department of Computer Science, Federal University Wukari, Wukari, Taraba State, Nigeria

[2] Department of Computer Science, University of Ilorin, Ilorin, Kwara State, Nigeria

[3] National Yunlin University of Science and Technology, Douliu 633102, Taiwan

[4] School of information Science and Technology, Yunnan Normal University, Kunming, China

Published online: 11 August 2022

selection model for selecting relevant and informative features from high dimensional Microarray cancer datasets and evaluate the approach with four classifiers - C4.5, Decision Tree, Random Forest and classification and regression tree (CART). The results obtained show that the proposed approach is promising for the classification of microarray cancer data. The random forest has 100% accuracy with few genes in all seven datasets used. Further investigations were also conducted to determine the optimal threshold for each of the datasets.

Keywords Feature selection · Binary bat algorithm · Information gain · Cancer classification · Microarray data · Random forest · Computational optimization

1 Introduction

In the last few decades, diagnosis and detection of cancer diseases were based on a thorough inspection of the organs where the tumour was developed. This sometimes may cause physical exhaustion and histopathological assessments of the tissues that harbour cancer. However, feasible diagnoses can only be carried out either by laboratory tests that are too expensive to bear or by surgical operations which risk the life of the patients [72]. In most situations, test results such as autopsy can only be available after certain periods. This result in delaying timely detection or diagnosis of cancer. With the emergence of Deoxyribonucleic Acid (DNA) Microarray technology in the last few years has presented a drastically changes in cancer research. With this recent innovation, analysis of several thousands of gene expressions can be carried out simultaneously at once and relate their pattern of expressions to clinical phenotypes [30, 37]. With this study, it is easy to identify cell patterns of expressions that can distinguish between infected (cancer) cells and uninfected (normal) cells. Therefore, it eases the task of predicting cancer without patients experience stress related to the examination of the organs or surgical and minimize the cost of laboratory tests.

Microarray data are commonly used in cancer researches nowadays where an early cancer diagnosis is required in determining the nature of treatment and its survivability [65]. Microarray Technology helps biologists in investigating the activity of thousands of genes in a single experiment and acquires vital information about cell's functionality. This explicit information can be employed for the detection of numerous diseases for instance Alzheimer's disease [52], diabetes diseases and cancer diseases [3, 29, 30, 32]. The gene expression data can be created from this technology, which is very germane for cancer prediction and classification. Nevertheless, gene expression data were characterized by high dimensionality data which are irrelevant, redundant, and noisy genes that are of no importance to the diagnosis of diseases. Huge amount of genes (features) versus few sample size, and the presence of redundancy in data expression are the major reasons that usually result in poor diagnosis and classification, in machine learning and data mining approaches employed in the field of medicine [5, 72]. In recent times, researchers had employed computational intelligence algorithms to investigate the most informative genes that contribute to cancer diagnosis [3].

Numerous literature has revealed that some of the genes present in DNA Microarray datasets are not pertinent to the correct diagnosis of different diseases [3, 30, 61, 66]. To circumvent the hitches of the 'curse of dimensionality', feature selection, which is sometimes called gene selection, is the technique employed to seek the most informative genes that can augment diagnosis and predictive accuracy of diseases [3, 29, 30]. Dimensionality reduction

can be accomplished in many ways. But, the choice of selecting the appropriate dimensionality reduction method is determined by the domain application and the peculiarity of the dataset. Feature selection techniques are categorized majorly into four (4) approaches, namely; filter, wrapper, embedded and hybrid approaches [72]. The selection of features in filter algorithms is done by using individual characteristic features. Wrapper approaches employ a certain machine learning algorithm or evolutionary approaches to select subset features. The filter methods are characterized by very fast computation and low accuracy while wrapper approaches tend to have better accuracy performance with less computation rate. In a domain with large datasets, filter-based methods have been established to be the best choice than wrapper methods due to their speed. A mutual drawback of the two methods is that both approaches disregard interaction between the classifier and the dependence among features. This may result in diverse classification accuracy when features selected are subjected to different classification algorithms. Embedded approaches, on the other hand, make use of specific learning algorithms. Embedded methods have the superiority advantage of interacting with the classification algorithm; while at the same time have lower computational costs compared to wrapper methods [72].

In data mining, machine learning, medical field and pattern recognition, gene (feature) selection has become a prominent research topic [67]. Feature selection is regarded as a non-deterministic polynomial (NP) hard problem since the growth of search space increase exponentially with the number of genes (2^N) present, where N signifies the number of features [67]. Furthermore, there are challenges in exploring the optimal solution from a vast number of possible solutions, because searching for the entire solution space requires a great computational accomplishment that entails an extreme quantity of computational time and cost [11]. Hence, this may not guarantee that the best subset of genes will be achieved. Moreover, due to the huge number of locally optimal solutions in a search space, classical methods of genes selection may not yield the best solution [35]. Therefore, Meta-heuristic techniques are the apt method of finding the optimal solution from the vast search space to achieve a noble solution without traversing the entire space of solutions. Several Meta-heuristic methods have been widely employed to deal with gene selection problems, and they have a record of better performance and effectiveness in addressing the problems [33, 67]. However, the Microarray data problem is not limited to the dimensionality issue alone but is also associated with redundancy and noise. Hence, a proficient gene selection algorithm is required to explore the best significant subset of genes by removing redundant or non-predictive genes from the dataset without compromising the accuracy of the learning algorithm.

Zhang et al. [9] performed feature selection using binary Bare Bones Particle Swarm Optimization algorithm (binary BPSO). Kabir et al. [33] employed a hybrid of Ant Colony Algorithm (ACO) for feature selection. Their approach not only emphasizes the selection of features alone at the same time attained a significant reduction in the number of features. Vieira et al. [73] designed some mechanisms for the BPSO to be able to deal with the problem of premature convergence and then proposed a Modified BPSO (MBPSO) algorithm for the feature selection. Forsati et al. [20] presented a new variant of ACO for feature selection and named it enriched Ant Colony Optimization (RACO). Also, Osama et al. [3] used a hybrid of Bat Algorithm and Minimum Redundancy Maximum Relevancy (MRMR) for feature selection.

In this study, InfoGain-MBA is proposed for selecting informative features from high dimensional Microarray cancer datasets to remove redundant and non-predictive genes from the Microarray cancer datasets and at the same time solve the problem of class imbalance in

the datasets without eliminating important genes that help in improving accuracy performance. Modification of BBA is done using Random forest as the fitness function, to make the algorithm robust for the selection of efficient genes in multiclass microarray data. Also, the proposed approach established the optimal threshold for all the datasets used.

The remaining part of this paper is organized as follows: Section 2 briefly reviews related approaches in the literature; followed by Section 3, which focuses on methods and materials used in this study. The section gives a brief description of performance metrics used, while section 5 presents the results and discussion. Finally, Section 6 concludes the paper.

2 Related works

Nature is a combination of inspiration and motivation. Researchers and computer scientists are motivated by nature and are trying to proffer solutions to numerous problems by observing nature. Nature-inspired algorithms proposed for providing solutions to multifaceted and hefty combinatorial problems such as Ant Colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Simulated Annealing (SA) are at the leading research areas in feature subset selection [21, 40, 45, 53]. These approaches proficiently eliminate feature redundancy and interaction and without involving the obstructive monotonicity assumption. Though, these approaches are computationally costly. Besides Artificial Neural Network (ANN), many Bio-inspired algorithms such as ACO, Bee Colony Optimization (BCO), Particle Swarm Optimization (PSO) and many others algorithms were designed mainly to proffer solutions to optimization problems. However, they are now extended to solve diverse problems. Bat algorithm is one of the Bio-inspired algorithms that has been employed in diverse applications [78]. Gandomi et al. [22] proposed a simple Bat algorithm to solve constrained based optimization problems. A new hybrid algorithm was built for global numerical optimization by using the Bat algorithm and Harmonic Search (HS) [74]. Bat algorithm was also used in solving optimization problems in the field of engineering [77]. Bat algorithm was employed to select features from numerous digital data sets for the first time in the literature [47]. Rodrigues et al., [60] presented a combination of the Optimum-path forest technique and Bat algorithm to select the features using a wrapper approach. Steadily, from the process of offering solutions to the optimization problem and feature selection, the focus of research shifts towards classification and clustering problems. Meta-heuristic Bat algorithm was used to classify Microarray data in [43]. Bat algorithm was also employed for clustering by combining traditional K-means clustering with a simple Bat algorithm [71].

In recent times, many researchers proposed hybrid approaches taping the benefits of both filter and wrapper feature selection techniques. Some common cases of hybrid algorithms found in the literature include correlation-based feature selection algorithm and genetic algorithm [49], t-statistics and GA [44], Principal Component Analysis (PCA) and ACO algorithm [50], mutual information and GA [48], Chi-Square approach and multi-objective optimization algorithm [50].

Osama et al. [3] presented a hybrid gene selection of filter technique - MRMR and a wrapper technique - BA using Microarray dataset. MRMR was employed to discover the most significant genes from the entire gene in the Microarray dataset, and the Bat algorithm was used to search for the most relevant feature subset from the reduced set produced by MRMR that aided in detecting cancers. The classifier approach used was a Support Vector Machine (SVM) with 10-fold cross-validation which helped in evaluating the Bat algorithm. To reveal

the performance accuracy of the proposed method, three different Microarray datasets were considered, which include: Breast, Colon and Ovarian cancer datasets. The same process was carried out with GA to perform the comparison with MRMR- Bat algorithm and the outcomes of their experiment revealed that their proposed method was able of finding the small size of gene subset with a high classification rate. An average of 18.6 genes from Breast data yielded 88.6% accuracy, 3.83 genes from Ovarian data yielded 100% accuracy while 8.13 genes from Colon data yielded 93.12% accuracy.

Doreswamy & UmmeSalma [16] designed a Binary Bat Algorithm (BBA) based on Feedforward Neural Network (FNN) hybrid model. The authors exploited the benefits of BBA and the efficiency of FNN to classify three standard breast cancer datasets into normal and diseased cases. In this work, BBA was utilized to produce a V-shaped hyperbolic tangent function for training the network and a fitness function was employed for error minimization. FNNBBA based approach yielded 92.61% accuracy rate for the training set and 89.95% for the testing set.

Alshamlan et al. [4] presented a hybrid feature selection technique called Genetic Bee Colony (GBC) algorithm, where a hybrid of both Genetic Algorithm and Artificial Bee Colony (ABC) algorithms were used. The GBC algorithm demonstrates the sign of having a better performance as it attains very high classifier accuracy with the small size number of genes. Their proposed method yielded 98.38% classification accuracy with 10 genes from Colon data, 100% accuracy rate with 4 genes from Leukemia1, 100% accuracy rate with 4 genes from Lung data, 100% classification accuracy with 6 genes from SBRCT data and 100% classification accuracy with 4 genes from Lymphoma data.

Shreem et al. [67] applied a new approach of hybridization that combines Harmony Search Algorithm (HSA) and the Markov Blanket (MB), termed HSA-MB for feature selection in classification tasks. HSA employs Naive Baye's classifier in its wrapper approach. Ten different Microarray datasets were employed to perform the experiments. The HSA-MB performance showed that the proposed approach yield 99.34% classification accuracy with 5 genes from ALL-AML data, 99.18% classification accuracy with 5.84 genes from ALL-AML-3C data, 84.17% classification accuracy with 7.43 genes from CNS data and 99.55% classification accuracy with 6.60 genes from MLL data.

Abeer & Basma [38] proposed a new fusion machine learning (ML) reduction method to improve the cancer diagnosis accuracy of Microarray data using two different gene ranking methods: t-test and Class Separability (CS). They combined the genes ranking methods with two ML classifiers – K-Nearest Neighbor (KNN) and SVM; to evaluate Microarray datasets. Four public cancer datasets - Lymphoma, Leukemia, SRBCT and Lung, were utilized to measure the proposed approaches, and they succeeded in accomplishing their aim. The combination of t-test and SVM was reported to have produced higher classification accuracy compared to other combination approaches of t-test and KNN, CS and SVM, and CS and KNN, where they reported classifier accuracy of 100% using peak ranked value of 2 and 3 genes from Lymphoma and SRBCT datasets respectively. They also obtained 94.12% from Leukemia using 6 genes as peak ranked value. Their results revealed that the proposed technique yielded favourable results by selecting a fewer number of genes complemented with the better accuracy of the classifiers.

Abeer et al. [1] developed Differentially Expressed Genes (DEGs) in Microarray data to build a reliable and robust classifier. A t-test was used for gene selection and KNN for the machine learning classifier. The dataset used was the Lymphoma dataset to achieve the DEGs and to analyse the influence of these on genes learning accuracy performance respectively.

Sahu & Mishra [62] presented a novel hybrid feature selection method for high dimensional cancer Microarray dataset before performing classification. The authors ranked the genes using the filtering method of Signal-to-Noise Ratio (SNR) score and the optimization technique was carried out using PSO for dimensionality reduction. KNN, Probabilistic Neural Network (PNN) and SVM were utilized as classifiers. PNN recorded 96%, 96.34%, 87.32% and 98.12% classification accuracy from Leukemia, Colon, DLBCL and Breast cancer datasets respectively.

Swathi et al. [70] designed the default Adaptive Resonance Theory (ART1) network model for breast cancer detection. This model consisted of three stages which included recognition, comparison and search phases. In their work, the Backpropagation algorithm was applied to minimize the error and training of network architecture. The experimental results revealed that ART1 was an efficient network in classifying cancer datasets. The model achieved better results for the unsupervised machine learning breast cancer dataset with 92% accuracy.

Dev et al. [14] designed three different algorithms for cancer tumour diagnosis. In this research work, Backpropagation Network (BPN), Functional Link Artificial Neural Network (FLANN) and PSO-FLANN were employed for the classification of breast cancer. In their work, they examined and compared the outcomes of the three classifiers. They reported 56.12% Classification rate for BPN and 63.34% FLANN, while PSOFLANN gave the best classification rate of 92.36%.

El Akadi et al. [18] presented a proposal that comprises a two-stage feature selection approach for Microarray data. The first phase involves the filtering of genes using MRMR. While GA is used in the second stage to select the best gene subsets. Naïve Bayes (NB) and SVM were employed for the classification and assessment analysis. The best results obtained from their study showed that the top 15 genes from Lymphoma yield 100% classification accuracy, 97.26% classification accuracy from Lung data and 100% classification accuracy from Leukemia data.

Rajeswari & Sophia [56] proposed an approach for tumour cells identification. They used SVM and Fuzzy Neural Network (FNN) to solve the classification task. The liver cancer dataset was utilized to analyse the proposed approach. Their results revealed that the proposed technique could diagnose cancer diseases considerably when it was compared with the conservative procedures of a cancer diagnosis. FNN recorded performance accuracy between 92 and 96%.

Ghorai et al. [25] developed a hybrid of filters and wrapper Computer-Aided Diagnosis (CAD) for cancer classification using Microarray gene expression profiles. MRMR ranking algorithm was utilized for gene selection and wrapper technique based on Nonparallel Plane Proximal Classifier (NPPC) was applied on selected genes to reduce the computational burden. Their results showed that 12 genes from ALL-AML data attained 96.56% classification accuracy and also, 9 genes from Lung data yielded 99.44% classification accuracy.

Revathy & Amalraj [59] proposed a method that hybridizes the enrichment score and SVM classifier for the Microarray cancer classification task. The Lymphoma dataset used is randomly partitioned into train and test datasets. The relevant genes were selected using a ranking approach and then passed selected genes into the classifier one after the other. But, if better accuracy was not achieved, the gene combination is performed on the ranked dataset. The report of their work showed that the combination of SVM and enrichment score performed better than SVM with T-Score. With 5 genes, SVM and enrichment score and SVM with T-Score yielded 99.5% and 97% classification accuracy respectively.

Rangasamy & Venketraman [58] developed a model that ranked the genes using classical statistical approach and two different learning algorithms. The researchers utilized the datasets that contained two classes like Liver and Leukemia datasets, and the one that had more than two classes in the dataset such as Lymphoma. They used Analysis of Variance (ANOVA) and Linear Discriminant Analysis (LDA) for genes selection and SVM-One-Against-All (SVM-OAA) and Radial Basis Function (RBF) Kernel-based on the suitability of dataset type. The learning algorithm was evaluated utilizing all possible combination of genes and the report of optimal genes combination were recorded. Their results showed that SVM-OAA yield average testing accuracy of 99.39% and while LDA: SVM-OAA yield 98.72% average testing accuracy.

Mart'ın-Merino & De Las Rivas [41] employed a Kernel Alignment KNN for cancer detection using gene expression profiles data. The Kernel alignment KNN achieves better performance and enhances the performance of classical KNN.

Wang et al. [75] presented a technique that involves two main steps of Microarray data analysis. In the first step, T-test and Class Separability (CS) scoring were used for genes selection and Fine classifier were used in the second stage which is gene classification. The divide and conquer technique was employed to achieve better accuracy performance. They used two different datasets in this research, Lymphoma Data and SRBCT Data. They employed an FNN technique to handle missing values in Microarray data. The best marker genes selected were passed one after the other to the classifier until better accuracy of 95% was attained with 3 genes from SRBCT. Also, they compared the result they obtained with FNN and SVM classifiers.

From the literature review, it has been established that Microarray data are characterized with challenges of irrelevant, high dimensional and redundant features. To select relevant and informative genes from these features is usually a herculean task, a lot of statistical and machine learning approaches had been proposed and utilized. The meta-heuristic or bio-inspired approach is not left out of the methods used in the past. In this research, Modification of BA was done using Random forest as the fitness function, in order to make the BA robust for the selection of informative genes in multiclass microarray data and also, establish the optimal threshold for each of the datasets used for this research work.

3 Material and methods

The methodology for this study is summarized in the conceptual framework shown in Fig. 1.

3.1 Datasets

Microarray dataset is typically denoted as M by N matrix, where N is columns (the corresponding gene expressions) and M is rows (the number of experimental samples). A brief summary of the datasets used for this work is shown in Table 1. The performance evaluation of the proposed method was done by performing experiments on selected 7 Microarray cancer datasets with different sizes of sample, features, and classes. The selected datasets comprise both binary and multiclass data which were characterized with the problems of class imbalance, missing value and dataset shift. The reason for including multiclass datasets is because the classification tasks are more complex. These datasets are available online [79].

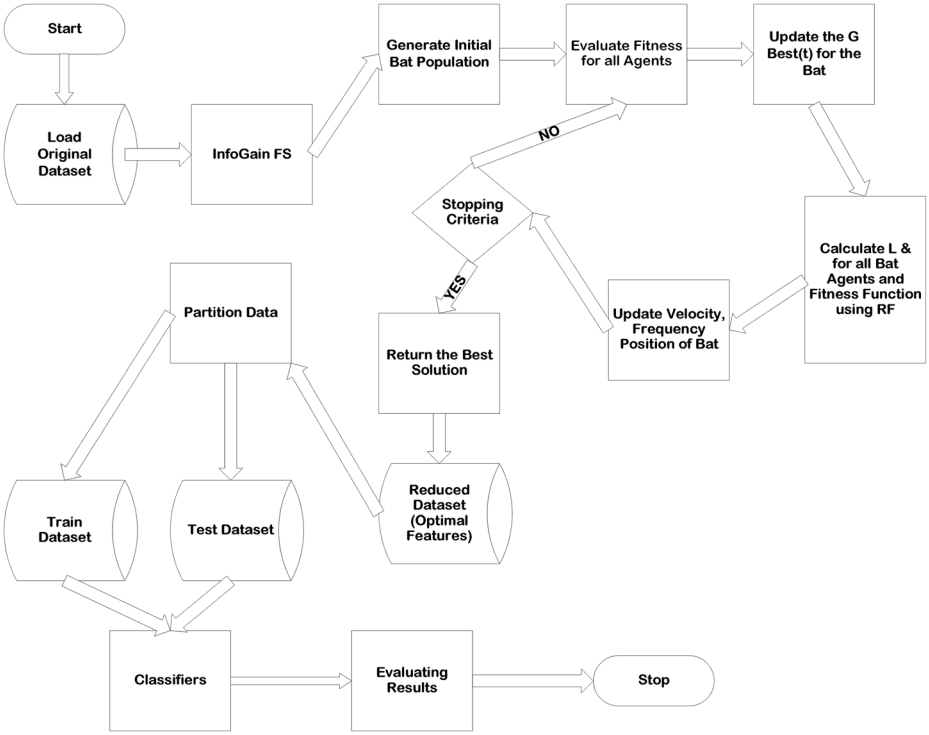


Fig. 1 Conceptual framework

3.2 Data pre-processing

Data pre-processing is the first crucial task after the selection of suitable datasets. This stage is very vital to treat the data that contain missing values. The reason is due to the decision trees algorithm proposed at the classification stage did not work with a dataset with missing values, therefore, the datasets were first treated to take care of missing values it contained.

In the pre-processing phase, the missing values in the whole dataset are recovered through the process called imputation. Imputation is a process of replacing the missing values in the dataset with various feasible values. Imputed values are reliable as the real observed data. Imputation with a mean approach was adopted to treat the missing value in this work. Substituting the mean for missing values is one of the common approaches and usually, gives

Table 1 Datasets summary

	Datasets	#Gene	#Instance	#Class
1	Lung Cancer	12,600	203	5
2	Ovarian Cancer	15,154	253	2
3	Leukemia_3c	7129	72	3
4	Leukemia_4c	7129	72	4
5	Lymphoma	4026	66	3
6	MLL	12,582	72	3
7	SRBCT	2308	83	4

satisfactory results. This approach was done by computing the mean of each column that contains the missing value and then assigned the computed mean to the missing value.

3.3 Filter stage

In this stage, each of the datasets was subjected to InfoGain-MBA to select relevant and informative features from the original datasets. A single feature may be considered irrelevant if treated in isolation, but when combined with other features it becomes a significant feature. This is due to the existence of interactions between features. In this study, instead of dealing with all single features as independent, the proposed methodology deal with feature subsets based on feature interdependency that prevents accidental elimination of relevant features that can result in the loss of useful information and at the same time resulted in poor diagnosis performance.

3.3.1 Information gain

Information gain (InfoGain) is one of the prominent attribute evaluation techniques. It is a univariate filter technique that offers an ordered ranking of all the features and selects based on the specified threshold [28]. InfoGain measures the number of information gain from the class prediction by the presence or absence of the informative genes in a dataset. The entire range of data expression of the gene j is initially divided into two regions based on the specified threshold. The region with expression values lower than the threshold is labelled as region 1, and the other as region 2. Let n_{vm} be the number of the expression values of gene j in region m, and let i_{km} be the number of instances with class k in region m. Then, the InfoGain of gene j is computed as shown in Eq. 1.

$$Gain_j = \sum_{k=0}^{1}\sum_{m=1}^{2}\frac{i_{km}}{N}\log_2\frac{i_{km}}{n_{vm}} - \sum_{k=0}^{1}\frac{i_{k1}+i_{k2}}{N}\log_2\frac{i_{k1}+i_{k2}}{N} \quad (1)$$

A gene with a high InfoGain indicates that the gene has more information to minimize the uncertainty in classification. In this study, the selection threshold of InfoGain was set to top 40%, 60% and 80% in all the datasets.

3.3.2 Bat algorithm

The concept of the Binary Bat Algorithm (BBA) cannot be comprehended except to firstly discuss the original bat algorithm. Bats are captivating creatures that possess a robust sound propagation technique known as Bio-Sonar. Bio-sonar, also called echolocation, is used while searching for food and/or navigating through the obstacles in the dark. Echolocation is a kind of phenomenon that describes how animals use their sound for ranging. Microbat is one of the animals that mostly use echolocation features. They send out a loud and short pulse of sound, waiting for the fraction of time till it hits into an object, and the echo reflects back to their ears [26, 47]. Hence, bats can determine their distance from an object [42]. Apart from this, bats use this amazing phenomenon to be able to differentiate between an obstacle and food and this gives them the chance to search for food in absolute darkness [63]. This feature of microbats' echolocation has fascinated the attention of many scholars from various fields. Based on this characteristic of microbats behaviour, Yang [76] has designed a novel and fascinating Meta-heuristic optimization technique named Bat algorithm. This technique was designed to mimic

the group of bats chasing target - foods employing the aptitude of their echolocation. To model this behaviour mathematically, Yang [76] has formulated some rubrics as follows:

1) All bats utilize echolocation to determine their location, and also use it to differentiate between food/prey and obstacle barriers in a marvelous way;
2) A bat b_i flies arbitrarily with velocity $\mathcal{V}$ at position D with a fixed frequency $\mathcal{F}_{min}$, adjust wavelength λ and loudness L_0 to hunt for food. They regularly vary the wavelength (or frequency) of the sound produced and regulate the rate of pulse emission $\mathcal{R} \in [0, 1]$, based on the closeness to their target;
3) Though the loudness can be adjusted in different manners, Yang [76] assumes that the loudness fluctuates from a high positive L_0 value to a least constant L_{min} value.

The search space in the Bat algorithm can be explored through varying positions and velocity vectors. Each bat b_i can be defined with its position D_i, frequency $\mathcal{F}_i$, and Velocity $\mathcal{V}_i$ in a d-dimensional search space. These vector parameters are updated as in Eq. 2, Eq. 3 and Eq. 4.

$$\mathcal{V}_i(t+1) = \mathcal{V}_i(t) + (D_i(t) - \mathcal{G}_b)\mathcal{F}_i \tag{2}$$

$$D_i(t+1) = D_i(t) + \mathcal{V}_i(t+1) \tag{3}$$

where t is time, $\mathcal{G}_b$ is the best solution attained so far and $\mathcal{F}_i$ denotes the frequency of the i-th bat, which is constantly updating in the course of iteration as denoted in Eq. 4:

$$\mathcal{F}_i = \mathcal{F}_{min} + (\mathcal{F}_{max} - \mathcal{F}_{min})\varphi \tag{4}$$

where φ is a uniform distribution random number in [0, 1]. The Bat algorithm used a random pace to enhance its ability to search as in Eq. 5.

$$D_{new} = D_{old} + \varepsilon L^t \tag{5}$$

where ϵ is a random number in [−1, 1], and L^t is the average peak loudness of sound emitted from all bats at iteration t. For each iteration, the loudness and pulse emission i are updated as in Eq. 6 and Eq. 7:

$$L_i(t+1) = \propto L_i(t) \tag{6}$$

$$\mathcal{R}_i(t+1) = \mathcal{R}_i(0) + \left[1 - e^{(-\tau t)}\right] \tag{7}$$

where α and τ are constant values which lie between 0 and 1 employed to update loudness rate L_i and pulse rate ($\mathcal{R}_i$). The pseudo-code of the BA is shown in Algorithm 1

3.4 Binary bat algorithm

In an incessant version of the Bat algorithm, the artificial bat navigates around in searching for a space to update its position and velocity vectors within the continuous real domain, though, to deal with binary search space where the vector position or solution is in a sequence of 0's

and 1's binary bits. Since the binary search space deals with only two numbers of 0 and 1, performing the positions update process may not be achieve using Eq. 3. Hence, a transformation approach needs to be established to mirror the change in velocity vector values and the components of position vector from'0′ to '1' or vice versa.

Nakamura et al. [47] have presented a binary version of the Bat algorithm limiting the new bat's position by a sigmoid function to only binary values as shown in Eq. 8.

$$S\left(\mathcal{V}_i^j\right) = \frac{1}{1 + e^{-\mathcal{V}_i}} \tag{8}$$

So, substitute Eq. 4 with Eq. 9

$$D_i^j = \begin{cases} 1, & if\ S\left(\mathcal{V}_i^j\right) > \sigma, \\ 0, & otherwise, \end{cases} \tag{9}$$

where σ ~ U(0,1). Hence, Eq. 8 can yield binary values only for each bat's coordinates in the Boolean lattice, which indicate that features are selected or discarded.

Algorithm 1: Pseudocode of bat algorithm [76]

Initialize the bat population: D_i *(I = 1, 2, ..., n) and* $\mathcal{V}_i$

Define pulse frequency $\mathcal{F}_i$

Initialize pulse rate $\mathcal{R}_i$ *and the loudness* L_i

While *(t < Max number of iterations)*

Generate new solutions by adjusting frequency,

Updating velocities and positions [equations (2) to (5)]

If *(rand >* $\mathcal{R}_i$*)*

Select a solution among the best solutions randomly

Generate a local solution around the selected best solution

End if

Generate a new solution by flying randomly

If*(* $rand < L_i$ & $f(D_i) < f(\mathcal{G}_b)$

Accept the new solutions

Increase $\mathcal{R}_i$ *and reduce* L_i

End if

Rank the bats and find the current $\mathcal{G}_b$

End while

3.4.1 Description of the proposed modified binary bat algorithm

There is a high tendency for Bat algorithm based feature selection to trap into local optima during the process. Hence, this motivates the design of MBA which used a global search approach with some additional techniques, targeting at enhancing the outcomes of Bat algorithm for feature selection. The details of the proposed MBBA for feature selection is given as follows.

3.4.2 Encoding and fitness functions

In MBA, binary string is used to encode each bat in the population and the string length is determined by the number of features in the dataset. Therefore, each bat bit represents a feature. For instance, in a dataset that has n features, n–bit string will stand for each bat as depicts by Eq. 10.

$$D_i^j = \left(D_i^1, D_1^2, D_i^3, \ldots D_i^n\right), \mathrm{i} = 1, 2, 3, \ldots, \mathrm{m} \tag{10}$$

where m denotes the bats' size, and the $D_i^j = \{0, 1\}$ indicates the jth feature selected in the feature subset.

Each of the candidate solutions is estimated by a fitness function defined in Eq. 11, where *Acc* represents the classification accuracy attained by the Random Forest classifier and *S(D)* is the total amount of selected features. From Eq. 11, two different components determine the fitness of each bat and these components are weighted differently. The value of α is assigned at the beginning of the experiment.

$$\mathrm{fit(D)} = \alpha \times \mathrm{Acc} + (1-\alpha) \times \frac{\mathrm{n-S(D)}}{\mathrm{n}} \tag{11}$$

The pseudo-code of the InfoGain-MBA used in the work has shown in Algorithm 2.

3.5 Classification algorithms

The classification model serves as a means of determining the class labels of a given dataset. Classification is considered a prognostic data mining task for the detection and diagnosis of different varieties of diseases. Usually, the model can be generated in two steps namely: learning and classification. The learning phase is achieved by training with pre-defined labelled data to build the model. The model is also evaluated by estimating the accuracy of the model on the testing dataset. If the model yields acceptable accuracy, then it can be employed to predict the unknown class of the new dataset [31].

Algorithm 2: Pseudocode of proposed modified binary bat algorithm (MBA)

Input: *Data Sample D with features* $F(x_1 \ldots x_n)$, Θ *Threshold value,* $\mathcal{V}_i = 0$, *Bat population:* D_i $(i = 1, 2, \ldots, n)$, *rand (0 , 1);*
Output: *Optimal features subset* $F_{\mathcal{G}_b}$;
1. *Define pulse frequency* $\mathcal{F}_i$
2. *Initialize pulse rates* $\mathcal{R}_i$ *and the loudness* L_i
3. *the feature subset* $F_{\mathcal{G}}$; $F_{\mathcal{G}_b} \leftarrow \phi$
4. *Apply attribute filter on training dataset D*
5. *Compute InfoGain by using Equation 1*
6. *If (InfoGain >* Θ*)*
7. *Select subset x with highest InfoGain*
8. $F_{\mathcal{G}} \leftarrow$ *subset x*
9. ***while*** $t <$ *Maximum iterations do*
10. $D_i \leftarrow F_{\mathcal{G}}$
11. *update velocities and adjust frequencies (using Eqn. 2 to 5)*
12. *Using Equation (11), Calculate fitness function value*
13. *Using Equation (9), update* D_i
14. ***if*** *(rand >* $\mathcal{R}_i$ *) then 0*
15. *Select the global best solution (*$\mathcal{G}_b$*) among the available best solutions and with the available* $\mathcal{G}_b$ *dimensions, modify the dimensions of* D_i *randomly*
16. ***end if***
17. *Generate new solution randomly*
18. ***if*** *((rand <* L_i *)and (f(*D_i *) < f(*$\mathcal{G}_b$*))* ***then***
19. *Accept the new solutions, Increase* $\mathcal{R}_i$ *and reduce* L_i
20. ***end if***
21. *Find the current* $\mathcal{G}_b$ *and Rank the Bat*
22. $F_{\mathcal{G}_b} \leftarrow \mathcal{G}_b$
23. ***end while***
24. *return* $F_{\mathcal{G}_b}$
25. ***end***

3.5.1 Decision tree classifiers

The procedure for learning in a decision tree algorithm is centred on the splitting criteria principle. Decision trees had a structure similar to a flow chart, with a tree-like arrangement where samples are classified based on the value of their features. An instance in classification is denoted by a node in a decision tree and the branches depict the outcome of the tests, where the class label is represented by each leaf node. In literature, there are a lot of decision tree classifiers, the famous one includes, Iterative Dichotomiser 3 (ID3), C4.5 (J48), C5, and Classification and Regression Tree (CART), are the widely employed because to their high precision and accuracy performance [51]. The benefits of the decision tree approach are

interpretability and comprehensibility. In terms of capability, decision tree is a rapid and effective method of classifying data set entries, handles numeric and categorical attributes, take care of interaction effects between variables, robust to outliers and missing values, and offer good decision support abilities [2, 13, 15].

Generally, a decision tree classifier involves two major steps; a growing phase and a pruning phase [68]. In the first phase, the decision tree is built in a top-down manner with the full dataset corresponding to the tree root node, a feature and a splitting weight are determined based on the splitting criterion and the dataset is partitioned into two different subsets subject to the feature value.

Afterwards, the built tree may result in a complex model due to the size of the dataset used for training [31]. The pruning phase is employed to minimize consecutive branches so that the tree is built to adequately generalize the model and reduce overfitting glitches. In this stage, individual nodes are compared against pruning criteria. Once a node satisfies the pruning criteria, its sub-tree is discarded and that particular node becomes a leaf node. This procedure continues until the built tree is reduced to a single node and the optimal sub-tree obtained (in terms of accuracy) during this procedure is retained to be the final tree. Finally, the leaf node represents each region (or class) in the dataset with an associated decision rule that corresponds to allot the most frequent class label to its corresponding region of the observations it belongs to [8]. The pseudocode for the decision tree is shown in Algorithm 3.

Algorithm 3: Pseudocode of decision tree

Input: *GenDecTree(Sample S, Features F);*

Output: *root;*

Steps:

1. ***If*** *stopping_condition(S, F) = true* ***then***
   a. *Leaf = createNode()*
   b. *leafLabel = classify(s)*
   c. ***return*** *leaf*
2. *root = createNode()*
3. *root.test_condition =findBestSpilt(S,F)*
4. *V = {v | v a possible outcomecf root.test_condition}*
5. ***For each*** *value v ∈ V:*
   a. S_v *= {s | root.test_condition(s) = v and s ∈ S };*
   b. *Child = TreeGrowth (*S_v*, F);*
   c. *Add child as descent of root and label the edge {root → child} as v*
6. ***return*** *root*

3.5.2 Classification and regression tree

Classification and Regression Tree (CART) [8] is based on Hunt's algorithm; it can process both continuous and categorical attributes to generate a decision tree. Also, it deals with missing values in the dataset and builds a decision tree using the Gini Index as an attribute

selection measure. CART splits training datasets into binary; therefore, it generates binary trees. Gini Index measure does not involve probabilistic assumptions used in ID3, C4.5. However, CART uses cost complexity pruning to eliminate the erratic branches from the decision tree to enhance the classification accuracy.

The tree grows in the CART algorithm by carrying out a thorough search of all variables for each decision node and all possible splitting values, then selecting the optimal split. It generates an estimate for the misclassification rate.

3.5.3 C4.5

C4.5 Decision tree also referred to as the J48 Classifier algorithm, is one of the famous employed and practical techniques for inductive inference [54]. It is an approach that used the approximation of discrete-valued functions that is robust to noisy data and has a proficiency for learning disjunctive expressions [44, 55].

In this algorithm, the approximating discrete-valued learning function is symbolized by a decision tree. The decision trees generated can likewise be denoted as, sets of if-then rubrics to ease human readability. C4.5 classifies a new sample by building a decision tree from the features of the provided training set. When it comes across the training set, it learns the features that are capable of classifying the various samples accurately. The likely attribute values which are devoid of ambiguity are allotted to the concerned branch by terminating it.

3.5.4 Random Forest

A well-known classification method employed in numerous classification domains is Random Forest (RF) [23, 36, 64]. RF is also considered as bagged decision trees [6], that comprises a group of weak classifiers to articulate a robust learner [23]. RF adopts two approaches:

1. Classification approach.
2. Regression Tree (CART) approach [24].

This technique is the development of several decision trees centred on the bootstrap aggregation (bagging) method [34]. CART algorithm permits the learning of accurate classification among some independent variables (x) and some dependent variables (y), and the relationship between them. RF method randomly chooses a subset to model an autonomous decision tree. The process is iterative which splits the randomly selected subset from the root node to a child node [69]. The split continues iterating until each tree attains a leaf node without cropping. Individual tree models the classification of the samples and the objective variable self-sufficiently contribute to the final tree class (vote) [69]. The overall classification is based on the majority obtained from tree voting. Algorithm 4 depicts the steps involved in building an RF technique.

Algorithm 4: Pseudocode of random Forest classification [23]

Input: *N - Quantitative amount of bootstrap samples*
M - Total number of features
m - Sample size
k - Next node
Output: A Random Forest (RF)
Steps:
1. ***Creates*** *N bootstrap samples from the dataset.*
2. Every node (sample) takings a feature randomly of size m where m<M.
3. Builds a split for the m features selected in Step 2 and detects the k node by using the best split point.
4. Split the tree iteratively until one leaf node is attained and the tree remains completed.
5. The algorithm is trained on each bootstrapped independently.
6. Using trees classification voting predicted data is collected from the trained trees (n).
7. The final RF model is build using the peak voted features.
8. ***return*** *RF*
End.

3.5.5 Distribution of training and test datasets

After dimensionality reduction of original datasets by InfoGain-MBB Algorithm, the reduced datasets were partitioned into train and test sets with 70–30%. 70% for training and 30% for testing.

3.5.6 Cross validation

To guarantee a good assessment for all the learner classifiers and to prevent overfitting during the evaluation, a double loop of cross-validation [41] was employed. The external loop is built on stratified 10-fold cross-validation that repeatedly splitting the data into ten different sets, the first nine for training while the last one for testing. In the inner loop, stratified 9-fold cross-validation was employed for the training set to determine the best parameters and preventing bias in the error computation. The stratified variant of cross-validation preserves the same percentage of samples for each class in training and test sets. This is essential for microarray data because of the imbalance class proportions and also with small samples of data.

4 Performance metrics used for the classification algorithms

The following are the metrics used to appraise the performance of classification algorithms employed in this study.

True Positives (TP): True positives are the scenarios when the actual class of the data point is True and predicted to be True.

True Negatives (TN): True negatives are the situations when the actual class of the data point is False and also predicted to be False.
False Positives (FP): False positives are the scenarios when the actual class of the data point is False but predicted to be True. It is False because the classifier has predicted incorrectly and positive. After all, the class predicted was a positive one.
False Negatives (FN): False negatives are the situations when the actual class of the data point is True but predicted to be False. It is False because the classifier has predicted incorrectly and negative because the class predicted was a negative one.

i. Accuracy

Accuracy is the percentage of correctly classifies instances out of all instances that are, the number of correct predictions achieved by the model over all types of predictions made (Eq. 12). Accuracy is a good measure when the target variable classes in the data are nearly balanced.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{12}$$

ii. Recall or Sensitivity

Sensitivity is the true positive rate (also known as recall (Eq. 13)). It is the number of instances from the positive class that predicted correctly. In cancer classification, it is a measure of the proportion of the patients that had cancer and was predicted by the model as having cancer.

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{13}$$

iii. Specificity

Specificity is known as the true negative rate (Eq. 14). It measures the proportion of patients that did NOT have cancer and were predicted by the model as non-cancerous. The actual negatives (People NOT having cancer are FP and TN) and the people diagnosed as not having cancer are TN. Specificity is the exact opposite of Recall.

$$Specificity = \frac{TN}{TN + FP} \tag{14}$$

iv. Kappa

Kappa or Cohen's Kappa (Eq. 15) is similar to the Accuracy score, except that it is normalized at the baseline of random chance on your dataset (that is, it takes into account

the accuracy that would have happened anyway through random predictions). It is a more useful measure to use on problems that have an imbalance in the classes (e.g. 70–30 split for classes 0 and 1).

$$Kappa = \frac{(Observed\ Accuracy - Expected\ Accuracy)}{(1 - Expected\ Accuracy)} \tag{15}$$

xxii. Detection Rate

The detection rate is the proportion of the whole sample where the events were detected correctly.

vi. Balanced Accuracy

Balanced Accuracy is defined as the average of recall obtained on each class that is, balance between correctly predicted classes (Eq. 16).

$$Balance\ Accuracy = \frac{(Sensitivity + Specificity)}{2} \tag{16}$$

vii. Time Taken to Extract

It is time taken by the algorithm to extract/select the feature from the complete dataset. It is measure in sec.

viii. Estimated Error Rate

The error rate is the percentage of samples that have been incorrectly classified by a classifier model.

5 Results and discussion

The strength of the proposed method hinges on the outcome of experiments obtained under different circumstances. Hence, the outcome of experiments is crucial in the research domain. The proposed method was coded in R Studio version 3.5.1 and run in a 64-bit Core i3-6100U (2.30 GHz) PC equipped with 8GB of RAM.

5.1 Results

The performance of the proposed algorithms was assessed on 7 different Microarray cancer datasets with a diverse number of samples, features (number of genes), and classes. The results of different datasets and algorithms were also critically compared.

5.2 InfoGain-MBA results

The parameters used for MBA were chosen based on the preliminary experiments conducted. They offer a good trade-off between the computational time and the quality of the solution desired. The preliminary investigations as shown in Fig. 2 (see Appendix Table 13) revealed that an increase in the bat population leads to a trade-off that influences the algorithm performance by increasing the computational time of the algorithm.

The appropriate value was chosen for pulse rate and loudness based on preliminary experiments conducted that generates a good quality solution. Table 2 shows the parameters used for the implementation of MBBA.

Table 3 presents the results of InfoGain-MBA feature selection for each dataset. Three different thresholds were used for each dataset and tagged 1st, 2nd and 3rd Run. That is 40%, 60 and 80% threshold values.

Figure 3 show the graphical representation of the average time taken by InfoGain-MBA to extract features for all the datasets used. Generally, InfoGain-MBA has the low average time taken to extract the features in all the datasets except in MLL, Lymphoma, and Lung datasets.

5.3 Classification results

The classification performance of the selected features of microarray data was evaluated using four classifiers: C4.5, DT, RF and CART. The results were presented based on the dataset by dataset. Each classifier was designated with three 'Runs', 1st, 2nd and 3rd 'Run' referred to top 40%, 60% and 80% selected features respectively.

5.3.1 Case 1: Leukemia_3c dataset

The leukemia_3c dataset has three classes of B Cell, T Cell and Acute Myeloid Leukemia (AML). Leukemia_3c has 72 samples with 7129 features and Class label of B Cell with 38

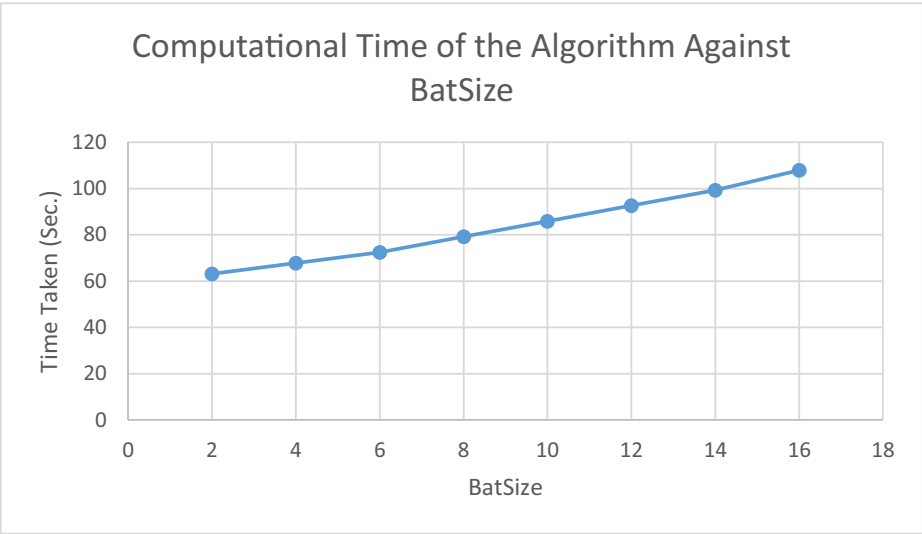


Fig. 2 Computation time of MBA against BatSize for Leukemia_4c dataset

Table 2 Parameters setting for MBA

Parameter	Value selected
Nofbats	4
F_{min}	20
F_{max}	50
L_0	1
$\mathcal{R}$	0.5
A	0.9
τ	1

Table 3 Number of features extracted by InfoGain-MBA

Dataset	1st Run	2nd Run	3rd Run
Leukemia_3c	88	13	3
Leukemia_4c	118	17	4
Lung	425	26	1
Lymphoma	852	266	66
MLL	996	163	24
Ovarian	62	21	1
SRBCT	140	45	7

instances, T Cell with 9 instances and AML with 25 instances. After feature selection processing, the final features used for classification evaluation were reduced to 101, 22 and 3 for the top 40%, 60% and 80% respectively. Table 4 shows the performance of four classifiers on the reduced Leukemia_3c dataset.

All the classifiers perform well with reduced features. In C4.5 100% accuracy was obtained with 88 genes but dropped to 76% with 13 genes and finally increase to 91% with only 3 genes. This is due to the loss of some complementary features in the 2nd 'Run' which affect its accuracy. The same scenario occurred in DT. While RF has 100% accuracy with all the three 'Run'. But in CART the accuracy increase from 76% in the first 'Run' to 91% in the 3rd 'Run'. Only 3 genes out of 7129 genes were able to give good classification accuracy in the

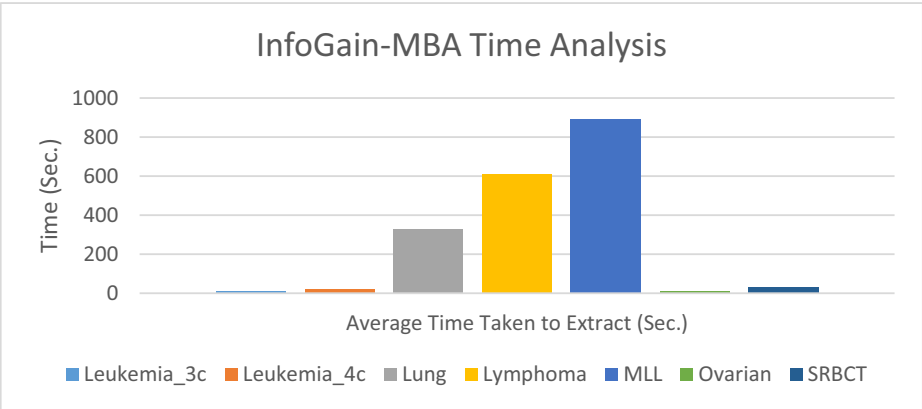


Fig. 3 Average time taken to extract features for all datasets using InfoGain-MBA

Table 4 Result of classifiers for Leukemia_3c dataset

Algorithms	No of selected features	Accuracy	Kappa	Ave. sensitivity	Ave. specificity	Ave. Pos Pred value	Ave. Neg Pred value
C4.5	88	1	1	1	1	1	1
	13	0.76	0.61	0.67	0.86	0.82	0.92
	3	0.91	0.83	0.94	0.62	0.95	0.94
DT	88	0.86	0.73	0.89	0.9	0.93	0.93
	13	0.81	0.70	0.78	0.90	0.80	0.91
	3	0.81	0.61	0.56	0.86	0.82	0.89
RF	88	1	1	1	1	1	1
	13	1	1	1	1	1	1
	3	1	1	1	1	1	1
CART	88	0.76	0.55	0.54	0.85	0.76	0.86
	13	0.76	0.61	0.67	0.85	0.55	0.92
	3	0.91	0.81	0.64	0.94	0.90	0.95

four classifiers. For this dataset, RF performed excellently in all the metrics used for evaluation.

Figure 4 shows the detection rate for the four classifiers with the Leukemia_3c dataset. The lowest average detection rate of 25.4% was obtained in the 2nd 'Run' of C4.5 and 1st 'Run of CART classifiers.

Figure 5 shows the average balanced accuracy for all the classifiers. It illustrates the balance between correctly predicted classes for all the classifiers. 100% balanced accuracy was obtained in all 'Run' of RF and 1st 'Run' of C4.5, and the lowest value of 63.9% was recorded in the 2nd 'Run' of CART.

5.3.2 Case 2: Leukemia_4c dataset

Leukemia_4c dataset has four classes of B Cell, T Cell, Bone Marrow (BM) and Peripheral Blood (PB). Leukemia_4c has 72 samples with 7129 features and Class label of B Cell with 38 instances, T Cell with 9 instances, BM with 21 instances and PB with 4 instances. After feature selection processing, the final features used for classification evaluation were reduced to 118,

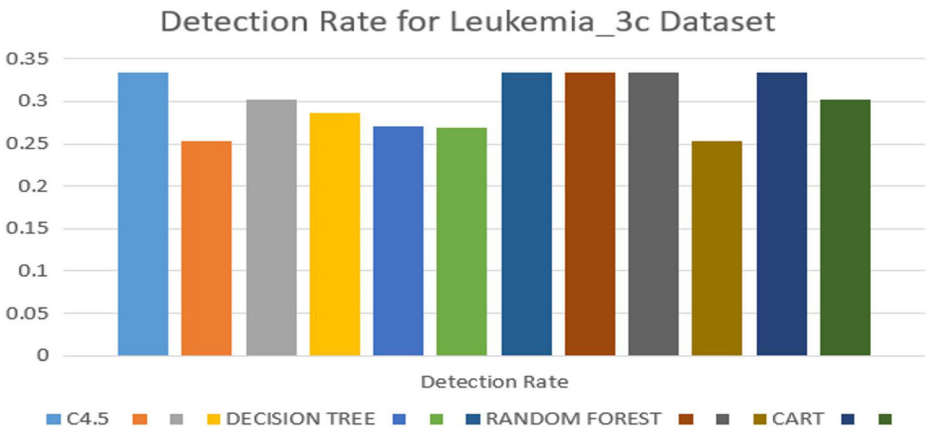


Fig. 4 Detection rate for Leukemia_3c dataset for the four classifiers

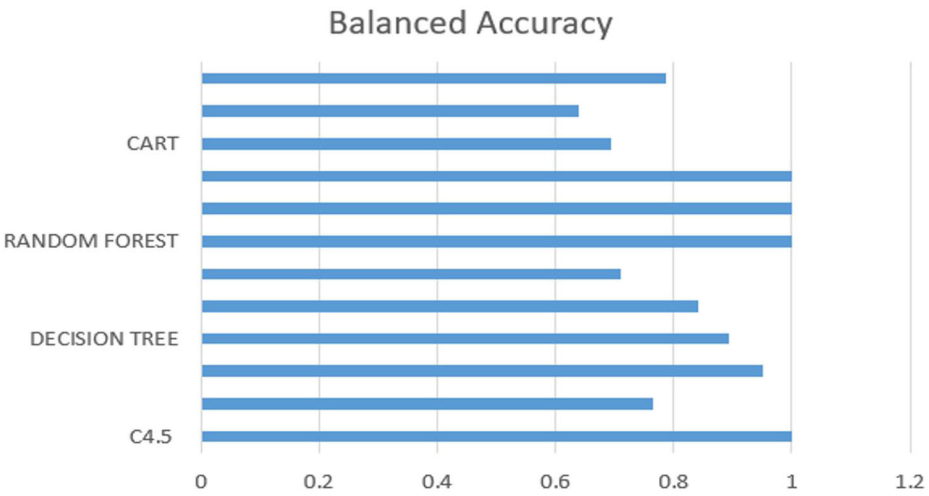


Fig. 5 Balanced accuracy for Leukemia_3c dataset for the four classifiers

17 and 4 for the top 40%, 60% and 80% respectively. Table 5 shows the performance of four classifiers on the reduced Leukemia_4c dataset.

All the classifiers perform well with reduced features. In C4.5 100% accuracy was obtained with 118 genes but dropped to 95% with 17 genes and finally back to 100% with only 4 genes. This is due to the loss of some complementary features in the 2nd 'Run' which affect its accuracy. The DT performed worst with 67% accuracy in the 3rd 'Run'. While RF has 100% accuracy with all the three 'Run'. But in CART the accuracy increase from 76% in the first 'Run' to 81% in the 3rd 'Run'. Only 4 genes out of 7129 genes were able to give good classification accuracy in the four classifiers. For this dataset, RF performed excellently in all the metrics used for evaluation.

Figure 6 shows the average detection rate for the four classifiers with the Leukemia_4c dataset. The lowest average detection rate of 16.67% was obtained in the 3rd 'Run' of DT classifiers.

Figure 7 shows the average balanced accuracy for all the classifiers. It illustrates the balance between correctly predicted classes for all the classifiers. 100% average balanced accuracy was

Table 5 Result of classifiers for Leukemia_4c dataset

Algorithms	No of selected features	Accuracy	Kappa	Ave. sensitivity	Ave. specificity	Ave. Pos Pred value	Ave. Neg Pred value
C4.5	118	1	1	1	1	1	1
	17	0.95	0.90	0.95	0.97	0.98	0.98
	4	1	1	1	1	1	1
DT	118	0.76	0.60	0.58	0.90	0.78	0.90
	17	1	1	1	1	1	1
	4	0.67	0.42	0.41	0.85	0.62	0.86
RF	118	1	1	1	1	1	1
	17	1	1	1	1	1	1
	4	1	1	1	1	1	1
CART	118	0.76	0.64	0.67	0.93	0.75	0.91
	17	0.76	0.44	0.54	0.84	0.86	0.90
	4	0.81	0.67	0.5	0.93	0.79	0.95

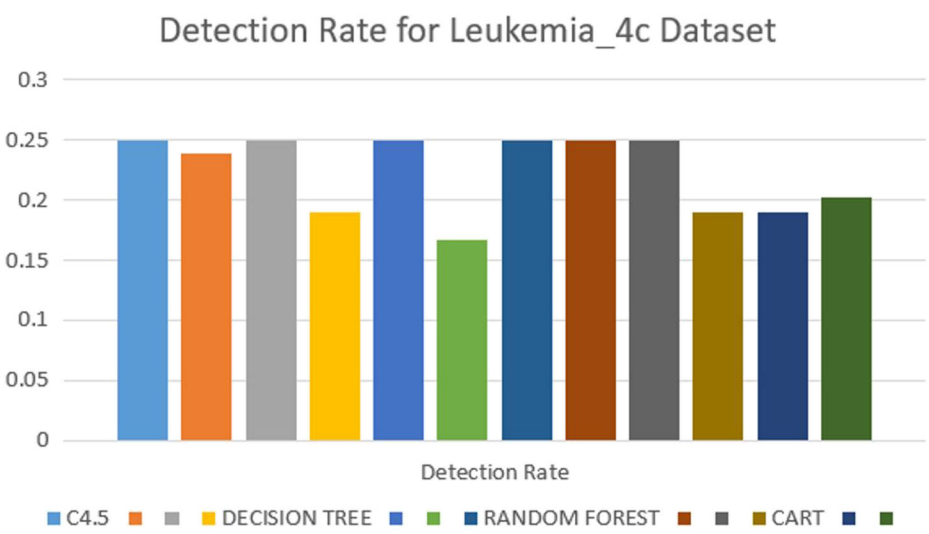


Fig. 6 Detection rate for Leukemia_4c dataset for the four classifiers

obtained in all 'Run' of RF and 1st and 3rd 'Run' of C4.5 and the lowest value of 63.15% was recorded in 3rd 'Run' of DT.

5.3.3 Case 3: Lung dataset

The lung dataset has five classes which are indicated as stage 1 to 5. It has 203 samples with 12,600 features and Class label distribution is as follows: 1 = 139, 2 = 17, 3 = 6, 4 = 21 and 5 = 20 instances. After feature selection processing, the final features used for classification evaluation were reduced to 425, 26 and 1 for the top 40%, 60% and 80% respectively. Table 6 shows the performance of four classifiers on the reduced Lung dataset.

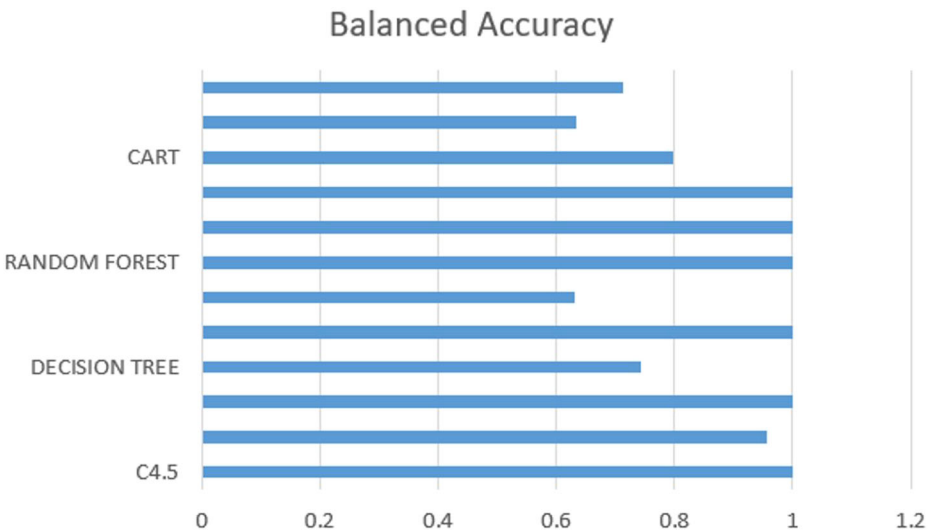


Fig. 7 Balanced accuracy for Leukemia_4c dataset for the four classifiers

Table 6 Result of classifiers for lung dataset

Algorithms	No of selected features	Accuracy	Kappa	Ave. sensitivity	Ave. specificity	Ave. Pos Pred value	Ave. Neg Pred value
C4.5	425	1	1	1	1	1	1
	26	1	1	1	1	1	1
	1	1	1	1	1	1	1
DT	425	0.95	0.89	0.71	0.99	0.76	0.99
	26	1	1	1	1	1	1
	1	0.98	0.97	1	1	1	1
RF	425	1	1	1	1	1	1
	26	1	1	1	1	1	1
	1	1	1	1	1	1	1
CART	425	0.98	0.96	0.95	1	0.9	1
	26	1	1	1	1	1	1
	1	0.98	0.97	0.97	1	0.9	1

All the classifiers performed excellently with reduced features. The least accuracy of 95% was obtained in the 1st 'Run' of DT. Even with 1 gene out of 12,600 genes, all the classifiers were able to yield good performance accuracy and all other evaluation metrics.

Figure 8 shows the average detection rate for the four classifiers with the Lung dataset. The lowest average detection rate of 19% was obtained in the 1st 'Run' of DT classifiers while other classifiers have equal values of average detection rate.

Figure 9 shows the average balanced accuracy for all the classifiers. 100% average balanced accuracy was obtained in almost all classifiers except in 1st 'Run' of DT, 1st and 3rd 'Run' of CART. The lowest value of 84.98% was recorded in the 1st 'Run' of DT.

5.3.4 Case 4: Lymphoma dataset

The lymphoma dataset has three classes consisting of Diffuse Large B Cell Lymphoma (DLBCL), Follicular Lymphoma (FL) and Chronic Lymphocytic Leukemia (CLL). It has 66

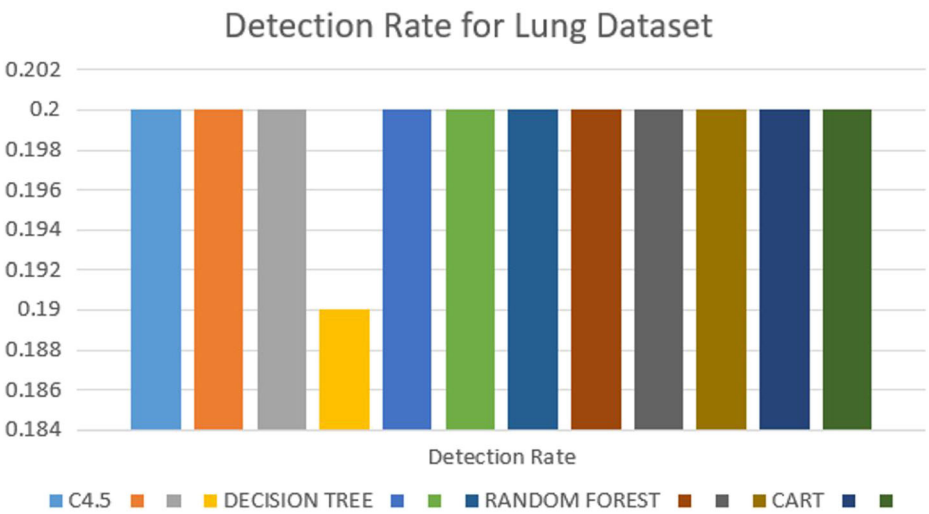


Fig. 8 Detection rate for lung dataset for the four classifiers

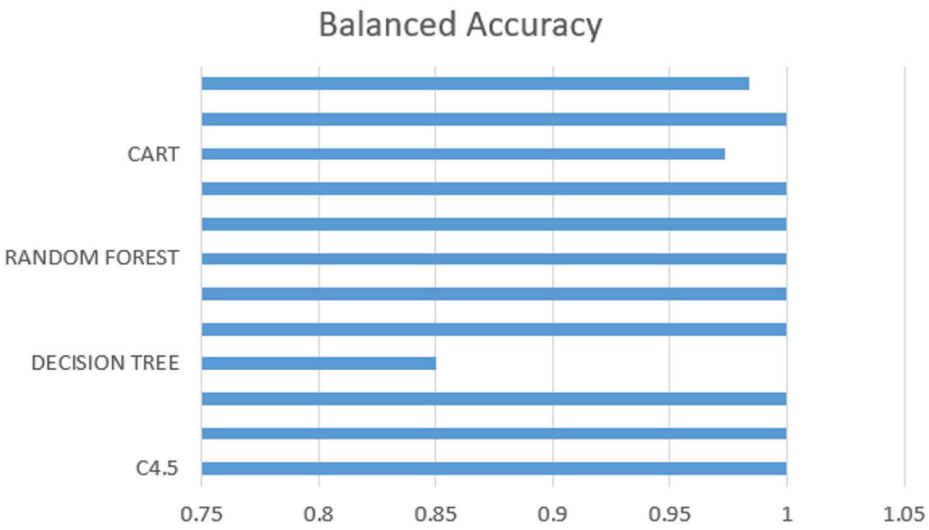


Fig. 9 Balanced accuracy for lung dataset for the four classifiers

samples with 4026 features and Class label distribution is as follows: DLBCL = 46, FL = 9 and CLL = 11 instances. After feature selection processing, the final features used for classification evaluation were reduced to 852, 266 and 66 for the top 40%, 60% and 80% respectively. Table 7 shows the performance of four classifiers on the reduced Lymphoma dataset.

All the classifiers performed very well with reduced features. The least accuracy obtained was 84% in the 1st 'Run' of CART with 825 genes. Even with 66 genes out of 4026 genes, all the classifiers were able to yield good performance accuracy and all other evaluation metrics used in this study.

Figure 10 shows the average detection rate for the four classifiers with the Lymphoma dataset. The lowest average detection rate of 28.07% was obtained in the 1st 'Run' of CART classifiers while other classifiers have roughly equal values of average detection rate.

Table 7 Result of classifiers for lymphoma dataset

Algorithms	No of Selected Features	Accuracy	Kappa	Ave. Sensitivity	Ave. Specificity	Ave. Pos Pred Value	Ave. Neg Pred Value
C4.5	852	1	1	1	1	1	1
	266	1	1	1	1	1	1
	66	1	1	1	1	1	1
DT	852	1	1	1	1	1	1
	266	1	1	1	1	1	1
	66	0.95	0.91	1	1	1	1
RF	852	1	1	1	1	1	1
	266	1	1	1	1	1	1
	66	1	1	1	1	1	1
CART	852	0.84	0.70	0.67	0.93	0.79	0.95
	266	1	1	1	1	1	1
	66	1	1	1	1	1	1

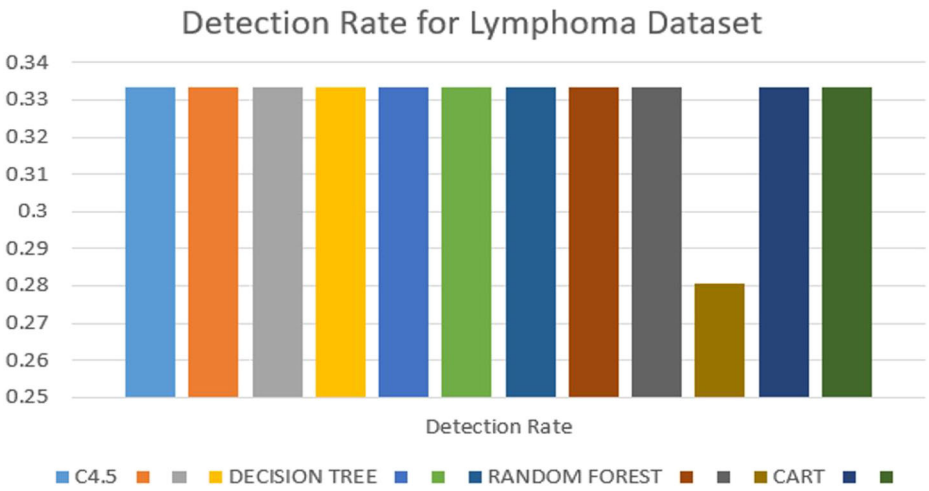


Fig. 10 Detection rate for lymphoma dataset for the four classifiers

Figure 11 shows the average balanced accuracy for all the classifiers. Almost all the classifiers have 100% average balanced accuracy except in the 1st 'Run' of the CART classifier where the average balanced accuracy was 80%.

5.3.5 Case 5: MLL dataset

The mixed-Lineage Leukemias (MLL) dataset has three classes consisting of Acute Lymphoblastic Leukemia (ALL), Acute Myeloid Leukemia (AML) and MLL. It has 72 instances with 12,582 features and Class label distribution is as follows: ALL = 24, AML = 28 and MLL = 20 instances. After feature selection processing, the final features used for classification evaluation were reduced to 996, 163 and 24 for the top 40%, 60% and 80% respectively. Table 8 shows the performance of four classifiers on the reduced MLL dataset.

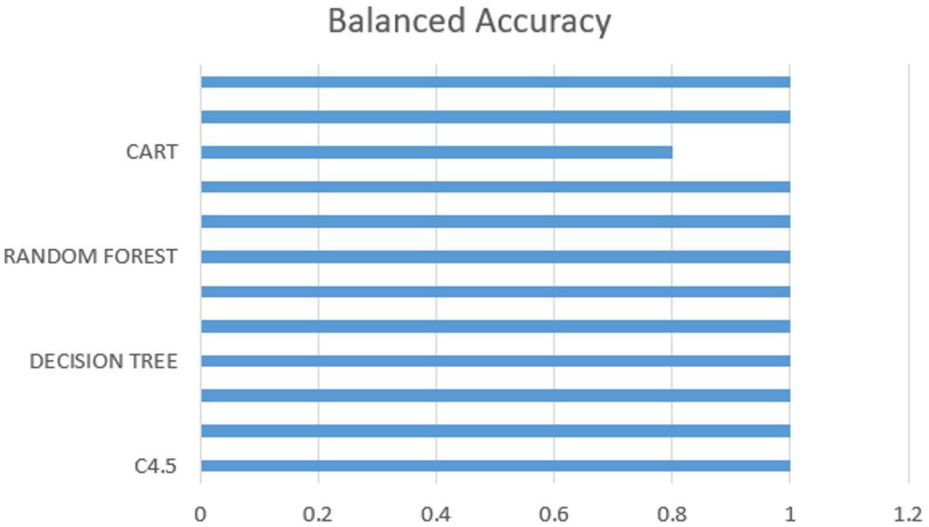


Fig. 11 Balanced accuracy for lymphoma dataset for the four classifiers

Table 8 Result of classifiers for MLL dataset

Algorithms	No of selected features	Accuracy	Kappa	Ave. sensitivity	Ave. specificity	Ave. Pos Pred value	Ave. Neg Pred value
C4.5	996	1	1	1	1	1	1
	163	1	1	1	1	1	1
	24	1	1	1	1	1	1
DT	996	0.81	0.72	0.83	0.92	0.83	0.92
	163	0.86	0.78	0.86	0.94	0.85	0.93
	24	0.95	0.93	0.94	0.97	0.97	0.98
RF	996	1	1	1	1	1	1
	163	1	1	1	1	1	1
	24	1	1	1	1	1	1
CART	996	0.91	0.86	0.92	0.96	0.89	0.95
	163	0.91	0.85	0.90	0.95	0.90	0.95
	24	0.95	0.93	0.94	0.97	0.97	0.98

All the classifiers performed very well with reduced features. The least accuracy obtained was 81% in the 1st 'Run' of DT with 996 genes. Even with 24 genes out of 12,582 genes, all the classifiers were able to yield good performance accuracy and all other evaluation metrics used in this study.

Figure 12 shows the average detection rate for the four classifiers with the Lymphoma dataset. The lowest average detection rate of 26.99% was obtained in the 1st 'Run' of the DT classifier.

Figure 13 shows the average balanced accuracy for all the classifiers. Almost all the classifiers have very good performance of average balanced accuracy. The lowest value of the average balanced accuracy of 87.75% was obtained in the 1st 'Run' of the DT classifier.

5.3.6 Case 6: Ovarian dataset

Selected features from the Ovarian dataset were evaluated by four different classifiers. It has 253 samples with 15,154 features and the Class label is Cancer and Normal. Cancer class indicates a cancerous instance of about 162 instances and Normal class indicates normal instances of about 91 instances. After feature selection processing, the final features used for

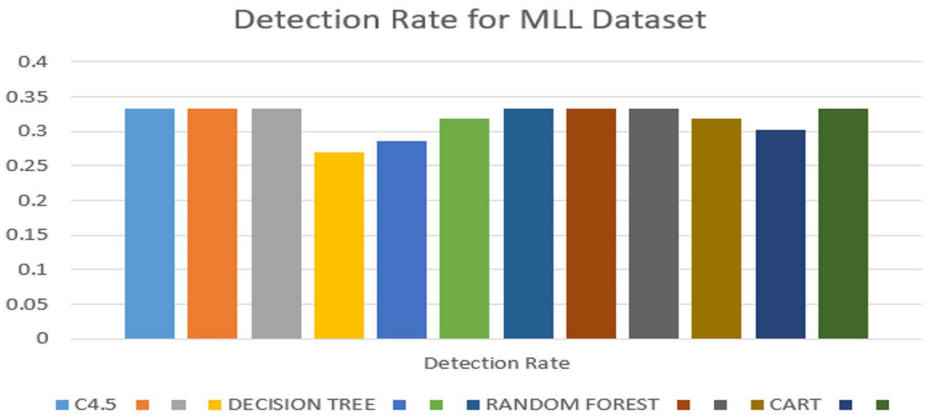


Fig. 12 Detection rate for MLL dataset for the four classifiers

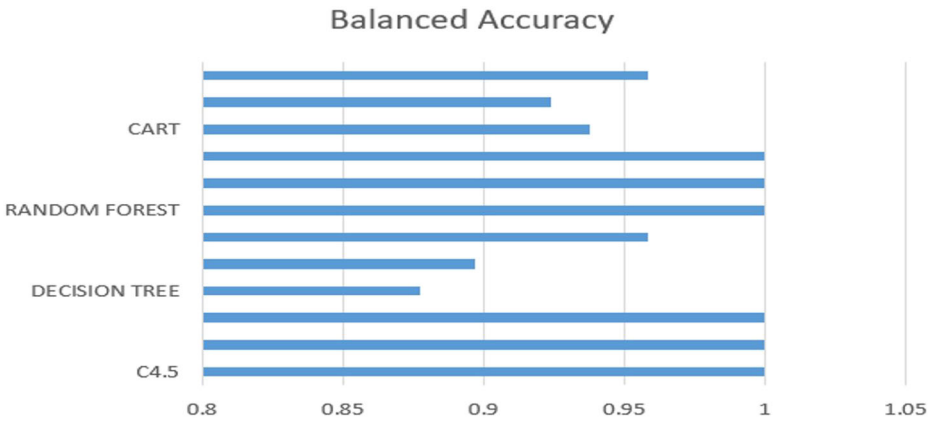


Fig. 13 Balanced accuracy for MLL dataset for the four classifiers

classification were reduced to 62, 21 and 1. Table 9 shows the performance of four classifiers on the reduced Ovarian dataset.

All the classifiers perform excellently with reduced features even with 1 gene selected yields 100% classification accuracy. Figure 14 shows the detection rate for the four classifiers. All the classifiers have a very close range of detection rates of 62.67% to 65.33%. The 2nd 'Run' of all the classifiers have the highest detection rate due to a sizeable number of features to train the classifiers.

Figure 15 shows a balanced accuracy for all the classifiers. It depicts the balance between correctly predicted classes, that is, the Cancer or Normal in all the classifiers. All classifiers recorded better performance.

5.3.7 Case 7: SRBCT dataset

Small Round Blue Cell Tumors (SRBCT) dataset has four classes consisting of Ewing's family of tumours (EWS), Burkitt's Lymphoma (BL), Neuroblastoma (NB) and Rhabdomyosarcoma (RMS) which coded as1 to 4. It has 82 samples with 2308 features and Class label distribution is as follows: EWS = 29, BL = 11, NB = 18 and RMS = 25 instances. After

Table 9 Result of classifiers for ovarian dataset

Algorithms	No of selected features	Accuracy	Kappa	Sensitivity	Specificity	Pos Pred value	Neg Pred value
C4.5	62	1	1	1	1	1	1
	21	1	1	1	1	1	1
	1	1	1	1	1	1	1
DT	62	1	1	1	1	1	1
	21	1	1	1	1	1	1
	1	1	1	1	1	1	1
RF	62	1	1	1	1	1	1
	21	1	1	1	1	1	1
	1	1	1	1	1	1	1
CART	62	1	1	1	1	1	1
	21	1	1	1	1	1	1
	1	1	1	1	1	1	1

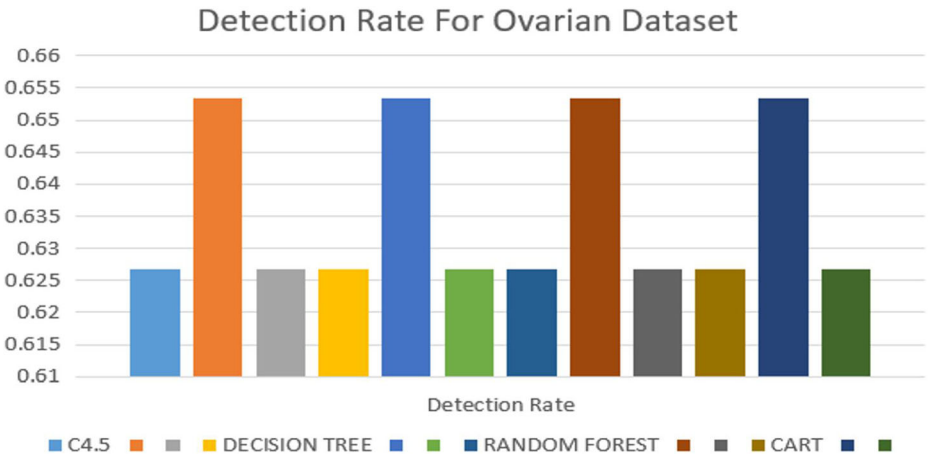


Fig. 14 Detection rate for ovarian dataset for the four classifiers

feature selection processing, the final features used for classification evaluation were reduced to 140, 45 and 7 for the top 40%, 60% and 80% respectively. Table 10 shows the performance of four classifiers on the reduced SRBCT dataset.

All the classifiers performed very well with reduced features. The least accuracy obtained was 75% in the 1st 'Run' of DT and 3rd 'Run' of CART with 140 and 7 genes respectively. Even with 7 genes out of 3208 genes, all the classifiers were able to yield good performance accuracy and all other evaluation metrics used in this study except in the CART classifier.

Figure 16 shows the average detection rate for the four classifiers with the SRBCT dataset. The lowest average detection rate of 18.75% was obtained in the 1st 'Run' of DT and 3rd 'Run' of CART classifiers.

Figure 17 shows the average balanced accuracy for all the classifiers. Almost all the classifiers have very good performance of average balanced accuracy. The lowest value of the average balanced accuracy of 80.1% was obtained in the 3rd 'Run' of the CART classifier.

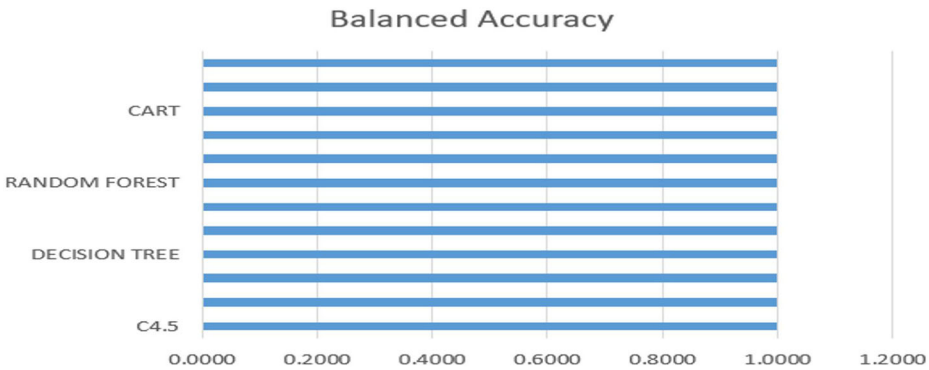


Fig. 15 Balanced accuracy for ovarian dataset for the four classifiers

Table 10 Result of classifiers for SRBCT dataset

Algorithms	No of Selected Features	Accuracy	Kappa	Ave. Sensitivity	Ave. Specificity	Ave. Pos Pred Value	Ave. Neg Pred Value
C4.5	140	**1**	**1**	1	1	1	1
	45	1	1	1	1	1	1
	7	1	1	1	1	1	1
DT	140	0.75	0.66	0.83	0.91	0.79	0.92
	45	0.92	0.88	0.90	0.97	0.94	0.98
	7	0.92	0.87	0.93	0.96	0.96	0.97
RF	140	**1**	**1**	1	1	1	1
	45	1	1	1	1	1	1
	7	1	1	1	1	1	1
CART	140	0.88	0.83	0.93	0.96	0.92	0.96
	45	0.96	0.94	0.94	0.99	0.96	0.99
	7	0.75	0.64	0.68	0.92	0.73	0.92

5.4 Comparison of performance of InfoGain and InfoGain-MBA

Also, a set of experiments were conducted using InfoGain only to select features from original datasets. The features selected were evaluated with a decision tree classifier and compared with InfoGain-MBA feature selection. The results were presented in Table 11. The results here showed that there is a great improvement in the classifier accuracy in InfoGain-MBA compare to InfoGain only, despite the fewer number of features selected.

5.5 Error rate for random Forest

Since RF performed excellently in all the datasets used for this study and the effect of eliminating some important features is hardly shown in its performance because of its robustness. Therefore there is a need for further investigation that will give insight into this effect. This lead to the consideration of an additional metric called estimated error rate per class for RF. Figures 18, 19, 20, 21, 22, 23 and 24 depict the estimated error per class for all the datasets used and its confusion matrix can be found in Appendix Table 14.

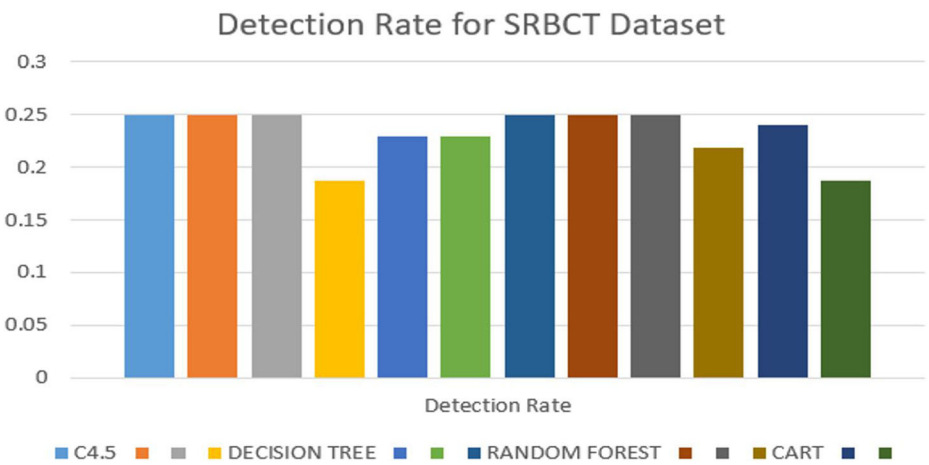


Fig. 16 Detection rate for SRBCT dataset for the four classifiers

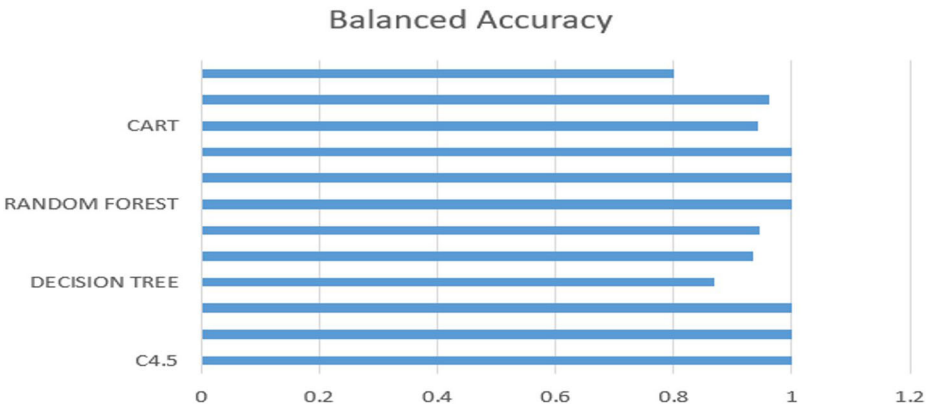


Fig. 17 Balanced accuracy for SRBCT dataset for the four classifiers

Figure 18 shows the estimated error rate per class for Leukemia_3c. The average estimated error obtained for Leukemia_3c was 19.44%, 29.17% and 27.78% respectively for all three 'Runs'. The value increases from the 1st 'Run' to 2nd 'Run' and then decrease in the 3rd 'Run', this indicates that the estimated error increase as the number of features selected is reduced. Therefore, this reveals that some complementary features have been eliminated during the selection.

Figure 19 shows the estimated error rate per class for Leukemia_4c. The average estimated error obtained for Leukemia_4c was 20.89%, 29.17% and 30.56% respectively for all three 'Runs'. The value increases from the 1st 'Run' to the 3rd 'Run', this indicate that the estimated error increase as the number of features selected is reduced. Therefore, this reveals that some complementary features have been eliminated during the selection.

Figure 20 shows the estimated error rate per class for Lung. The average estimated error obtained for Lung was 0.99%, 0.49% and 1.97% respectively for all three 'Runs'. The value of the average estimated error decrease from the 1st 'Run' to the 2rd 'Run', this indicates that the

Table 11 Performance Evaluation of InfoGain and InfoGain-MBBA with Decision Tree Algorithm

Datasets	Run	Features Extracted with InfoGain		Features Extracted with InfoGain-MBA	
Leukemia_3c	1st	901	0.8571	88	0.86
	2nd	268	0.8095	13	0.81
	3rd	38	0.7143	3	0.81
Leukemia_4c	1st	787	0.7524	118	0.76
	2nd	396	0.8519	17	1
	3rd	62	0.671	4	0.67
Lung	1st	8118	0.9667	425	0.95
	2nd	1192	0.9707	26	1
	3rd	161	0.9647	1	0.98
Lymphoma	1st	2151	0.9474	852	1
	2nd	1461	0.9474	266	1
	3rd	403	0.9233	66	0.95
MLL	1st	5224	0.8095	996	0.81
	2nd	2246	0.8048	163	0.86
	3rd	171	0.9324	24	0.95
SRBCT	1st	667	0.7083	140	0.75
	2nd	334	0.8867	45	0.92
	3rd	45	0.9067	7	0.92

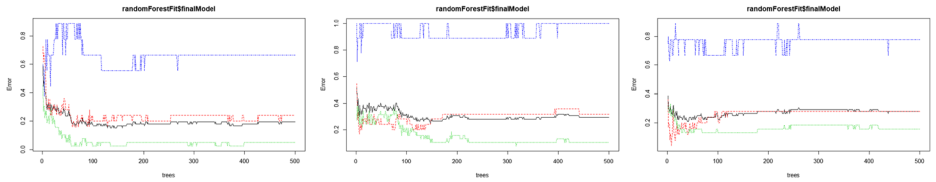


Fig. 18 Estimated error per class for Leukemia_3c Dataset. Note: The black colour in the graph indicate the average estimated error rate for all the class

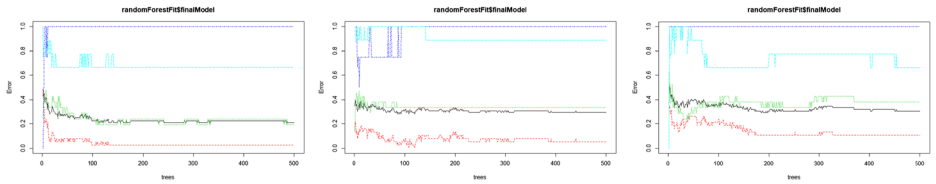


Fig. 19 Estimated error per class for Leukemia_4c Dataset. Note: The black colour in the graph indicate the average estimated error rate for all the class

estimated error decrease as the number of features selected is reduced. Therefore, this reveals that the best selection for Lung dataset is the 2nd 'Run' because the error increased at the 3rd 'Run'when the number of genes falls below 26 genes.

Figure 21 shows the estimated error rate per class for Lymphoma. The average estimated error obtained for Lymphoma was 6.06%, 7.58% and 3.03% respectively for all three 'Runs'. The value increases slightly from the 1st 'Run' to the 2rd 'Run' and later decrease in the 3rd 'Run'. Therefore, optimal feature selection for the Lymphoma dataset was obtained in the 3rd 'Run' when the number of genes is 66.

Figure 22 shows the estimated error rate per class for MLL. The average estimated error obtained for MLL was 15.28%, 6.94% and 12.50% respectively for all three 'Runs'. The value decreases from the 1st 'Run' to the 2rd 'Run', and later increase in the 3rd 'Run'. Therefore, optimal feature selection for the MLL dataset was obtained in the 2nd 'Run' when the number of genes selected is 163.

Figure 23 shows the estimated error rate per class for Ovarian. The average estimated error obtained for Ovarian were 0%, 0.79% and 0.40% respectively for all three 'Runs'. The value increase from the 1st 'Run' to the 2nd 'Run', and later decrease at the 3rd 'Run'. Therefore, optimal feature selection for the Ovarian dataset was obtained in the 1st 'Run' when the number of genes selected is 62.

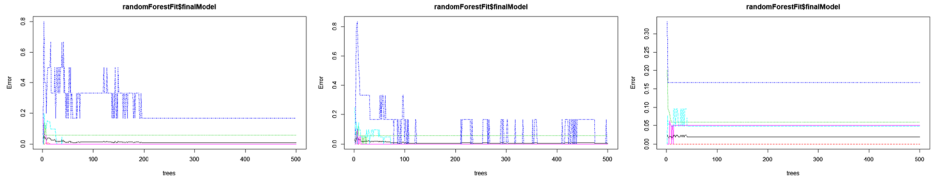


Fig. 20 Estimated error per class for lung dataset. Note: The black colour in the graph indicate the average estimated error rate for all the class

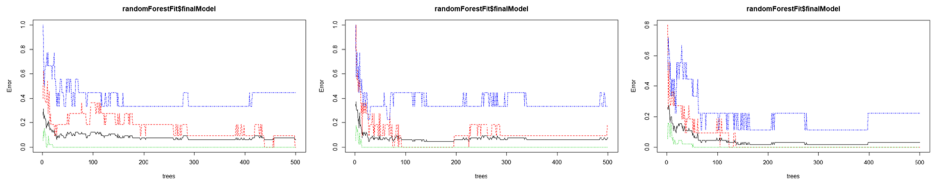


Fig. 21 Estimated error per class for lymphoma dataset. Note: The black colour in the graph indicate the average estimated error rate for all the class

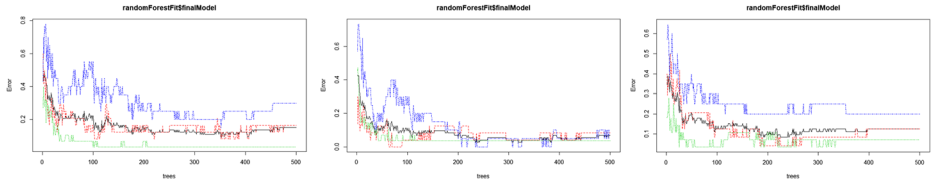


Fig. 22 Estimated error per class for MLL dataset. Note: The black colour in the graph indicate the average estimated error rate for all the class

Figure 24 shows the estimated error rate per class for SRBCT. The average estimated error obtained for SRBCT was 7.2%, 10.84% and 24.10% respectively for all three 'Runs'. The value increase from the 1st 'Run' to the 3rd 'Run' Therefore, optimal feature selection for the SRBCT dataset was obtained in the 1st 'Run' when the number of genes selected is 140.

Generally, only the Lymphoma dataset out of seven datasets used in this study did not has its estimated error increase as the number of features selected reduced when it reaches optimal feature selected.

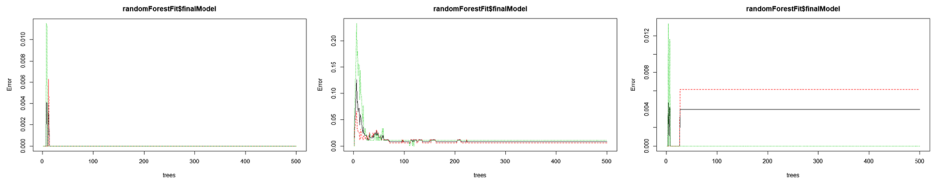


Fig. 23 Estimated error per class for ovarian dataset. Note: The black colour in the graph indicate the average estimated error rate for all the class

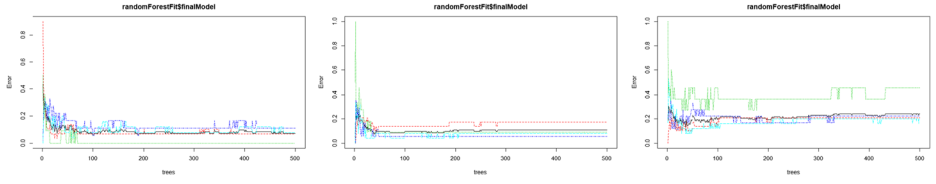


Fig. 24 Estimated error per class for SRBCT dataset. Note: The black colour in the graph indicate the average estimated error rate for all the class

5.6 Discussion

Section 5.2 presented the results of InfoGain-MBA for all the datasets that showed the number of features selected and the average time is taken to select the feature. The least time of 8.79 seconds was obtained in the Ovarian dataset while the highest time taken of 892.34 seconds was recorded in the MLL dataset. Figure 3 showed the graphical representation of the time analysis for all the datasets. In section 5.3, the results of classifiers were presented dataset by dataset. Four different classifiers (C4.5, DT, RF, and CART) were used in this research work with 3 'Run' for each dataset, because, 3 different outputs were available from the feature selection approach used (InfoGain-MBA). That is features from a threshold of ≥0.4, ≥0.6 and ≥ 0.8 respectively.

Table 4 presented the classifiers' results of the Leukemia_3c dataset where 100% accuracy was recorded in the 1st 'Run' of C4.5 and all the three 'Run' of RF with the highest value of Kappa statistic, sensitivity, specificity, positive predicted value, and negative predicted value. All other classifiers performed better than average and the least accuracy recorded was 76% in the 2nd 'Run' of C4.5, 1st and 2nd 'Run' of CART classifiers. Fig. 4, showed the average detection rate per class for all the classifiers. The least average detection rates per class of 25.4% were obtained in the 2nd 'Run' of C4.5 and 1st 'Run' of CART classifiers. There was a decrease in average detection rates per class from the 1st 'Run' of all the classifiers to the next 'Run' except in CART where it increased and remain constant in RF for all the three 'Run'. Also, Fig. 5 presented a balanced accuracy of all the classifiers for the Leukemia_3c dataset. The least average balanced accuracy of 63.9% was recorded in the 2nd 'Run' of the CART classifier and 100% were recorded in all the three 'Run' of RF and 1st 'Run' of C4.5 classifiers. Apart from the RF classifier that has the same values of average balanced accuracy for all the 'Run's, the remaining classifiers were decreased after the 1st 'Run' to 2nd and later increase in the 3rd 'Run', except in the DT classifier where it keeps on decreasing up to the 3rd 'Run'. This showed that the average detection rate and balanced accuracy for Leukemia_3c decreased when the genes selected is less than 88 in all the classifiers except in the RF classifier. Further investigation with estimated error per class of RF (Section 5.5 and Fig. 19) revealed the same scenario. Therefore, this study concluded that the best threshold for the Leukemia_3c data set is ≥0.4, that is, the top 40% features.

Table 5 presented the classifiers' results of the Leukemia_4c dataset where 100% accuracy was recorded in the 1st and 3rd 'Run' of C4.5, 2nd 'Run' of DT and all the three 'Run' of RF with the highest value of Kappa statistic, sensitivity, specificity, positive predicted value, and negative predicted value. All other classifiers performed better than average with the least accuracy value of 67% was recorded in the 3rd 'Run' of DT classifiers. Fig. 6, showed the average detection rate per class for all the classifiers. The least average detection rate per class of 16.67% was obtained in the 3rd 'Run' of DT classifiers. There were a varying increase and decrease in average detection rates per class from the 1st 'Run' of all the classifiers to the next 'Run' except RF where it remains constant in all the three 'Run'. Also, Fig. 7 presented a balanced accuracy of all the classifiers for the Leukemia_4c dataset. The least average balanced accuracy of 63.15% was recorded in the 3rd 'Run' of the DT classifier and 100% recorded for all the three 'Run' of RF and 1st and 3rd 'Run' of C4.5 classifiers. Apart from the RF classifier that has the same values of average balanced accuracy for all the three 'Run's, the remaining classifiers were decreasing after the 1st 'Run' to 2nd and later increase in the 3rd 'Run', except in the CART classifier where it keeps on decrease up to the 3rd 'Run'. This

showed that the average detection rate and balanced accuracy for Leukemia_4c decrease when the genes selected were less than 118 in all the classifiers except in the RF classifier. Further investigation with estimated error per class of RF (Section 5.5 and Fig. 20) revealed the same scenario of an increase in estimated error per class from 1st 'Run' to the 3rd 'Run'. Therefore, this study concluded that the best threshold for the Leukemia_4c dataset is ≥0.4, that is, the top 40% features.

Table 6 presented the classifiers' results of the Lung dataset where 100% accuracy was recorded in the three 'Run' of C4.5, 2nd 'Run' of DT, all the three 'Run' of RF and 2nd 'Run' of CART classifiers with the highest value of Kappa statistic, sensitivity, specificity, positive predicted value, and negative predicted value. The least accuracy value of 95% was recorded in the 1st 'Run' of DT classifiers. Fig. 8, showed an average detection rate per class for all the classifiers. The least average detection rate per class of 19% was obtained in the 1st 'Run' of DT classifiers and others have the same values of average detection rate per class. Also, Fig. 9 presented a balanced accuracy of all the classifiers for the Lung dataset. The least average balanced accuracy of 84.98% was recorded in the 1st 'Run' of DT classifier and 100% were recorded in all the three 'Run' of other classifiers except in 1st and 3rd 'Run' of the CART classifier where 97.33% and 98.4% were obtained respectively. This showed that the average detection rate and balanced accuracy for Lung increase when the genes selected are less than 425 in all the classifiers except in the RF classifier where it remains constant. Further investigation with estimated error per class of RF (Section 5.5 and Fig. 21) revealed that there is a decrease in estimated error per class from 1st 'Run' to the 2nd 'Run' and later increase in the 3rd 'Run'. Therefore, this study concluded that the best threshold for the Lung dataset is ≥0.6, that is, the top 60% features.

Table 7 presented the classifiers' results of the Lymphoma dataset where 100% accuracy was recorded in the three 'Run' of all the classifiers except in the 3rd 'Run' of DT and 1st 'Run' of CART where accuracy is 95% and 84% respectively. Better performance of Kappa statistic, sensitivity, specificity, positive predicted value, and negative predicted value were obtained in all the classifiers. Figure 10 showed the average detection rate per class for all the classifiers. The least average detection rate per class of 28.07% was obtained in the 1st 'Run' of CART classifiers and others have the same values of average detection rate per class. Also, Fig. 11 presented a balanced accuracy of all the classifiers for the Lymphoma dataset. The least average balanced accuracy of 80% was recorded in 1st 'Run' of the CART classifier and 100% were recorded in all the three 'Run' of other classifiers Further investigation with estimated error per class of RF (Section 5.5 and Fig. 22) revealed that there is a slight increase in estimated error per class from 1st 'Run' to the 2nd 'Run' and later decrease drastically to 3.03% in the 3rd 'Run'. Therefore, this study concluded that the best threshold for the Lymphoma dataset is ≥0.8, that is, the top 80% features.

Table 8 presented the classifiers' results of the MLL dataset where 100% accuracy was recorded in all the three 'Run' of C4.5 and RF classifiers. The least value of 81% was obtained in the 1st 'Run' of DT with good performance of Kappa statistic, sensitivity, specificity, positive predicted value and negative predicted value in all the classifiers. Figure 12 showed the average detection rate per class for all the classifiers. The least average detection rate per class of 26.99% was obtained in the 1st 'Run' of DT classifiers and others have the very close range values of average detection rate per class. Also, Fig. 15 presented a balanced accuracy of all the classifiers for the MLL dataset. The least average balanced accuracy of 87.75% was recorded in 1st 'Run' of DT

classifier and 100% were recorded in all the three 'Run' of C4.5 and RF classifiers, others have the value increase from 1st 'Run' to next 'Run' except in where there is a slight decrease in balanced accuracy from 1st 'Run' to 2nd 'Run' and then increased in the 3rd 'Run'. Further investigation with estimated error per class of RF (Section 5.5 and Fig. 23) revealed that there was a decrease in estimated error per class from 1st 'Run' to the 2nd 'Run' and later increased drastically to 12.50% in the 3rd 'Run'. Therefore, this study concluded that the best threshold for the MLL dataset is ≥0.6, that is, the top 60% features.

Table 9 presented the classifiers' results of the Ovarian dataset where all the classifiers yield 100% accuracy for all the three 'Run' with the highest value of Kappa statistic, sensitivity, specificity, positive predicted value, and negative predicted value. In Fig. 14, all the classifiers have equal average detection rate in 1st and 3rd 'Run', and the detection rate increased from 62.67% to 65.33% in the 1st 'Run' to 2nd 'Run' and later decreased back to 62.67% in 3rd 'Run' for all the classifiers. Also, Fig. 15 presented a balanced accuracy of all the classifiers for the Ovarian dataset. The balanced accuracy recorded was 100% for all the three 'Run' for each classifier. They showed that all the classifiers were able to predict both normal and cancer cells correctly. Further investigation with estimated error per class of RF (Section 5.5 and Fig. 24) revealed that the estimated error per class increased from 0% in the 1st 'Run' to 0.79% in the 2nd 'Run' and later decreased to 0.40% in the 3rd 'Run'. Therefore, this study concluded that the best threshold for the Ovarian data set is ≥0.4, that is, the top 40% features.

Table 10 presented the classifiers' results of the SRBCT dataset where 100% accuracy was recorded in all the three 'Run' of C4.5 and RF classifiers. The least value of 75% was obtained in the 1st 'Run' of DT and 3rd 'Run' of CART with good performance of Kappa statistic, sensitivity, specificity, positive predicted value and negative predicted value in all the classifiers. Fig. 16 showed an average detection rate per class for all the classifiers. The least average detection rate per class of 18.75% was obtained in the 1st 'Run' of DT and 3rd 'Run' of CART classifiers and others have the very close range values of average detection rate per class. Also, Fig. 17 presented a balanced accuracy of all the classifiers for the SRBCT dataset. The least average balanced accuracy of 80.1% was recorded in 3rd 'Run' of CART classifier and 100% were recorded in all the three 'Run' of C4.5 and RF classifiers, others have the value increase from 1st 'Run' to next 'Run' except in CART where there is increased in balanced accuracy from 1st 'Run' to 2nd 'Run' and then decreased in the 3rd 'Run'. Further investigation with estimated error per class of RF (Section 5.5 and Fig. 24) revealed that there was an increase in estimated error per class from 1st 'Run' to the 3rd 'Run'. Therefore, this study concluded that the best threshold for the SRBCT dataset is ≥0.4, that is, the top 40% features.

5.7 Comparison of proposed approach with other approaches in the literature

In this section, the proposed approach is compared with other methods in the literature. The comparison is based on accuracy and the number of genes selected, where the author(s) used more than one approach in their work. The best accuracy and number of features selected are considered for comparison. Table 12 summarizes the comparison between the approaches used in this study with other available works in the literature.

Table 12 Comparison of classification accuracy with other classifiers in the literature

Approach Used	Leukemia_3c	Leukemia_4c	Lung	Lymphoma	MLL	Ovarian	SRBCT
1. Proposed InfoGain-MBBA	100% (3)	100% (4)	100% (1)	100% (66)	100% (24)	100% (1)	100% (7)
2. CCFS [17]						98% (32.8)	
3. SVDD & CS-SVM [39]							95.3% (250); 75.9% (50)
4. Distributed Feature selection (DF) [7]						100% (10% of dataset); 98.81% (25% Dataset)	
5. CLACOFS [19]		86.38% (13.08)			98.95% (12.40)		
6. MOBBA-LS [12]							86% (6); 100% (6); 100% (6)
7. GALA [46]					93.96% (3)		99.34 (6)
8. mRMR-based ANN [57]			95% (100)	90% (100)	100% (100)	100% (100)	100% (500)
9. GBC [4]	77.78% (50); 100% (250)		89.56% (50); 100% (200)	93.73% (50); 100% (200)			62.65 (50); 100% (250)
10. CBFS with Cluster [10]	94.04% (40% Dataset)				93.70% (40% Dataset)		
11. Cuckoo Search (CS) [27]			100% (50)			100% (100)	
12. mRMR-BA [3]						100% (3.83)	

1. Leukemia_3c Dataset: There are several approaches in the literature that had explored the Leukemia_3c dataset, among them are, [4] used Genetic Bee Colony (GBC) algorithm for feature selection and [10] employed Correlation-based feature selection (CBFS) with k-NN Cluster to select the relevant features. InfoGain-MBBA was able to select 3 genes that yielded 100% classification accuracy, which outperformed other approaches used in the past.
2. Leukemia_4c Dataset: This dataset is not among the popular dataset frequently used by researchers may be due to its nature of multiclass that has 4 different classes. [19] employed cellular learning automata (CLA) optimized with ant colony method (ACO) tagged it as CLACOFS to select relevant features from the Leukemia_4c dataset. They were able to select an average number of 13.08 genes that yielded 86.38% classification accuracy which performed below the approach used in this research work.
3. Lung Dataset: There are several approaches available in the literature that had explored the Lung dataset, a few among them include [27] employed Cuckoo Search (CS) feature selection and [4] used GBC algorithm to select informative genes. Both the approaches yielded 100% accuracy with not less than 50 genes but InfoGain-MBBA had able to produce 100% with one gene selected.
4. Lymphoma Dataset: [4] used the GBC algorithm to select informative genes which produce 93.73% and 100% classification accuracy from 50 and 200 genes respectively. InfoGain-MBBA outperformed this approach by producing 100% classification accuracy with 66 genes selected.
5. MLL Dataset: few among the authors who employed the MLL dataset include [46]. They used the integration of Genetic Algorithm and Learning Automata (GALA) for feature selection to obtained 93.96% classification accuracy with 3 genes. Another approach that used the MLL dataset is [57] that employed mRMR-based ANN to yield 100% accuracy with 100 genes selected. CBFS with Cluster was employed by [10] to select 40% of features from the MLL dataset that yielded 93.70% classification accuracy. But InfoGain-MBBA yielded 100% classification accuracy with 24 genes.
6. Ovarian Dataset: [3] used mRMR-BA to select an average of 3.83 genes to yield 100% classification accuracy. [27] employed Cuckoo Search (CS) to select 100 genes that produce 100% classification accuracy. Also, DF was employed by [7] to produce 100% accuracy with 10% of the dataset and 98.81% accuracy with 25% of the dataset, [17] got 98% classification accuracy using CCFS to select an average of 32.8 genes.
7. SRBCT Dataset: [4] explored GBC to extract 50 and 100 genes from SRBCT that yield 62.65% and 100% classification accuracy respectively. [57] used mRMR-based ANN to select 500 genes that gave 100% classification accuracy. [46] used GALA to select 6 genes to produce 99.34% accuracy. [12] employed Multi-objective binary bat algorithm with specific local searches (MOBBA-LS) to 6 genes that yielded 100% classification accuracy. Also, [39] used both Support Vector Data Description (SVDD) and Cost-Sensitive SVM (CS-SVM) to select 250 genes that produced 95.3% accuracy and 50 genes that produced 75.9% classification accuracy respectively.

From Table 12 only the work of [12] performs better than the approach used in this work in the SRBCT dataset where the author used 6 genes to obtain 100% classification accuracy which is very close to the one obtained in this work with 7 genes.

6 Conclusion

The InfoGain-MBBA is a feature selection method focusing on selecting relevant and informative genes that will help in the timely diagnosis and prediction of cancer diseases without eliminating vital features from the original dataset.

The experimental results showed that the approach attains better performance improvement compared with the related approaches in the literature. It also requires less time to obtain accurate results. To validate the effectiveness of the InfoGain-MBBA, the evaluation was carried out using four different classifiers on seven different microarray datasets, which include both binary and multi-classes datasets. The efficiency of the approach was measured and compared. The results obtained showed that the approaches were good for the classification of cancer microarray data. For example, Random forest has 100% accuracy in all seven datasets used and the least accuracy obtained was 67% from the DT classifier using the Leuemia_4c dataset, which demonstrated that all the classifiers perform excellently with few genes selected. Furthermore, the method can be directly employed for biomarker discovery in large scale proteomics datasets.

7 Future work

While this research work is successful at attaining good results with improving accuracies, there were certain hindrances that build-up while carrying out this study. The computation time of MBA is a major concern, as one of the bio-inspired algorithms; usually takes a lot of time to implement on microarray data. This limitation is worth noting to improve future research in this domain. In future work, the proposed method can be used with datasets with very large samples to verify and extend the algorithm can cope. Likewise, an improvement to BBA can be done by hybridizing other bio-inspired algorithms such as to empower the searching process that results in enhancing the performance of BBA. Also, to reduce the computational time of the hybridization, a parallel or distributed version of the proposed approach can be implemented with MBA to reduce its computation time.

Appendix

Table 13 Computation time of MBB algorithm against BatSize for Leukemia_4c dataset with 118 features

BatSize	Time Taken
2	63.18
4	67.86
6	72.45
8	79.22
10	85.9
12	92.58
14	99.26
16	107.94

Table 14 Statistic of estimated error rate per class for all the datasets

(a): CNS

	d≥0.1 Confusion matrix		
	0	**1**	**Class.error**
0	38	1	0.0256
1	14	7	0.6667
	d≥0.2 Confusion matrix		
	0	**1**	**Class.error**
0	38	1	0.0256
1	1	20	0.0476
	d≥0.3 Confusion matrix		
	0	**1**	**Class.error**
0	38	1	0.0256
1	0	21	0

(b): Ovarian

	d≥0.4 Confusion matrix		
	Cancer	**Normal**	**Class.error**
Cancer	162	0	0
Normal	0	91	0
	d≥0.6 Confusion matrix		
	Cancer	**Normal**	**Class.error**
Cancer	161	1	0.0062
Normal	1	90	0.011
	d≥0.8 Confusion matrix		
	Cancer	**Normal**	**Class.error**
Cancer	161	1	0.0062
Normal	0	91	0

(c): Leukemia_3c

	d≥0.4 Confusion matrix			
	AML	**B cell**	**T cell**	**Class.error**
AML	19	6	0	0.24
B cell	2	36	0	0.0526
T cell	1	5	3	0.6667
	d≥0.6 Confusion matrix			
	AML	**B cell**	**T cell**	**Class.error**
AML	17	8	0	0.3200
B cell	4	34	0	0.1053
T cell	0	9	0	1.0000
	d≥0.8 Confusion matrix			
	AML	**B cell**	**T cell**	**Class.error**
AML	18	7	0	0.2800
B cell	4	32	2	0.1579
T cell	1	6	2	0.7778

(d): Leukemia_4c

	d≥0.4 Confusion matrix				
	B cell	**BM**	**PB**	**T cell**	**Class.error**
B cell	37	1	0	0	0.2632
BM	4	17	0	0	0.1905
PB	0	4	0	0	1.0000
T cell	5	1	0	3	0.6667
	d≥0.6 Confusion matrix				
	B cell	**BM**	**PB**	**T cell**	**Class.error**
B cell	36	1	0	1	0.0526
BM	7	14	0	0	0.3333
PB	0	4	0	0	1.0000
T cell	7	1	0	1	0.8889
	d≥0.8 Confusion matrix				
	B cell	**BM**	**PB**	**T cell**	**Class.error**
B cell	34	2	0	2	0.1053
BM	7	13	1	0	0.3810

Table 14 (continued)

PB	3	1	0	0	1.0000	
T cell	6	0	0	3	0.6667	
(e): Lung						
	d ≥ 0.4 Confusion matrix					
	1	**2**	**3**	**4**	**5**	**Class.error**
1	139	0	0	0	0	0.0000
2	1	16	0	0	0	0.0588
3	0	0	5	1	0	0.1667
4	0	0	0	21	0	0.0000
5	0	0	0	0	20	0.0000
	d ≥ 0.6 Confusion matrix					
	1	**2**	**3**	**4**	**5**	**Class.error**
1	139	0	0	0	0	0.0000
2	1	16	0	0	0	0.0588
3	0	0	6	0	0	0.0000
4	0	0	0	21	0	0.0000
5	0	0	0	0	20	0.0000
	d ≥ 0.8 Confusion matrix					
	1	**2**	**3**	**4**	**5**	**Class.error**
1	139	0	0	0	0	0.0000
2	1	16	0	0	0	0.0588
3	0	1	5	0	0	0.1667
4	0	0	1	20	0	0.0476
5	0	0	0	1	19	0.0500
(f): Lymphoma						
	d ≥ 0.4 Confusion matrix					
	CLL		**DLBCL**	**FL**	**Class.error**	
CLL	11		0	0	0.0000	
DLBCL	0		46	0	0.0000	
FL	0		4	5	0.4444	
	d ≥ 0.6 Confusion matrix					
	CLL		**DLBCL**	**FL**	**Class.error**	
CLL	9		2	0	0.1818	
DLBCL	0		46	0	0.0000	
FL	0		3	6	0.3333	
	d ≥ 0.8 Confusion matrix					
	CLL		**DLBCL**	**FL**	**Class.error**	
CLL	11		0	0	0.0000	
DLBCL	0		46	0	0.0000	
FL	0		2	7	0.2222	
(g): MLL						
	d ≥ 0.4 Confusion matrix					
	ALL		**AML**	**MLL**	**Class.error**	
ALL	20		0	4	0.1667	
AML	0		27	1	0.0357	
MLL	4		2	14	0.3000	
d ≥ 0.6 Confusion matrix						
	ALL		**AML**	**MLL**	**Class.error**	
ALL	22		0	2	0.0833	
AML	0		27	1	0.0357	
MLL	0		2	18	0.1000	
	d ≥ 0.8 Confusion matrix					
	ALL		**AML**	**MLL**	**Class.error**	
ALL	21		2	1	0.1250	
AML	2		26	0	0.0714	
MLL	3		1	16	0.2000	
(h): SRBCT						
	d ≥ 0.4 Confusion matrix					
	1	**2**	**3**	**4**	**Class.error**	

Table 14 (continued)

1	27	0	0	2	0.0690
2	0	11	0	0	0.0000
3	0	0	16	2	0.1111
4	1	0	1	23	0.0800
	d ≥ 0.6 Confusion matrix				
	1	**2**	**3**	**4**	**Class.error**
1	23	1	1	4	0.2069
2	2	6	2	1	0.4546
3	0	1	14	3	0.2222
4	1	0	4	20	0.2000
	d ≥ 0.8 Confusion matrix				
	1	**2**	**3**	**4**	**Class.error**
1	24	0	0	5	0.1724
2	1	10	0	0	0.0909
3	0	0	17	1	0.0556
4	0	1	1	23	0.0800

Acknowledgments This research has been financially supported in part by Tertiary Education Trust Fund (TETFUND) with Reference FUW/REG/T.5/VOL.1/T11. We also acknowledge the support of Ministry of Education of the People's Republic of China.

Funding Tertiary Education Trust Fund (TETFUND). Ministry of Education of the People's Republic of China.

Declarations

Conflicts of interest Not Applicable.

References

1. Abeer MM, Basma AM, El-Sayed ME, Abdel-Badeeh MS (2013) Applying a statistical technique for the discovery of differentially expressed genes in microarray data. Int Conf. on Recent Advances in Circuits, Systems, Telecommunications and Control, pp 220–227.
2. Aitkenhead MJ (2008) A co-evolving decision tree classification method. Expert Syst Appl 34:18–25. https://doi.org/10.1016/j.eswa.2006.08.008
3. Alomari OA, Khader AT, Al-Betar MA, Abualigah LM (2017) MRMR BA: a hybrid gene selection algorithm for cancer classification. J Theor Appl Inf Technol 95(12):2610–2618
4. Alshamlan HM, Badr GH, Alohali YA (2015) Genetic bee Colony (GBC) algorithm: a new gene selection method for microarray cancer classification. Comput Biol Chem 56:49–60. https://doi.org/10.1016/j.compbiolchem.2015.03.001
5. Bennet J, Ganaprakasam C, Kumar N (2015) A hybrid approach for gene selection and classification using support vector machine. Int Arab J Inf Technol 12(6A):695–700
6. Biau G (2012) Analysis of a random forests model. J Mach Learn Res 13:1063–1095
7. Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A (2015) Distributed feature selection: an application to microarray data classification. Appl Soft Comput J 30:136–150. https://doi.org/10.1016/j.asoc.2015.01.035
8. Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees, vol 432. Wadsworth International Group, Belmont
9. Cao J, Zhang L, Wang B, Li F, Yang J (2015) A fast gene selection method for multi-cancer classification using multiple support vector data description. J Biomed Inform 53:381–389
10. Chormunge S, Jena S (2018) Correlation based feature selection with clustering for high dimensional data. J Electr Syst Inf Technol 5(3):542–549
11. Chuang LY, Yang CH, Li JC (2011) A hybrid BPSO-CGA approach for gene selection and classification of microarray data. J Comput Biol 19:1–14

12. Dashtban M, Balafar M, Suravajhala P (2018) Gene selection for tumor classification using a novel bio-inspired multi-objective approach. Genomics 110(1):10–17. https://doi.org/10.1016/j.ygeno.2017.07.010
13. De Caigny A, Coussement K, De Bock KW (2018) A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. Eur J Oper Res 269(2):760–772. https://doi.org/10.1016/j.ejor.2018.02.009
14. Dev J, Dash SK, Dash S, Swain M (2012) A classification technique for microarray gene expression data using PSO-FLANN. Int J Comput Sci Eng 4(09):1534–1539
15. Doddipalli L, Rani KU (2016) Ensemble decision tree classifier for breast Cancer data. Int J Inf Technol Converg Serv 2(1):16–24. https://doi.org/10.5121/ijitcs.2012.2103
16. Doreswamy H, Salma UM (2016) A binary bat inspired algorithm for the classification of breast Cancer data. Int J Soft Comput Intell Appl 5(2/3):1–21
17. Ebrahimpour MK, Nezamabadi-Pour H, Eftekhari M (2018) CCFS: a cooperating coevolution technique for large scale feature selection on microarray datasets. Comput Biol Chem 73:171–178
18. El Akadi A, Amine A, El Ouardighi A, Aboutajdine D (2011) A two-stage gene selection scheme utilizing MRMR filter and GA wrapper. Knowl Inf Syst 26(3):487–500
19. Fatemeh VS, Sara M, Mohammad HM (2016) A hybrid gene selection approach for microarray data classification using cellular learning automata and ant Colony optimization. Genomics 107:231–238
20. Forsati R, Moayedikia A, Jensen R, Shamsfard M, Meybodi MR (2014) Enriched ant Colony optimization and its application in feature selection. Neurocomputing 142:354–371
21. Friedman J, Hastie T, Tibshirani R (2001) The elements of statistical learning, vol 1. Springer Series in Statistics, New York
22. Gandomi AH, Yang X-S, Alavi AH, Talatahari S (2013) Bat algorithm for constrained optimization tasks. Neural computing and applications. Neural Comput & Applic 22(6):1239–1255
23. Geetha R, Sivasubramanian S, Kaliappan M, Vimal S, Annamalai S (2019) Cervical cancer identification with synthetic minority oversampling technique and PCA analysis using random forest classifier. J Med Syst 43(9):286
24. Genuer R, Poggi J-M, Tuleau C (2008) Random Forests: some methodological insights. Retrieved January 18, 2020, from https://doi.org/10.48550/arXiv.0811.3619
25. Ghorai S, Mukherjee A, Sengupta S, Dutta PK (2010) Cancer classification from gene expression data by NPPC ensemble. IEEE/ACM Trans Comput Biol Bioinforma 8(3):659–671
26. Griffin DR, Webster FA, Michael CR (1960) The echolocation of flying insects by bats. Animal Behaviour, 8(3):141–154.
27. Gunavathi C, Premalatha K (2015) Cuckoo search optimisation for feature selection in cancer classification: a new approach. Int J Data Min Bioinform 13(3):248–265
28. Hall MA, Smith LA (1998) Practical feature subset selection for machine learning. In C. McDonald(Ed.), Computer Science 98 Proceedings of the 21st Australasian Computer Science Conference ACSC'98, Perth, 4-6 February 1998 (pp 181–191). Berlin: Springer.
29. Hambali MA, Gbolagade MD (2016) Ovarian cancer classification using hybrid synthetic minority over-sampling technique and neural network. J Adv Comput Res 7(4):109–124
30. Hambali M, Saheed Y, Oladele T, Gbolagade M (2019) ADABOOST ensemble algorithms for breast cancer classification. J Adv Comput Res 10(2):31–52. http://jacr.iausari.ac.ir/article_663924.html. Accessed 18 Jan 2020
31. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Morgan Kaufmann, 340:94104–3205.
32. Hira ZM, Gillies DF (2015) A review of feature selection and feature extraction methods applied on microarray data. Adv Bioinforma 2015:1–13. Retrieved January 18, 2020, from https://doi.org/10.1155/2015/198363
33. Kabir MM, Shahjahan M, Murase K (2012) A new hybrid ant colony optimization algorithm for feature selection. Expert Syst Appl 39(3):3747–3763
34. Liaw A, Wiener M (2002) Classification and regression by randomForest. R News 2(3):18–22
35. Lin S-W, Chen S-C (2012) Parameter determination and feature selection for C4. 5 algorithm using scatter search approach. Soft Comput 16(1):63–75
36. Lin W-Z, Fang J-A, Xiao X, Chou K-C (2011) iDNA-Prot: identification of DNA binding proteins using random forest with grey model. PLoS One 6(9):e24756
37. Lønning PE, Sørlie T, Børresen-Dale A-L (2005) Genomics in breast cancer—therapeutic implications. Nat Clin Pract Oncol 2(1):26–33
38. Mahmoud AM, Maher BA (2014) A hybrid reduction approach for enhancing cancer classification of microarray data. Int J Adv Res Artif Intell 3(10):1–10
39. Maldonado S, López J (2018) Dealing with high-dimensional class-imbalanced datasets: embedded feature selection for SVM classification. Appl Soft Comput 67:94–105

40. Martens H (2001) Reliable and relevant modelling of real world data: a personal account of the development of PLS regression. Chemom Intell Lab Syst 58(2):85–95
41. Martín-Merino M, De Las Rivas J (2009) Improving k-nn for human cancer classification using the gene expression profiles. In: International Symposium on Intelligent Data Analysis, pp. 107–118
42. Metzner W (1991) Echolocation behaviour in bats. Sci Prog Edinburgh 75(298):453–465. http://www.files/27/ADABOOSTEnsembleAlgorithmsforBreastCancerClassification.ris. Accessed 18 Jan 2020
43. Mishra S, Shaw K, Mishra D (2012) A new meta-heuristic bat inspired classification approach for microarray data. Procedia Technol 4:802–806
44. Mitchell TM (1997) Machine learning
45. Moody J, Darken CJ (1989) Fast learning in networks of locally-tuned processing units. Neural Comput 1(2):281–294
46. Motieghader H, Najafi A, Sadeghi B, Masoudi-Nejad A (2017) A hybrid gene selection algorithm for microarray cancer classification using genetic algorithm and learning automata. Informatics Med Unlocked 9:246–254
47. Nakamura RYM, Pereira LAM, Costa KA, Rodrigues D, Papa JP, Yang XS (2012) BBA: a binary bat algorithm for feature selection. Braz Symp Comput Graph Image Process 291–297. Retrieved January 18, 2020, from https://doi.org/10.1109/SIBGRAPI.2012.47
48. Narayanan A, Keedwell EC, Olsson B (2002) Artificial intelligence techniques for bioinformatics. Appl Bioinforma 1:191–222
49. Nguyen DV, Rocke DM (2002) Tumor classification by partial least squares using microarray gene expression data. Bioinformatics 18(1):39–50
50. Nguyen DV, Rocke DM (2002) Multi-class cancer classification via partial least squares with gene expression profiles. Bioinformatics 18(9):1216–1226
51. Panigrahi R, Borah S (2018) Rank allocation to J48 group of decision tree classifiers using binary and multiclass intrusion detection datasets. Procedia Comput Sci 132:323–332
52. Panigrahi PP, Singh TR (2013) Computational studies on Alzheimer's disease associated pathways and regulatory patterns using microarray gene expression and network data: revealed association with aging and other diseases. J Theor Biol 334:109–121
53. Pirooznia M, Yang JY, Yang MQ, Deng Y (2008) A comparative study of different machine learning methods on microarray gene expression data. BMC Genomics 9(S1):S13
54. Polat K, Güneş S (2009) A novel hybrid intelligent method based on C4. 5 decision tree classifier and one-against-all approach for multi-class classification problems. Expert Syst Appl 36(2):1587–1592
55. Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106
56. Rajeswari P, Reena GS (2011) Human liver cancer classification using microarray gene expression data. Int J Comput Appl 34(6):25–37
57. Rana MM, Ahmed K (2020) Feature selection and biomedical signal classification using minimum redundancy maximum relevance and artificial neural network. In: Proceedings of International Joint Conference on Computational Intelligence, pp. 207–214
58. Rangasamy M (2009) An efficient statistical model based classification algorithm for classifying cancer gene expression data with minimal gene subsets. Int J Cyber Soc Educ 2(2):51–66
59. Revathy N, Amalraj R (2011) Accurate cancer classification using expressions of very few genes. Int J Comput Appl 14(4):19–22
60. Rodrigues D, Pereira LAM, Nakamura RYM, Costa KAP, Yang XS, Souza AN, Papa JP (2014) A wrapper approach for feature selection based on bat algorithm and optimum-path forest. Expert Syst Appl 41(5): 2250–2258
61. Saeid MM, Nossair ZB, Saleh MA (2020) A microarray cancer classification technique based on discrete wavelet transform for data reduction and genetic algorithm for feature selection. In: 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), pp. 857–861
62. Sahu B, Mishra D (2012) A novel feature selection algorithm using particle swarm optimization for cancer microarray data. Procedia Eng 38:27–31
63. Schnitzler H-U, Kalko EKV (2001) 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
64. Seera M, Lim CP (2014) A hybrid intelligent system for medical data classification. Expert Syst Appl 41(5): 2239–2249
65. Selvaraj S, Natarajan J (2011) Microarray data analysis and mining tools. Bioinformation 6(3):95
66. Shafi ASM, Molla MMI, Jui JJ, Rahman MM (2020) Detection of colon cancer based on microarray dataset using machine learning as a feature selection and classification techniques. SN Appl Sci 2(7):1–8
67. Shreem SS, Abdullah S, Nazri MZA (2014) Hybridising harmony search with a Markov blanket for gene selection problems. Inf Sci (NY) 258:108–121

68. Sulaiman A, Akinbowale B, Ronke B, Moshood H (2015) Comparative analysis of decision tree algorithms for predicting undergraduate students' performance in computer programming. J Adv Sci Res Appl 2(20): 79–92
69. Suresh A, Udendhran R, Balamurgan M (2020) Hybridized neural network and decision tree based classifier for prognostic decision making in breast cancers. Soft Comput 24:7947–7953
70. Swathi S, Babu GA, Sendhilkumar R, Bhukya SN (2012) Performance of ART1 network in the detection of breast cancer. In: Proceedings of International Conference on Computer design and Engineering (ICCDE 2012), vol. 49, pp. 100–105.
71. Tang R, Fong S, Yang X-S, Deb S (2012) Integrating nature-inspired optimization algorithms to K-means clustering. In: Seventh International Conference on Digital Information Management (ICDIM 2012), pp. 116–123.
72. Veerabhadrappa, Rangarajan L (2010) Bi-level dimensionality reduction methods using feature selection and feature extraction. Int J Comput Appl 4(2):33–38
73. Vieira SM, Mendonça LF, Farinha GJ, Sousa JMC (2013) Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. Appl Soft Comput 13(8):3494–3504
74. Wang G, Guo L (2013) A novel hybrid bat algorithm with harmony search for global numerical optimization. J Appl Math vol. 2013. Retrieved January 18, 2020, from https://doi.org/10.1155/2013/696491
75. Wang L, Chu F, Xie W (2007) Accurate cancer classification using expressions of very few genes. IEEE/ACM Trans Comput Biol Bioinforma 4(1):40–53
76. Yang X-S (2011) Bat algorithm for multi-objective optimisation. Int J Bio-Inspired Comput 3(5):267–274. https://doi.org/10.1504/IJBIC.2011.042259
77. Yang X, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput 29(5):464–483. Retrieved January 18, 2020, from https://doi.org/10.1108/02644401211235834
78. Yang XS, He X (2013) Bat algorithm: literature review and applications. Int J Bio-Inspired Comput 5(3): 141. https://doi.org/10.1504/IJBIC.2013.055093
79. Zhu Z, Ong Y-S, Dash M (2007) Markov blanket-embedded genetic algorithm for gene selection. Pattern Recogn 40(11):3236–3248

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

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.