

**INFOGAIN-MODIFIED BINARY BAT ALGORITHM FOR FEATURE
SELECTION IN HIGH-DIMENSIONAL MICROARRAY CANCER
DATASETS**

BY

HAMBALI, Moshood A.

(13/52HE015)

B. Sc. (Al-Hikmah), M. Sc. (Ilorin)

**BEING A PhD. THESIS SUBMITTED TO THE DEPARTMENT OF
COMPUTER SCIENCE, FACULTY OF COMMUNICATION AND
INFORMATION SCIENCES, UNIVERSITY OF ILORIN,**

ILORIN, NIGERIA

**IN PARTIAL FULFILMENT OF THE REQUIRMENTS FOR AWARD OF
PhD. IN COMPUTER SCIENCE**

JUNE, 2021

CERTIFICATION

This is to certify that research reported in this thesis was carryout by HAMBALI, Moshood Abiola with Matriculation number 13/52HE015 in the Department of Computer Science, Faculty of Communication and Information Sciences, University of Ilorin, Ilorin, Nigeria for the award of Doctor of Philosophy (Ph.D.) Degree in Computer Science.

.....
Dr. Tinuke O. Oladele
(Supervisor)

.....
DATE

.....
Dr. A.O. Ameen
(Postgraduate Coordinator)

.....
DATE

.....
Dr. Oluwakemi C. Abikoye
(Head of Department)

.....
DATE

.....
Prof. W. B. Yahaya
(Internal/External Examiner)

.....
DATE

.....
(External Examiner)

.....
DATE

DEDICATION

This research work is dedicated to **Almighty Allah** and the **suffering people** of the world.

ACKNOWLEDGEMENT

All thanks, praises and adoration belong to Almighty Allah, who created and sustains me up till the moment. If not for His mercy, I wouldn't have got to this level. I thank You for your mercy and blessing upon me.

The Prophet (S.A.W.) says whoever does not appreciate peoples' kindness will not appreciate the blessings of Allah. With this, I am very much grateful to Dr. Tinuke O. Oladele for her mentorship, supervision, and motherly advice all the time. The impact you make in my life can never be forgotten. May Almighty God bless you in abundance.

Moreover, I am greatly indebted to Dr K.S. Adewole for his advice and total support during the course of this research; I say Jaza Kumullahu khaira to you. Also, I will not forget the contribution of our able Dean Prof. R.G. Jimoh for your assistance and advice, our able HOD Dr. (Mrs) O.C. Abikoye, Dr. R.O. Oladele, Dr. A.O. Ameen, , Dr. A.O. Babatunde, Dr. D.R. Aremu, Dr. A.O. Bajeh, Dr. (Mrs) M.A. Mabayoje, Dr. A.R. Ajiboye, Dr. (Mrs) A.G. Akintola, Dr. (Mrs) F.E. Usman-Hamzah, Dr. (Mrs) G.B. Balogun, Dr. M. Abdulraheem, Dr. J.B. Awotunde, Dr. I.D. Oladipupo, Dr. A.W. Asaju-Gbolagade, Mr P.O. Sadiku, Mrs S.A. Saliu, Mr. A.O. Balogun, Mr, A.M. Balogun, Mr H.A. Majid, and the entire staff of Computer Science Department, University of Ilorin, for making this programme a successful one.

I also express my profound gratitude to my parents Alhaji and Alhaja Hambali Alabere for their enormous contributions to my education. May Almighty Allah forgive the dead one and let the living one live long to reap the fruits of your efforts. I acknowledge the effort of my siblings, the entire family of Alabere and the brethren in faith, I said thank you all.

Part of this research study was sponsored by Tertiary Education Trust Fund (TETFUND). I am very grateful for the opportunity given me to be able to carry out this research work. Also, to the management of Federal University Wukari, Taraba State, my colleagues in Computer Science Department Federal University Wukari, Prof. S. Amoo, Dr. Ajayi, entire Muslim Ummah of FUW and my PhD colleagues in Computer Science Dept., University of Ilorin, most especially Mr Alarape Moshood; millions of thanks cannot express my gratitude, thank you all for your concern and contribution in my life.

Also, to my Love Zulfah, Adunni and the children, for their sacrifice. Had it been you did not bear it with me, I wouldn't have achieved the little am able to do. Thank you so much; may Almighty Allah make it easy for me to be able to pay back the price of your sacrifice.

ABSTRACT

Microarray data (MD) have a lot of application areas in the health sector that include diseases prediction and diagnosis. The enormous quantity of raw gene in MD usually results in analytical and computational complexities and poor performance of classifiers' algorithm. To achieve a better accuracy of cancer classification with the complete set of genes from MD remains a great challenge, due to the high dimensionality of features and presence of noise in MD. Hence, there is a need for the design of Modified Binary Bat (MBB) algorithm which used a global search technique with some additional mechanisms, targeting at enhancing the results of Bat Algorithm (BA) - based algorithm for feature selection (FS) and also to be able to cater for multiclass datasets. The study aimed at developing InfoGain-modified Binary Bat (InfoGain-MBB) algorithm FS model for selecting informative features from high dimensional Microarray cancer datasets. The objectives of the study were to: (i) analyse series of filter algorithms (FA) to select relevant features from datasets and compare their performances; (ii) develop InfoGain-MBB feature subset selection model; (iii) implement the developed model; and (iv) evaluate the performance of the developed model and compare it with other existing models.

Series of FA were analyzed using Chi-squared (Chi), Information gain (InfoGain) and Gain ratio (GR) algorithm to reduce the number of genes by selecting the top-ranked relevant genes from eight (8) difference MD obtained from <http://csse.szu.edu.cn/staff/zhuzx/Datasets.html> by setting up threshold values; and their results were compared using number of genes selected, time taken and classifiers' performance. The best FA (InfoGain) was combined with MBB algorithm to form InfoGain-MBB algorithm which was used to select the subset of highly relevant and informative genes. All the approaches were implemented using RStudio. The performance of the approach was evaluated with four different classifiers – C4.5, Decision Tree (DT), Random Forest (RF) and Classification and Regression Algorithm (CART). Further investigation was also conducted to determine the optimal threshold for each dataset used.

The findings of this study were that:

- i. FA showed that InfoGain performed better by selecting few numbers of genes (5 genes) with shortest time taken to extract features (24.07 sec.) in 6 out of 8 datasets used;
- ii. MBB Algorithm was developed with $BatSize=4$, $F_{min}=20$, $F_{max}=50$, $R=0.5$ and $A=0.9$ parameters;
- iii. InfoGain-MBB Algorithm was able to select few informative genes as low as 1 from CNS, Ovarian and Lung, 3 from Leukemia_3c, 4 from Leukemia_4c, 66 from Lymphoma, 24 from MLL and 7 from SRBCT dataset;
- iv. 100% Classification Accuracy (CA) and Balanced Accuracy (BA) with detection rate of 61% were obtained in all classifiers with CNS and Ovarian datasets;
- v. RF recorded 100% CA and BA for all datasets;
- vi. C4.5 yielded 100% CA and BA in Leukemia_3c, Leukemia_4c, Lung, Lymphoma, MLL and SRBCT datasets.

The study concluded that the InfoGain-MBB Algorithm was an effective FS technique for MD that outperformed the existing models. The study recommended that InfoGain-MBB should be used as part of the "standard tool-box" methods for gene selection in MD.

TABLE OF CONTENTS

TITLE PAGE	i
CERTIFICATION	ii
DEDICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	vi
Table of Contents	vii
List of Figures	x
List of Tables	xiii
List of Algorithms	viii
List of Abbreviations	xiii
CHAPTER ONE	1
INTRODUCTION	1
1.1 Background to the Study	1
1.2 Statement of the Problem	5
1.3 Aim and Objectives	8
1.4 Significance of the Study	8
1.5 Scope of the Study	9
1.6 Definition of Operational Terms	9
CHAPTER TWO: LITERATURE REVIEW	13
2.1 Microarray Technology	13
2.1.1 DNA Microarray	14
2.1.2 Microarray Experiment	15
2.2 Microarray Data	16
2.3 Microarray Data Analysis	17
2.4 Cluster Analysis	18
2.5 Classification	19
2.6 Feature Selection Techniques	20
2.6.1 Label Status	22
2.6.2 Search Strategy	28
2.7 Algorithms Used in this Research Work	43
2.7.1 Information Gain	43

2.7.2	Gain Ratio	44
2.7.3	Chi-Squared Test	45
2.7.4	Bat Algorithm	46
2.7.5	Classification Algorithms	49
2.7.6	Decision Tree Classifier	50
2.7.7	Classification and Regression Tree	54
2.7.8	C4.5	55
2.7.9	Random Forest	55
2.8	Related Work	57
CHAPTER THREE: RESEARCH METHODOLOGY		74
3.1	Conceptual Framework	74
3.2	Dataset	76
3.2.1	Data Pre-processing	77
3.3	Filter Stage	78
3.3.1	Threshold Values	78
3.4	Subset Selection Stage	79
3.4.1	Binary Bat Algorithm	79
3.4.2	Description of the Modified Binary Bat Algorithm	80
3.5	Distribution of Training and Test Datasets	83
3.5.1	Cross Validation	83
3.6	Performance Metrics used for the Classification Algorithms	83
CHAPTER FOUR: RESULTS AND DISCUSSION		87
4.1	Experiment Setup Environment	87
4.2	Results	87
4.3	Filter-Based Algorithm Results	87
4.4	InfoGain-MBB Algorithm Results	94
4.5	Classification Results	98
4.5.1	Comparison of Original Approach and Modified BBA with InfoGain	98
4.5.2	Performance Evaluation of InfoGain-MBB Algorithm	99
4.6	Error Rate for Random Forest	118
4.7	Discussion	123
4.7.1	Comparing Studies Algorithms	123

4.7.2	Comparison of Proposed Approach with other Existing Approaches in the Literature	130
CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATIONS		137
5.1	Summary	137
5.2	Conclusion	138
5.3	Recommendations	138
5.4	Contribution to Knowledge	139
5.5	Limitation of the Study	140
5.6	Future Work	140
REFERENCES		141
APPENDIX I		163
APPENDIX II		167
APPENDIX III		167
APPENDIX IV		172
APPENDIX V		181
APPENDIX VI		1943

LIST OF FIGURES

Figure 2.1: Generic Procedure for Acquiring the Gene Expression Data	15
Figure 2.2: Steps of a Microarray Experiment	16
Figure 2.3: Feature Selection Category	25
Figure 3.1: Framework of the Proposed Method	75
Figure 3.2: Confusion Matrix	84
Figure 4.1: Average Time Taken to Extract Features for All Datasets using Chi-Squared Algorithm	92
Figure 4.2: Average Time Taken to Extract Features for All Datasets using Gain Ratio Algorithm	92
Figure 4.3: Average Time Taken to Extract Features for All Datasets using Information Gain Algorithm	93
Figure 4.4: Computation Time of MBB Algorithm against BatSize for Leukemia_4c Dataset	95
Figure 4.5: Average Time Taken to Extract Features for All Datasets using InfoGain-MBB Algorithm	98
Figure 4.6: Detection Rate for CNS Dataset for all the Classifiers	101
Figure 4.7: Balanced Accuracy for CNS Dataset for all the Classifiers	101
Figure 4.8: Detection Rate for Leukemia_3c Dataset for all the Classifiers	103
Figure 4.9: Balanced Accuracy for Leukemia_3c Dataset for all the Classifiers	104
Figure 4.10: Detection Rate for Leukemia_4c Dataset for all the Classifiers	106
Figure 4.11: Balanced Accuracy for Leukemia_4c Dataset for all the Classifiers	106

Figure 4.12: Detection Rate for Lung Dataset for all the Classifiers	108
Figure 4.13: Balanced Accuracy for Lung Dataset for all the Classifiers	109
Figure 4.14: Detection Rate for Lymphoma Dataset for all the Classifiers	111
Figure 4.15: Balanced Accuracy for Lymphoma Dataset for all the Classifiers	111
Figure 4.16: Detection Rate for MLL Dataset for all the Classifiers	113
Figure 4.17: Balanced Accuracy for MLL Dataset for all the Classifiers	113
Figure 4.18: Detection Rate for Ovarian Dataset for all the Classifiers	115
Figure 4.19: Balanced Accuracy for Ovarian Dataset for all the Classifiers	115
Figure 4.20: Detection Rate for SRBCT Dataset for all the Classifiers	117
Figure 4.21: Balanced Accuracy for SRBCT Dataset for all the Classifiers	117
Figure 4.22: Estimated Error Per Class for CNS Dataset	119
Figure 4.23: Estimated Error Per Class for Leukemia_3c Dataset	119
Figure 4.24: Estimated Error per class for Leukemia_4c Dataset	119
Figure 4.25: Estimated Error Per Class for Lung Dataset	120
Figure 4.26: Estimated Error Per Class for Lymphoma Dataset	121
Figure 4.27: Estimated Error Per Class for MLL Dataset	121
Figure 4.28: Estimated Error Per Class for Ovarian Dataset	122
Figure 4.29: Estimated Error Per Class for SRBCT Dataset	123

LIST OF TABLES

Table 2.1:	Microarray Data Matrix Format	17
Table 2.2:	Summary of Other Filter Techniques Used in Microarray Data	35
Table 2.3	Summary of Wrapper Techniques Used in Microarray	38
Table 2.4:	Summary of Embedded Techniques Used in Microarray Data	41
Table 2.5:	Summary of Other Hybrid Techniques for Feature Selection Used in Microarray Data	42
Table 2.6:	Summary of Other Related Works	65
Table 3.1:	Dataset Description	77
Table 4.1:	Result of Filter Algorithms for CNS Dataset	88
Table 4.2:	Result of Filter Algorithms for Leukemia_3c Dataset	88
Table 4.3:	Result of Filter Algorithms for Leukemia_4c Dataset	89
Table 4.4:	Result of Filter Algorithms for Lung Dataset	89
Table 4.5:	Result of Filter Algorithms for Lymphoma Dataset	90
Table 4.6:	Result of Filter Algorithms for MLL Dataset	90
Table 4.7:	Result of Filter Algorithms for Ovarian Dataset	91
Table 4.8:	Result of Filter Algorithms for SRBCT Dataset	91
Table 4.9:	Preliminary Evaluation with Decision Tree Algorithm	94
Table 4.10:	Parameters setting for MBB Algorithm	95
Table 4.11:	Result of InfoGain-MBB Algorithm for All Datasets	96
Table 4.12:	Comparison of BBA and Modified BBA	99
Table 4.13:	Result of Classifiers for CNS Dataset	100
Table 4.14:	Result of Classifiers for Leukemia_3c Dataset	102
Table 4.15:	Result of Classifiers for Leukemia_4c Dataset	105
Table 4.16:	Result of Classifiers for Lung Dataset	107
Table 4.17:	Result of Classifiers for Lymphoma Dataset	110
Table 4.18:	Result of Classifiers for MLL Dataset	112
Table 4.19:	Result of Classifiers for Ovarian Dataset	114
Table 4.20:	Result of Classifiers for SRBCT Dataset	116
Table 4.21:	Comparison of Classification Accuracy with Other Classifiers in the Literature	119

LIST OF ALGORITHMS

Algorithm 2.1: Pseudocode of Information Gain	44
Algorithm 2.2: Pseudocode of Gain Ratio	45
Algorithm 2.3: Pseudocode of Bat Algorithm	49
Algorithm 2.4: Pseudocode of Decision Tree	54
Algorithm 2.5: Pseudocode of Random Forest Classification	56
Algorithm 3.1: Pseudocode of Modified Binary Bat Algorithm	82

List of Abbreviations

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ADSRPCL-SVM	Distance Sensitive Rival Penalized Competitive Learning – Support Vector Machine
ALL	Acute Lymphoblastic Leukemia
AML	Acute Myeloid Leukemia
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
ART	Adaptive Resonance Theory
BA	Bat Algorithm
BAHSIC	Backward Elimination using HSIC
BBA	Binary Bat Algorithm
BCO	Bee Colony Optimization
BIRS	Best Incremental Ranked Subset
BL	Burkitt’s Lymphoma
BM	Bone Marrow
BPN	Backpropagation network
CAD	Computer-Aided Diagnosis
CART	Classification and Regression Tree
CBFS	Correlation-Based Feature Selection
Coevolving Feature Selection	CCFS Cooperative
cDNA	Complementary DNA
CFS	Correlation Feature Selection
CFS-TGA Algorithm	Correlation-based Feature Selection (CFS) and the Taguchi-Genetic
CLA	Cellular Learning Automata
CLL	Chronic Lymphocytic Leukemia
CNS	Central Nervous System

CS	Cuckoo Search
CS	Class Separability
CS-SVM	Cost-Sensitive SVM
DEGs	Differentially Expressed Genes
DF	Distributed Feature selection
DLBCL	Diffuse Large B-Cell Lymphoma
DNA	Deoxyribonucleic Acid
DT	Decision Tree
EF	Ensemble of Filters
EFA	Entropic Filtering Algorithm
EUFS	Embedded Unsupervised Feature Selection
EWS	Ewing's family of tumours
FAST	Fast Clustering-Based Feature Selection Algorithm
FCBF	Fast Correlation-Based Filter
FL	Follicular Lymphoma
FLANN	Functional Link Artificial Neural Network
FN	False Negatives
FNN	Fuzzy Neural Network
FNN	Feedforward Neural Network
FOIL	First Order Inductive Learner
FP	False Positives
FS	Feature Selection
GA	Genetic Algorithm
GADP	Genetic Algorithm with Dynamic Parameter setting
GA-KDE	Genetic Algorithm-Kernel Density Estimation
GALA	Genetic Algorithm and Learning Automata
GBC	Genetic Bee Colony
GC	Counting Grid

GR	Gain Ratio
HS	Harmonic Search
HSA	Harmony Search Algorithm
HSIC	Hilbert-Schmidt Independence Criterion
InfoGain	Information Gain
JELSR	Joint Embedding Learning and Sparse Regression
KFDA	kernel Fisher discriminant analysis
KNN	K Nearest Neighbours
LDA	Linear Discriminant Analysis
MA	Memetic Algorithm
MASSIVE	Matrix of Average Sub-Subset Information for Variable Elimination
MB	Markov Blanket
MBB	Modified Binary Bat
MCFS	Multi-Cluster Feature Selection
MCF-RFE	Multi-Criterion Fusion-based Recursive Feature Elimination
MFMW	Multiple-Filter-Multiple-Wrapper
MFS	Multi-task Feature Selection Filter
ML	Machine Learning
MLL	Mixed-Lineage Leukemia
MMC	Maximum Margin Criterion
MOBBA-LS	Multi-Objective Binary Bat Algorithm with Specific Local Searches
mRMR	minimum Redundancy Maximum Relevance
mRNA	Messenger Ribonucleic Acid
MRSF	Minimum Redundancy Spectral Feature Selection
MT	Microarray Technology
MTL	Multi-Task Learning
MWMR	Maximum Weight and Minimum Redundancy
NB	Naïve Bayes

NB	Neuroblastoma
NDFS	Nonnegative Discriminative Feature Selection
NP	Non-Deterministic Polynomial
NPPC	Nonparallel Plane Proximal Classifier
PAC	Probably Approximately Correct
PB	Peripheral Blood
PCA	Principal Component Analysis
PcaScor	PCA Score
PLS	Partial Least Squares
PNN	Probabilistic Neural Network
PSO	Particle Swarm Optimization
RBF	Radial Basis Function
RF	Random Forest
RFE	Recursive Feature Elimination
RFS	Relevance Feature Selection
RFS	Robust Feature Selection
RMS	Rhabdomyosarcoma
R-m-GA	ReliefF mRMR and GA
RNA	Ribonucleic Acid
RRFS	Relevance Redundancy Feature Selection
SA	Simulated Annealing
SAGE	Serial Analysis of Gene Expression
SFS	Successive Feature Selection
SMOTE	Synthetic Minority Oversampling Technique
SPEC	Spectral Feature Selection
SPFS	Similarity Preserving Feature Selection
SRBCT	Small Round Blue Cell Tumors
SRF	Stratified Random Forest

SVDD	Support Vector Data Description
SVM	Support Vector Machines
SVM-OAA	SVM-One-Against-All
TN	True Negatives
TP	True Positives
TRCFS	Trace Ratio Criterion for Feature Selection
UDFS	Unsupervised Discriminative Feature Selection
WHO	World health Organisation

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

Cancer, seldom referred to as malignant neoplasm, is one of the complex diseases that occur as a result of some cells manifesting certain traits of unrestrained development and perhaps later invades other sections of the body (American Cancer Society, 2017). According to WHO Fact sheet, about 14 million new incidences of cancer was recorded in 2012, which is one of the prominent causes of morbidity and mortality globally (WHO, 2018). In the next 2 decades, the anticipated new cases are likely to rise to about 70%. Globally, virtually 1 out of 6 deaths reported is as a result of cancer disease, about 8.8 million deaths documented in 2015 which place cancer as the second prominent cause of death in the world (WHO, 2018).

In Nigeria, WHO statistics revealed that, annually over 100,000 Nigerians are identified to be living with cancer, and roughly 80,000 die as a result. On the average, 240 Nigerians pass away every day from cancer diseases. The Nigerian cancer rate is one of the worst in the world as the death ratio is documented to be 4 out of 5 (Ogundipe, 2017).

However, the Cancer mortality rate can be cut down if diagnosed and treated early. Medical data mining is one of the branches of data mining that help in analysing, extracting, transforming, interpreting and visualizing medical records stored in repositories. Medical data mining is not only interesting but also challenging, because diagnosis and prediction of diseases are not the issues of accuracy alone but it also a matter of life and death. A wrong classification or prediction can be disastrous to the

life of patients and their relatives. Hence, medical data mining is considered sometimes as an expert system that employs machine learning to make decisions that help experts to easily and quickly diagnose and predict diseases (Doreswamy & UmmeSalma, 2016).

Microarray data are mostly employed in cancer researches where early diagnosis of cancer disease is very important in determining the nature of treatment and its survivability (Selvaraj & Natarajan, 2011). Microarray Technology (MT) assists biologists in examining the activity of ten thousands of genes in a single experiment and gets important information about cell's functionality. This specific information can be explored for diagnosis of many diseases such as Alzheimer's disease (Panigrahi & Singh, 2013), diabetes diseases and cancer diseases (Alomari et al., 2017; Hira & Gillies, 2015; Shekar & Dagneu, 2020). The gene expression data can be produced from this technology, which is highly germane for cancer classification and prediction. Though, 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. Large number of genes (features) versus small size of sample, and presence of redundancy in data expression are the major reasons that usually result into poor diagnosis and classification, in machine learning and data mining approaches employed in the field of medicine (Bennet, Ganaprakasam, & Kumar, 2015; Veerabhadrappe & Rangarajan, 2010). In recent times, researchers had employed computational intelligence algorithms to investigate the most informative genes that contribute to cancer diagnosis (Alomari et al., 2017).

Numerous studies have revealed that most genes present in DNA (Deoxyribonucleic Acid) Microarray datasets are not pertinent in the accurate diagnosis of

different diseases (Alomari et al., 2017; Hambali, Saheed, Oladele, & Gbolagade, 2019; Hambali & Gbolagade, 2016; Liang, Ma, Yang, Wang, & Ma, 2018; Saeid, Nossair & Saleh, 2020; Saqib, Qamar, Khan, & Aslam, 2020). To circumvent the problem 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 (Alomari et al., 2017; Hambali et al., 2019; Hambali & Gbolagade, 2016).

Therefore, dimensionality reduction is one of the exciting research areas in the domain of data mining, pattern recognition, machine learning, and statistics. The main drive of dimensionality reduction is to augment the classification accuracy performance of classification algorithm through the removal of redundancy and irrelevant features in the Microarray dataset. Dimensionality reduction can be accomplished in many ways. But, the choice of selecting appropriate dimensionality reduction method is determined by the domain application and the peculiarity of the dataset. Feature selection methods are categorized into four (4) approaches, namely; filter, wrapper, embedded and hybrid approaches (Veerabhadrapa & Rangarajan, 2010).

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 with 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 method due to their speed. A mutual drawback of the two approaches is that they both ignore the interaction

between the classifier and the dependence among features, and this result to diverse classification accuracy when features selected are subjected to different classification algorithms. While embedded approaches, 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 cost compared to wrapper methods (Veerabhadrapa & Rangarajan, 2010).

Meta-heuristic algorithms employ updating optimum decision function known as the cost function to design non-parametric classifiers. These algorithms have the capability of providing the best solution even to a complex problem. Also, they are free from the problem of local minima or maxima (Yang, 2010a). These two main qualities make meta-heuristic algorithms to produce high accuracy and robust solutions within the shortest period of time. The nature of the problem at hand and the solution required determine the choice of nature-inspired algorithm to employ. Bat algorithm (BA) (Yang, 2010b) is one of the novel meta-heuristic and nature-inspired algorithms which have numerous applications. BA has an additional benefit of combining the idea of the population-based algorithm with local search. This combination is the major advantage of the Meta-heuristic algorithm that leads to better result due to its capability of local intensive exploitation and global diverse exploration. Furthermore, it has been successfully employed to accomplish variety of tasks such as Classification (Sahu & Mishra, 2012), Clustering (Komarasamy & Wahi, 2012), Image processing (Akhtar, Ahmad, & Abdel-Rahman, 2012), optimization problems like Continuous Optimization (Yang, 2010b; Yang & Gandomi, 2012), combinatorial optimization and scheduling (Musikapun & Pongcharoen, 2012; Ramesh, Mohan & Reddy, 2013), Inverse problem and parameter estimation and so on (Dashtbana,

Mohammadali & Prashanth, 2018; Doreswamy & UmmeSalma, 2016; Gunavathi & Premalatha, 2015; Lin, Chou, Yang & Tsai, 2012; Wanderley, Gardeux, Natowicz & de Pádua Braga, 2013; Yang & Gandomi, 2012; Zhang, Gong, Hu & Zhang, 2015).

BA imitates Bat's manner of searching for its target (food) based upon echolocation technique. Bat uses echolocation to vary its velocity and direction based on the reflection of sound that strikes back from the target (obstacle). Its velocity is updated randomly in order to get to its prey within the shortest time. Earlier research shows that BA performs better than Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in proffering solution to the unrestricted optimization problems (Doreswamy & UmmeSalma, 2016). Due to the strength of search capabilities of BA, in this study, the researcher has applied Modified Binary Bat (MBB) Algorithm for selection of informative genes from Microarray datasets. The continuous nature of BA is changed to binary nature. Furthermore, when MBB Algorithm is applied to Microarray datasets, it is expected to encounter some computational problems, like other evolutionary algorithms. To prevent these challenges and further enhance the performance of MBB Algorithm, filter methods (Chi-Squared, Information Gain, Gain Ratio) are employed separately as a pre-processing technique to reduce the dimensionality of Microarray datasets before passing it to the classifier algorithms.

1.2 Statement of the Problem

Microarray data are mostly employed in cancer researches where the ultimate goal is to prevent the outbreak and advancement of cancer, proffer early diagnosis of the disease and determining the nature of treatment and its survivability. Cancer is one of the complex diseases and most devastating health conditions in the world. Cancer is like other diseases

that influenced by genes, therefore, detecting the genetic profile and landscape of this disease is of greatest importance and challenging (Bozek, Lengauer, Sierra, Kaiser & Domingues, 2013; Mohammadi, Noghabi, Hodtani & Mashhadi, 2016; Saqib et al., 2020; Shafi, Molla, Jui & Rahman, 2020). Recent improvements in high-throughput technologies such as Microarray, Mass Spectrometry (MS) and Next Generation Sequencing (NGS) have provided breakthrough for researchers to recognise genetic origins of diseases (Agarwal & Owzar, 2014; Li, Peng, Chen, Lü, Zhang & Lai, 2012; Taylor, Steinberg, Andrews & Webber, 2015). These innovative high-throughput representations need suitable computational approaches in order to derive and acquire knowledge from them (Mohammadi et al., 2016; Saeid et al., 2020).

Though these new innovations are noticeably useful and beneficial, but they are suffering from the problem of “Curse of Dimensionality” if they have few number of samples with the high number of observed genes (dimensions) (Alshamlan, Badr & Alohal, 2015; Dashtban & Balafar, 2017; Song, Bedo, Borgwardt, Gretton & Smola, 2007). To achieve satisfactory cancer classification accuracy with the complete set of genes remains a great challenge, due to the high dimensions, small sample size, and the presence of noise in gene expression data (Bolón-Canedo, Sánchez-Marño & Alonso-Betanzos, 2015; Dash, 2017; Hambali et al., 2019). Feature reduction is critical and sensitive in the classification task. One of the major drawbacks in cancer study is recognizing informative genes among thousands of others in microarray data.

Furthermore, feature selection is typically useful in reducing the computation time and memory complexities, which are always been the challenges with microarray data. Besides, the high complexity on the memory space or time as a result of high dimension,

noise effect and outliers but it also has adverse impacts on the performance of the classification algorithms (Elhamifar & Vidal, 2013; Hambali et al., 2019; Zhang, Nie, Li & Wei, 2019).

Feature selection is also regarded as a non-deterministic polynomial (NP)-hard problem since the growth of search space increase exponentially with the number of genes 2^N , where N signifies the number of features (genes) (Shreem, Abdullah & Nazri, 2014). 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 involves an extreme quantity of computational time and space (Chuang, Yang & Li, 2011). Hence, this may not guarantee that the best subset of genes will be achieved. Moreover, due to the huge number of local optimal solutions in a search space, classical methods of genes selection do not yield the best solution (Lin & Chen, 2012; Zhang et al., 2019).

Therefore, Meta-heuristic techniques are the apt method of finding the optimal solution from the vast search space in order to achieve effective solution without traversing the entire space of solutions. Several Meta-heuristic techniques have been widely employed to deal with gene selection problems, and they have the record of better performance and effectiveness in addressing the problems (Dashtbana et al., 2018; Kabir, Shahjahan & Murase, 2012; Shreem et al., 2014). Sometimes it is not enough to find the highly informative genes alone but, it is a great task to build a classification model that is capable of classifying unknown samples using a few highly predictive gene subset of the original set of genes and, at the same time minimize misclassifications rate (Zhang et al., 2019). This study therefore, aimed at developing Information Gain-Modified Binary Bat

(InfoGain-MBB) algorithm for selecting informative features from high dimensional Microarray cancer datasets in order to remove redundant and non-predictive genes from the Microarray cancer datasets at the same time solved the problem of class imbalance in the datasets without eliminating important genes that help in improving accuracy performance.

1.3 Aim and Objectives

The aim of this study is to design an InfoGain-MBB Algorithm features selection model for selecting relevant features from high dimensional Microarray cancer datasets.

The objectives are, to:

- i. analyse series of filter algorithms (FA) to select relevant features from datasets and compare their performances;
- ii. develop InfoGain-MBB feature subset selection model;
- iii. implement the developed model; and
- iv. evaluate the performance of the developed model and compare it with other existing models..

1.4 Significance of the Study

In the past few decades, diagnosis and detection of cancer diseases were based on the examination of the organs where the tumour was developed. This sometimes resulted in the exhaustive physical and histopathological assessments of the organs that harbour the tumour. However, diagnoses are only achievable either through laboratory tests which

might be too costly to bear or through surgical operations which might expose the patients to different kinds of risk (Veerabhadrapa & Rangarajan, 2010).

In some instances, some of the test results, like autopsy can be available only after the passage of time, thus causing some delay before any detection or diagnosis of cancer could be performed. Fortunately, the advent of DNA Microarray technology in the recent past has introduced dramatic changes into cancer research. With this new technology, it is possible to simultaneously analyse the expressions of several thousands of genes at once and relate their expression patterns to clinical phenotypes (Lønning, Sørli & Børresen-Dale, 2005). With this study, it is easy to identify cell expression patterns that are capable of discriminating between infected (cancer) cells and uninfected (normal) cells. Therefore, it eases the task of predicting (diagnosing) cancer without patients going through the rigour and expensive laboratory tests or surgery.

Also, the developed method helps in producing high accuracy and robust solutions within the shortest period of time without eliminating vital genes.

1.5 Scope of the Study

The scope of the study is to develop an InfoGain-MBB Algorithm features selection model for selecting relevant features from high dimensional Microarray cancer datasets using a combination of filter algorithms and MBB Algorithm to select relevant features that will help in improving classification of Microarray cancer datasets.

1.6 Definition of Operational Terms

Biomarker: or biological marker is a substance that is used as an indicator of a biological state.

Classification: Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data.

Data mining: It is the task of extracting hidden, interesting and useful patterns from large amounts of data in databases or other information repositories.

Deoxyribonucleic Acid (DNA): It is hereditary material found in the nucleus of cells and composes genes.

Feature selection: This is a process, through which no new set of features will be generated, but only a subset of original features are selected and feature space is reduced.

Feature extraction/transformation: This is a process through which a new set of features are created. The feature transformation may be a linear or nonlinear combination of original features.

Filter techniques: This is the process of evaluating the relevance of features by considering only the inherent properties of the data. In most cases, a feature relevance score is calculated, and low-scoring features are removed.

Gene: it is basic physical unit of heredity and it is made up of DNA which encodes instructions to make protein molecules. It determines what the organism will be like, its appearance and how it will behave in the environment.

Gene Expression: It is a process by which gene information in the DNA is transcribed into messenger RNA (mRNA) molecules and is translated into proteins. Disruption in gene expression is responsible for many diseases.

Metastasis: It is a process of spreading tumour to distant areas of the body.

Microarray: It is a laboratory tool used to detect the expression of thousands of genes at the same time. DNA microarrays are microscope slides that are printed with thousands of tiny spots in defined positions, with each spot containing a known DNA sequence or gene.

Oligonucleotide or oligo: This is a short fragment of a single-stranded DNA that is typically 5 to 50 nucleotides long.

Prediction: It is an act of constructing and using a model to determine the class of unlabelled objects, or to evaluate the value or value ranges of an attribute that a given object is likely to have.

Predictive accuracy: It is referred to as the capability of the model to correctly predict the class label of new or previously unseen data.

Supervised learning: It is the machine learning algorithm in which the class to which each training sample belongs is known in advance.

Test Set: It is the labelled data used for predictive data mining tasks that are reserved to evaluate the effectiveness (that is, accuracy) of the predictive model built using the training data.

Training set: It is the data provided as input to a data mining algorithm that is used to train or build a model.

Unsupervised learning: It is a machine learning algorithm in which the class labels of the training samples are not known and the number or set of classes to be learned may not be known in advance.

Wrapper technique: This is the process of selecting feature subset based on the classifier and ranks feature using a learner to allot fitness to each subset. It is more computationally costly than the filter model. The wrapper technique often employs evolutionary approaches to lead their searches.

CHAPTER TWO

LITERATURE REVIEW

2.1 Microarray Technology

The advent of Microarray technology has brought about innovative changes into the biological fields. It is regarded as a motivating advent for valuable researches. It has allowed concurrent examination of hundreds to thousands of gene activities at the same time. Though most of biologists and researchers from other fields find hitches while mining and working with this type of data, likewise Microarray experimental results are stored in diverse and multiple databases.

Microarray technology first appeared in the research arena in the late of 1980s (Rafii, Hassani & Kbir, 2017). Augenlicht, Wahrman, Halsey, Anderson, Taylor & Lipkin, (1987) were the first researchers to describe DNA Microarrays, where about 4000 of complementary DNA (cDNA) sequences were spotted on nitrocellulose (Augenlicht, Taylor, Anderson, & Lipkin, 1991). Microarray has provided a robust opportunity for the biologists to examine and measure hundreds of thousands of gene expressions simultaneously (Chen, Shu, Tsai, Cheng, & Hsu, 2012; Rafii et al., 2017; Remeseiro & Bolon-Canedo, 2019). The Microarray technology has inspired a new line of research in bioinformatics, medical fields and machine learning (Hira & Gillies, 2015; Ventimiglia & Petralia, 2013). DNA Microarrays are typically referred to as DNA chips or biochips that contain a sequence of microscopic DNA spots mounted on a solid surface. The biologists are making uses of DNA Microarrays as a platform to examine the expression levels of thousands of genes simultaneously, or the numerous parts of the genes genotype (Rafii et al., 2017).

2.1.1 DNA Microarray

Cells consist of a core part called nucleus, and in this nucleus, there is DNA that contains encoded “program” for upcoming generations. DNA comprises of coding and non-coding components. The coding parts known to be genes, stipulate the protein structures which perform the vital job in every organism. Proteins are generated in genes in two phases: the first stage is the transcription of DNA into mRNA and then followed by mRNA being translated into proteins. The development of molecular genetics technologies, for example, DNA Microarrays, has provided a platform to successfully view the functionality of the cells and with potential to examine the expression of tens of thousands of genes simultaneously (Bolón-Canedo, Sánchez-Marono, Alonso-Betanzos, Benítez, & Herrera, 2014). Figure 2.1 shows the generic process of obtaining the gene expression profiles from a DNA Microarray. These gene expression data can serve as inputs to large-scale data analysis, such as, to improve understanding and classifying the genes into diseased or normal cells. Gene expression is the process of portraying the manner in which genotype metamorphoses into phenotype by examining the total number of transcribed mRNA in a genomic system. There are lots of standardized approaches for recognizing the variation in gene expression which include the differential display, Microarray hybridization, RNA-seq sequencing, Serial analysis of gene expression (SAGE), subtractive hybridization. Detailed information about all these approaches can be found in (Acunzo, Romano, Wernicke & Croce, 2015; Hammond, 2015; Monobe & Silva, 2016).

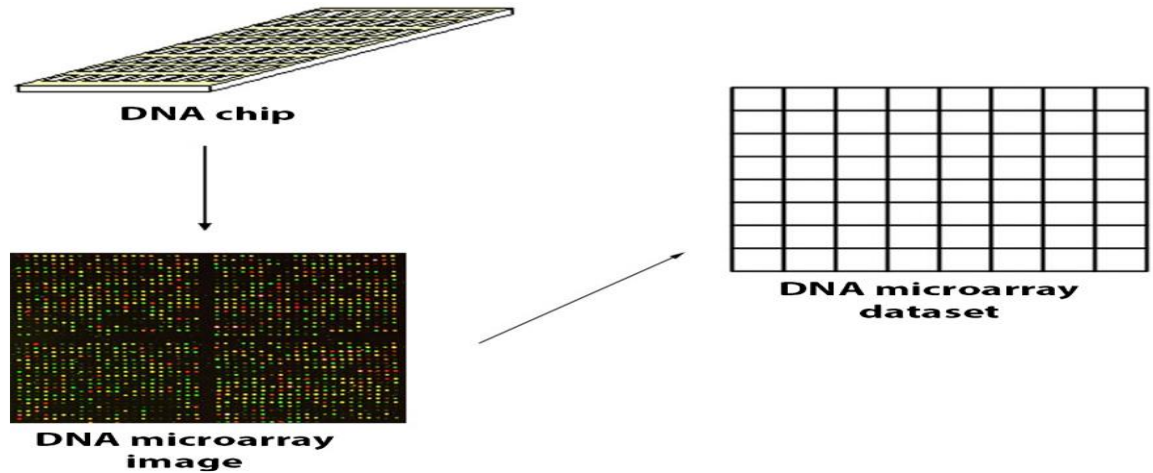


Figure 2.1: Generic Procedure for Acquiring the Gene Expression Data (Bolón-Canedo et al., 2014)

Figure 2.1 depicts the procedure for acquiring microarray data which involves collection of sample tissues from both cancerous and healthy human tissues, isolate mRNA from samples using either a column or a solvent such as phenol-chloroform. After that, cDNA label is created and then hybridized by combining both tissues together on a Microarray plate. The last stage is detection of the relative intensities of fluorescence under Microarray scanner to view the results and also save the data in form of array for future use.

2.1.2 Microarray Experiment

Microarray experiment is the hybridization reaction that involves in the comparison of relative cellular mRNA acquired from two tissue samples. The hybridization reaction takes place when RNA molecules or the single-stranded DNA join in order to integrate into double-stranded complexes. The importance of gene expression Microarray experiments is to investigate and examine the differential gene expression between groups, that is, class comparison, or by classification studies, that is, class prediction and class discovery (Rafii et al., 2017).

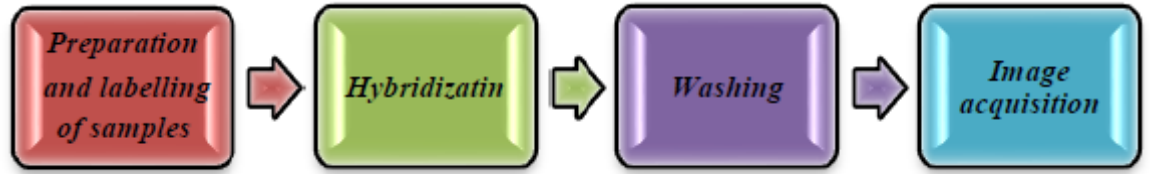


Figure 2.2: Steps of a Microarray Experiment (Rafii et al., 2017)

There are basically four stages that are involved in measuring the gene expressions in Microarray experiment as shown in Figure 2.2. The first stage is referred to as preparation and labelling of samples, where RNA extraction is done from a specific tissue and the labelling depend on the selected technology. The second phase is hybridization; it is the stage where the probes and labels of DNA or RNA targeted for heteroduplexes are base-paired on the surface of glass through Watson-Crick. Detection of hybridization can be done optically, electrochemically or by employing mass sensitive devices (Rafii et al., 2017). Washing is the third stage where excess solution from hybridization array is eliminated. It is a stage where nonspecifically cRNA bound are removed from the microarray surface in order to reduce the effect of the sensitivity and background level of the whole microarray. The final phase is the image acquisition stage where the hybridized image of the array is produced.

2.2 Microarray Data

The data generated by Microarray experiments are arranged and kept as large matrices ($M \times N$). Each Microarray data matrix comprises of the samples illustrated in rows and the genes (features) in the columns as shown in Table 2.1.

Table 2.1: Microarray Data Matrix Format

	G₁	G₂	...	G_M
S₁	x ₁₁	x ₁₂		x _{1M}
S₂	x ₂₁	x ₂₂		x _{2M}
.				
.				
.				
S_N	x _{N1}	x _{N2}		x _{NM}

Microarray data is of the form of M by N matrix and is very huge; where M is rows and it indicates the samples, N is columns and it indicates the genes and every single cell has a certain value of gene expression in a sample (Kong, Yu, Minion & Rajan, 2011). x_{ij} denotes the expression level of the gene j and the condition or sample i . Where j ranges between 1 and M , and i from 1 to N .

2.3 Microarray Data Analysis

Biomedicine field has gained popularity in machine learning in the last few decades mainly as a result of the enormous amount of data available for extraction from genetic tissues. Especially with the advent of DNA Microarray datasets has paved way to the development of a novel and active area of research in machine learning and bioinformatics. For the purpose of machine learning, Microarray data are usually considered as structured data that are characterized by few samples (most of the time they are less than 100) but has numerous features (in the order of thousands). Dealing with this nature of data that have just a few samples with a large number of features pose serious challenges for researchers in the field of machine learning; because of the likelihood of "false positives", which may occur perchance at the time of building the model for prediction or during the selection of relevant features (genes) (Remeseiro & Bolon-Canedo, 2019). It has been established from

the literature that very few genes from these enormous number of genes available in a DNA Microarray are pertinent for classification purpose. In this situation, feature selection is crucial in order to remove redundant and irrelevant features, and also to assist experts to uncover fundamental connections between gene expression and a particular disease.

Therefore, it is necessary to reduce the dataset of those genes in order to reduce the cost of identifying the best genes that can distinguish between the cell classes (normal or abnormal cells). There are two famous techniques for in-depth Microarray data analysis: clustering and classification (Mutch, Berger, Mansourian, Rytz & Roberts, 2001). Clustering is an unsupervised technique of classifying data into clusters of genes or samples with related patterns or characteristics. Classification is a supervised technique that learns from examples. Provided with a set of pre-classified samples, the classifier learns to assign unknown sample cases to one of the predefined classes (Selvaraj & Natarajan, 2011).

2.4 Cluster Analysis

Clustering is among the famous methods employed in the Microarray data analysis. It is employed to seek co-regulated and functionality of related groups (Svrakic, Nesic, Dasu, Herndon & Perez-Polo, 2003). Clustering is generally useful in the instances when the whole sets of an organism's genes are not available. Cluster techniques can be categorized into three common types such as hierarchical clustering, k-means clustering, and self-organizing maps. Hierarchical clustering is a popularly employed unsupervised technique that models clusters of genes with identical patterns of expression (Eisen, Spellman, Brown, & Botstein, 1998). Its process is performed in iteration where genes that are highly correlated in terms of the extent of their expression are grouped together. It is a

way of modelling cluster in a hierarchical manner. Genes are represented as leaves of tree branches in the dendrogram. In general, hierarchical clustering is usually modelled in two ways; that is agglomerative and divisive. Agglomerative uses a bottom-up approach where each individual that has a similar pattern is clustered together and different pairs of clusters are amalgamated as one moves up the hierarchy. Divisive employs top to down approach that is, all patterns (Datasets) start in a single cluster and splits are done recursively as one traverse down the hierarchy (Eisen et al., 1998).

K-means clustering is a machine learning algorithm that clusters datasets into groups of similar observations without the need of having prior knowledge of the relationships that exist in the data (Tavazoie, Hughes, Campbell, Cho & Church, 1999). It is a clustering technique that is very easy to accomplish and is commonly employed in biometrics and medical imaging applications. Euclidean properties of the vector space are usually applied in the K-means clustering algorithm. The vector space is initially partitioned into K parts then; the algorithm computes the focal positions in each of the subspaces and modifies the partition in order to assign each vector into the cluster that is closest to the midpoint. This iteration is recurring until either the partitioning is steady or the numbers of specified iterations are reached (Brazma & Vilo, 2000).

The third category of the cluster is the Self-Organizing Map (SOM) which is a neural network-based non-hierarchical clustering approach. SOMs work in the same manner as K-means clustering (Tamayo et al., 1999).

2.5 Classification

Classification is a supervised learning approach which is sometimes referred to as class prediction or discriminant analysis. When given a set of pre-classified dataset, for

instance, cancer dataset with different classes such as normal and diseased cells or Acute myeloid Leukemia (AML) and Acute Lymphoblastic Leukemia (ALL); a classifier will seek for rules that will enable it to assign unknown samples supply to any of the above classes (or diagnose the diseased cell) (Quackenbush, 2001). In the classification task, there is a need for sufficient numbers of samples to allow the algorithm to be able to use a particular sample as training dataset and other fresh independent samples are used to test the algorithm which will serve as the test set. There are lots of classification algorithms available in the literature, such as Artificial Neural Networks (ANN), Decision Trees, K Nearest Neighbours (KNN), Weighted voting, Support vector machines (SVM) and many others. One of the interesting areas of classification approach is in the clinical diagnostic application which predicts and classifies disease types and subtypes. Prevalent examples include diagnosis of Leukemia into ALL or AML (Golub et al., 1999), brain tumour diagnosis (Wang, Bø, Jonassen, Myklebost & Hovig, 2003) and Lymphoma (Kim et al., 2005) and breast cancer diagnosis (Hambali & Gbolagade, 2016).

2.6 Feature Selection Techniques

Many classification algorithms were initially not designed to handle huge volumes of irrelevant features; merging them with feature selection (FS) methods have become inevitability in several applications (Guyon & Elisseeff, 2003; Yu & Liu, 2004).

The major focus of FS is to select a subset of features from the input data, and that subset can successfully describe the input data. Feature selection can also diminish negative impacts from irrelevant or noise features (Bolón-Canedo et al., 2015; Brown, Pocock, Zhao & Luján, 2012; García et al., 2016; Zhang et al., 2019; Zhou, Hu, Li & Wu, 2017). No extra information can be obtained from redundant features except that they

create noise for the classification algorithm. In the nutshell, few unique features can provide crucial information that can be achieved by using the whole dataset as input, which can serve as class representative. Therefore, by discarding the redundant features, the volume of data can be reduced to enhance the performance of the classification algorithm. In some instances, some features did not have any correlation with the classes' at all. They are just pure noise and hence, introduce bias to the classification algorithm and weaken its performance. By employing FS techniques, insight is provided into the feature selection process and therefore the classification accuracy is improved (Zhang et al., 2019).

In summary, the goals of FS are multifarious; the vital ones include the following:

- (a) It prevents over-fitting and therefore, enhances model performance,
- (b) It brings about the faster processing and reduces the model cost-effectively and
- (c) It helps in gaining deeper knowledge into the core processes that generate the data.

Nevertheless, the benefits of FS methods come with a trade-off. The process of searching for the subset of relevant features brings about an extra layer of complexity in the processing task. There is need to seek the best model parameters for the best feature subset rather than optimizing the model's parameters for the full feature subset since assurance for the optimization of parameters for the complete feature set that can equally lead to the best feature subset is not guaranteed (Daelemans, Hoste, De Meulder & Naudts, 2003).

Quite a lot of statistical approaches have been used to select the genes for medical disease diagnosis, prognosis, and therapeutic targets (Tu, Yu, Guo & Li, 2004). Apart from the statistical methods, in recent times, data mining and machine learning approaches have

been widely adopted in genomic data analysis (Khan & Alam, 2012; Zahiri, Yaghoubi, Mohammad-Noori, Ebrahimpour & Masoudi-Nejad, 2013). For instance, Cho, Lee, Park & Lee, (2004) proposed a modified kernel Fisher discriminant analysis (KFDA) for the analysis of the hereditary breast cancer dataset. The KFDA classifier employed the mean-squared-error as the gene selection criterion.

Furthermore, numerous hybrid evolutionary algorithms are available in the literature for the purpose of improving the accuracy of the classification techniques (Huang & Chang, 2007; Tan, Fu, Zhang & Bourgeois, 2008; Tong & Schierz, 2011). The goal of many evolutionary algorithms are to seek for best features subset by adopting bio-inspired approaches such as particle swarm optimization (PSO), Honey Bee, Firefly algorithms and so on. These types of algorithms show suitable performance improvement over several problems but require experts' involvement to attain the desired performance outcome.

FS methods differ from one another in the mode of integrating their search in the added space of feature subsets, in the model selection. Gene selection is a subclass of larger machine learning class of feature selection. There are two major ways of categorizing FS and the related taxonomy – Label Status and Search strategy (Hambali et al., 2020; Zhang et al., 2019) as illustrated in Figure 2.3.

2.6.1 Label Status

In this category, methods are categorized based on the samples' feature labelled or not. The labelled samples offer hands-on information, such that relevant and informative features are selected to differentiate samples from different classes through the supervised FS methods (Nie, Huang, Cai & Ding, 2010). In label status category, FS technique can be further divided into three groups, such as, supervised methods (Huang, 2015; Li, Chen,

Wei, Xu & Kou, 2007; Nie et al., 2010; Zhang, Nie & Li, 2018; Zhao & Liu, 2007), semi-supervised methods (Xu et al., 2010; Zhao & Liu, 2007a; Zhou & Wang, 2007), and unsupervised methods (Cai, Zhang & He, 2010; He, Cheng, Hu, Zhu & Wen, 2017; Hu et al., 2017; Liu, Rallo & Cohen, 2011; Yang, Shen, Ma, Huang & Zhou, 2011). In semi-supervised FS approach, parts of the data are labelled while others are not labelled. Most of the famous semi-supervised FS methods make use of similarity matrix construction, such that the selected features are based on graph structure (Cheng, Zhou & Cheng, 2010; Zhao & Liu, 2007b). Due to the nonappearance of labels that can help to guide the exploration of informative genes, unsupervised FS make use of some tricky approaches (Zhang et al., 2019) to select the relevant features from the unlabelled dataset. In its simplest form, the FS procedure is performed by evaluating individual features and then ranks them using their correlation with class labels (Zhang et al., 2019). Though, Hall, (1999) has revealed that the best features subsets are those subsets whose features did not correlate with one another. Therefore, it is desirable to evaluate and consider the features as a subset rather than the individual. Liu, Motoda, Setiono & Zhao (2010) categorised a summary of the subset feature selection procedure into three main phases:

- i. **Search:** Producing features subset to be evaluated.
- ii. **Evaluation:** Evaluating the efficacy of the produced subset.
- iii. **Stop:** Determining whether to continue the search or to halt the search.

In order to achieve the aim of feature selection, a lot of criteria have been suggested to estimate the relevance of the feature (Zhang et al., 2019; Zhao & Liu, 2007).

2.6.1.1 Supervised Method

Supervised FS methods are more related to the labelled data points. Conservative supervised approaches such as Fisher Score (Duda, Hart & Stork, 2012) do the ranking of the features individually and they did not take feature correlation into consideration. Linear discriminant analysis (LDA) (Fisher, 1936) was employed to select features by maximizing the ratio between the 'between-class scatter and within-class scatter'. However, with the small sample size data, LDA encountered some challenges (Fukunaga, 2013). In order to solve this problem, maximum margin criterion (MMC) is used in Li, Jiang & Zhang, (2004), which converts the computation of ratio of traces between the 'between-class scatter and within-class scatter' into the subtraction form. In summary, all supervised methods need appropriately labelled data to achieve the task of feature selection. But when the data for training are not sufficiently labelled, the performance of the method will degenerate (Luo, Tao, Xu, Li & Xu, 2013).

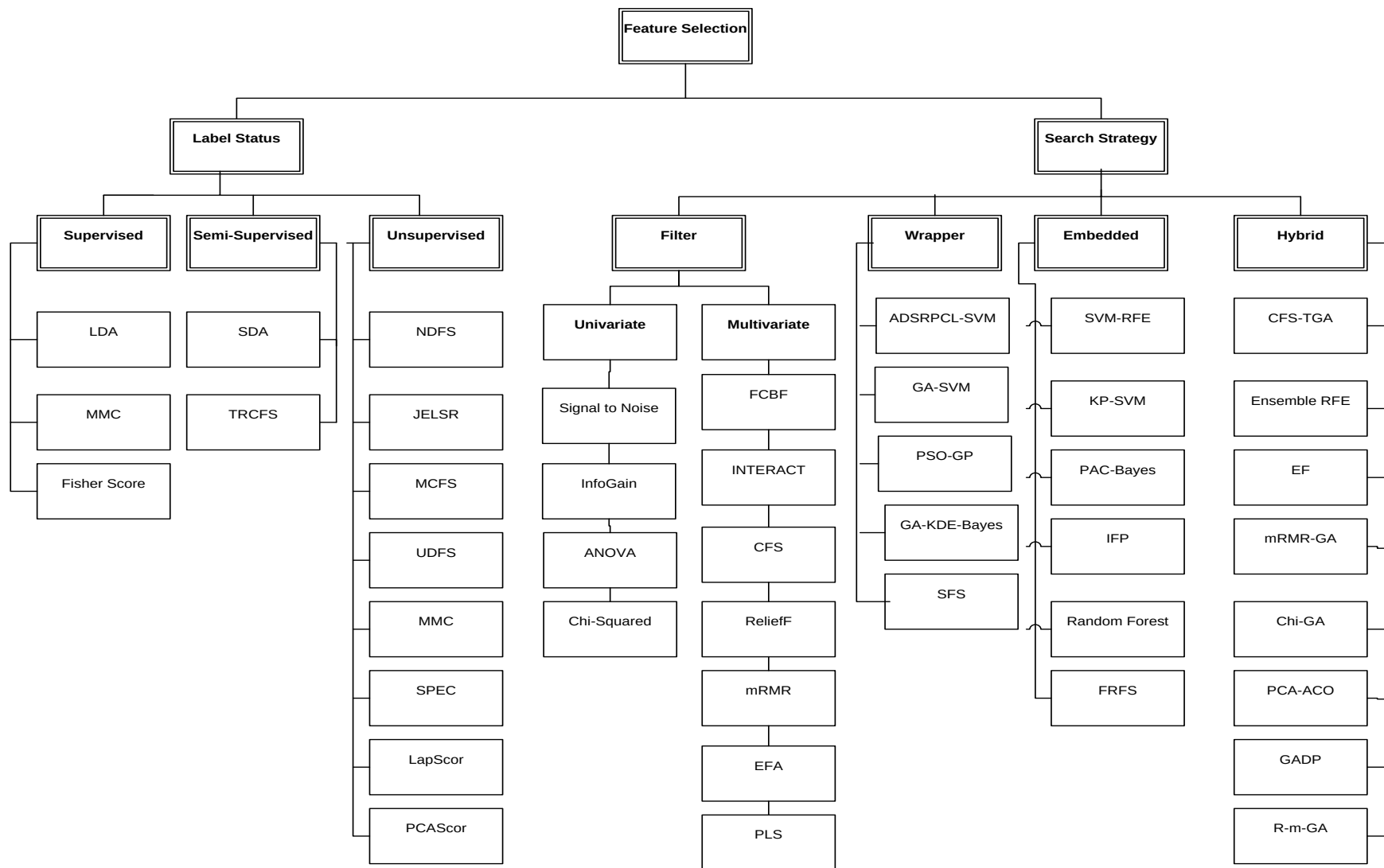


Figure 2.3: Feature Selection Category (Hambali, Oladele & Adewole, 2020)

2.6.1.2 Semi-Supervised Method

In semi-supervised FS methods, Luo, Hou, Nie, Tao & Yi, (2018) and Han, Yang, Yan, Ma, Sebe & Zhou, (2014) worked on both labelled and unlabelled data. Consequently, semi-supervised approaches demonstrated the ability to select features from unlabelled data when provided with few numbers of labelled data. Graph Laplacian based semi-supervised techniques assumed that most data samples lay on a low-dimensional manifold, such as semi-supervised Discriminant Analysis (SDA) (Cai, He & Han, 2007). In Graph Laplacian based approaches; unlabelled samples are harnessed through uses of graph Laplacian matrix. For example, Trace Ratio Criterion for Feature Selection (TRCFS) is one of the effective semi-supervised FS algorithms to select informative features (Liu, Nie, Wu & Chen, 2013). However, semi-supervised FS techniques are typically less effective in handling huge dataset due to the cost of computation time of the graph (Chang, Nie, Yang & Huang, 2014).

2.6.1.3 Unsupervised Method

As a result of the missing label information that can help in guiding the search for informative features in the data and the problem of getting the actual label information, unsupervised FS is regarded as a challenging problem (Dy & Brodley, 2004; Zhang et al., 2019). Multiple criteria were considered for the evaluation of feature relevance. One of the famous criteria used is to select those features that retain the original data manifold structure. Another popularly employed criterion is to find cluster indicators through clustering algorithms and then follow a supervised approach of problem-solving. Unsupervised FS can be categorized into two different approaches. One of the techniques is to find cluster indicators which concurrently execute the supervised feature selection.

For instance, Yang et al., (2011) and Li, Piao, Piao, Li & Ryu, (2014) designed a combined framework that consists of nonnegative spectral clustering and structural learning to select features from unlabelled data. The second one is hierarchical form, which the first search for cluster indicators, then executing feature selection, and lastly iterating two previous steps until a certain criterion is satisfied. Zhao, Wang, Liu & Ye, (2011) proposed Similarity Preserving Feature Selection (SPFS) for feature selection by conserving sample similarity, which can handle feature redundancy successfully.

Other examples of unsupervised FS methods include Nonnegative Discriminative Feature Selection (NDFS), Embedded Unsupervised Feature Selection (EUFS) (Wang, Tang & Liu, 2015), Multi-Cluster Feature Selection (MCFS) (Cai et al., 2010), Joint Embedding Learning and Sparse Regression (JELSR) (Hou, Nie, Li, Yi & Wu, 2013) and Unsupervised Discriminative Feature Selection (UDFS) (Yang et al., 2011).

In recent times, some investigations have appeared that leverage on both the learning mechanism and manifold structure. The typical techniques include; PCA Score (PcaScor) (Krzanowski, 1987), Laplacian Score (LapScor) (He, Cai & Niyogi, 2006), Spectral Feature Selection (SPEC) (Zhao & Liu, 2007), Minimum Redundancy Spectral Feature Selection (MRSF) (Zhao, Wang & Liu, 2010) and Multi-Cluster Feature Selection (MCFS) (Cai et al., 2010). Generally, these approaches first employ numerous graphs to depict the manifold structure. LapScor and SPEC then utilize other metrics for the ranking of each feature. MCFS and MRSF, however, enhance the process with sparse constraints in multi-output regression (Hou et al., 2013).

2.6.2 Search Strategy

Based on the different searching strategies, FS can further be grouped into four approaches, that is, filter techniques, wrapper techniques, embedded techniques and hybrid (ensemble) techniques (Khan & Alam, 2012; Remeseiro & Bolon-Canedo, 2019; Saeys, Inza & Larrañaga, 2007; Zhang et al., 2019). Filter techniques accomplish the FS task before classification or clustering of the data and usually execute in a two-step approach. In the first step, all the features are ranked based on defined criteria and the second step involves the retaining of the top-ranking features. Filter techniques evaluate the relevance of the feature genes by searching for the inherent properties of the data. In some of the filter approaches, a feature relevance score is estimated, and low-scoring features are disregarded. Subsequently, the features with high rank are chosen (Pihur, Datta & Datta, 2008; Qi, Sun, Sun & Pan, 2011; Tan, Fu, Zhang & Bourgeois, 2008). The famous Filter methods include ReliefF (Kira & Rendell, 1992a; Raileanu & Stoffel, 2004), signal-to-noise ratio (Golub et al., 1999), receiver operating characteristics analysis (Khodarev et al., 2003), information gain (Peng, Long & Ding, 2005; Raileanu & Stoffel, 2004), Laplacian score (He et al., 2006), F-statistic (Ding & Peng, 2005), mRMR (Peng et al., 2005), fuzzy logic for removing redundant features (Huang & Chang, 2007), T-test feature ranking for gene selection (Zhou & Wang, 2007), consensus independent component analysis that employs gene expression value for cancer classification (Zheng, Huang, Kong & Zhao, 2008), mutual information (Cai, Hao, Yang & Wen, 2009) and maximum-minimum correntropy criterion (Mohammadi et al., 2016).

Filter approaches are machine learning algorithm independent. It focuses on the general inherent property of the data. Due to their independence from the learning

algorithm, they have less computational cost and have a good generalization capacity. Wrappers and embedded approaches both need a machine-learning algorithm to accomplish feature selection. In wrappers, evaluation of candidate feature subsets is done by a learning algorithm. The interaction wrappers with the classifier make them more computationally expensive than filters, though they perform better than filters. Wrapper methods use the learning algorithm to evaluate the FS. Guyon, Weston, Barnhill & Vapnik, (2002) uses SVM based Recursive Feature Elimination (RFE) to choose significant genes associated with cancer diseases. Embedded methods implement FS in relations to optimization model (Zhang et al., 2019). Embedded techniques maintain mid-course between filters and wrappers since FS is part of the training process for the learning algorithm. The exploration of the best features subset is done during the training of the learning algorithm (for instance, during weights optimization in a neural network). Due to this reason, embedded approaches are less computationally expensive compared to wrappers. There are also other approaches to combining algorithms. This is referred to as hybrid approaches, which typically combines two or more FS algorithms of the different search strategy in a sequential manner. For example, a less computationally expensive algorithm such as filter may be first applied to eliminate some features and later employ a more complex and costly algorithm (such as, wrapper) for fine-tuning.

2.6.2.1 Filter Techniques

Filter techniques evaluate the gene subsets informative by observing the inherent data characteristics only, that is, statistical measures by which a single gene (a subset of genes) is typically assessed against the class label. Classical filter techniques are commonly employed in Microarray data analysis such as consistency-based filter, Fast

Correlation-Based Filter (FCBF), Correlation Feature Selection (CFS) and ReliefF. In these approaches, individual features are considered separately; hence it does not give room for feature dependencies, as a result of that, it usually gives poor diagnostic accuracy when compared to other types of FS methods. Filter methods have some pros which are not limited to the following: they are computationally simple and easy, efficiently scaled through very high-dimensional datasets, and classifiers independent. FS techniques can also be classified into univariate and multivariate types. Univariate approaches treat each feature individually of each other, a hitch that multivariate methods proffer solution to by incorporating feature dependencies to some certain level, but with an increase in demanding for more computational resources. Signal-to-noise ratio (Chow, Moler & Mian, 2001; Golub et al., 1999) is one of the common examples of univariate approach; it evaluates each feature independently based on a certain criteria. Thus, univariate methods are simple, fast and the most available methods in the literature (Ben-Dor, Bruhn, Friedman, Nachman, Schummer & Yakhini, 2000; Bolón-Canedo, Sánchez-Marroño & Alonso-Betanzos, 2013; Bolón-Canedo et al., 2014; Xing, Jordan & Karp, 2001). On the other hand, Forward Feature Selection is one of the example of the multivariate techniques. It evaluates relevance and dependency of the features in a group subset (Bø & Jonassen, 2002; Grate, Bhattacharyya, Jordan & Mian, 2002). It starts with a single discriminative feature and continuously adds the new discriminative features using greedy search technique. Another multivariate approach is Base-Pair Selection; the relevance of the features is estimated via pairs of features. Though, there are several snags limiting the application of multivariate methods. Mainly, they are characterized with the overtraining of the data and normally are of the high computational cost.

Filter techniques that have been employed in various tasks can be found in the work of Bekkerman, El-Yaniv, Tishby & Winter, (2003), Caruana & Sa, (2003) and Davidson & Jalan, (2010). Javed, Babri & Saeed, (2010) and Peng et al., (2005), developed a novel ranking criterion based on the class density of the binary data. The related algorithm applies the filter technique to rank the features and later the wrapper approach to further remove irrelevant features. Caruana & Sa, (2003), eliminated features are employed for Multi-Task Learning (MTL). In Stoppiglia, Dreyfus, Dubois & Oussar, (2003), a random variable known as a probe is utilized to rank the features in terms of Gram-Schmidt orthogonalization.

The noticeable drawback of the filter approach is the disregard of interaction between the classification algorithm and filter algorithm; that is, the feature subset space search is separated from the search in the hypothesis space. Since filter approaches do not rely on the learning algorithms, the selected features subset may not yield optimal subset, such that a redundant subset could be attained respectively (Zhang et al., 2019). Brief descriptions of common filter techniques are as follows:

- i. **Correlation-based Feature Selection (CFS):** is one of the multivariate filter algorithms that do feature subsets ranking by applying correlation based heuristic evaluation function (Hall & Smith, 1998). The evaluation function usually biases toward the subsets that contain features that are uncorrelated within themselves but demonstrated a high correlation with the class. Irrelevant features are eliminated as a result of their low correlation with the class and also, redundant features are discarded due to the fact that they are highly correlated with one or more of the selected features. The choice of a selected feature depends on to what

level in which it is able to predict classes in areas of the instance space not already predicted by other features.

- ii. **The Fast Correlation-Based Filter (FCBF) technique:** is a multivariate algorithm that evaluates feature-class and feature-feature correlation (Yu & Liu, 2003). FCBF begins selection using symmetrical uncertainty (SU) to select a set of features that are highly correlated with the class. SU is described as the ratio between the information gain and the entropy of two features. Also, it employs three heuristic approaches to eliminate the redundant features and retain those features which are more pertinent to the class. FCBF is suitable for high-dimensionality data and has been proven to be efficient in eliminating both irrelevant and redundant features. Though, it neglects interaction between features.
- iii. **The INTERACT algorithm:** utilizes the same SU goodness measurement employed by the FCBF, but in addition to that, it includes the consistency contribution, which reveals how significantly the exclusion of a feature will likely affect consistency (Zhao & Liu, 2007b). The algorithm entails two major steps. The first step involves the ranking of features in descending order based on their evaluated SU values; while the second step performs the evaluation of features consistency by individually estimating the features consistency contribution of the feature beginning from the last on the ranked feature list and it selects only those features whose consistency contribution exceeds an established threshold and discard those with the less value. The authors established that this technique takes care of feature interaction and effectively selects relevant features.

- iv. **Information Gain:** is one of the famous attribute evaluation approaches that computes mutual information for each class and attribute of the data. It is a univariate filter which provides orderly ranked features with the guide of a threshold (Kira & Rendell, 1992b). The selection of feature is done based on those features with a positive information gain value.
- v. **ReliefF:** is one of the famous multivariate filters based on nearest neighbours (Kononenko, 1994). Its original version is known as Relief algorithm developed by Kira & Rendell, (1992a). The Relief algorithm works by sampling an instance randomly from the data and seeking for nearest neighbours from the same class and other class present in the data. The nearest neighbour values of the selected attributes are compared to the sampled instance and then, utilized it to update the relevance scores for each attribute. The basis is that an informative attribute should be distinguished between instances from different classes and have the identical value with instances of the same class. ReliefF has the capability of efficiently solving the problem of multiclass and also more robust and proficient in dealing with missing and noisy data. This algorithm can be employed in all circumstances with low bias; it accommodates features interaction and captures local dependencies which are lacking in other approaches. The weakness of the ReliefF algorithm includes the process of choosing a threshold. Acuna, Coaquira & Gonzalez, (2003), compared ReliefF with other wrapper techniques under different datasets.
- vi. **The minimum Redundancy Maximum Relevance (mRMR) algorithm:** chooses features with the highest relevance to the target class and also with minimal

redundancy; that is, it chooses features that have maximum dissimilarity to each other (Peng et al., 2005). Both criteria employed are based on mutual information that is, maximum-relevance and minimum-redundancy.

- vii. Consistency-Based Filter:** is a multivariate method which selects subsets of the features, but the selection is done based on the degree of consistency with the class (Dash & Liu, 2003). The acceptability of the selected feature rate is determined using an inconsistency criterion. Summary of other Filter techniques provided in Table 2.2 based on evaluation approach used (either rank or subset) and type of data.

Table 2.2: Summary of Other Filter Techniques Used in Microarray Data (Bolón-Canedo et al., 2014)

Method	Original Reference	Type of evaluations (R/S)	Type of Data (B/M)
Backward Elimination using Hilbert-Schmidt Independence Criterion (HSIC) (BAHSIC)	Song, Smola, Gretton, Bedo & Borgwardt, (2012)	R	M
Discritizer+filter	Bolón-Canedo, Sánchez-Maróño & Alonso-Betanzos, (2010); Sánchez-Maróño, Alonso-Betanzos, García-González & Bolón-Canedo, (2010)	S	M
Entropic Filtering Algorithm (EFA)	Navarro & Muñoz, (2009)	S	B
Monotone dependence	Bolón-Canedo, Seth, Sánchez-Marono, Alonso-Betanzos & Príncipe, (2011)	R	M
Multi-task feature selection filter (MFS)	Lan & Vucetic, (2011)	R	M
Matrix of Average Sub-Subset Information for Variable Elimination (MASSIVE)	Meyer, Schretter & Bontempi, (2008)	R	M
Maximum Weight And Minimum Redundancy (MWMR)	Wang, Song, Xu, & Zhou, (2013)	R	B
Partial Least Squares (PLS)	Student & Fajarewicz, (2012)	R	M
Relevance Feature Selection (RFS)	Ferreira & Figueiredo, (2012)	R	M
Robust Feature Selection method (RFS)	Nie et al., (2010)	R	M
Relevance Redundancy Feature Selection (RRFS)	Ferreira & Figueiredo, (2012)	R	M

Note: R/S for ranker/subset and B/M for binary/multiclass.

2.6.2.2 Wrapper Techniques

Evolutionary approaches were normally used to lead their searches in wrapper techniques. They usually commence with a population of solutions that contains a subset

of the features. Then each subset would be computed using a learner algorithm to allot fitness to each subset. Generally, the feature subsets selection is improved by an iterating process to come out with optimal solutions. Some of the famous wrapper methods found in literature are Ant Colony Optimization (Yu, Gu, Liu, Shen & Zhao, 2009), Distance Sensitive Rival Penalized Competitive Learning – Support Vector Machine (ADSRPCL-SVM) (Xiong, Cai & Ma, 2008), Artificial Bee Colony (ABC) algorithm (Garro, Rodríguez & Vázquez, 2016; Hancer, Xue, Karaboga & Zhang, 2015), Genetic algorithm with SVM (Wang et al., 2011), Particle Swarm Optimization (Kar, Sharma, Das & Maitra, 2015) and Genetic Programming (employed to predict alternative mRNA splice variants) (Vukusic, Grellscheid & Wiehe, 2007). The wrapper technique's performance is usually better than filter techniques since wrapper techniques allow interactions between the solutions and predictors. On the other hand, filter methods address the challenges of seeking a better feature subset individually for the model selection phase; wrapper approaches integrate the model assumption search within the feature subset search. In this approach, possible feature subsets space is defined in a search procedure, and many features subsets are generated and weighed. The particular features subset evaluation is attained by training and testing the subset with a specific classification algorithm, making this method designed for a particular learning algorithm. The search algorithm is enfolded around the classification technique to explore all the feature subset space. Though, as the feature space subsets upsurge exponentially with the number of features, heuristic search approaches are employed to achieve the search for the best subset. These search techniques can be categorized into two groups: deterministic and randomized search techniques. Among the benefits of wrapper techniques are the capability to take feature dependencies

into consideration and existence of an interaction between feature subset search and model selection. A prominent setback for these approaches is that they are prone to over-fitting than filter methods due to the few sample size of Microarray data, and are characterized with high computational expenses, particularly if modelling the classifier requires high computational cost.

Wrapper method did not receive much attention as received by the filter approaches as a result of its high computational cost. As the feature number increases, the feature searching space of subsets upsurges exponentially. This sometimes becomes critical when the features grow to tens of thousands or more. For this reason, the wrapper method has been mostly avoided in the literature.

Most of the works which utilized wrapper method can be found in the early years of investigating Microarray data. In a classical wrapper, a search is conducted in the space of genes and the goodness of each gene subset is determined by evaluating the accuracy achieved by the specific learning algorithm. This learning algorithm is trained only with the selected genes. For instance, Inza, Sierra, Blanco & Larrañaga, (2002) employed typical wrapper techniques such as floating selection, sequential forward and backward selection and best-first search on three Microarray datasets. Another case can be found in Ruiz, Riquelme & Aguilar-Ruiz, (2006) that made use of an incremental wrapper named BIRS (Best Incremental Ranked Subset) for the selection of genes. Though the application of wrappers on Microarray data is not well embraced compare to the other feature selection approaches, other examples were found in recent years.

Sharma, Imoto, & Miyano, (2011) presented an approach called Successive Feature Selection (SFS). It is a famous conventional feature selection algorithm that tries to

overcome the weakness of individual ranking and forward selection schemes. The proposed SFS firstly arrange the features into smaller block partitions. From each of the blocks, peak features were obtained based on their classification accuracy values and then compared so as to select the optimal feature subset. This technique yielded high classification accuracy on many DNA Microarray datasets.

Wanderley et al., (2013), present an evolutionary wrapper technique called Genetic Algorithm-Kernel Density Estimation (GA-KDE-Bayes). It employs a Bayesian classification algorithm and a non-parametric density estimation technique. The authors explain that non-parametric techniques are a good alternative for scarce and sparse data such as the bioinformatics analysis because assumptions about its structure are not predefined and all the information originates from the data itself. The results of their works on six Microarray datasets revealed their approach performs better. Table 2.3 shows the summary of the wrapper technique.

Table 2.3 Summary of Wrapper Techniques Used in Microarray (Bolón-Canedo et al., 2014)

Method	Original Reference	Type of Evaluation (R/S)	Type of Data(B/M)
GA-KDE-Bayes	Wanderley et al., (2013)	S	B
SFS	Sharma et al., (2011)	S	M

2.6.2.3 Embedded Technique

In spite of its lower computational time, the major drawback of the filter method is that it lacks interaction with the classifier, mostly leading to poor performance of classifier when compared with those obtained with wrappers. Though, wrapper technique is also characterized by a huge computational cost with Microarray data. A middle course solution for researchers is the application of embedded approaches which employ the core of the

learning algorithm for ranking of feature criteria. In this approach, optimization techniques are employed to rank the features embedded in the projection subspace. The goal of embedded technique (Guyon & Elisseeff, 2003; Zhang et al., 2019) is to minimize the computational time for reclassifying different feature subsets. The integration of feature selection as part of the learning process is the key purpose of embedded techniques. In Battiti, (1994), a greedy search algorithm is utilized to evaluate the feature subsets. Support Vector Machine based on Recursive Feature Elimination (SVM-RFE), is one of the most popular embedded techniques proposed by Guyon et al., (2002). It was specifically designed to select genes for cancer classification. Feature selection was done in this approach by repeatedly training SVM classifier with the presented set of features and eliminating the less significant feature as shown by the classifier. Table 2.4 shows a description of a few embedded methods used for Microarray data analysis.

Maldonado, Weber & Basak, (2011) introduced a novel embedded technique that concurrently selects significant features during the classifier modelling by penalizing individual features used in the dual formulation of SVM. This technique is referred to as kernel penalized SVM (KP-SVM) and it enhances an anisotropic radial basis function (RBF) Kernel shape by removing features that have low significance for the learning algorithm. In their experiment, two benchmark Microarray datasets and two real-world datasets were used and it revealed that KP-SVM performed better than the alternative approaches with steadily fewer relevant features.

In order to solve the problem of data imbalance in some Microarray datasets, Anaissi, Kennedy & Goyal, (2011) proposed a new embedded approach based on the random forest algorithm. The approach used different strategy methods and algorithms to

overcome the different problem of complex gene expression of Leukemia dataset. Firstly, a technique is used to seek for the best training error cost for an individual class, and also, to treat data imbalance. Then, the random forest is employed to select the significant features and lastly, a strategy to prevent over-fitting is also applied. The experimental outcome proved to be a very acceptable result.

Shah, Marchand & Corbeil, (2011) proposed a Probably Approximately Correct (PAC) Bayes feature selection embedded in decision stumps and three other formulations based on different learning principles. PAC-Bayes yields a viable classification performance and at the same time using fewer significant features.

Canul-Reich, Hall, Goldgof, Korecki & Eschrich, (2012) presented an embedded gene selector called Iterative Perturbation method (IFP). They applied it on four different Microarray datasets. The least significant features were determined by employing a backward elimination approach and a criterion, which rely on classifier performance of each feature when perturbed by noise. A feature is considered significant if when adding noise produces a great change in the classifier performance. The IFP algorithm results showed similar or superior average class accuracy when compared to the SVM-RFE on three out of the four datasets used.

Wang et al., (2013) presented a First Order Inductive Learner (FOIL) Rule-based feature subset selection algorithm (FRFS). This technique, in the first instance produces the FOIL classification rules using a modified propositional implementation of the FOIL algorithm. It then merges the subset features that are obtained in the antecedents of rules and achieves a candidate feature subset that has eliminates redundant features and preserves the interactive and informative ones. In the last stage, it evaluates the

significance of the features in the selected feature subset by their new metric called Cover Ratio and detects and eliminates the irrelevant features.

Table 2.4: Summary of Embedded Techniques used in Microarray Data (Bolón-Canedo et al., 2014)

Method	Original Reference	Type of Evaluation (R/S)	Type of Data(B/M)
FRFS	Wang et al., (2013)	S	M
IFP	Canul-Reich et al., (2012)	S	B
KP-SVM	Maldonado et al., (2011)	S	M
PAC-Bayes	Shah et al., (2011)	R	B
Random Forest	Anaissi et al., (2011)	S	M

2.6.2.4 Hybrid Techniques

Today, the trend has moved away from using only typical FS techniques (such as filters, wrappers and embedded) but now focusing on combinations of typical FS techniques such as hybrid or ensemble methods. Hybrid approaches typically combine two or more FS algorithms of diverse conceptual origin in a serial manner (linear combination). The hybrid technique is a novel approach which tries to seek the benefits of both filter and wrapper techniques. A hybrid method uses the combination of independent test and performance estimation function of the feature subset. Hybrid approaches are appropriate for high-dimensional data (such as Microarray dataset) in order to reduce the time complexity of selecting relevant features from the dataset. The notion behind the hybrid approach is that the filter approach is used to remove irrelevant features (dimensionality reduction) from the original dataset and it aimed to attain a trade-off between the time complexity and feature space size. Then, the wrapper technique is employed to find the best features subset from the selected feature pool. This approach speeds up feature

selection because the filter technique swiftly reduces the irrelevant features from the dataset. Advocates of hybrid techniques claim that the chance of removing fantastic predictors by filter approaches is very low if the filter threshold point for a ranked list of features is set low. Some common hybrid techniques include: Chi-Squared statistics with Genetic Algorithm (GA) (Lee & Leu, 2011), Information Gain with a novel Memetic algorithm (Zibakhsh & Abadeh, 2013), a novel similarity scheme with ABC (Hancer et al., 2015), MRMR with GA (El Akadi, Amine, El Ouardighi & Aboutajdine, 2011), and hybridization of GA and SVM (Li et al., 2005). Table 2.5 depicts a summary of other hybrid methods available in the literature.

Table 2.5: Summary of Other Hybrid Techniques for Feature Selection used in Microarray Data. (Bolón-Canedo et al., 2014)

Method	Original Reference	Type of Evaluation (R/S)	Type of Data (B/M)
Correlation-based Feature Selection (CFS) and the Taguchi-Genetic Algorithm (CFS-TGA)	Chuang et al., (2011)	S	M
Ensemble- Recursive Feature Elimination (RFE)	Abeel, Helleputte, Van de Peer, Dupont & Saeys, (2010)	S	B
Ensemble of Filters (EF)	Bolón-Canedo, Sánchez-Maróño, & Alonso-Betanzos, (2012)	S	M
Fast clustering-based feature selection algorithm (FAST)	Song, Ni & Wang, (2011)	S	M
Genetic Algorithm with Dynamic Parameter setting (GADP)	Lee & Leu, (2011)	S	M
Counting Grid (CG)	Lovato, Bicego, Cristani, Jojic, & Perina, (2012)	R	B
Multi-Criterion Fusion-based Recursive Feature Elimination (MCF-RFE)	Yang & Mao, (2010)	S	B
multiple-filter-multiple-wrapper (MFMW)	Leung & Hung, (2008)	S	B

ReliefF, mRMR and GA (R-m-GA)	Shreem, Abdullah, Nazri & Alzaqebah, (2012)	S	B
Stratified Random Forest (SRF)	Ye, Wu, Huang, Ng & Li, (2013)	S	M
SVM-RFE with MRMR	Mundra & Rajapakse, (2009)	R	B

2.7 Algorithms Used in this Research Work

This section presents a brief description of the algorithms used in this research work and some other related issue.

2.7.1 Information Gain

Information gain (InfoGain) is one of the prominent attribute evaluation methods. It is a univariate filter technique which offers an ordered ranking of all the features and select based on the specified threshold (Hall & Smith, 1998). InfoGain measures the number of information (relevant) got from the class prediction by the presence or absence of the informative genes in a dataset. Su, Murali, Pavlovic, Schaffer & Kasif, (2003) explained a measure they referred to as information gain used to evaluate the relevancy of a gene and the class value. The entire range of data expression of the gene j is initially divided into two regions based on the specified threshold (average of its maximum and minimum expression values). 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 Equation 2.1.

$$Gain_j = \sum_{k=0}^1 \sum_{m=1}^2 \frac{i_{km}}{N} \log_2 \frac{i_{km}}{n_{vm}} - \sum_{k=0}^1 \frac{i_{k1} + i_{k2}}{N} \log_2 \frac{i_{k1} + i_{k2}}{N} \quad (2.1)$$

A gene with a larger InfoGain implies that it can provide more information to reduce the uncertainty in classification. In each dataset, the features subset of with non-

zero InfoGain will be selected. InfoGain can be applied only to discrete features but numeric features require discretization which is essential before estimate InfoGain. Entropy-based discretization process is usually utilized for gene expression data. InfoGain operates in a similar manner in which t-Statistic work by selecting features with larger values of InfoGain. The higher the value of InfoGain, the higher the chances of having good genes selection in a target class if the split is done on the variable with the highest gain score (Chuang, Ke & Yang, 2016). The pseudo-code of the InfoGain algorithm is presented in Algorithm 2.1.

Algorithm: InfoGain

Input: Data Sample D with features $F(x_1 \dots x_n)$

Θ Threshold value

Output: Optimal Feature Subset

Steps:

1. Apply attribute filter on training dataset D
 2. Compute Information gain by using Equation 2.1
 3. If $(IG > \Theta)$
 4. Select subset x with highest IG
 5. End
-

Algorithm 2.1: Pseudocode of Information Gain

2.7.2 Gain Ratio

Gain Ratio (GR) improves InfoGain by trying to overcome the bias of the InfoGain filter that selects attributes with a large number of outcomes by integrating “split information” of features into Information Gain measurement. The “split information” of a gene is achieved by computing how broadly and consistently it splits the data (Mitchell, 1997). GR also employs an iterative process where the selection of smaller sets of features is done in an incremental manner. These iterations halt when the specific predefined numbers of genes remain. Let us assume that a Microarray dataset has a set of classes

represented as k_i , ($i = 1, \dots, m$), and each gene j has a set of possible values denoted as V ; the discriminative power of a gene j is computed as shown in Equation 2.2:

$$GR(j) = \frac{Gain(j)}{split(j)} \quad (2.2)$$

in which:

$$split(j) = - \sum_{v \in V} \sum_{i=1}^m \frac{|s_v|}{|S|} \log \frac{|s_v|}{|S|} \quad (2.3)$$

where s_v is the subset of S , of which gene j has value. GR usually performs better than InfoGain feature selection (Turney, 2002). The major hitch of GR is that of its sensitivity to the number of the feature selected (Sharma & Dey, 2012). The pseudo-code of the GR algorithm is presented in Algorithm 2.2.

Algorithm: Gain Ratio

Input: Data Sample D with features $F(x_1 \dots x_n)$

Θ Threshold value

Output: Optimal Feature Subset

Steps:

1. Apply attribute filter on training dataset D
 2. Begin
 3. Compute Information gain by using Equation 2.1
 4. Compute Split by using Equation 2.3
 5. Compute Gain Ratio using Equation 2.2
 6. If ($GR > \Theta$)
 7. Select subset x with highest GR
 8. End
-

Algorithm 2.2: Pseudo-code of Gain Ratio

2.7.3 Chi-Squared Test

Chi-Squared (Chi) is a measurement test normally employed to compare the observed sample with presumed sample foreseen to obtain under a specific hypothesis. In gene features selection, it evaluates the association between the feature subset and its related class or category. This test is developed precisely for multiple tests having not less than two discrete outcomes (for example, normal and diseased gene). The Chi test is

usually testing for what scientists refer to as the null hypothesis, which postulates that there is an insignificant difference between the expected and observed outcome. The formula for evaluating Chi (χ^2) is shown in Equation 2.4.

$$\chi^2 = \sum_{i=1}^k \sum_{j=1}^r \left(\frac{ob_{ij} - ex_{ij}}{ex_{ij}} \right)^2 \quad (2.4)$$

where i is instance range from $i = 1, \dots, k$, j is gene range from $j = 1, \dots, r$, ob is observed outcome and ex is expected the outcome.

A Chi-Squared test is normally used to measure goodness fit between an observed and expected distribution of values, knowing that this helps to determine whether the expected outcomes fit into points of a curve, or testing a statistical hypothesis.

2.7.4 Bat Algorithm

The concept of Binary Bat Algorithm (BBA) cannot be comprehended except to firstly discuss the original bat algorithm (BA). 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 prey and/or navigating through the obstacles in the darkness. 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 feature. 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 returns to their ears (Griffin, Webster & Michael, 1965; Nakamura, Pereira, Costa, Rodrigues, Papa & Yang, 2012). Thus, bats can determine their distance from an object (Metzner, 1991). Apart from this, bats use this amazing phenomenon to be able to differentiate between an obstacle and prey and this gives them the chance to hunt even in absolute darkness (Schnitzler & Kalko, 2001). This feature of microbats' echolocation has

fascinated the attention of many researchers from various fields. Based on this characteristic of microbats behaviour, Yang, (2011) has developed a novel and fascinating Meta-heuristic optimization technique called Bat Algorithm (BA). This algorithm has been designed to mimic the band of bats chasing prey (foods) using their aptitude of echolocation. To model this behaviour mathematically, Yang, (2011) has idealized some rubrics as follows:

- 1) All bats utilize echolocation to determine their distance, and also use it to differentiate between food/prey and background barriers in some marvellous way;
- 2) A bat b_i flies arbitrarily with velocity $\mathcal{V}_i$ at position D_i with a fixed frequency $\mathcal{F}_{\min}$, adjust wavelength λ and loudness L_0 to hunt for prey. They regularly vary the wavelength (or frequency) of the sound produced and adjust 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 ways, Yang, (2011) assumes that the loudness fluctuates from a large (positive) L_0 to a minimum constant value $L_{\min}$.

The search space in BA can be explored through varying positions and velocity vectors (or updated positions 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 Equation 2.5, Equation 2.6 and Equation 2.7.

$$\mathcal{V}_i(t+1) = \mathcal{V}_i(t) + (D_i(t) - \mathcal{G}_b)\mathcal{F}_i \quad (2.5)$$

$$D_i(t+1) = D_i(t) + \mathcal{V}_i(t+1) \quad (2.6)$$

where t is time, $\mathcal{G}_b$ is the best solution attained so far and $\mathcal{F}_i$ represents the frequency of the i -th bat, which is updated in its course of iteration as in Equation 2.7:

$$\mathcal{F}_i = \mathcal{F}_{min} + (\mathcal{F}_{max} - \mathcal{F}_{min})\varphi \quad (2.7)$$

where φ is a random number of uniform distribution in $[0, 1]$. The BA used a random pace to enhance its ability in exploitation (possible solution) as in Equation 2.8. BA selects a solution amongst the best bats, and then random walks is employ to produce a new solution for individual bat:

$$D_{new} = D_{old} + \epsilon L^t \quad (2.8)$$

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 Equation 2.9 and Equation 2.10:

$$L_i(t + 1) = \alpha L_i(t) \quad (2.9)$$

$$\mathcal{R}_i(t + 1) = \mathcal{R}_i(0) + [1 - e^{(-\tau t)}] \quad (2.10)$$

where α and τ are constant parameters which lie between 0 and 1 used to update loudness rate L_i and pulse rate ($\mathcal{R}_i$). The pseudo-code of the Bat algorithm is presented in Algorithm 2.3.

In Algorithm 2.3, BA starts by initializing bat population and parameters such as frequency, loudness, velocity and pulse rate. It generates the best solution by iterating and updating frequency and other population parameters until it achieves the best and acceptable solution of $\mathcal{G}_b$.

Algorithm: Bat Algorithm

Initialize the bat population: D_i ($i = 1, 2, \dots, n$) and V_i

Define pulse frequency $\mathcal{F}_i$

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

While ($t < \text{Max number of iterations}$)

 Generate new solutions by adjusting frequency,

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

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

 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 ($\text{rand} < L_i$ & $f(D_i) < f(G_b)$)

 Accept the new solutions

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

End if

 Rank the bats and find the current G_b

End while

Algorithm 2.3: Pseudocode of Bat Algorithm (Yang, 2011)

2.7.5 Classification Algorithms

Classification is the technique for seeking a model that describes and differentiates classes or notion. The classification model serves as a means of determining class labels of a given (new) dataset. Classification is considered as a prognostic data mining task for 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 data (labelled class 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 used to predict the unknown class of the new dataset (Han, Pei & Kamber, 2011).

2.7.6 Decision Tree Classifiers

The process of learning in decision tree algorithm is centred on the splitting criteria principle. Decision trees had a structure similar to a flow chart, with a tree-like structure where instances 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. There are lots of decision tree classifiers available in the literature. The famous one includes, ID3, C4.5 (J48), C5 (Quinlan, 1986, 2014), and CART (Breiman, Friedman, Olshen & Stone, 1984), are widely used due to their high precision and accuracy rates (Panigrahi & Borah, 2018).

Decision trees are easy to build and comprehend due to their hierarchical structure. They are capable of modelling complex functions. They use the technique of recursively splitting of data into smaller subsets repeatedly by employing a greedy search approach to transverse the space of prospective decision tree branches and selecting optimum splits using the splitting criterion. The decision tree processes start from the root node, the first node with no parent nodes; then using best splitting criteria to determine and repeatedly divide the data into two nodes is usually referred to as child nodes. This process halts, when there are no more desirable or possible splits. The last set of nodes with no child nodes are known as leaves or terminal nodes. They achieve this task by employing some mathematical algorithms such as Information gain, Gini index, Chi-Squared test and so on, to determine variable and a corresponding threshold for splitting the data into two or more subclasses.

Generally, decision tree classifier involves two major steps; a growing phase and pruning phase (Abdulsalam, Babatunde, Hambali & Babatunde, 2015). 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 (Han et al., 2011). 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 process 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, leaf node represents each region (or class) in the dataset with an associated decision rule that corresponds to allot most frequent class label to its corresponding region of the observations it belongs to (Breiman et al., 1984).

2.7.6.1 Notation

The i -th observation in the sample dataset is denoted by the vector of its features (variables) x_i , and its dimensionality (the number of features) represented by p . The corresponding j -th value of the feature observation x_i , is represented by $x_{i,j}$, therefore, we refer to all the vector values as a single feature $x_{i,j}$.

The outcome variable (class label), corresponding to the observation x_i , is denoted by y_i . This is the best representation for classification problems, most especially

classification with the discrete variable that assumes a finite quantity of different values, each matching to a different class. The entire dataset is represented by the $n \times p$ matrix, where n is the number of dataset samples and X obtained by stacking all the feature vectors. Also, y denoted the n -dimensional vector of outcomes, which is achieved by concatenating the outcome values of each dataset observation.

2.7.6.2 Splitting Criteria

In the training process, each step of the growing stage seeks out for a feature $x_{i,j}$ and a split value sto , then split the current observation sets X^1 into two smaller and more coherent (pure) subsets $S_1 = \{x_i \mid |x_{i,j} \leq s\}$ and $S_2 = \{x_i \mid |x_{i,j} > s\}$. The famously used purity criteria for classification tasks are the Gini index (Equ. 2.11) and the Cross-entropy index (Equ. 2.12) (Panigrahi & Borah, 2018):

$$\text{Gini index} = \sum_{k=1}^k \hat{p}_k (1 - \hat{p}_k) \quad (2.11)$$

$$\text{Cross entropy index} = \sum_{k=1}^k \hat{p}_k \log \hat{p}_k \quad (2.12)$$

where $\hat{p}_k$ is the sample fractions for each class k associated to a specified subset S_1 . Both indexes, equal to 0 when $\hat{p}_k = 0$ or $\hat{p}_k = 1$ and attain maximum value when $\hat{p}_k = 1/2$ which mean that subsets are homogeneous with the identical value of the class label (outcome variable) are preferred.

2.7.6.3 Stopping Criteria

Obviously, it is impossible to have leaf nodes that are more than samples in the training set. This configuration helps in determining corresponding maximal pure leaf nodes that can be obtained. Nevertheless, in this circumstance, generalization performances of new samples may be poor. To prevent this situation of tree complexity explosion, the growing stage must be on halt early in some nodes. Popular stopping criteria

include restricting the number of samples per node (minimum samples per node) or the minimize impurity level to split a node.

2.7.6.4 Pruning Criterion

It was reported that imposing strict stopping criteria may lead to underfitting trees (Breiman et al., 1984). However, generalization performances can be improved by training the decision tree without restriction and then execute pruning with the weakest link pruning technique. The weakest link pruning technique works on the contrary direction of training procedure: begin with the complete tree, a series of smaller trees is built let say $T_0, T_1, \dots, T_k$, where T_0 is the complete tree and T_k is a single node tree. Sub-tree T_{i+1} can be generated by collapsing internal node of the individual sub-tree T_i into a leaf node, one at a time, and the error rate increase are estimating over the pruned leaves using Equation 2.13:

$$\alpha = \frac{err(T_{i+1}, X^i) - err(T_i, X^i)}{leaves(T_i) - leaves(T_{i+1})} \quad (2.13)$$

where $err(T_i, X^i)$ is the tree T error rate on the observation sets X^i , and $leaves(T)$ denote to the number of leaves on tree T . Once the estimation of all the collapsed sub-trees T_i has been performed, the sub-tree with the lowest value of α is chosen as T_{i+1} . The criterion α is also referred to as complexity parameter which is used in determining the apt quantity of complexity (the overall number of leaves) to be retained from the original tree. Lastly, the best-pruned tree is selected from the series of estimated sub-trees performance from a pruning dataset that is set aside during the training of T_0 . But in a situation when there is no enough data to be divided into training and pruning datasets, cross-validation techniques can be applied instead, although this results in higher computational cost. The pseudocode for decision tree is shown in Algorithm 2.4.

Algorithm: Decision Tree

GenDecTree(Sample S, Features F)

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. $V = \{v \mid v \text{ a possible outcome of } root.test_condition\}$
5. **For each** value $v \in V$:
 - a. $S_v = \{s \mid root.test_condition(s) = v \text{ and } s \in S\}$;
 - b. *Child = TreeGrowth (S_v, F);*
 - c. *Add child as descent of root and label the edge {root $\rightarrow$ child} as v*
6. **return** *root*

Algorithm 2.4: Pseudocode of Decision Tree

Decision tree algorithm (Algorithm 2.4) begins by testing for stopping criteria and then create the leaf with the class label, if the criteria are met its return leaf. But if the criteria are false, it labels the leaf as root and continues splitting the leaves until splitting criteria are met and then label the leaf with the class they belong to.

2.7.7 Classification and Regression Tree

Classification and Regression Tree (CART) (Breiman et al., 1984) is based on Hunt's algorithm; it can process both categorical and continuous 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 attribute selection measure. CART splits training datasets into binary; therefore, it generates binary trees. Gini Index measure does not involve in 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 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.

2.7.8 C4.5

C4.5 Decision tree also known as J48 Classifier algorithm is one of the famous employed and practical techniques for inductive inference (Polat & Güneş, 2009). It is an approach that used the approximation of discrete-valued functions, which is robust to noisy data and has proficiency for learning disjunctive expressions (Mitchell, 1997; Quinlan, 1986).

In this algorithm, the approximating discrete-valued learning function is represented by a decision tree. The learned trees can likewise be denoted as sets of if-then rules to ease human readability. C4.5 classifies a new instance by building a decision tree from the attribute values of the given training set. The moment it comes across the training set, it recognizes the attribute that is responsible for categorizing the various instances most accurately. The possible feature values with no ambiguity are assigned to the concerned branch by terminating it.

2.7.9 Random Forest

A well-known classification method employed in numerous classification domains is Random Forest (RF) (Geetha, Sivasubramanian, Kaliappan, Vimal & Annamalai, 2019; Lin, Fang, Xiao & Chou, 2011; Seera & Lim, 2014). RF is also considered as bagged

decision trees (Biau, 2012), that comprises a group of weak classifiers to articulate a robust learner (Geetha et al., 2019). RF adopts two approaches

1. Classification technique
2. Regression Tree (CART) technique (Genuer et al., 2008).

This technique is the development of several decision trees centred on bootstrap aggregation (bagging) method (Liaw & Wiener, 2002). 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 chooses a subset randomly to model an independent decision tree. The process is iterative which splits the randomly selected subset from the root node to a child node (Suresh, Udendhran & Balamurgan, 2019). 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) (Suresh et al., 2019). The overall classification is based on the majority obtained trees voting. Algorithm 2.5 depicts the steps involved in building an RF technique.

Algorithm: Random Forest

Parameters: *N* - Quantitative amount of bootstrap samples
 M - Total number of features
 m - Sample size
 k - Next node

Step:

1. **Creates** *N* bootstrap samples from the dataset.
2. Every node (sample) takings 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.

End.

Algorithm 2.5: Pseudocode of Random Forest Classification (Geetha et al., 2019)

In Algorithm 2.5, RF starts with creation of N bootstrap samples of the dataset and each node taking random features of m size to build the best split for the tree. The split continues repeatedly until the leaf node is attained and the tree is completed. The algorithm trains each bootstrap separately and selecting the best tree through voting.

2.8 Related Work

Nature is a combination of inspiration and motivation. Researchers and computer scientists are motivated by nature and are trying to proffer solution to numerous problems by observing nature. Nature-inspired (Stochastic) algorithms are proposed for providing solutions to multifaceted and difficult combinatorial problems. Algorithms such as genetic algorithm (GA), Ant Colony Optimization (ACO), Simulated Annealing (SA) and Particle Swarm Optimization (PSO) are at the leading research areas in feature subset selection (Friedman, Hastie & Tibshirani, 2001; Martens, 2001; Moody & Darken, 1989; Pirooznia, Yang, Yang & Deng, 2008). These approaches proficiently eliminate feature redundancy and interaction and without involving the obstructive monotonicity assumption. Though, these approaches are computationally costly. The best instance is an Artificial Neural Network (ANN) which is built to mimic the design and functionality of the human brain. Beside ANN, many Bio-inspired algorithms such as Ant Colony Optimization (ACO) algorithm, Bee Colony Optimization (BCO) algorithm, Particle Swarm Optimization (PSO) algorithm, and many others were designed mainly with the intention of proffering solutions to optimization problems. However, they are now extended to solve diverse problems. Bat algorithm is one Bio-inspired algorithm that has been used in diverse applications (Yang, 2013).

Gandomi, Yang, Alavi & Talatahari, (2013) 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) (Wang & Guo, 2013). Bat algorithm was also used in solving optimization problems in the field of engineering (Yang & Gandomi, 2012). Bat algorithm was employed to select features from numerous digital data sets for the first time in the literature (Nakamura et al., 2012). Rodrigues et al., (2014) presented a combination of 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 problem. Meta-heuristic Bat algorithm was used to classify Microarray data by Mishra, Shaw & Mishra, (2012). Bat algorithm was also employed for clustering by combining traditional K-means clustering with simple Bat algorithm (Tang, Fong, Yang, & Deb, 2012).

In recent times, many researchers proposed hybrid approaches tapping the advantages of both filter and wrapper methods. Some common cases of hybrid algorithms found in literatures include correlation-based feature selection algorithm and genetic algorithm (Nguyen & Rocke, 2002a), t-statistics and GA (Mitchell, 1997), Principal Component Analysis (PCA) and ACO algorithm (Nguyen & Rocke, 2002b), mutual information and GA (Narayanan, Keedwell & Olsson, 2002), Chi-Squared approach and multi-objective optimization algorithm (Nguyen & Rocke, 2002b).

Wang, Chu & Xie, (2007) 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 the second stage involved classification of genes using Fine

classifier. Divide and conquer technique were employed to achieve better accuracy performance. They used two different datasets in this research, Lymphoma Data and SRBCT Data. They employed a FNN algorithm 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 a FNN and SVM classifiers.

Veerabhadrapa & Rangarajan, (2010) developed a model that ranked the genes based on a 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 suitability of dataset type. The learning algorithm was evaluated utilizing all possible gene combinations and the optimal gene combination was reported. Their results showed that SVM-OAA yielded average testing accuracy of 99.39% and while LDA: SVM-OAA yielded 98.72% average testing accuracy.

Ghorai, Mukherjee, Sengupta & Dutta, (2010) developed a hybrid computer-aided diagnosis (CAD) framework based on filters and wrapper techniques for cancer classification using Microarray gene expression profiles. Minimum redundancy maximum relevance (MRMR) ranking algorithm was utilized for feature 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, (2011) proposed a method that hybridized the enrichment score and SVM classifier for cancer classification in Microarray data. The Lymphoma dataset used is randomly partitioned into train and test dataset. 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.

El Akadi et al., (2011) presented a framework that comprised of a two-stage feature selection approach for Microarray data. The first stage involved the filtering of genes using Minimum Redundancy Maximum Relevance (MRMR). While GA is used in the second stage to generate the gene subsets and both Naïve Bayes (NB) and support vector machine (SVM) were employed for the classification and assessment analysis. The best results obtained from their study showed that top 15 genes from Lymphoma yield 100% classification accuracy, 97.26% classification accuracy from Lung data and 100% classification accuracy from Leukemia data.

Rajeswari & Reena, (2011) proposed an approach for tumour cells identification. They used Support Vector Machine (SVM) and Fuzzy Neural Network (FNN) to solve the classification problem. Liver cancer dataset was utilized to analyse the proposed method. Their results revealed that the proposed method had the capability to diagnose cancer

diseases considerably when it was compared with the conservative procedures of a cancer diagnosis. Fuzzy Neural network recorded performance accuracy between 92-96%.

Sahu & Mishra, (2012) 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 optimization technique was carried out using Particle Swarm Optimization (PSO) for dimensionality reduction. K-nearest neighbour (KNN), Probabilistic Neural Network (PNN) and Support vector machine (SVM) were utilized as classifiers. PNN recorded 96%, 96.34%, 87.32% and 98.12% classification accuracy from Leukemia, Colon, DLBCL and Breast cancer dataset respectively.

Swathi, Babu, Sendhilkumar & Bhukya, (2012) designed 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 dataset. The model achieved better results for unsupervised machine learning breast cancer datasets with 92% accuracy.

Dev, Dash, Dash & Swain, (2012) 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 breast cancer classification. In this work, the outcomes of the three classifiers were examined and compared. They reported 56.12% classification rate for BPN and 63.34% for FLANN, while PSOFLLANN gave the best classification rate of 92.36%.

Shreem et al., (2014) applied a new approach of hybridization that combines Harmony Search Algorithm (HSA) and the Markov Blanket (MB), termed HSA-MB for gene selection in classification tasks. HSA employs Naive Bayes'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 yielded 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, (2014) proposed a new fusion machine learning (ML) reduction method to improve the cancer classification 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 - KNN and SVM; to evaluate Microarray gene expression profiles. Four public cancer Microarray (Lymphoma, Leukemia SRBCT, and Lung) datasets were utilized to measure the proposed approaches, and they succeeded in accomplishing their aim. The approach of selecting genes only used for training samples and completely excluding the testing samples from the process of classifying was employed to obtain better performance and valid the results. The combination of (t-test + SVM) was reported to have produced higher classification accuracy compared to other combination approaches of (t-test + KNN, CS + SVM and CS + KNN), where they reported classifier accuracy of 100% using peak ranked value of 2 and 3 genes from Lymphoma and SRBCT data respectively. They also obtained 94.12% from Leukemia using 6 genes as peak ranked value. Their results revealed that the proposed reduction

approach reached favourable results of the number of genes complemented to the learner classifiers at the same time with the accuracy of the classifier.

Alshamlan et al., (2015) presented a new hybrid gene selection method 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 attained 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% classification accuracy with 4 genes from Leukemia1, 100% classification accuracy 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.

Doreswamy & UmmeSalma, (2016) designed a Binary Bat Algorithm (BBA) based Feedforward Neural Network (FNN) hybrid model. The authors exploited the advantages of BBA and efficiency of FNN to classify three standard breast cancer datasets into malignant and benign 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 classification yielded 92.61% accuracy rate for training data and 89.95% for testing data.

Alomari et al., (2017) presented a hybrid gene selection of filter technique - Minimum Redundancy Maximum Relevancy (MRMR) and a wrapper technique - Bat algorithm (BA) using Microarray dataset. MRMR was employed to discover the most significant genes from the entire gene in gene expression dataset, and BA was used to search for the most informative gene subset from the reduced set produced by MRMR that

aided in detecting cancers. The classifier method used was a support vector machine (SVM) with 10-fold cross-validation which helped in evaluating the BA. To reveal the performance accuracy of the proposed method, three different Microarray datasets were used, which include: colon, Breast, and Ovarian cancer dataset. The same procedure was carried out with Genetic algorithm (GA) to perform the comparison with MRMR-BA and the results revealed that their proposed approach was capable of finding the small size of gene subset with high classification rate. 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. Table 2.6 presents a summary of other recent related approaches.

Table 2.6: Summary of Related Works

No.	Author/Year	Title of Work	Method Used	Conclusion	Limitation
1.	Alelyani, (2021)	Stable Bagging Feature Selection on Medical Data	The author employed ensemble technique based on bagging method for feature selection on 4 microarrays datasets. Also, to ensure that the improvement of selection stability is not due to random behavior, they used SVM and KNN classifiers and Jaccard Index was used to evaluate selection stability.	The proposed approach was able to improve selection stability while maintaining classification accuracy due to the intrinsic power of reducing learning variance.	To choose the proper number of selected features is an open research problem where stability might be an indicator of the optimal number. Also, The datasets used were binary class datasets
2.	Wisesty, Sthevanie & Rismala, (2021)	Momentum Backpropagation Optimization for Cancer Detection Based on DNA Microarray Data	Genetic Algorithm was employed as a feature selection technique and Neural Network with the Momentum Backpropagation training algorithm as a microarray data classification method. The optimization of Momentum Backpropagation algorithm was done by adding an adaptive learning rate scheme.	The proposed method proved that it capable of reducing the number of epochs required in the training process from 390 epochs to 76 epochs compared to the Momentum Backpropagation algorithm. The results of proposed approach record high accuracy of 90.51% for Colon Tumor data, and 100% for Leukemia, Lung Cancer, and Ovarian Cancer data.	Datasets used were binary class data except Leukemia.
3.	Sathya & Priya, (2020)	A Search Space Enhanced Modified Whale Optimization Algorithm for	The authors proposed a Search space enhanced Modified Whale Optimization	The authors concluded that the classification accuracy was improved by the processes. Overall result	Binary Class dataset was used and SMWOA may get stuck in a part of the

		Feature Selection in Large-Scale Microarray Datasets	Algorithm (SMWOA) for cancer detection. To enhance the exploration and exploitation strategy of MWOA. A non-linear dynamic strategy was introduced in SMWOA to update the control parameters, and balances the exploration and exploitation abilities of MWOA. Also, to proffer solution to problem of local optima for cancer detection, a Levy-flight strategy was introduced in SMWOA.	showed that accuracy, specificity, sensitivity, F1-score and average error of SMWOA-ANN are 6.7%, 5.6%, 7.3% and 5.6% greater than MWOA-ANN respectively for cancer detection.	Pareto-optimal problem because multiple objectives were used in the SMWOA.
4.	Abdulsalam et al., (2020)	Performance Evaluation of ANOVA and RFE Algorithms for Classifying Microarray Dataset Using SVM	The study presented a comparison study on Analysis of Variance (ANOVA) and Recursive Feature Elimination (RFE) feature selection techniques. They evaluated their relative performance using SVM kernel classification approach.	The result showed that SVM-RFE achieved 93% compared to ANOVA with 87% classification accuracy.	Only colon dataset was used.
5.	Ebrahimpour, Nezamabadi-Pour & Eftekhari, (2018)	CCFS: A Cooperating Coevolution Technique for Large Scale Feature Selection on	Cooperative Coevolving feature selection (CCFS) for feature selection and C4.5 and Naïve Baye as classifiers with six different microarray	The proposed approach can be applied on hefty datasets without any pre-processing and It converges to near-global optimum since Cooperative Coevolving	All the datasets used were binary Class and redundancy data is not taken care of.

		Microarray Datasets	datasets	approach to do the searches.	
6.	Maldonado & López, (2018)	Dealing with high-dimensional class-imbalanced datasets: embedded feature selection for SVM classification	They proposed embedded strategy penalizes cardinality of the feature set via the scaling factors method, and used the approach with two SVM formulations designed to deal with the class-imbalanced Problem, (Cost-Sensitive SVM, and Support Vector Data Description). Their proposed concave formulations are solved through a Quasi-Newton update and Armijo line search.	They suggested that KP-CSSVM is an excellent alternative for feature selection and class-imbalance classification. Since the technique was designed to solve the latter issue and penalizes the use of features by adjusting the kernel width in an iterative process.	AUC was the only metric used and datasets used were all binary Class data
7.	Dashtbana et al., (2018)	Gene Selection for Tumor Classification Using a Novel Bio-inspired Multiobjective Approach	Their presented approach is a hybrid model using the Fisher criterion and the traditional Bat Algorithm with refined formulations, effective multi-objective operators, and novel local search strategies employing social learning concepts in designing random walks. The prediction accuracy of each subset was evaluated using four widely-used classifiers including SVM, KNN, NB and Decision Tree (DT)	Experimental results unveil new combinations of informative biomarkers have an association with other studies. Whereas in prostate cancer dataset, the proposed method achieved the highest reported accuracy with a significantly lower number of genes	The performance of the SRBCT dataset is not too good because it is only multiclass dataset used for the experiment.

8.	Dash, (2018)	An Adaptive Harmony Search Approach for Gene Selection and Classification of High Dimensional Medical Data	The author proposed a hybridized harmony search and Pareto optimization approach for feature selection in high dimensional data classification problem. In the first stage, an adaptive harmony search algorithm for gene selection with probability distribution factor for optimal gene ranking is implemented. This selection is further refined applying a bi-objective Pareto based feature selection technique to select an optimal minimum number of top-ranked genes (a wrapper-filter approach for feature selection on high dimensional databases). Four classifiers are used such as ANN, KNN, NB and SVM	As the presence of relevant genes minimizes the classification error, so it is used as a fitness function for feature evaluation. The classification result showed that the performance of the proposed model with SVM classifier is better than that of the result obtained using other classifiers.	All datasets used were binary class data
9.	Gao, Ye, Lu & Huang, (2017)	Hybrid Method Based on information Gain and Support Vector Machine for Gene Selection in Cancer Classification	They proposed a hybrid gene selection method, Information Gain-Support Vector Machine (IG-SVM). IG was initially employed to filter irrelevant and redundant genes. Then, further removal of redundant genes was performed using SVM to eliminate the noise in the	IG-SVM achieved a classification accuracy of 90.32% for colon cancer, which is difficult to be accurately classified, only based on three genes including CSRP1, MYL9, and GUCA2B	Only Binary class dataset was used.

			datasets more effectively. Finally, the informative genes selected by IG-SVM served as the input for the LIBSVM classifier.		
10.	Passi, Nour & Jain, (2017)	Markov Blanket: Efficient Strategy For Feature Subset Selection Method For High Dimensional Microarray Cancer Datasets	In this work, two techniques were used. The first one is dependent on the Memetic Algorithm (MA), while the second one is dependent on MRMR for feature subset selection methods. Both techniques were hybridized with GA. Also, four common classifiers were employed Lazy-nearest neighbour, linear SVM, Naïve Bayes, and AdaBoost ensemble to evaluate both approaches. They compared the performance of both the techniques with Bayesian Network-based Markov Blanket for feature selection method and KNN classifier.	The main contribution of this paper is to show the importance of the Bayesian network and Markov Blanket for feature subset selection and learning classification models. The results of the experiment revealed that the approach can build high quality and effective learning classification algorithms.	Only Running time and Average test Error are the performance metric used and the approach perform better only in binary class datasets.
11.	Dashtban & Balafar, (2017)	Gene Selection for Microarray Cancer Classification Using a New Evolutionary Method Employing Artificial	They proposed a novel evolutionary method based on genetic algorithms and artificial intelligence to identify predictive genes for cancer classification. A filter	Their presented method practically exploited the concept of random restart hill climbing to avoid biased or improper initializations by initiating a few populations	Most of the datasets used have features that are less than 10,000 and there are lot of dissimilarity among the genes

		Intelligence Concepts	method was first applied to reduce the dimensionality of feature space followed by employing an integer-coded genetic algorithm with dynamic-length genotype, intelligent parameter settings, and modified operators. Three classifiers -SVM, Naïve Bayes and KNN were used to evaluate the performance of the proposed algorithm.	of individuals with randomly generated length. The proposed method obtained its best results in combination with Fisher score and SVM.	selected (stability in question here).
12.	Dash, (2017)	A Two-Stage Grading Approach for Feature Selection and Classification of Microarray Data Using Pareto Based Feature Ranking Techniques: A Case Study	The author proposed a bi-objective ranked based Pareto front technique. Performances of the models are evaluated using three classifiers, such as ANN, NB and KNN.	The model performance changes with the datasets and classifier. Thus a two-stage grading approach was used to obtain a superior optimization model.	All the datasets used were binary Class and most of the datasets have features that are less than ten thousand.
13.	Mohammadi et al., (2016)	Robust and stable gene selection via Maximum–Minimum Correntropy Criterion	Filter-based feature selection via Maximum–Minimum Correntropy Criterion and SVM with linear kernel as classifier was used with 25 different datasets. Time, Accuracy and stability of	The results revealed that less than 10 genes among thousands of genes can possible to achieve significantly higher accuracy in cancer classification	The Author trade-off accuracy performance with time and memory space.

			feature selected were the metrics used for performance evaluation of the approach		
14.	Sharbaf, Mosafer & Moattar, (2016)	A Hybrid Gene Selection Approach for Microarray Data Classification Using Cellular Learning Automata and Ant Colony Optimization	They employed Hybrid of Fisher criterion as filter approach and wrapper approach which is based on cellular learning automata (CLA) optimized with ant colony method (ACO), the performance is evaluated with K-nearest neighbour; support vector machine and naïve Bayes on four microarray data.	The evaluations of the approach confirm that the proposed approach can find the smallest subset of genes while approaching the maximum accuracy.	Their experiment is time-consuming and only used ROC and Accuracy as the performance metrics.
15.	Bolón-Canedo et al., (2015)	Distributed feature selection: An application to microarray data classification	Distributed filter approach was adopted with 5 different filter algorithms such as InfoGain, ReliefF, INTERACT, Consistency-based Filter, and CFS and 4 classifiers (C4.5, K-NN, SVM and Naive Bayes) were used to evaluate the performance of the approach.	The most important advantage of the distributed approach utilized is that, it greatly minimizes the execution time while still maintaining accuracy at reasonable levels, and sometimes even improving it. This outcome is based on the idea of divide-and-conquer since, in some cases, the result obtained by the learner can be more accurate if it is focused on a local region of	Test and Train dataset were selected randomly and the number is not uniform for all the datasets used. Also, Binary class datasets were used for the experiment.

				the data.	
16.	Cao, Zhang, Wang, Li & Yang, (2015)	A Fast Gene Selection Method for Multi-Cancer Classification Using Multiple Support Vector Data Description	This paper proposed a novel fast feature selection method based on multiple SVDD and applies it to multi-class microarray data. Recursive feature elimination (RFE) scheme was introduced to iteratively remove irrelevant features, the proposed method is called multiple SVDD-RFE (MSVDD-RFE). KNN and SVM classifiers were used to evaluate the performance of the proposed method.	SVDD cannot address the problems with multiple classes since it only considers the target class. In addition, it is time-consuming when applying SVDD to gene selection.	The result of the Lung Cancer dataset was unsatisfying and SVDD is time-consuming for gene selection.
17.	Santos, Datia & Pato, (2014)	Ensemble feature ranking applied to medical data	The FS algorithms used in the ensemble are: InfoGain, Gain Ratio, Symmetrical Uncertainty and Chi-Squared and learning algorithms used SVM, Bagging using the RPART function (BAG), Random Forest (RF) and Naive Bayes (NB) on KDD Cup 2008 -Breast Cancer dataset	They make uses of ensemble FS by combining 5 different Feature Ranking algorithm to select relevant feature for high dimensional data and they were able to develop better classification models.	Only breast dataset was used.
18.	Mao, Cai & Shao, (2013)	Selecting Significant Genes	In this work, Randomization test (RT) is used as a gene	The rationality of the results is validated by multiple	All datasets used were binary class data

		by Randomization Test for Cancer Classification Using Gene Expression Data	selection method for dealing with gene expression data and a statistic derived from the statistics of the regression coefficients in a series of partial least squares discriminant analysis (PLSDA) models is used to evaluate the significance of the genes.	linear regression (MLR) modelling and PCA, Compared with the results obtained in previous studies, the superiority of the method is proved. Therefore, the method may be an alternative tool for classification using the expression data.	
--	--	--	--	--	--

CHAPTER THREE

RESEARCH METHODOLOGY

In this chapter, details of the proposed algorithms, datasets, and stages involved in this study were discussed. Critical analysis to support the choice of the datasets were addressed. Furthermore, the data preparation process which includes the treatment of missing data and feature selection were also discussed.

3.1 Conceptual Framework

The methodology for this study consists of three main steps as shown in the conceptual framework in Figure 3.1. Though there is no concrete proof in the literature that indicates the kind of features selection method that is best for Microarray expression data (Li, Liu, Yang, Zhou & Lu, 2013), therefore, different methods of implementation were selected, analysed and their results were compared.

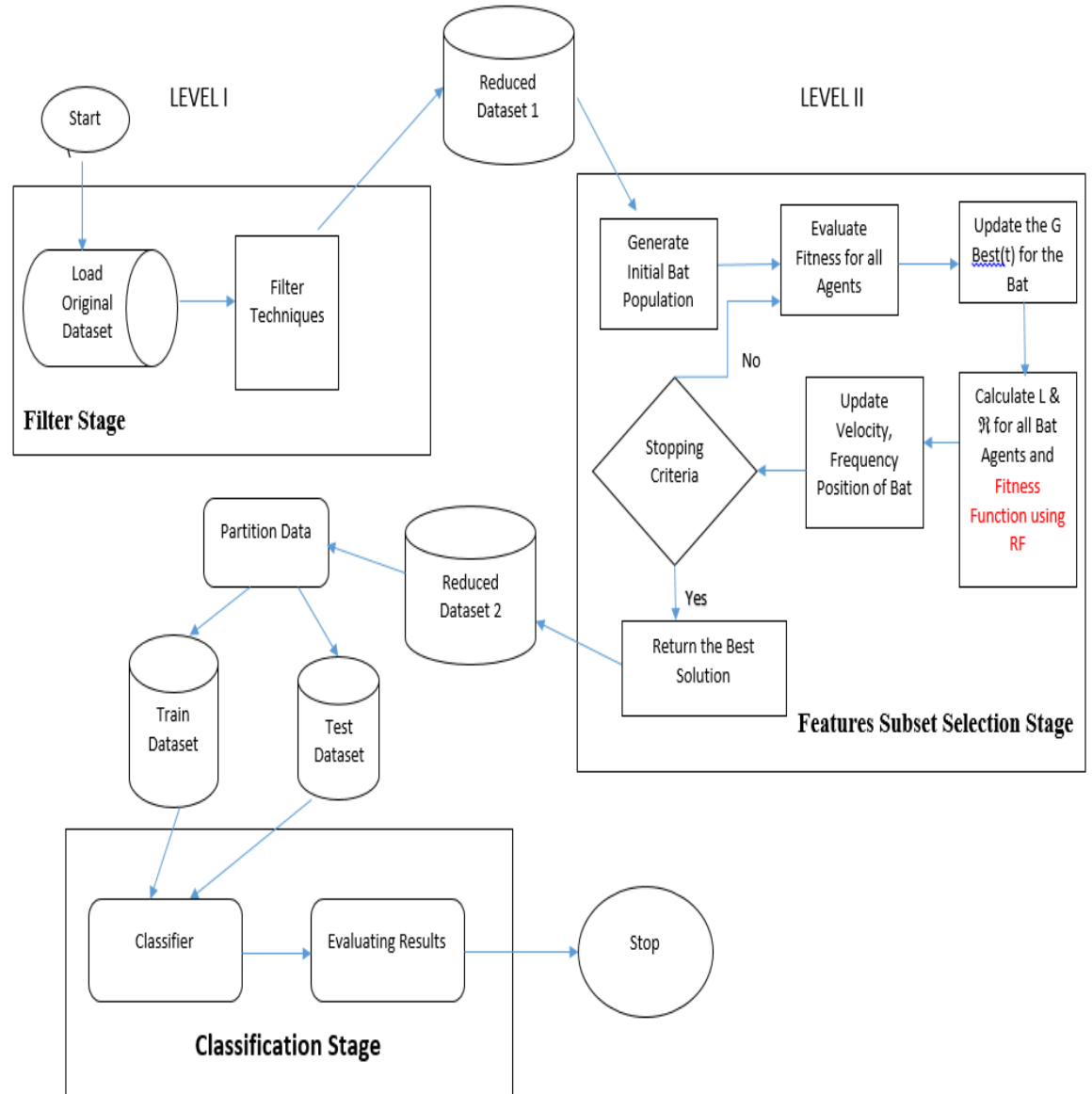


Figure 3.1: Conceptual Framework

Note: Filter algorithms used were: Information Gain, Gain Ratio, and Chi-Squared, while Modified Binary Bat (MBB) Algorithm was used as subset selection algorithm and Random Forest (RF) as the fitness function.

In the first phase (Figure 3.1), a filtering technique was employed to reduce the number of genes and to select the top-ranked relevant genes; three famous filter techniques (Chi-Squared, Information Gain and Gain Ratio) were utilized, compared, and exploited in

the whole process. Then, an evolutionary algorithm for features subset technique, which is based on the concept of Binary Bat algorithm, was designed to exploit the subset of highly relevant genes in the second phase and decision trees algorithm were also utilized for classification at the last stage. R programming language was used to implement the algorithms.

The following are a brief description of the stages involved in the conceptual framework.

3.2 Datasets

Microarray dataset is usually represented as M by N matrix, where N is columns (the corresponding gene expressions) and M is rows (the number of experimental samples). In this study, eight public available Microarray cancer datasets were utilized to evaluate the proposed method for gene selection and cancer classification using four different algorithms. A brief description of these datasets is shown in Table 3.1.

The performance evaluation of the proposed method was done by performing numerical experiments on selected 8 Microarray cancer datasets with different sizes of sample, features, and classes. The selected datasets comprise of both Binary and Multiclass data which were characterized with the problems of class imbalance, missing value and overlapping or dataset shift. The reason for including multiclass datasets is because the classification tasks are more complex.

Table 3.1: Datasets Description

	Datasets	#Gene	#Instance	#Class
1	Central Nervous System (CNS)	7129	60	2
2	Lung Cancer	12600	203	5
3	Ovarian Cancer	15154	253	2
4	Leukemia_3c	7129	72	3
5	Leukemia_4c	7129	72	4
6	Lymphoma	4026	66	3
7	MLL	12582	72	3
8	SRBCT	2308	83	4

All these datasets are available at <http://csse.szu.edu.cn/staff/zhuzx/Datasets.html> (Zhu, Ong & Dash, 2007)

3.2.1 Data Pre-processing

Data pre-processing is the first critical 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 contains missing values, therefore, the datasets were first treated to take care of missing values it contained.

In the pre-processing stage, 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, though they are just simple assumption applied to build them. Imputation with mean approach was adopted to treat the missing value in this work. Substituting the mean for missing values is acceptable and usually, gives satisfactory results. This approach was done by computing the mean of each column that contains the missing value and then assigned computed mean to the missing value.

3.3 Filter Stage

In this stage, each of the datasets was subjected to three different filter algorithms (Chi-Squared, Gain Ratio, and InfoGain) to select relevant features from the original datasets. Section 2.7 described the details of all filter algorithms used in this research work. All three filter algorithms used were ranker filters. One of the wrong assumptions about learning algorithms and feature selection algorithms is the independence of features. 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 interactions that prevent accidental elimination of relevant features that can result in the loss of useful information and at the same time resulted in poor diagnosis performance.

In each of the algorithms, threshold of ≥ 0.4 , ≥ 0.6 and ≥ 0.8 were set to obtain the features above that ranking threshold except in CNS dataset where ≥ 0.3 , ≥ 0.4 and ≥ 0.6 to select top 30%, 40% and 60% respectively for Chi-Squared; ≥ 0.2 , ≥ 0.3 and ≥ 0.4 to select top 20%, 30% and 40% respectively for Gain Ratio and ≥ 0.1 , ≥ 0.2 and ≥ 0.3 to select top 10%, 20% and 30% respectively for InfoGain were used, because the dataset did not have many features above 0.5 ranking.

3.3.1 Threshold Values

In this research work, the effort is on trying to improve computational efficiency and accuracy of the classification, ranking approaches that order all initial features are preferred, and then an additional thresholding step is also vital. As stated earlier, the FS approaches used in this study are all rankers (they sort all features) and therefore, it is

essential to establish a threshold cut-off to select a practical subset of features. Most studies in the literature use thresholds that retain different percentages of features (Seijo-Pardo, Bolón-Canedo & Alonso-Betanzos, 2019).

Fixed thresholds of 0.8, 0.6 and 0.4% were used to select the top 80%, 60% and 40% of the features of the ordered ranking respectively. After selection of relevant features with the three filter algorithms, their results were compared and the best filter algorithm was chose to combine with MBB algorithm for further subset selection.

3.4 Subset Selection Stage

In this stage, the Modified Binary Bat (MBB) Algorithm was employed in the proposed method. However, there is a need to briefly discuss the concept of Binary Bat Algorithm first before explaining MBB Algorithm.

3.4.1 Binary Bat Algorithm

In an incessant version of BA, the artificial bat can be moved 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 position vector or solution is in series of 0's and 1's binary bits. As binary search space deals with two numbers (0 and 1) only, performing updating positions process cannot be done using Equation 2.6 (in Section 2.7.4). Hence, a transfer approach needs to be established to mirror the change in velocity vector values and the elements of position vector from '0' to '1' or vice versa.

Nakamura et al. (2012) have presented a binary version of the Bat Algorithm limiting the new bat's position to only binary values using a sigmoid function as shown in Equation 3.1.

$$S(v_i^j) = \frac{1}{1 + e^{-v_i}} \quad (3.1)$$

So, substitute Equation 2.7 with equation 3.2

$$D_i^j = \begin{cases} 1, & \text{if } S(v_i^j) > \sigma, \\ 0, & \text{otherwise,} \end{cases} \quad (3.2)$$

where $\sigma \sim U(0,1)$. Therefore, Equation 3.2 can yield binary values only for each bat's coordinates in the Boolean lattice, which indicate the presence or absence of the features. Alomari et al., (2017) computed fitness function with Equation 3.3 and employed SVM classifier to determine its accuracy.

$$\text{Fitness (R)} = \alpha \gamma R(D) + \beta \frac{|C|}{|C| - |R|} \quad (3.3)$$

Where average classification accuracy rate was denoted by $\alpha \gamma R(D)$ and it was obtained by SVM classifiers with ten cross-validation. In each cross-validation, prediction model was built by SVM on the training dataset, based on gene subset R and class label (Decision D), and evaluation was performed on the testing dataset to obtain classification accuracy for prediction model.

$|R|$ is the length of selected gene subset. $|C|$ is the total number of genes. α and β are two parameters related to the significance of classification quality and subset length. $\alpha \in [0;1]$ and $\beta = (1-\alpha)$. The classification accuracy is more vital than subset length.

3.4.2 Description of the Modified Binary Bat Algorithm

There is a high tendency for BA based feature selection to trap into local optima during the process. Hence, this motivates the design of MBB Algorithm which used a global search technique with some additional mechanisms, targeting at enhancing the

results of BA-based algorithm for feature selection. The details of the proposed modified BBA for feature selection is given as follows.

3.4.2.1 Encoding and Fitness Functions

In MBB Algorithm, 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 depicted by Equation 3.4.

$$D_i = (D_i^1, D_i^2, \dots, D_i^j, \dots, D_i^n), \quad i = 1, 2, 3, \dots, m \quad (3.4)$$

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

Each of the candidate solutions is determined by a fitness function defined in Equation 3.5, where Acc represents the classification accuracy attained by Random forest classifier with ten cross-validation and $S(D)$ is the total amount of selected features. From Equation 3.5, two different components determine the fitness of each bat and these components weighted differently. The value of α is assigned at the beginning of the experiment. Equation 3.5 was used to compute the fitness function so as to solve the problem of data imbalance in the datasets as well as to cope with multiclass data. The previous approach was used for binary class datasets only.

$$fit(D) = \alpha \times Acc + (1 - \alpha) \times \frac{n - S(D)}{n} \quad (3.5)$$

In MBB algorithm, Equation 3.4 is used to replace Equation 3.3 in BBA. The pseudocode of the MBB algorithm is shown in Algorithm 3.1.

Algorithm: Modified Binary Bat Algorithm

1. **Initialize Bat population:** D_i ($i = 1, 2, \dots, n$), rand (0 or 1) and $\mathcal{V}_i = 0$
 2. Define pulse frequency $\mathcal{F}_i$
 3. Initialize pulse rates $\mathcal{R}_i$ and the loudness L_i
 4. **while** $t < \text{Maximum iterations}$ **do**
 5. update velocities and adjust frequencies (using Equ. 2.5 to 2.7)
 6. Using Equation (3.5), Calculate fitness function value
 7. Using Equation (3.2), update D_i
 8. **if** ($\text{rand} > \mathcal{R}_i$) **then**
 9. 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
 10. **end if**
 11. Generate new solution randomly
 12. **if** ($(\text{rand} < L_i)$ and $(f(D_i) < f(\mathcal{G}_b))$) **then**
 13. Accept the new solutions, Increase $\mathcal{R}_i$ and reduce L_i
 14. **end if**
 15. Find the current $\mathcal{G}_b$ and Rank the Bat
 16. **end while**
 17. **end**
-

Algorithm 3.1: Pseudocode of Modified Binary Bat Algorithm (MBB Algorithm)

In Algorithm 3.1, MBB Algorithm starts by initializing bat population and parameters such as frequency, loudness, velocity and pulse rate. Its firstly determine if the maximum iteration condition is met, if false, it adjust frequency and update other population parameters, and then compute fitness function using Equation 3.5. The Acc was obtained from RF classifier used as fitness function. The process continue until it achieves the best and an acceptable solution of $\mathcal{G}_b$ and terminate the process.

3.5 Distribution of Training and Test Datasets

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

3.5.1 Cross Validation

In order to guarantee a good evaluation for all the classifiers and to prevent over-fitting during the evaluation, a double loop of cross-validation (Martín-Merino & De Las Rivas, 2009) was employed. The external loop is based on stratified tenfold cross-validation that repeatedly splits the data in ten sets, the first nine for training while the last one for testing. In the inner loop, stratified nine fold cross-validation over the training set was used to determine the best parameters and preventing bias in the error estimation. 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 the class proportions are not equal and also with small samples of data.

3.6 Performance Metrics used for the Classification Algorithms

The following are the metrics used to evaluate the performance of classification algorithms used in this study. However, there is a need to define confusion matrix and related term before explaining the performance metrics.

Confusion Matrix: Confusion matrix is not a performance metric but the easiest means to find the correctness and accuracy of the model. It is usually employed in Classification problem where the outcome can be two or more types of classes. Most of the performance metrics are based on the Confusion Matrix. Figure 3.2 depicts the confusion matrix.

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

Figure 3.2: Confusion Matrix

Terms Associated with Confusion Matrix

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 because 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.

1. Accuracy

Accuracy is the percentage of correctly classified instances out of all instances that are, the number of correct predictions achieved by the model over all types of predictions

made. Accuracy is a good measure when the target variable classes in the data are nearly balanced.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (3.5)$$

2. Recall or Sensitivity

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

$$Sensitivity = \frac{TP}{(TP+FN)} \quad (3.6)$$

3. Specificity

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

4. Kappa

Kappa or Cohen's Kappa is a statistic that is used to measure inter-rater reliability (and also Intra-rater reliability) for qualitative (categorical) items. It is generally thought to be a more robust measure than simple percent agreement calculation, as κ takes into account the possibility of the agreement occurring by chance. 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).

$$Kappa = \frac{(Observed\ Accuracy - Expected\ Accuracy)}{(1 - Expected\ Accuracy)} \quad (3.7)$$

5. Detection Rate

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

6. Balanced Accuracy

Balanced Accuracy is defined as the average of recall obtained on each class. That is, it represents the average of sensitivity and specificity which is the portion of negatives which are classified as negatives

$$Balance\ Accuracy = \frac{(Sensitivity + Specificity)}{2} \quad (3.8)$$

7. Time Taken to Extract

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

8. Estimated Error Rate

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

CHAPTER FOUR

RESULTS AND DISCUSSION

The validity of the proposed approach depends on the experimental results obtained under different conditions. Therefore, experimental results are vital in the research domain. The main subject of this chapter is the detailed analysis of different experiment using the proposed algorithms.

4.1 Experimental Setup Environment

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 under a Windows 10 environment.

4.2 Results

In this section, the numerical experimental performance of the proposed algorithms was evaluated on 8 different Microarray cancer datasets with a diverse number of samples, features (number of genes), and classes. The experimental results were presented in tabular as well as graphical formats. The results of different datasets and filter algorithms were also critically compared. In each experiment, the number of features selected and the time taken to select the features were reported.

4.3 Filter-Based Algorithm Results

In this stage, Information Gain, Gain Ratio, and Chi-Squared were used to filter relevant genes from the original Microarray cancer datasets. The reports were presented based on the dataset by dataset. Sample of raw datasets used can be found in Appendix IV.

Case 1: CNS Dataset

Table 4.1 shows the comparative analysis of feature selection for CNS dataset of various approaches. The least feature of 1 was obtained in GR and InfoGain with the threshold of ≥ 0.4 and ≥ 0.3 respectively, while the highest feature selected was obtained in InfoGain with the threshold of ≥ 0.1 . Chi-Squared has the least average time taken to extract the features (Sample of reduced datasets can be found in Appendix IV).

Table 4.1: Result of Filter Algorithms for CNS Dataset

Algorithms	No. of Original Features	Features Extracted, Threshold	Features Extracted, Threshold	Features Extracted, Threshold	Average Time Taken to Extract
Chi-Squared	7129	$d \geq 0.6 = 2$	$d \geq 0.4 = 54$	$d \geq 0.3 = 69$	21.9
Gain Ratio	7129	$d \geq 0.4 = 1$	$d \geq 0.3 = 12$	$d \geq 0.2 = 54$	23.47
Information Gain	7129	$d \geq 0.3 = 1$	$d \geq 0.2 = 13$	$d \geq 0.1 = 73$	25.5

Case 2: Leukemia_3c Dataset

Table 4.2 shows the comparative analysis of feature selection for Leukemia_3c dataset of various approaches. The least feature of 3 were obtained in InfoGain with the threshold of ≥ 0.8 , while the highest feature of 901 were obtained in Chi-Squared with the threshold of ≥ 0.4 . Chi-Squared has the least average time taken to extract the features.

Table 4.2: Result of Filter Algorithms for Leukemia_3c Dataset

Algorithms	No. of Original Features	Features Extracted, Threshold (d) ≥ 0.4	Features Extracted, Threshold ≥ 0.6	Features Extracted, Threshold ≥ 0.8	Average Time Taken to Extract
Chi-Squared	7129	901	268	38	21.95
Gain Ratio	7129	173	45	4	24.04
Information Gain	7129	101	22	3	24.12

Case 3: Leukemia_4c Dataset

Table 4.3 shows the comparative analysis of feature selection for Leukemia_4c dataset of various approaches. The least feature of 5 were obtained in InfoGain with the threshold of ≥ 0.8 , while the highest feature of 787 were obtained in Chi-Squared with the threshold of ≥ 0.4 . Chi-Squared has the least average time taken to extract the features.

Table 4.3: Result of Filter Algorithms for Leukemia_4c Dataset

Algorithms	No. of Original Features	Features Extracted, Threshold (d) ≥ 0.4	Features Extracted, Threshold ≥ 0.6	Features Extracted, Threshold ≥ 0.8	Average Time Taken to Extract
Chi-Squared	7129	787	396	62	22.23
Gain Ratio	7129	278	73	25	23.83
Information Gain	7129	127	25	5	24.07

Case 4: Lung Dataset

Table 4.4 shows the comparative analysis of feature selection for Lung dataset of various approaches. The least feature of 1 was obtained in InfoGain with the threshold of ≥ 0.8 , while the highest feature of 8,118 were obtained in Chi-Squared with the threshold of ≥ 0.4 . Chi-Squared has the least average time taken to extract the features.

Table 4.4: Result of Filter Algorithms for Lung Dataset

Algorithms	No. of Original Features	Features Extracted, Threshold (d) ≥ 0.4	Features Extracted, Threshold ≥ 0.6	Features Extracted, Threshold ≥ 0.8	Average Time Taken to Extract
Chi-Squared	12600	8,118	1,192	161	88.35
Gain Ratio	12600	779	191	37	94.73
Information Gain	12600	434	28	1	105.6

Case 5: Lymphoma Dataset

Table 4.5 shows the comparative analysis of feature selection for Lymphoma dataset of various approaches. The least feature of 33 were obtained in GR with the threshold of ≥ 0.8 , while the highest feature of 2,151 were obtained in Chi-Squared with the threshold of ≥ 0.4 . Chi-Squared has the least average time taken to extract the features.

Table 4.5: Result of Filter Algorithms for Lymphoma Dataset

Algorithms	No. of Original Features	Features Extracted, Threshold (d) ≥ 0.4	Features Extracted, Threshold ≥ 0.6	Features Extracted, Threshold ≥ 0.8	Average Time Taken to Extract
Chi-Squared	4026	2,151	1,461	403	10.24
Gain Ratio	4026	878	248	33	10.28
Information Gain	4026	861	273	72	10.31

Case 6: MLL Dataset

Table 4.6 shows the comparative analysis of feature selection for MLL dataset of various approaches. The least feature of 4 were obtained in GR with the threshold of ≥ 0.8 , while the highest feature of 5,224 were obtained in Chi-Squared with the threshold of ≥ 0.4 . GR has the least average time taken to extract the features.

Table 4.6: Result of Filter Algorithms for MLL Dataset

Algorithms	No. of Original Features	Features Extracted, Threshold (d) ≥ 0.4	Features Extracted, Threshold ≥ 0.6	Features Extracted, Threshold ≥ 0.8	Average Time Taken to Extract
Chi-Squared	12582	5,224	2,246	171	79.34
Gain Ratio	12582	1314	134	4	74.5
Information Gain	12582	1010	170	28	87.91

Case 7: Ovarian Dataset

Table 4.7 shows the comparative analysis of feature selection for Ovarian dataset of various approaches. The least feature of 1 was obtained in InfoGain with the threshold of ≥ 0.8 , while the highest feature of 2,381 were obtained in Chi-Squared with the threshold of ≥ 0.4 . InfoGain has the least average time taken to extract the features.

Table 4.7: Result of Filter Algorithms for Ovarian Dataset

Algorithms	No. of Original Features	Features Extracted, Threshold (d) ≥ 0.4	Features Extracted, Threshold ≥ 0.6	Features Extracted, Threshold ≥ 0.8	Average Time Taken to Extract
Chi-Squared	15154	2381	245	32	141.62
Gain Ratio	15154	38	6	3	151.47
Information Gain	15154	66	21	1	115.53

Case 8: SRBCT Dataset

Table 4.8 shows the comparative analysis of feature selection for SRBCT dataset of various approaches. The least feature of 7 were obtained in InfoGain with the threshold of ≥ 0.8 , while the highest feature of 667 were obtained in Chi-Squared with the threshold of ≥ 0.4 . Chi-Squared has the least average time taken to extract the features.

Table 4.8: Result of Filter Algorithms for SRBCT Dataset

Algorithms	No. of Original Features	Features Extracted, Threshold (d) ≥ 0.4	Features Extracted, Threshold ≥ 0.6	Features Extracted, Threshold ≥ 0.8	Average Time Taken to Extract
Chi-Squared	2308	667	334	45	4.05
Gain Ratio	2308	227	49	11	5.09
Information Gain	2308	150	49	7	4.66

Figure 4.1 – 4.3 show the average time taken to extract the features from Chi-Squared, Gain Ratio and InfoGain algorithms respectively. Generally, Chi-Squared has

least average time taken to extract the features in all the datasets except in MLL and Ovarian datasets where GR and InfoGain respectively have least average time taken.

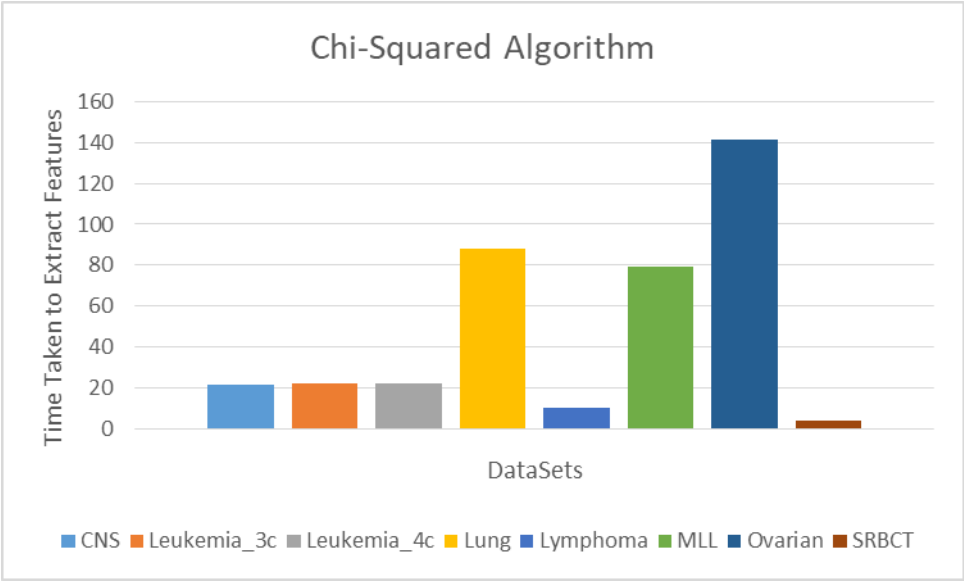


Figure 4.1: Average Time Taken to Extract Features for All Datasets Applying Chi-Squared Algorithm

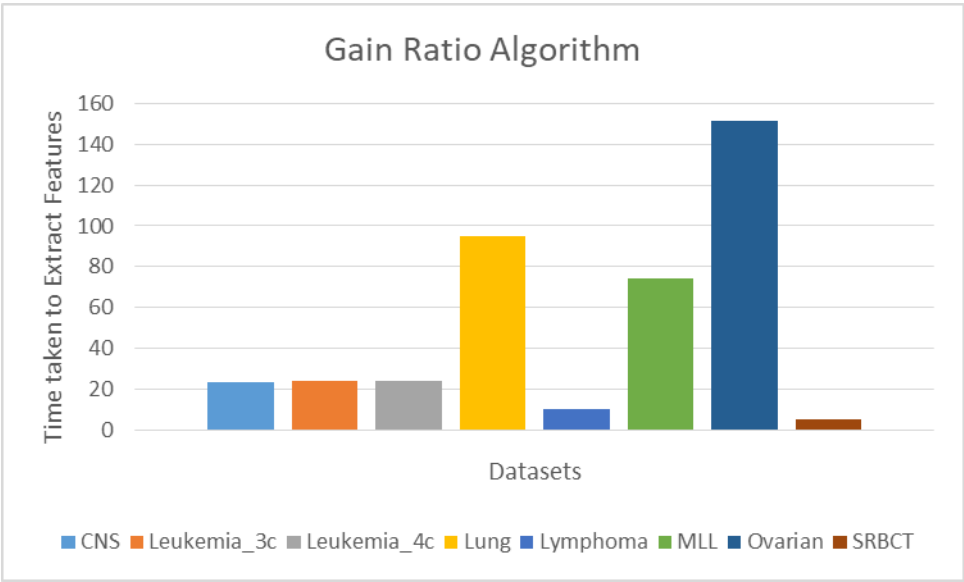


Figure 4.2: Average Time Taken to Extract Features for All Datasets Applying Gain Ratio Algorithm

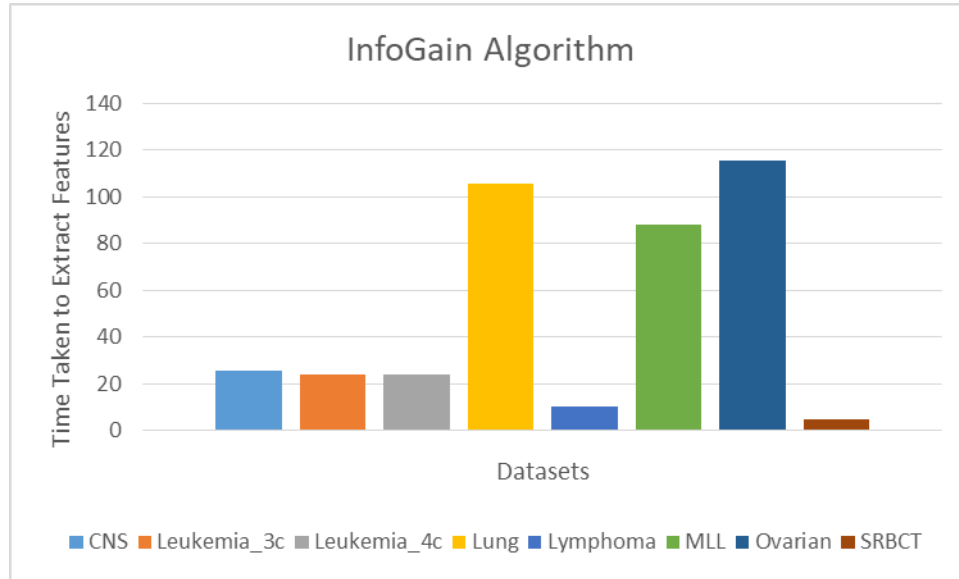


Figure 4.3: Average Time Taken to Extract Features for All Datasets Applying InfoGain Algorithm

Chi-Squared recorded highest features selected for the datasets used for this experiment with the relative least average time taken to select except in MLL and Ovarian datasets where GR and InfoGain have the least time taken to extract respectively. Also, InfoGain recorded the lowest value of features selected in all the dataset except in MLL and Lymphoma datasets. The preliminary evaluation revealed that InfoGain performs better with classifiers in some of the datasets evaluated (see Table 4.9). Therefore it was considered for combination with MBB Algorithm.

Table: 4.9: Preliminary Evaluation with Decision Tree Algorithm

Datasets	Features Extracted with Threshold	Classification Accuracy		
		Chi-Squared	Gain Ratio	InfoGain
Leukemia_3c	$d \geq 0.4 = 901$	0.8371	0.8095	0.8571
	$d \geq 0.6 = 268$	0.7995	0.8048	0.8095
	$d \geq 0.8 = 38$	0.7095	0.7095	0.7143
Leukemia_4c	$d \geq 0.4 = 787$	0.7571	0.7571	0.7524
	$d \geq 0.6 = 396$	0.8571	0.8519	0.8519
	$d \geq 0.8 = 62$	0.6195	0.619	0.671
Lung	$d \geq 0.4 = 8118$	0.9333	0.9633	0.9667
	$d \geq 0.6 = 1192$	0.9633	0.9633	0.9707
	$d \geq 0.8 = 161$	0.9533	0.9577	0.9647
Lymphoma	$d \geq 0.4 = 2151$	0.9474	0.8752	0.9474
	$d \geq 0.6 = 1461$	0.8421	0.8421	0.9474
	$d \geq 0.8 = 403$	0.9074	0.8947	0.9233
MLL	$d \geq 0.4 = 5224$	0.9024	0.8048	0.8095
	$d \geq 0.6 = 2246$	0.7571	0.7571	0.8048
	$d \geq 0.8 = 171$	0.6667	0.8748	0.9324
SRBCT	$d \geq 0.4 = 667$	0.9167	0.7083	0.7083
	$d \geq 0.6 = 334$	0.855	0.7917	0.8867
	$d \geq 0.8 = 45$	0.7917	0.875	0.9067

4.4 InfoGain-MBB Algorithm Results

The parameters used for MBB algorithm were chosen based on the preliminary experiments conducted. They offer a good trade-off between the computational cost time and quality solution desired to achieve worthy quality solutions. The preliminary investigations as showed in Figure 4.4 (see Appendix II) revealed that an increase in the number of BatSize (that is number of Iteration) influences the algorithm performance, by increasing the computational time of the algorithm. The appropriate value was chosen for pulse rate and loudness generates a good quality solution. Table 4.10 shows the parameters used for the implementation of MBB Algorithm.

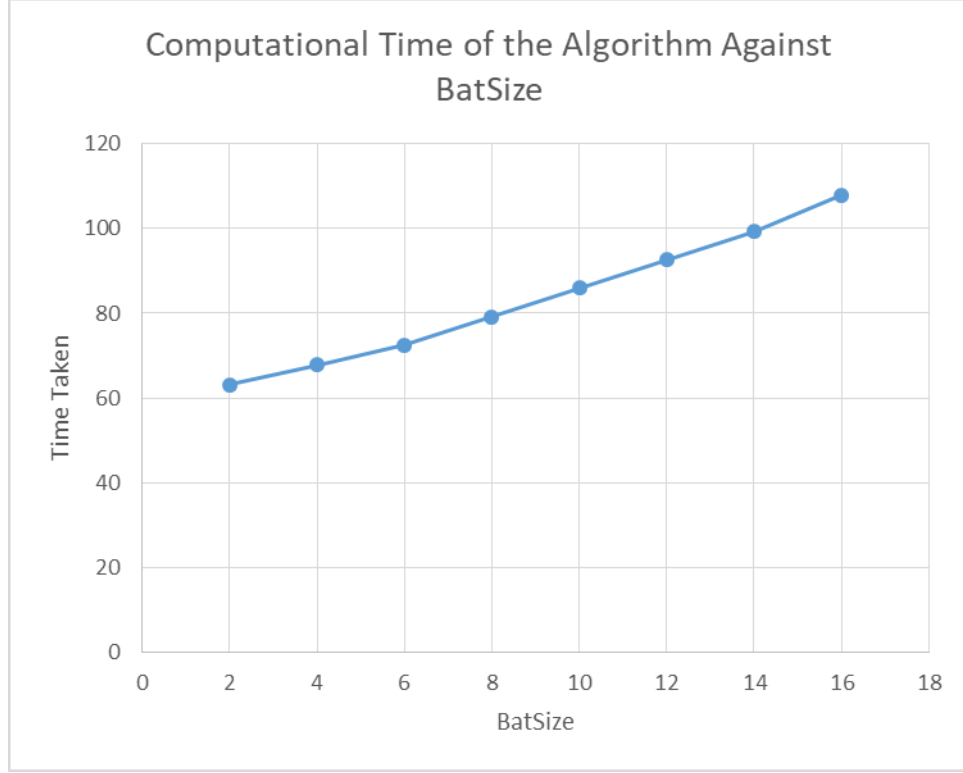


Figure 4.4: Computation Time of MBB Algorithm against BatSize for Leukemia_4c Dataset

Note: Preliminary experiment to determine BatSize was conducted using Leukemia_4c dataset of threshold ≥ 0.4 (That is, Top 40%) which is 118 features.

Table 4.10: Parameters Setting for MBB Algorithm

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

Table 4.11 shows the performance of InfoGain-MBB features selection algorithm for all datasets.

Case 1: CNS Dataset

In CNS Dataset, 73 genes were selected in the 1st Run (≥ 0.1 threshold), 13 genes in 2nd Run (≥ 0.2 threshold) and 1 gene in 3rd Run (≥ 0.3 threshold).

Table 4.11: Result of InfoGain-MBB Algorithm for All Datasets

Dataset	1st Run	2nd Run	3rd Run	Average Time Taken to Extract
CNS	73	13	1	2.76
Leukemia_3c	88	13	3	12.79
Leukemia_4c	118	17	4	19.52
Lung	425	26	1	329.39
Lymphoma	852	266	66	608.71
MLL	996	163	24	892.34
Ovarian	62	21	1	8.79
SRBCT	140	45	7	30.45

Case 2: Leukemia_3c Dataset

In Leukemia_3c dataset, 88 genes were selected in the 1st Run (≥ 0.4 threshold), 13 genes in 2nd Run (≥ 0.6 threshold) and 3 genes in 3rd Run (≥ 0.8 threshold).

Case 3: Leukemia_4c Dataset

In Leukemia_4c dataset, 118 genes were selected in the 1st Run (≥ 0.4 threshold), 17 genes in 2nd Run (≥ 0.6 threshold) and 4 genes in 3rd Run (≥ 0.8 threshold).

Case 4: Lung Dataset

In Lung dataset, 425 genes were selected in the 1st Run (≥ 0.4 threshold), 26 genes in 2nd Run (≥ 0.6 threshold) and 1 gene in 3rd Run (≥ 0.8 threshold).

Case 5: Lymphoma Dataset

In Lymphoma dataset, 852 genes were selected in the 1st Run (≥ 0.4 threshold), 266 genes in 2nd Run (≥ 0.6 threshold) and 66 genes in 3rd Run (≥ 0.8 threshold).

Case 6: MLL Dataset

In MLL dataset, 996 genes were selected in the 1st Run (≥ 0.4 threshold), 163 genes in 2nd Run (≥ 0.6 threshold) and 24 genes in 3rd Run (≥ 0.8 threshold).

Case 7: Ovarian Dataset

In Ovarian dataset, 62 genes were selected in the 1st Run (≥ 0.4 threshold), 24 genes in 2nd Run (≥ 0.6 threshold) and 1 gene in 3rd Run (≥ 0.8 threshold).

Case 8: SRBCT Dataset

In SRBCT dataset, 140 genes were selected in the 1st Run (≥ 0.4 threshold), 45 genes in 2nd Run (≥ 0.6 threshold) and 7 genes in 3rd Run (≥ 0.8 threshold).

Figure 4.5 shows the average time taken by InfoGain-MBB Algorithm to extract features. Generally, InfoGain-MBB Algorithm has the least average time taken to extract the features in all the datasets except in MLL, Lymphoma, and Lung datasets.

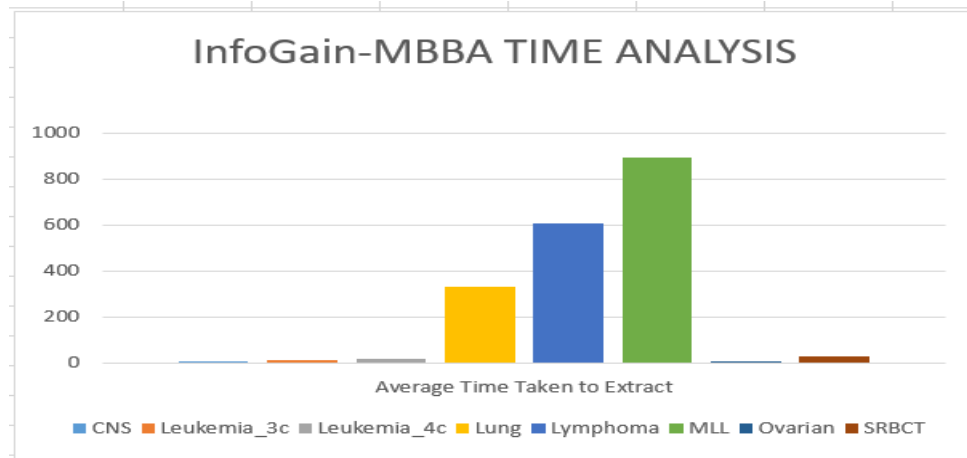


Figure 4.5: Average Time Taken to Extract Features for All Datasets Using InfoGain-MBB Algorithm

4.5 Classification Results

To evaluate the classification performance of the selected features of microarray data, four simple, effective and well-known classifiers were chosen: C4.5, Decision Tree (DT), Random Forest (RF) and Classification and Regression Tree (CART). The results were presented based on the dataset by dataset. Each classifier designated with three 'Runs', 1st, 2nd and 3rd 'Run' referred to top 40%, 60% and 80% selected features respectively.

4.5.1 Comparison of Original Approach and Modified BBA with InfoGain

In this section, the performance of original BBA and Modified BBA was compared by conducting experiment with three multiclass datasets – Leukemia_3c, Leukemia_4c and Lung. The summary of results was presented in Table 4.12. The results showed the improvement in classification accuracy in all the datasets considered from at least 2 - 6%.

Table 4.12: Comparison of BBA and MBBA

Datasets	No Features Selected	Classification Accuracy for BBA	Classification Accuracy for MBB
Leukemia_3c	88	0.84	0.86
	13	0.79	0.81
	3	0.79	0.81
Leukemia_4c	118	0.74	0.76
	17	0.94	1
	4	0.66	0.67
Lung	425	0.92	0.95
	26	0.96	1
	1	0.93	0.98

4.5.2 Performance Evaluation of InfoGain-MBB Algorithm

Case 1: CNS Dataset

Selected features from Central Nervous System (CNS) dataset were evaluated by four different classifiers. CNS has 60 samples with 7129 features and Class label of 0 and 1. 0 indicates a cancerous instance of about 39 instances and 1 indicates normal instances of about 21 instances. After feature selection processing, the final features used for classification were reduced to 73, 13 and 1. Table 4.13 shows the performance of four classifiers on reduced CNS dataset.

Table 4.13: Result of Classifiers for CNS Dataset

Algorithms	No of Selected Features	Accuracy	Kappa	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value
C4.5	73	1	1	1	1	1	1
	13	1	1	1	1	1	1
	1	1	1	1	1	1	1
DT	73	1	1	1	1	1	1
	13	1	1	1	1	1	1
	1	1	1	1	1	1	1
RF	73	1	1	1	1	1	1
	13	1	1	1	1	1	1
	1	1	1	1	1	1	1
CART	73	1	1	1	1	1	1
	13	1	1	1	1	1	1
	1	1	1	1	1	1	1

All the classifiers perform excellently well with reduced features even with 1 gene selected yields 100% classification accuracy. Figure 4.6 shows the detection rate for the four classifiers. All the classifiers have an equal detection rate of 61% and 56%. The reduction in detection rate in the last 'Run' (where the feature is 1) as a result of a few features to train the classifiers.

Figure 4.7 shows the balanced accuracy for all the classifiers. It depicts the balance between correctly predicted classes, that is, the 1's and 0's (cancer / normal) in all the classifiers.

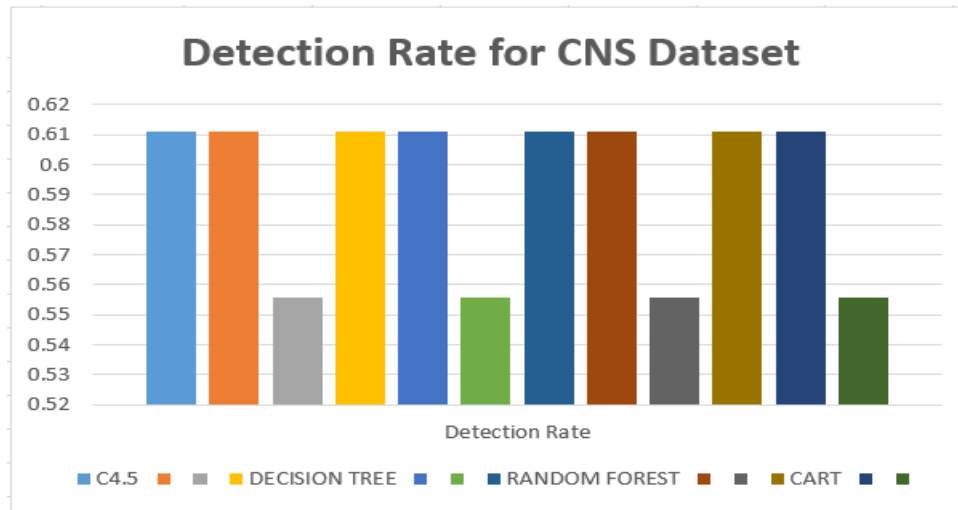


Figure 4.6: Detection Rate for CNS Dataset for all the Classifiers

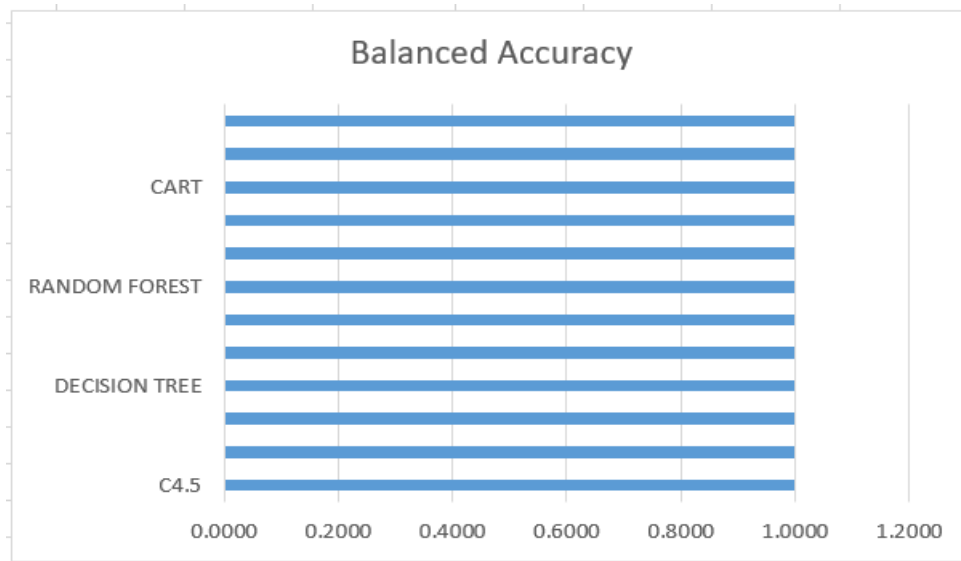


Figure 4.7: Balanced Accuracy for CNS Dataset for all the Classifiers

Note: See Appendix III for the computation of Balanced Accuracy and Detection Rate and Appendix V for Decision tree models built.

Case 2: Leukemia_3c Dataset

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 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 4.14 shows the performance of four classifiers on reduced Leukemia_3c dataset.

Table 4.14: 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

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

Figure 4.8 shows the detection rate for the four classifiers with Leukemia_3c dataset. The lowest average detection rate of 25.4% was obtained in 2nd 'Run' of C4.5 and 1st 'Run' of CART classifiers.

Figure 4.9 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 2nd 'Run' of CART.

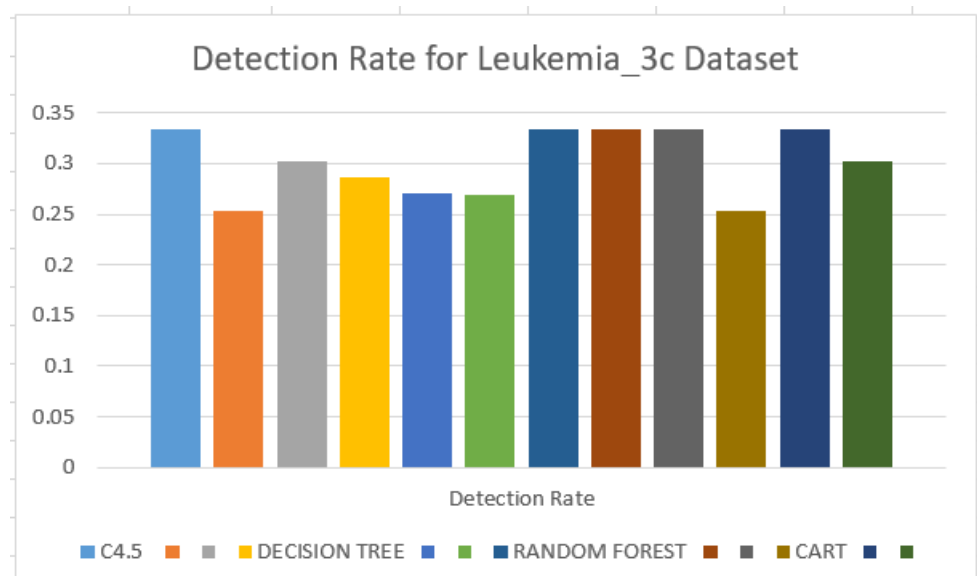


Figure 4.8: Detection Rate for Leukemia_3c Dataset for all the Classifiers

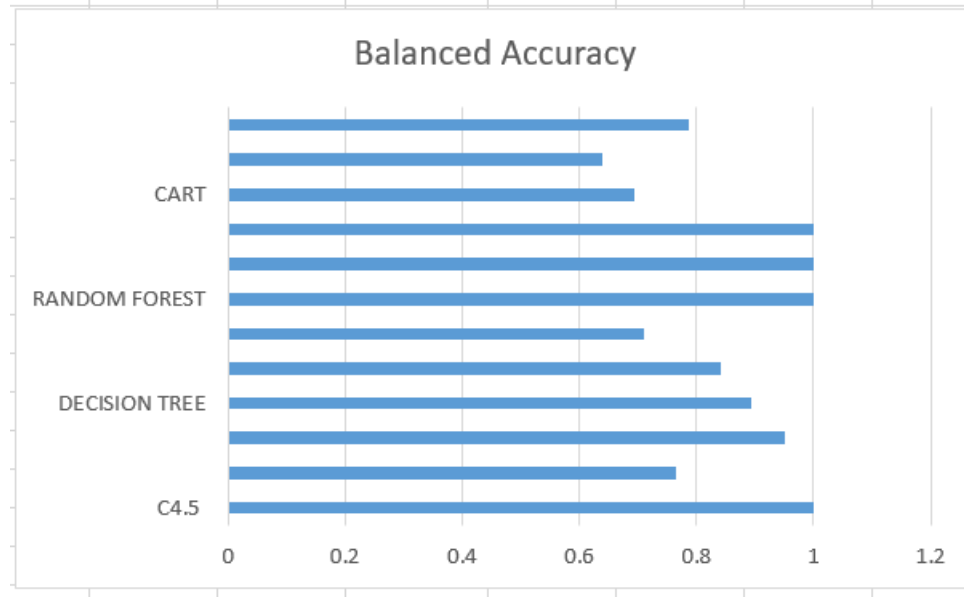


Figure 4.9: Balanced Accuracy for Leukemia_3c Dataset for all the Classifiers

Case 3: 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, 17 and 4 for the top 40%, 60% and 80% respectively. Table 4.17 shows the performance of four classifiers on reduced Leukemia_4c dataset.

Table 4.15: 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

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'. 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 well in all the metrics used for evaluation.

Figure 4.10 shows the average detection rate for the four classifiers with Leukemia_4c dataset. The lowest average detection rate of 16.67% was obtained in 3rd 'Run' of DT classifiers.

Figure 4.11 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 obtained in all 'Run' of RF and 1st and 3rd 'Run' of C4.5 and lowest value of 63.15% was recorded in 3rd 'Run' of DT.

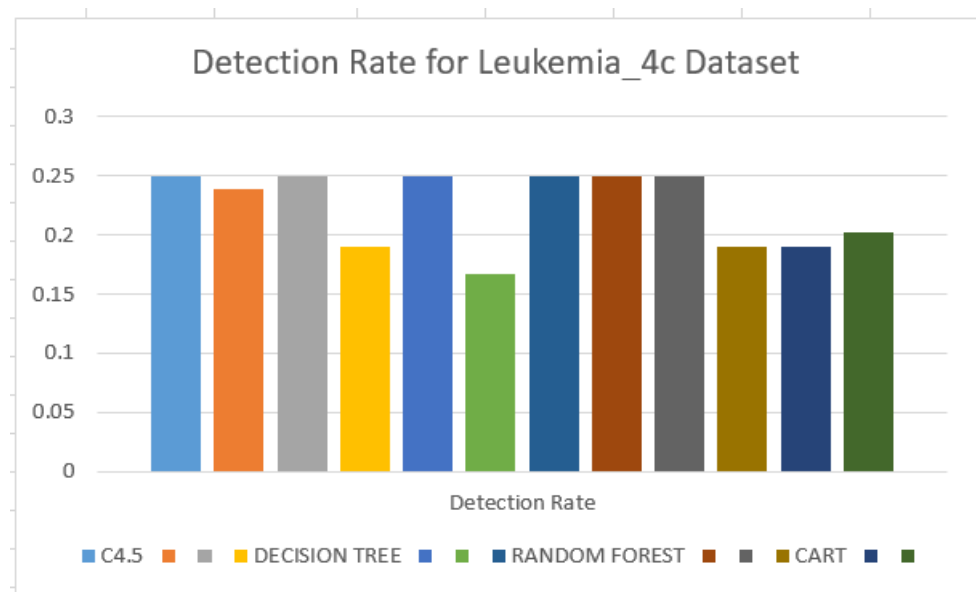


Figure 4.10: Detection Rate for Leukemia_4c Dataset for all the Classifiers

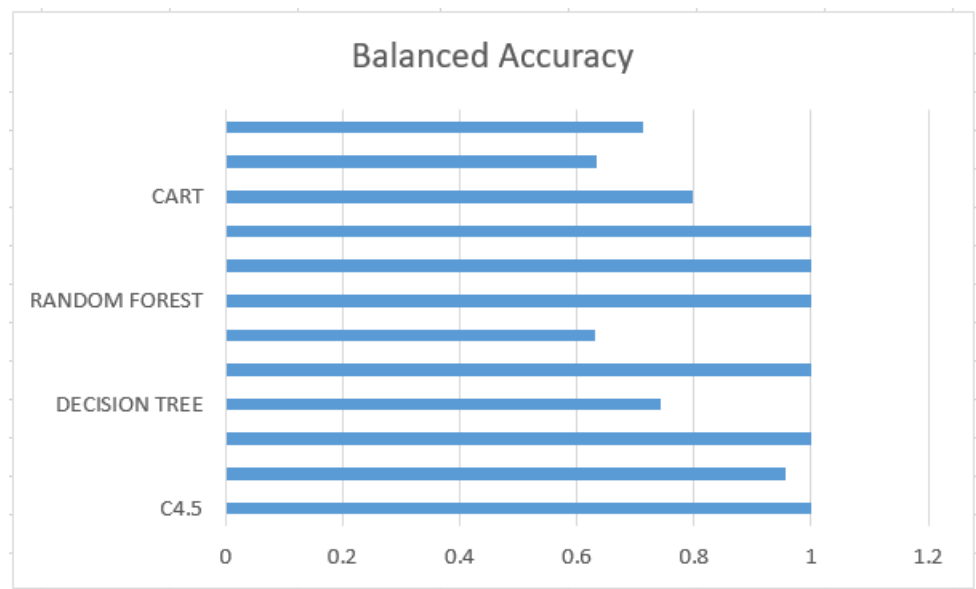


Figure 4.11: Balanced Accuracy for Leukemia_4c Dataset for all the Classifiers

Case 4: Lung Dataset

Lung dataset has five classes which indicate 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 4.16 shows the performance of four classifiers on reduced Lung dataset.

Table 4.16: 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 well with reduced features. The least accuracy of 95% was obtained in 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 4.12 shows the average detection rate for the four classifiers with Lung dataset. The lowest average detection rate of 19% was obtained in 1st ‘Run’ of DT classifiers while other classifiers have equal values of average detection rate.

Figure 4.13 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 1st ‘Run’ of DT.

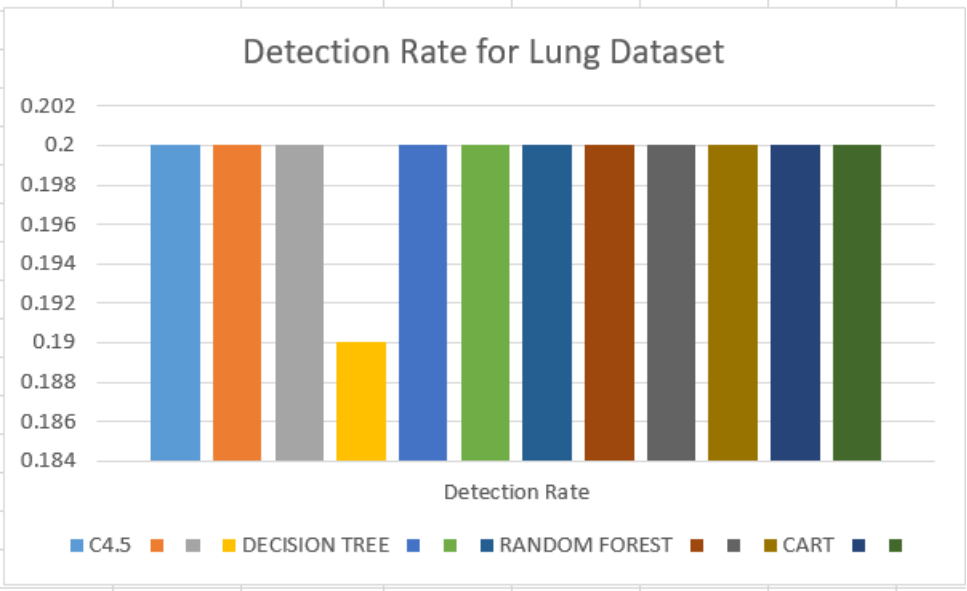


Figure 4.12: Detection Rate for Lung Dataset for all the Classifiers

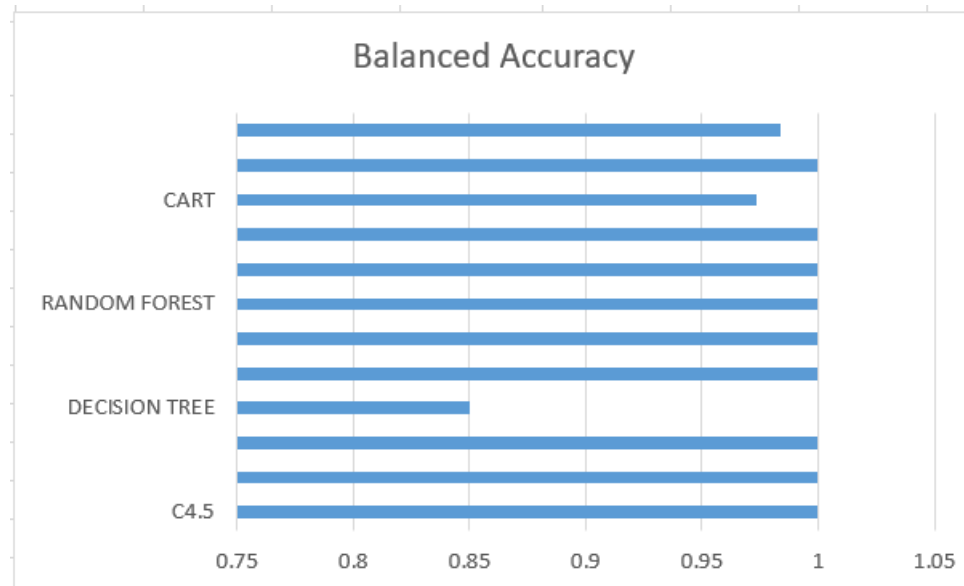


Figure 4.13: Balanced Accuracy for Lung Dataset for all the Classifiers

Case 5: Lymphoma Dataset

Lymphoma dataset has three classes consisting of Diffuse Large B-Cell Lymphoma (DLBCL), Follicular Lymphoma (FL) and Chronic Lymphocytic Leukemia (CLL). It has 66 samples with 4,026 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 4.17 shows the performance of four classifiers on reduced Lymphoma dataset.

Table 4.17: 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

All the classifiers performed very well with reduced features. The least accuracy obtained was 84% in 1st 'Run' of CART with 825 genes. Even with 66 genes out of 4,026 genes, all the classifiers were able to yield good performance accuracy and all other evaluation metrics used in this study.

Figure 4.14 shows the average detection rate for the four classifiers with Lymphoma dataset. The lowest average detection rate of 28.07% was obtained in 1st 'Run' of CART classifiers while other classifiers have roughly equal values of average detection rate.

Figure 4.15 shows the average balanced accuracy for all the classifiers. Almost all the classifiers have 100% average balanced accuracy except in 1st 'Run' of CART classifier where the average balanced accuracy was 80%.

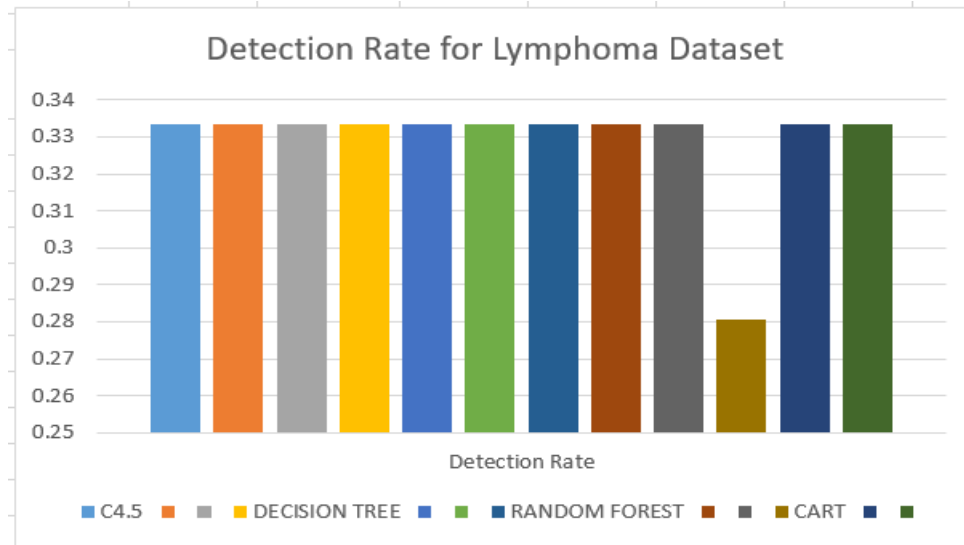


Figure 4.14: Detection Rate for Lymphoma Dataset for all the Classifiers

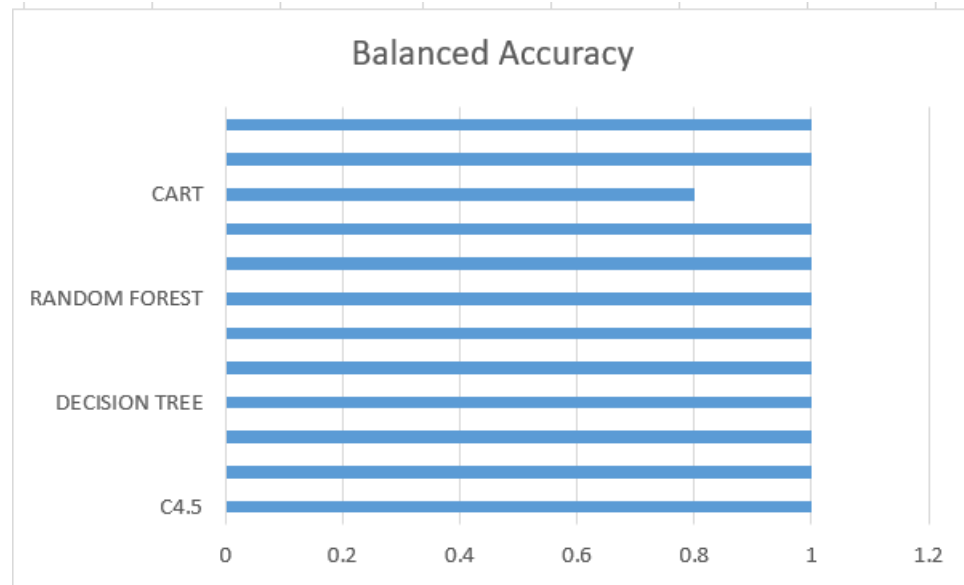


Figure 4.15: Balanced Accuracy for Lymphoma Dataset for all the Classifiers

Case 6: MLL Dataset

Mixed-Lineage Leukemias (MLL) dataset has three classes consisting of Acute Lymphoblastic Leukemia (ALL), Acute Myeloid Leukemia (AML) and MLL. It has 72 samples 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 4.18 shows the performance of four classifiers on reduced MLL dataset.

Table 4.18: 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 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 4.16 shows the average detection rate for the four classifiers with Lymphoma dataset. The lowest average detection rate of 26.99% was obtained in 1st ‘Run’ of DT classifier.

Figure 4.17 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 87.75% was obtained in 1st ‘Run’ of DT classifier.

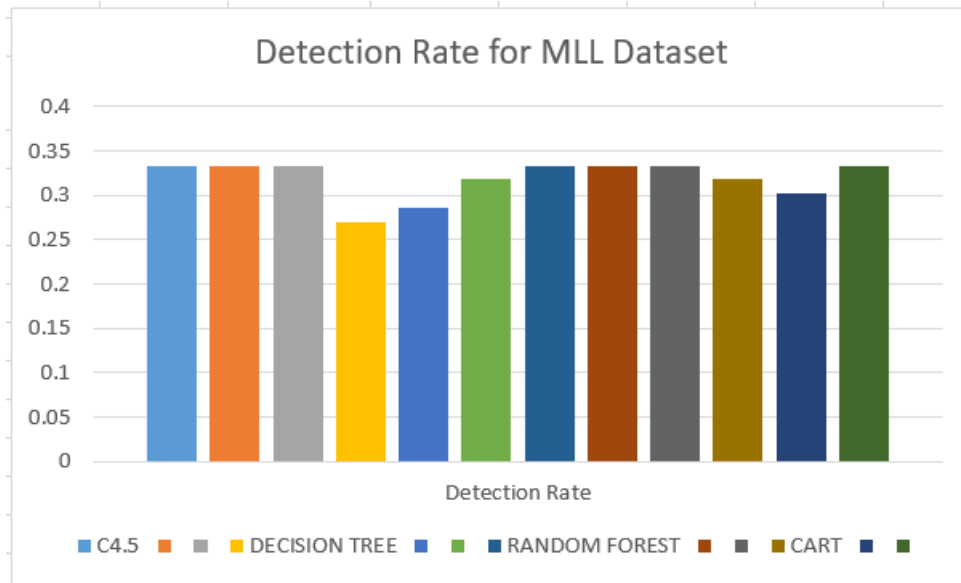


Figure 4.16: Detection Rate for MLL Dataset for all the Classifiers

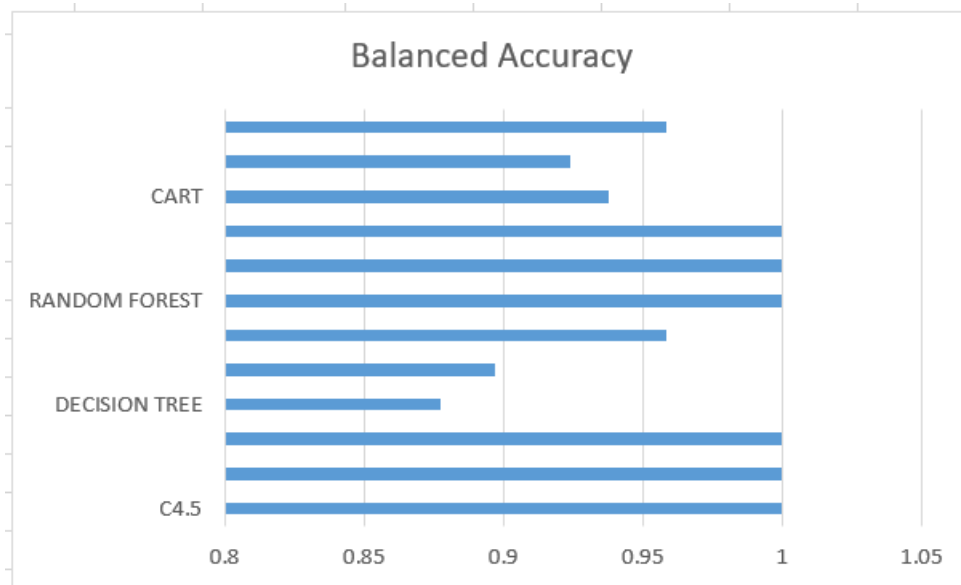


Figure 4.17: Balanced Accuracy for MLL Dataset for all the Classifiers

Case 7: Ovarian Dataset

Selected features from Ovarian dataset were evaluated by four difference 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 classification were reduced to 62, 21 and 1. Table 4.19 shows the performance of four classifiers on reduced Ovarian dataset.

Table 4.19: 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

All the classifiers perform excellently well with reduced features even with 1 gene selected yields 100% classification accuracy. Figure 4.18 shows the detection rate for the four classifiers. All the classifiers have a very close range of detection rate of 62.67% to 65.33%. The 2nd 'Run' of all the classifiers has the highest detection rate due to a sizeable number of features to train the classifiers.

Figure 4.19 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.

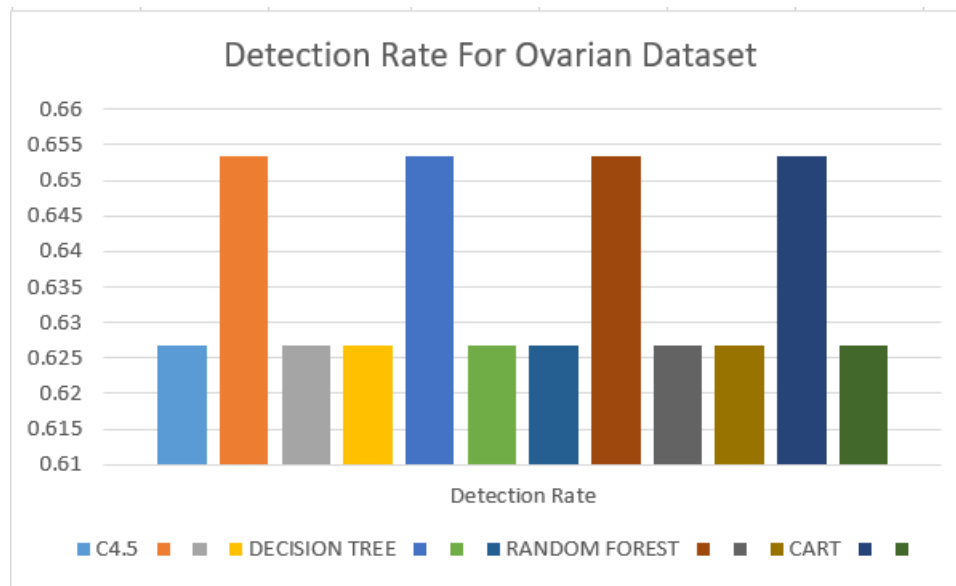


Figure 4.18: Detection Rate for Ovarian Dataset for all the Classifiers

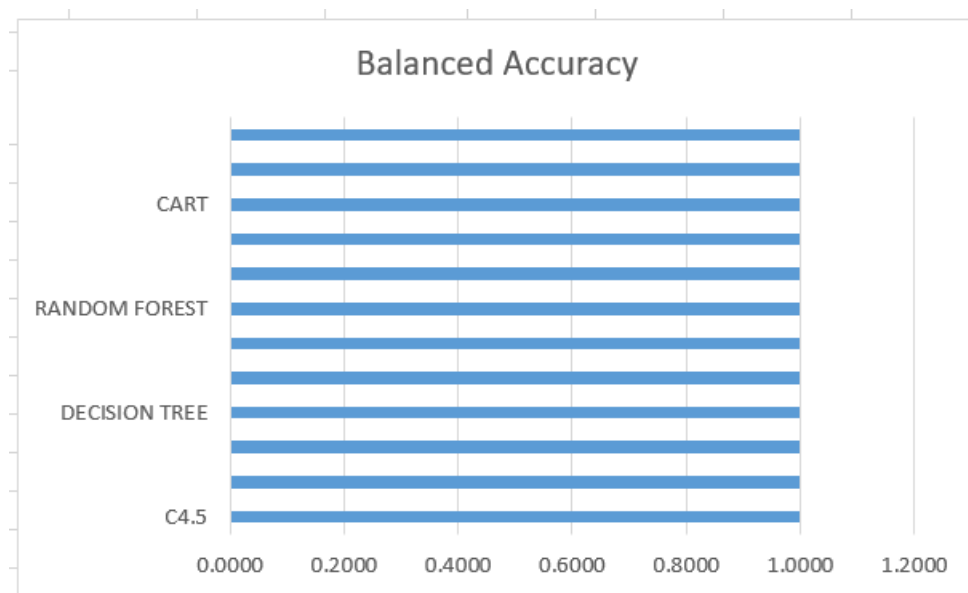


Figure 4.19: Balanced Accuracy for Ovarian Dataset for all the Classifiers

Case 8: 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 as 1 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 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 4.20 shows the performance of four classifiers on reduced SRBCT dataset.

Table 4.20: 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

All the classifiers performed very well with reduced features. The least accuracy obtained was 75% in 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 CART classifier.

Figure 4.20 shows the average detection rate for the four classifiers with SRBCT dataset. The lowest average detection rate of 18.75% was obtained in 1st ‘Run’ of DT and 3rd ‘Run’ of CART classifiers.

Figure 4.21 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 3rd ‘Run’ of CART classifier,

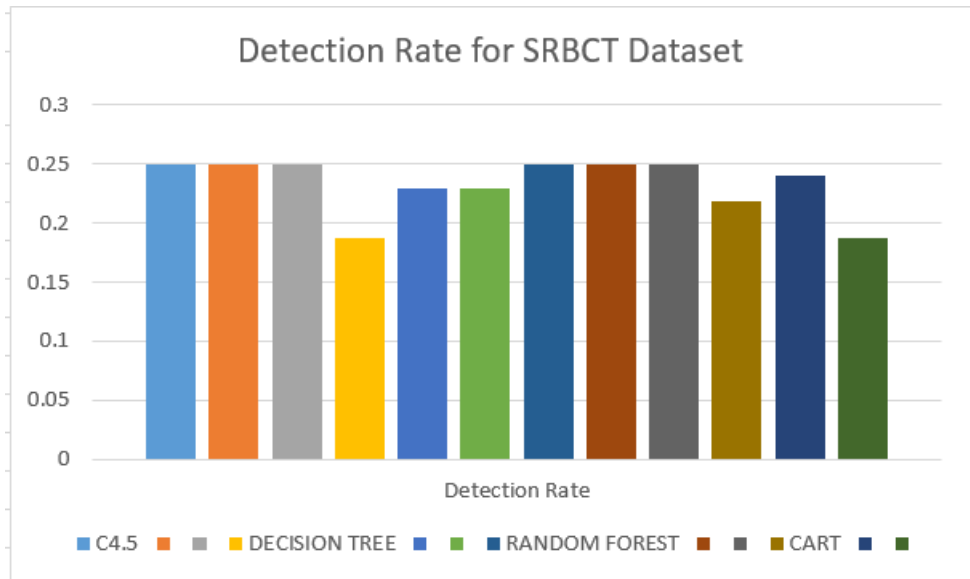


Figure 4.20: Detection Rate for SRBCT Dataset for all the Classifiers

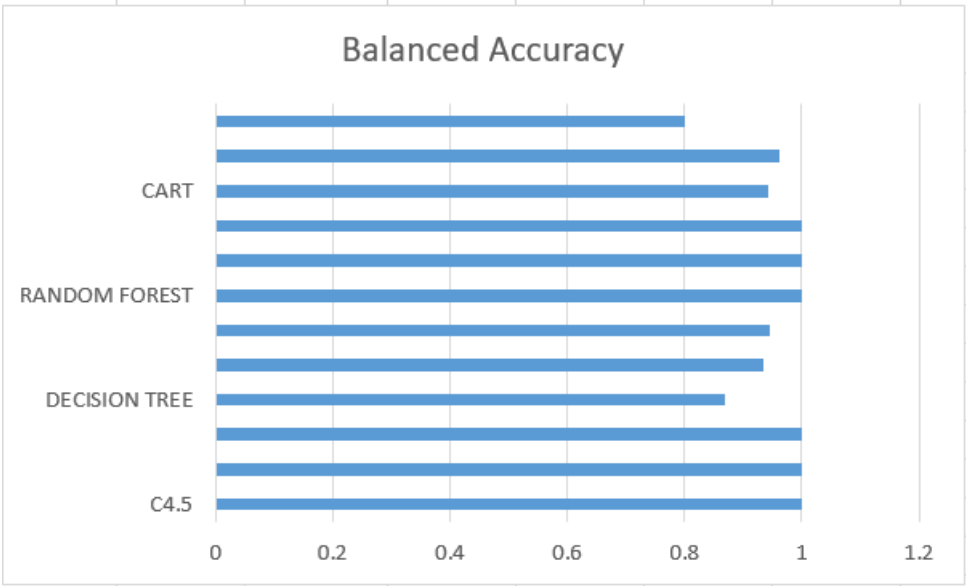


Figure 4.21: Balanced Accuracy for SRBCT Dataset for all the Classifiers

4.6 Error Rate for Random Forest

Since RF performed excellently well 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 additional metric called estimated error rate per class for RF. Figure 4.22 - 4.29 depict estimated error per class for all the datasets used and its confusion matrix can be found in Appendix I.

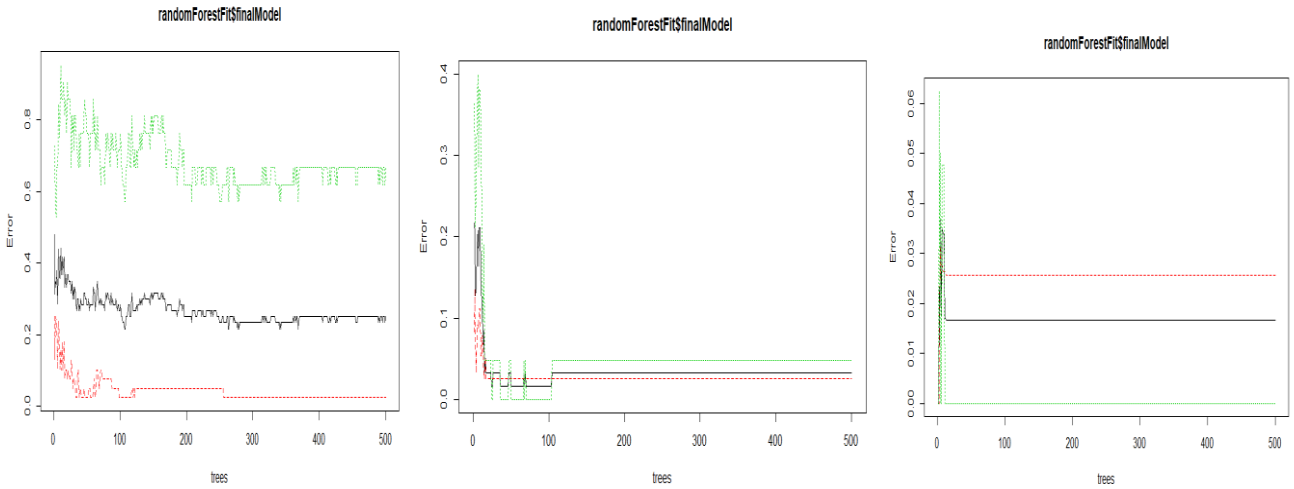


Figure 4.22: Estimated Error Per Class for CNS Dataset

Note: The black colour in the graph indicates the average estimated error rate for all the class

Figure 4.22 shows the estimated error rate per class for CNS. Average estimated error for CNS is 25%, 3.3% and 1.67% respectively for all three 'Run's. The value decreases from 1st 'Run' to the 3rd 'Run', this indicates that estimated error reduces as the number of features selected reduces. Optimal features selection was obtained in the 3rd 'Run'.

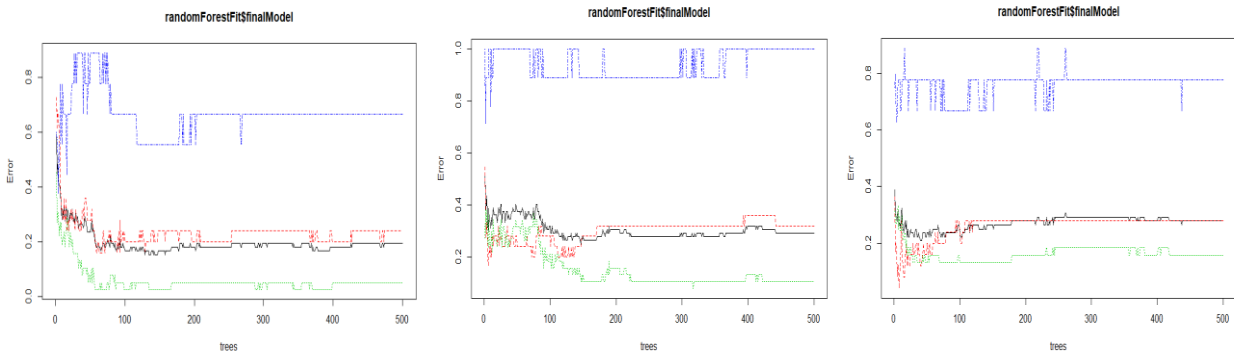


Figure 4.23: Estimated Error Per Class for Leukemia_3c Dataset

Note: The black colour in the graph indicates the average estimated error rate for all the class

Figure 4.23 shows the estimated error rate per class for Leukemia_3c. Average estimated error obtained for Leukemia_3c was 19.44%, 29.17% and 27.78% respectively for all three 'Run's. The value increases from the 1st 'Run' to 2nd 'Run' and then decreases in the 3rd 'Run', this indicates that estimated error increases as the number of features selected reduces. Therefore, this reveals that some complementary features have been eliminated during the selection.

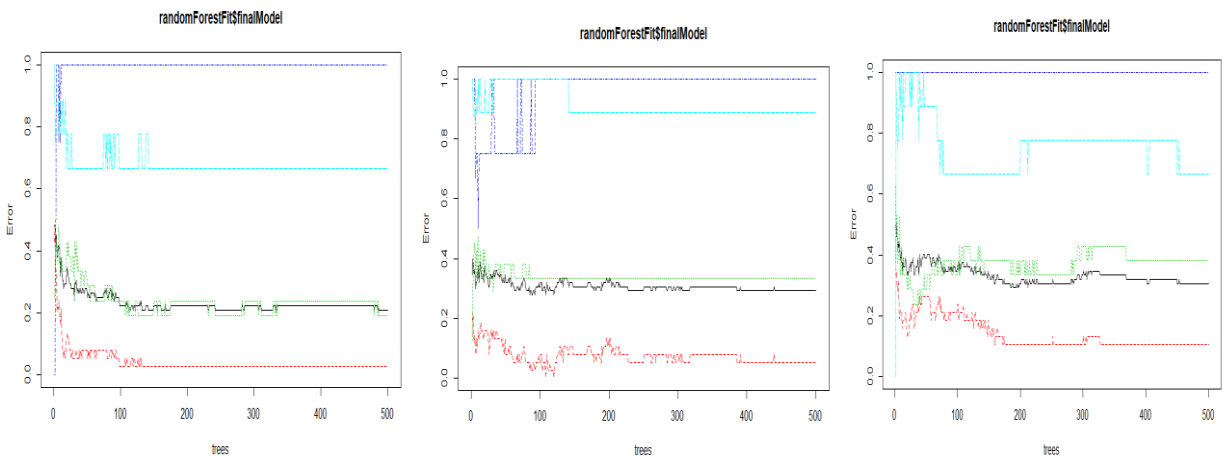


Figure 4.24: Estimated Error per class for Leukemia_4c Dataset

Note: The black colour in the graph indicates the average estimated error rate for all the class

Figure 4.24 shows the estimated error rate per class for Leukemia_4c. Average estimated error obtained for Leukemia_4c was 20.89%, 29.17% and 30.56% respectively for all three

'Run's. The value increases from the 1st 'Run' to the 3rd 'Run', this indicates that estimated error increases as the number of features selected reduces. Therefore, this reveals that some complementary features have been eliminated during the selection.

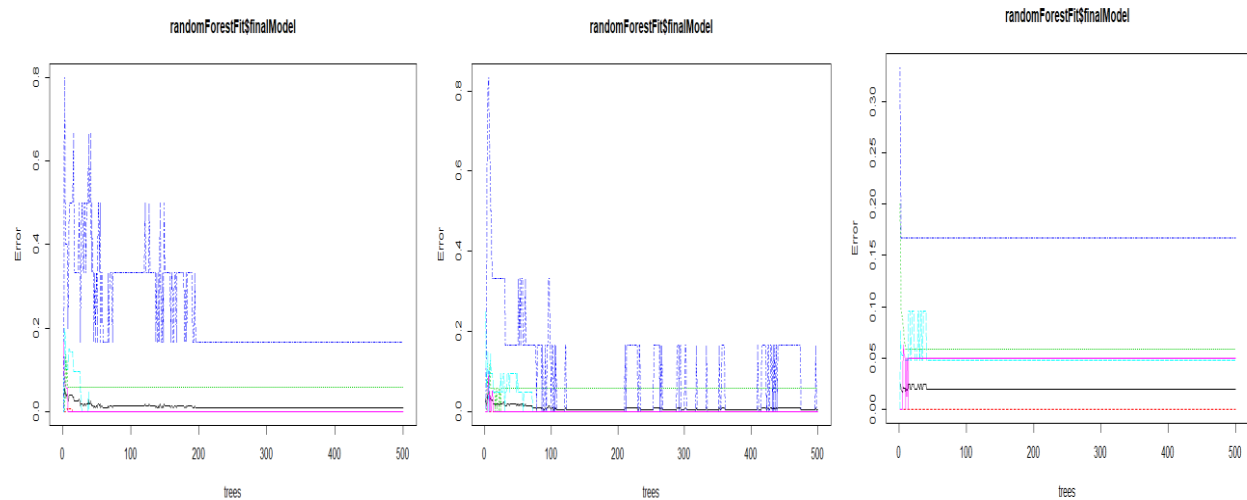


Figure 4.25: Estimated Error Per Class for Lung Dataset

Note: The black colour in the graph indicates the average estimated error rate for all the class

Figure 4.25 shows the estimated error rate per class for Lung. Average estimated error obtained for Lung was 0.99%, 0.49% and 1.97% respectively for all three 'Run's. The value decreases from the 1st 'Run' to the 2nd 'Run', this indicates that estimated error decreases as the number of features selected reduces. Therefore, this reveals that the best selection for Lung dataset is 2nd 'Run' because the error increased at the 3rd 'Run' when number of genes fall below 26 genes.

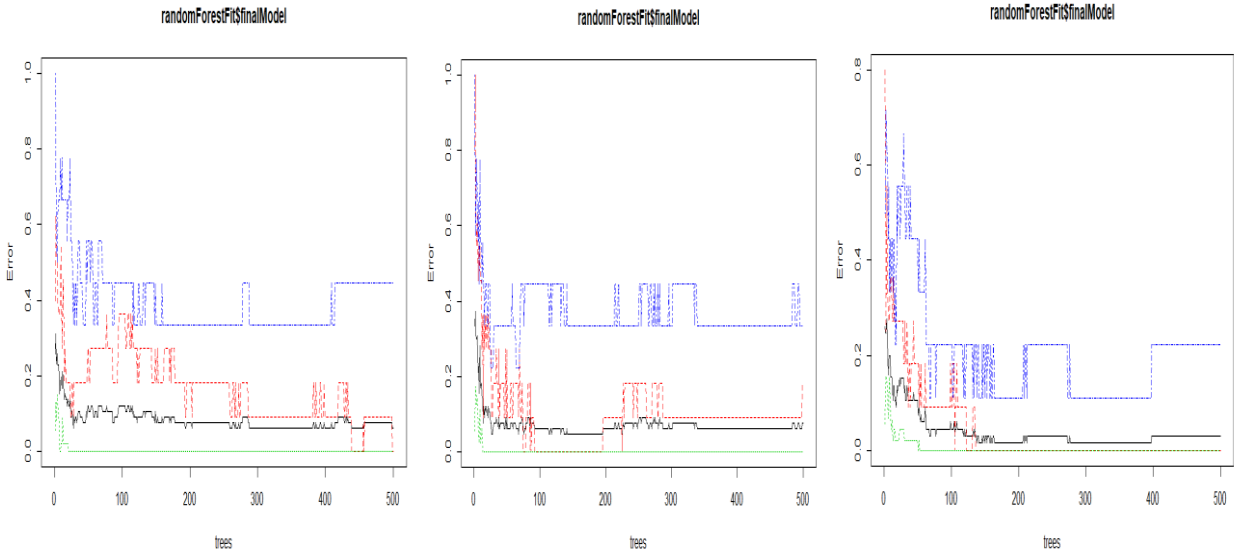


Figure 4.26: Estimated Error Per Class for Lymphoma Dataset

Note: The black colour in the graph indicates the average estimated error rate for all the class

Figure 4.26 shows the estimated error rate per class for Lymphoma. Average estimated error obtained for Lymphoma was 6.06%, 7.58% and 3.03% respectively for all three 'Run's. The value increases slightly from the 1st 'Run' to the 2nd 'Run', and later decreases in the 3rd 'Run'. Therefore, optimal feature selection for Lymphoma dataset was obtained in the 3rd 'Run' when the number of genes is 66.

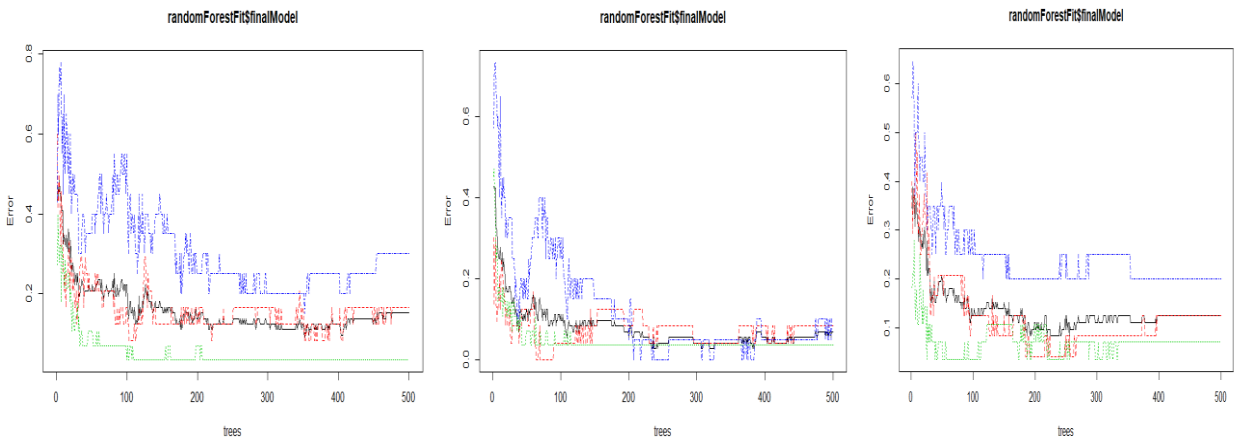


Figure 4.27: Estimated Error Per Class for MLL Dataset

Note: The black colour in the graph indicates the average estimated error rate for all the class

Figure 4.27 shows the estimated error rate per class for MLL. Average estimated error obtained for MLL was 15.28%, 6.94% and 12.50% respectively for all three 'Run's. The value decreases from the 1st 'Run' to the 2nd 'Run', and later increases in the 3rd 'Run'. Therefore, optimal feature selection for MLL dataset was obtained in the 2nd 'Run' when the number of genes selected is 163.

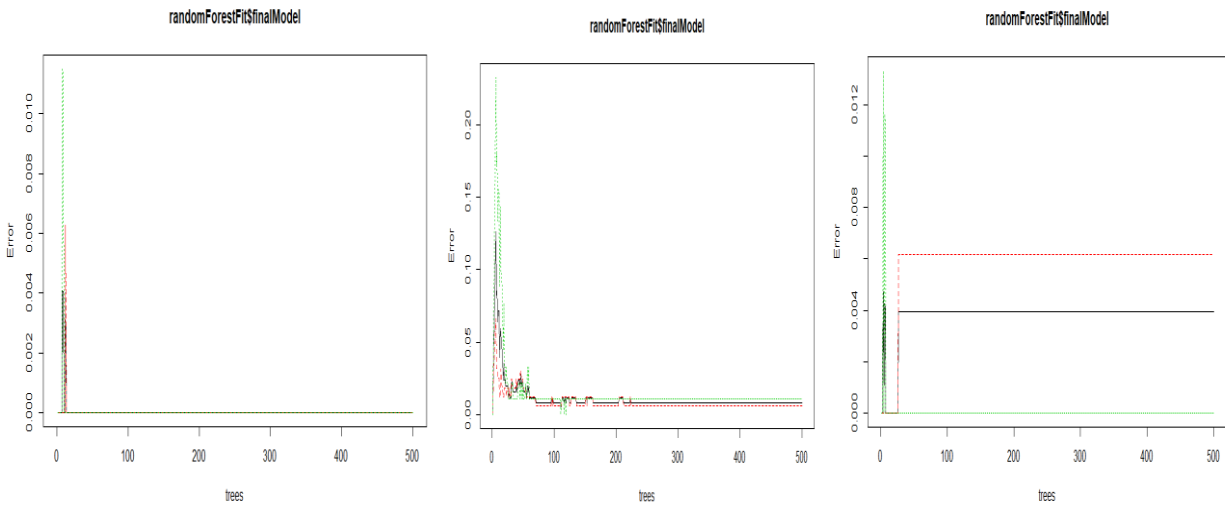


Figure 4.28: Estimated Error Per Class for Ovarian Dataset

Note: The black colour in the graph indicates the average estimated error rate for all the class

Figure 4.28 shows the estimated error rate per class for Ovarian. Average estimated error obtained for Ovarian were 0%, 0.79% and 0.40% respectively for all three 'Run's. The value increases from the 1st 'Run' to the 2nd 'Run', and later decreases at the 3rd 'Run'. Therefore, optimal feature selection for Ovarian dataset was obtained in the 1st 'Run' when the number of genes selected is 62.

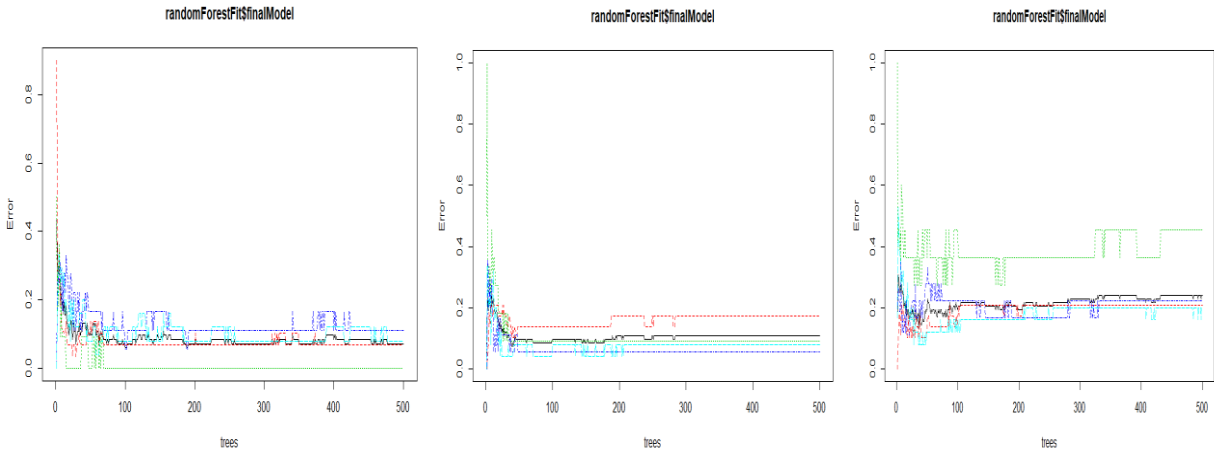


Figure 4.29: Estimated Error Per Class for SRBCT Dataset

Note: The black colour in the graph indicates the average estimated error rate for all the class

Figure 4.29 shows the estimated error rate per class for SRBCT. Average estimated error obtained for SRBCT was 7.2%, 10.84% and 24.10% respectively for all three 'Run's. The value increases from the 1st 'Run' to the 3rd 'Run' Therefore, optimal feature selection for SRBCT dataset was obtained in the 1st 'Run' when the number of genes selected is 140.

On a general note, only CNS and Lymphoma datasets are the only datasets out of eight datasets used in this study that did not have their estimated error increases as the number of features selected reduces when they reach optimal features selected.

4.7 Discussion

This section presents the discussion of the results obtained in the previous section.

4.7.1 Comparing Studies Algorithms

In section 4.3, the results of three filter algorithms (InfoGain, GR and Chi-Squared) were presented along with the number of features selected and time taken to extract the selected features. InfoGain has the least features selected with the relatively average time taken to extract the feature in five out of the eight datasets used. Preliminary experiments on selected features

with the classifiers also, showed that InfoGain performed better than GR and Chi-Squared. That is the reason why InfoGain was considered for the combination with MBB Algorithm to further remove redundant features from selected features.

Section 4.4 presented the results of InfoGain-MBB Algorithm for all the datasets that showed the number of features selected and the average time taken to select the feature. The least time of 2.76 seconds was obtained in the CNS dataset while the highest time taken of 892.34 seconds was recorded in the MLL dataset. Figure 4.5 showed the graphical representation of the time analysis for all the datasets. In section 4.5, 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 three 'Run' for each dataset, because, 3 different outputs were available from the feature selection approach used (InfoGain-MBB Algorithm). That is, features from a threshold of ≥ 0.4 , ≥ 0.6 and ≥ 0.8 respectively.

Table 4.13 presented the classifiers' results of the CNS 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 Figure 4.6, all the classifiers have an equal average detection rate in first and second 'Run', and the detection rate dropped from 61% to 56% in the third 'Run' for all the classifiers as a result of few features used to train the classifier algorithms. Also, Figure 4.7 presented a balanced accuracy of all the classifiers for the CNS dataset. The balanced accuracy recorded was 100% for all the three 'Run' for each classifier. These 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 4.6 and Figure 4.22) revealed that estimated

error per class reduced from 1st 'Run' to the 3rd 'Run'. Therefore, this study concluded that the best threshold for the CNS data set was ≥ 0.3 , that is, the top 30% features.

Table 4.14 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 2nd 'Run' of C4.5, 1st and 2nd 'Run' of CART classifiers. Figure 4.8, 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 1st 'Run' of all the classifiers to the next 'Run' except in CART where it increases and remains constant in RF for all the three 'Run'. Also, Figure 4.9 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 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 have 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 increased in the 3rd 'Run', except in DT classifier where it kept on decreasing up to the 3rd 'Run'. This showed that the average detection rate and balanced accuracy for Leukemia_3c decreases 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 4.6 and Figure 4.23) revealed the same scenario. Therefore, this study concluded that the best threshold for the Leukemia_3c data set was ≥ 0.4 , that is, the top 40% features.

Table 4.15 presented the classifiers' results of 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. Figure 4.10, 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 1st 'Run' of all the classifiers to the next 'Run' except RF where it remains constant in all the three 'Run'. Also, Figure 4.11 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 have 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 increasing in the 3rd 'Run', except in CART classifier where it kept 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 4.6 and Figure 4.24) revealed the same scenario of 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 was ≥ 0.4 , that is, the top 40% features.

Table 4.16 presented the classifiers' results of 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. Figure 4.12, 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, Figure 4.13 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 4.6 and Figure 4.25) 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 was ≥ 0.6 , that is, the top 60% features.

Table 4.17 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 4.14 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, Figure 4.15 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 CART classifier and 100% were recorded in all the three 'Run' of other classifiers. Further investigation with estimated error per class of RF (Section 4.6 and Figure 4.26) revealed that there is slight increase in estimated error per class from 1st 'Run' to the 2nd 'Run' and later decreases drastically to 3.03% in the 3rd 'Run'. Therefore, this study concluded that the best threshold for the Lymphoma dataset was ≥ 0.8 , that is, the top 80% features.

Table 4.18 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 4.17 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 a very close range values of average detection rate per class. Also, Figure 4.18 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 increased from 1st 'Run' to next 'Run' except in where there is 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 4.6 and Figure 4.27)

revealed that there was a decreased 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 was ≥ 0.6 , that is, the top 60% features.

Table 4.19 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 Figure 4.19, 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, Figure 4.20 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 4.6 and Figure 4.28) 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 was ≥ 0.4 , that is, the top 40% features.

Table 4.20 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. Figure 4.21 showed an average detection rate per class for all the classifiers. The least average detection rate per class of 18.75% was obtained in 1st

‘Run’ of DT and 3rd ‘Run’ of CART classifiers and others have a very close range values of average detection rate per class. Also, Figure 4.22 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 increased 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 4.6 and Figure 4.29) revealed that there was an increased in estimated error per class from 1st ‘Run’ to the 3rd ‘Run’. Therefore, this study concluded that the best threshold for the SRBCT dataset was ≥ 0.4 , that is, the top 40% features.

4.7.2 Comparison of Proposed Approach with other Existing Approaches in the Literature

In this section, the proposed approach is compared with other available approaches 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 were considered for comparison. Table 4.21 summarizes the comparison between the approaches used in this study with other available works in the literature.

- i. **CNS Dataset:** There are several approaches available in the literature that had explored CNS dataset, among them, are Yahya, Ulm, Ludwig & Hapflemeir, (2011) used sequential features selection (K-SS), Gunavathi & Premalatha, (2015) employed Cuckoo Search (CS) feature selection, Bolón-Canedo et al., (2015) employed Distributed Feature selection (DF), Ebrahimpour et al., (2018) used Cooperative Coevolving feature selection (CCFS), and Chormunge & Jena, (2018)

used CBFS with Cluster. The approach used (InfoGain-MBB Algorithm) in this research work outperforms other approaches used in the literature, with one gene selected and it is able to record 100% classification accuracy.

- ii. **Leukemia_3c Dataset:** There are several approaches available in the literature that had explored Leukemia_3c dataset, among them are Alshamlan et al., (2015) used Genetic Bee Colony (GBC) algorithm for feature selection and Chormunge & Jena, (2018) employed Correlation-based feature selection (CBFS) with k-NN Cluster to select the relevant features. InfoGain-MBB Algorithm was able to select 3 genes that yielded 100% classification accuracy, which outperformed other approaches used in the past.
- iii. **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. Fatemeh, Sara & Mohammad, (2016) employed Cellular Learning Automata (CLA) optimized with Ant Colony Optimization method (ACO) tagged it as CLACOFS to select relevant features from 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.

Table 4.21: Comparison of Classification Accuracy with Other Classifiers in the Literature

Approach Used	Datasets							
	CNS	Leukemia_ 3c	Leukemia_4c	Lung	Lympho ma	MLL	Ovarian	SRBCT
1. Proposed: InfoGain- MBB Algorithm	100% (1)	100% (3)	100% (4)	100% (1)	100% (66)	100% (24)	100% (1)	100% (7)
2. CCFS (Ebrahimpour et al., 2018)	55% (33.4)						98% (32.8)	
3. SVDD & CS-SVM (Maldonado & López, 2018)								95.3% (250); 75.9% (50)
4. Distributed Feature selection (DF) (Bolón- Canedo et al., 2015)	55% (10%) 65% (25%)						100% (10% of dataset); 98.81% (25% Dataset)	
5. CLACOFS (Fatemeh et al., 2016)			86.38% (13.08)			98.95% (12.40)		

6. MOBBA-LS (Dashtbana et al., 2018)							86% (6); 100% (6); 100% (6)
7. GALA (Motieghader, Najafi, Sadeghi & Masoudi-Nejad, 2017)					93.96% (3)		99.34 (6)
8. mRMR-based ANN (Rana & Ahmed, 2020)			95% (100)	90% (100)	100% (100)	100% (100)	100% (500)
9. GBC (Alshamlan et al., 2015)		77.78% (50); 100% (250)	89.56% (50); 100% (200)	93.73% (50); 100% (200)			62.65 (50); 100% (250)
10. CBFS with Cluster (Chormunge & Jena, 2018)	90.05% (40% Dataset)	94.04% (40% Dataset)			93.70% (40% Dataset)		
11. Cuckoo Search (CS) (Gunavathi & Premalatha, 2015)	87.5% (100)		100% (50)			100% (100)	

12. k-SS (Yahya et al., 2011)	96.43%(4)	
13. SMOTE (Hambali & Gbolagade, 2016)	96.8% (50% Dataset)	
13. mRMR-BA (Alomari et al., 2017)		100% (3.83)

- iv. **Lung Dataset:** There are several approaches available in the literature that had explored Lung dataset, few among them include Gunavathi & Premalatha, (2015) employed Cuckoo Search (CS) feature selection and Alshamlan et al., (2015) used GBC algorithm to select informative genes. Both the approaches yielded 100% accuracy with not less than 50 genes but InfoGain-MBB Algorithm had able to produce 100% with one gene selected.
- v. **Lymphoma Dataset:** Alshamlan et al., (2015) used GBC algorithm to select informative genes which produce 93.73% and 100% classification accuracy from 50 and 200 genes respectively. InfoGain-MBB Algorithm outperformed this approach by producing 100% classification accuracy with 66 genes selected.
- vi. **MLL Dataset:** Few among the authors who employed the MLL dataset include Motieghader et al., (2017). 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 MLL dataset is Rana & Ahmed, (2020) that employed mRMR-based ANN to yield 100% accuracy with 100 genes selected. CBFS with Cluster was employed by Chormunge & Jena, (2018) to select 40% of feature from MLL dataset that yielded 93.70% classification accuracy. But InfoGain-MBB Algorithm yielded 100% classification accuracy with 24 genes.
- vii. **Ovarian Dataset:** Alomari et al., (2017) used mRMR-BA to select an average of 3.83 genes to yield 100% classification accuracy. Hambali & Gbolagade, (2016) used Synthetic Minority Oversampling Technique (SMOTE) for solving problem of data imbalance and their result showed 96.8% accuracy with 50% dataset selected. Gunavathi & Premalatha, (2015) employed Cuckoo Search (CS) to select

100 genes that produce 100% classification accuracy. Also, DF was employed by Bolón-Canedo et al., (2015) to produce 100% accuracy with 10% of the dataset and 98.81% accuracy with 25% of the dataset, Ebrahimpour et al., (2018) got 98% classification accuracy using CCFS to select an average of 32.8 genes.

- viii. **SRBCT Dataset:** Alshamlan et al., (2015) explored GBC to extract 50 and 100 genes from SRBCT that yielded 62.65% and 100% classification accuracy respectively. Rana & Ahmed, (2020) used mRMR-based ANN to select 500 genes that gave 100% classification accuracy. Motieghader et al., (2017) used GALA to select 6 genes to produce 99.34% accuracy. Dashtbana et al., (2018) employed Multi-objective binary bat algorithm with specific local searches (MOBBA-LS) to 6 genes that yielded 100% classification accuracy. Also, Maldonado & López, (2018) 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 4.21 only the work of Dashtbana et al., (2018) perform better than the approached used in this work in 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.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary

Cancer disease is one of the deadly diseases and it has been identified as the second prominent cause of death in the world. Medically, cancer diagnosis and detection were based on the examination of the organs where the tumour was developed. This sometimes resulted in the exhaustive physical and histopathological assessments of the organs that harbour the tumour. However, diagnoses are only achievable either through laboratory tests which might be too costly to bear or through surgical operations which might expose the patients to different kinds of risk.

The methodology for this study consists of three main steps, the first phase, is filtering technique where the feature reduction takes place and select the top-ranked relevant genes; three famous filter techniques (Chi-Squared, Information Gain, Gain Ratio) were utilized, compared, and exploited in the whole process. Then, an evolutionary algorithm for features subset technique based on the concept of Binary Bat algorithm was designed to exploit the subset of highly relevant genes in the second phase and decision trees algorithm (C4.5, Decision tree, Random forest and CART) were also utilized for classification at the last stage.

The efficiency of the approach was measured and compared. The results obtained showed that the approaches were good for classification of cancer microarray data. For example, Random forest has 100% accuracy in all eight datasets used and least value

accuracy obtained is 76% which demonstrated that all the classifier perform excellently with few genes selected.

5.2 Conclusion

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

Simulation results showed that the approach achieved superior performance improvement compared with the existing algorithms. It also requires less time to get accurate results. To validate the effectiveness of the InfoGain-MBB Algorithm, the evaluation was carried out using four different popular classifiers on eight publicly available datasets which included binary classes and multi-classes datasets. The experimental results are very promising and supported by existing biological knowledge, which shows that InfoGain-MBB Algorithm method can effectively perform gene selection with high classification accuracy and a small number of selected genes in all datasets used. Furthermore, the method can be directly employed for biomarker discovery in large scale proteomics datasets.

5.3 Recommendations

Based on the outcome of this study, InfoGain-MBB Algorithm is highly recommended as an alternative or supportive means of cancer prediction or classification using high-dimensional microarray datasets. With this approach, early diagnosis of cancer can be achieved, save a lot of time for awaiting laboratory result and other means of cancer diagnosis that are not reliable, and also reduce the exposure of a patient to avoidable hardship. This study proves to be effective and efficient in selecting the best and most

informative genes that help in accurate classification of cancer diseases. Due to its excellent performance, the study recommends that InfoGain-MBB Algorithm feature selection is an effective technique for gene selection in microarray data and should be used as part of the "standard tool-box" methods.

5.4 Contribution to Knowledge

Microarray data are characterized with challenges of irrelevant, high dimensional and redundant features. Because selecting relevant and informative genes from these features is usually a herculean task, a lot of statistical and machine learning approaches have been proposed and utilized. Metaheuristic or Bio-inspired approach is not left out of the methods used in the past. The main contributions of this study are as follows:

- i. Modification of BBA was done using Random forest as the fitness function, in order to make the algorithm robust for the selection of efficient genes in multiclass microarray data.
- ii. This study was able to establish the optimal threshold for each of the datasets used.

5.5 Limitation of the Study

While this research work was successful at attaining good result performances, there were certain limitations which build-up while carrying out this study. One of them is the computation time of executing MBB Algorithm, as one of the Bio-inspired algorithms; it usually takes a lot of time to implement them on microarray data. Also, the issue of memory space during feature extraction was encountered.

5.6 Future Work

In future work, the proposed method can be used with very huge datasets such as NCBI GEO datasets and others, to verify and extend the algorithm. Likewise, an improvement to BBA can be done by hybridizing other Bio-inspired algorithms to empower the exploitation process that results in improving the performance of BBA. Also, Hadoop approach can be adopted with BBA to reduce its computation time.

REFERENCES

- Abdulsalam, S O, Babatunde, A. N., Hambali, M. A., & Babatunde, R. S. (2015). Comparative Analysis of Decision Tree Algorithms for Predicting Undergraduate Students' Performance in Computer Programming. *Journal of Advances in Scientific Research & Its Application*, 2, 79–92.
- Abdulsalam, Sulaiman Olaniyi, Mohammed, A. A., Ajao, J. F., Babatunde, R. S., Ogundokun, R. O., Nnodim, C. T., & Arowolo, M. O. (2020). Performance Evaluation of ANOVA and RFE Algorithms for Classifying Microarray Dataset Using SVM. *European, Mediterranean, and Middle Eastern Conference on Information Systems*, 480–492. https://doi.org/10.1007/978-3-030-63396-7_32
- Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., & Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3), 392–398.
- Abeer, M. M., & Basma, A. M. (2014). A Hybrid Reduction Approach for Enhancing Cancer Classification of Microarray Data. *International Journal of Advanced Research in Artificial Intelligence*, 3(10).
- Acuna, E., Coaquira, F., & Gonzalez, M. (2003). A comparison of feature selection procedures for classifiers based on kernel density estimation. *Proc. of the Int. Conf. on Computer, Communication and Control Technologies, CCCT*, 3, 468–472.
- Acunzo, M., Romano, G., Wernicke, D., & Croce, C. M. (2015). MicroRNA and cancer—a brief overview. *Advances in Biological Regulation*, 57, 1–9.
- Agarwal, P., & Owzar, K. (2014). Next generation distributed computing for cancer research. *Cancer Informatics*, 13, CIN-S16344.
- Akhtar, S., Ahmad, A., & Abdel-Rahman E. (2012). A Metaheuristic Bat-Inspired Algorithm for Full Body Human Pose Estimation. *9th Conference on Computer and Robot Vision (CRV)*, 369–375.
- Alelyani, S. (2021). Stable bagging feature selection on medical data. *Journal of Big Data*, 8(1). <https://doi.org/10.1186/s40537-020-00385-8>
- Alomari, O. A., Khader, A. T., Mohammed, A. A. B., Abualigah, L. M., Nugroho, H., Chandra, G. R., Katyayani, A., Sandhya, N., HOSSAIN, J., & SANI, N. O. R. F. M. (2017). Mrmr ba: a hybrid gene selection algorithm for cancer classification. *J*

- Theoretical Appl Inf Techn*, 95(12), 15.
- Alshamlan, H. M., Badr, G. H., & Alohal, Y. A. (2015). Genetic Bee Colony (GBC) Algorithm: A New Gene Selection Method for Microarray Cancer Classification. *Computational Biology and Chemistry*, 56, 49–60.
- American Cancer Society. (2017). Cancer Facts & Figures 2017. In *American Cancer Society*.
- Anaissi, A., Kennedy, P. J., & Goyal, M. (2011). Feature selection of imbalanced gene expression microarray data. *2011 12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 73–78.
- Augenlicht, L. H., Taylor, J., Anderson, L., & Lipkin, M. (1991). Patterns of gene expression that characterize the colonic mucosa in patients at genetic risk for colonic cancer. *Proceedings of the National Academy of Sciences*, 88(8), 3286–3289.
- Augenlicht, L. H., Wahrman, M. Z., Halsey, H., Anderson, L., Taylor, J., & Lipkin, M. (1987). Expression of cloned sequences in biopsies of human colonic tissue and in colonic carcinoma cells induced to differentiate in vitro. *Cancer Research*, 47(22), 6017–6021.
- Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4), 537–550.
- Bekkerman, R., El-Yaniv, R., Tishby, N., & Winter, Y. (2003). Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 3(Mar), 1183–1208.
- Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., & Yakhini, Z. (2000). Tissue classification with gene expression profiles. *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, 54–64.
- Bennet, J., Ganaprakasam, C., & Kumar, N. (2015). A Hybrid Approach for Gene Selection and Classification using Support Vector Machine. *International Arab Journal of Information Technology (IAJIT)*, 12.
- Biau, G. (2012). Analysis of a Random Forests Model. *Journal Machine Learning Res.*, 13, 1063–1095.
- Bø, T. H., & Jonassen, I. (2002). New feature subset selection procedures for classification

- of expression profiles. *Genome Biology*, 3(4), research0017-1.
- Bolón-Canedo, V, Sánchez-Maróño, N., & Alonso-Betanzos, A. (2015). Distributed Feature Selection: An Application to Microarray Data Classification. *Applied Soft computing*, 30, 136–150.
- Bolón-Canedo, Verónica, Sánchez-Maróño, N., & Alonso-Betanzos, A. (2012). An ensemble of filters and classifiers for microarray data classification. *Pattern Recognition*, 45(1), 531–539.
- Bolón-Canedo, Verónica, Sánchez-Maróño, N., & Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowledge and Information Systems*, 34(3), 483–519.
- Bolón-Canedo, Verónica, Sánchez-Maróño, N., & Alonso-Betanzos, A. (2010). On the effectiveness of discretization on gene selection of microarray data. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 1–8.
- Bolón-Canedo, Verónica, Sánchez-Marono, N., Alonso-Betanzos, A., Benítez, J. M., & Herrera, F. (2014). A review of microarray datasets and applied feature selection methods. *Information Sciences*, 282, 111–135.
- Bolón-Canedo, Verónica, Seth, S., Sánchez-Marono, N., Alonso-Betanzos, A., & Príncipe, J. C. (2011). Statistical dependence measure for feature selection in microarray datasets. *ESANN*.
- Bozek, K., Lengauer, T., Sierra, S., Kaiser, R., & Domingues, F. S. (2013). Analysis of physicochemical and structural properties determining HIV-1 coreceptor usage. *PLoS Computational Biology*, 9(3).
- Brazma, A., & Vilo, J. (2000). Gene expression data analysis. *FEBS Letters*, 480(1), 17–24.
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees* (Vol. 432). Wadsworth. International Group.
- Brown, G., Pocock, A., Zhao, M.-J., & Luján, M. (2012). Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13(Jan), 27–66.
- Cai, D., He, X., & Han, J. (2007). Semi-Supervised Discriminant Analysis. *11th International Conference on ICCV*, 1–7.

- Cai, Deng, Zhang, C., & He, X. (2010). Unsupervised feature selection for multi-cluster data. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 333–342.
- Cai, R., Hao, Z., Yang, X., & Wen, W. (2009). An efficient gene selection algorithm based on mutual information. *Neurocomputing*, 72(4–6), 991–999.
- Canul-Reich, J., Hall, L. O., Goldgof, D. B., Korecki, J. N., & Eschrich, S. (2012). Iterative feature perturbation as a gene selector for microarray data. *International Journal of Pattern Recognition and Artificial Intelligence*, 26(05), 1260003.
- 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. *Journal of Biomedical Informatics*, 53, 381–389.
- Caruana, R., & Sa, V. R. de. (2003). Benefitting from the variables that variable selection discards. *Journal of Machine Learning Research*, 3(Mar), 1245–1264.
- Chang, X., Nie, F., Yang, Y., & Huang, H. (2014). A convex formulation for semi-supervised multi-label feature selection. *Twenty-Eighth AAAI Conference on Artificial Intelligence*.
- Chen, C.-R., Shu, W.-Y., Tsai, M.-L., Cheng, W.-C., & Hsu, I. C. (2012). THEME: A web tool for loop-design microarray data analysis. *Computers in Biology and Medicine*, 42(2), 228–234.
- Cheng, Q., Zhou, H., & Cheng, J. (2010). The fisher-markov selector: fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6), 1217–1233.
- Cho, J.-H., Lee, D., Park, J. H., & Lee, I.-B. (2004). Gene selection and classification from microarray data using kernel machine. *FEBS Letters*, 571(1–3), 93–98.
- Chormunge, S., & Jena, S. (2018). Correlation based feature selection with clustering for high dimensional data. *Journal of Electrical Systems and Information Technology*, 5(3), 542–549.
- Chow, M. L., Moler, E. J., & Mian, I. S. (2001). Identifying marker genes in transcription profiling data using a mixture of feature relevance experts. *Physiological Genomics*, 5(2), 99–111.

- Chuang, L.-Y., Ke, C.-H., & Yang, C.-H. (2016). A hybrid both filter and wrapper feature selection method for microarray classification. *ArXiv Preprint ArXiv:1612.08669*.
- Chuang, L.-Y., Yang, C.-H., Wu, K.-C., & Yang, C.-H. (2011). A hybrid feature selection method for DNA microarray data. *Computers in Biology and Medicine*, 41(4), 228–237.
- Chuang, L. Y., Yang, C. H., & Li, J. C. (2011). A Hybrid BPSO-CGA Approach for Gene Selection and Classification of Microarray Data. *Journal of Computational Biology*, 19, 1–14.
- Daelemans, W., Hoste, V., De Meulder, F., & Naudts, B. (2003). Combined optimization of feature selection and algorithm parameters in machine learning of language. *European Conference on Machine Learning*, 84–95.
- Dash, M., & Liu, H. (2003). Consistency-based search in feature selection. *Artificial Intelligence*, 151(1–2), 155–176.
- Dash, R. (2017). A two stage grading approach for feature selection and classification of microarray data using Pareto based feature ranking techniques: A case study. *Journal of King Saud University-Computer and Information Sciences*.
- Dash, R. (2018). An adaptive harmony search approach for gene selection and classification of high dimensional medical data. *Journal of King Saud University-Computer and Information Sciences*.
- Dashtban, M., & Balafar, M. (2017). Gene selection for microarray cancer classification using a new evolutionary method employing artificial intelligence concepts. *Genomics*, 109(2), 91–107.
- Dashtbana, M., Mohammadali, B., & Prashanth, S. (2018). Gene selection for Tumor Classification using a Novel Bio-Inspired Multiobjective Approach. *Genomics*, 110, 10–17.
- Davidson, J. L., & Jalan, J. (2010). Feature selection for steganalysis using the Mahalanobis distance. *Media Forensics and Security II*, 7541, 754104.
- Dev, J., Dash, S., Dash, S., & Swain, M. (2012). A Classification Technique for Microarray Gene Expression Data using PSO-FLANN. *International Journal on Computer science and Engineering*, 4(9), 1534–1535.
- Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene

- expression data. *Journal of Bioinformatics and Computational Biology*, 3(02), 185–205.
- Doreswamy, & UmmeSalma, M. (2016). A Binary Bat Inspired Algorithm for the Classification of Breast Cancer Data. *International Journal on Soft Computing, Artificial Intelligence and Applications*, 5(2/3), 1–21.
- Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.
- Dy, J. G., & Brodley, C. E. (2004). Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5(Aug), 845–889.
- Ebrahimpour, M. K., Nezamabadi-Pour, H., & Eftekhari, M. (2018). CCFS: A cooperating coevolution technique for large scale feature selection on microarray datasets. *Computational Biology and Chemistry*, 73, 171–178.
- Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25), 14863–14868.
- El Akadi, A., Amine, A., El Ouardighi, A., & Aboutajdine, D. (2011). A Two-Stage Gene Selection Scheme Utilizing MRMR Filter and GA Wrapper. *Knowledge and Information Systems*, 26(3), 487–500.
- Elhamifar, E., & Vidal, R. (2013). Sparse Subspace Clustering: Algorithm, Theory and Applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11), 2765–2781.
- Fatemeh, V. S., Sara, M., & Mohammad, H. M. (2016). A Hybrid Gene Selection Approach for Microarray Data Classification using Cellular Learning Automata and Ant Colony Optimization. *Genomics*, 107, 231–238.
- Ferreira, A. J., & Figueiredo, M. A. T. (2012). An unsupervised approach to feature discretization and selection. *Pattern Recognition*, 45(9), 3048–3060.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2), 179–188.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer series in statistics New York.
- Fukunaga, K. (2013). *Introduction to statistical pattern recognition*. Elsevier.
- Gandomi, A. H., Yang, X.-S., Alavi, A. H., & Talatahari, S. (2013). Bat Algorithm for Constrained Optimization Tasks. *Neural Computing and Applications. Neural*

- Computing and Applications*, 22(6), 1239–1255.
- Gao, L., Ye, M., Lu, X., & Huang, D. (2017). Hybrid method based on information gain and support vector machine for gene selection in cancer classification. *Genomics, Proteomics & Bioinformatics*, 15(6), 389–395.
- García, S., Luengo, J., & Herrera, F. (2016). Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. *Knowledge-Based Systems*, 98, 1–29.
- Garro, B. A., Rodríguez, K., & Vázquez, R. A. (2016). Classification of DNA microarrays using artificial neural networks and ABC algorithm. *Applied Soft Computing*, 38, 548–560.
- 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. *Journal of Medical Systems*, 43(9), 286.
- Genuer, R., Poggi, J.-M., & Tuleau, C. (2008). Random Forests: some methodological insights. *ArXiv Preprint ArXiv:0811.3619*.
- Ghorai, S., Mukherjee, A., Sengupta, S., & Dutta, P. K. (2010). Cancer classification from gene expression data by NPPC ensemble. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(3), 659–671.
- Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., & Caligiuri, M. A. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439), 531–537.
- Grate, L. R., Bhattacharyya, C., Jordan, M. I., & Mian, I. S. (2002). Simultaneous relevant feature identification and classification in high-dimensional spaces. *International Workshop on Algorithms in Bioinformatics*, 1–9.
- Griffin, D. R., Webster, F. A., & Michael, C. R. (1965). The echolocation of flying insects by bats. *Readings in the Psychology of Perception*, 21.
- Gunavathi, C., & Premalatha, K. (2015). Cuckoo search optimisation for feature selection in cancer classification: a new approach. *International Journal of Data Mining and Bioinformatics*, 13(3), 248–265.
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection.

- Journal of Machine Learning Research*, 3(Mar), 1157–1182.
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1–3), 389–422.
- Hall, Mark A, & Smith, L. A. (1998). *Practical feature subset selection for machine learning*.
- Hall, Mark Andrew. (1999). *Correlation-based feature selection for machine learning*.
- Hambali, M. A., & Gbolagade, M. D. (2016). Ovarian cancer classification using hybrid synthetic minority over-sampling technique and neural network. *Journal of Advances in Computer Research*, 7(4), 109–124.
- Hambali, M. A., Oladele, T. O., & Adewole, K. S. (2020). Microarray cancer feature selection: Review , challenges and research directions. *International Journal of Cognitive Computing in Engineering*, 1(October), 78–97.
<https://doi.org/10.1016/j.ijcce.2020.11.001>
- Hambali, M. A., Saheed, Y. K., Oladele, T. O., & Gbolagade, M. D. (2019). ADABOOST ENSEMBLE ALGORITHMS FOR BREAST CANCER. *Journal of Advances in Computer Research*, 10(2), 31–52.
- Hambali, M., Saheed, Y., Oladele, T., & Gbolagade, M. (2019). ADABOOST Ensemble Algorithms for Breast Cancer Classification. *Journal of Advances in Computer Research*, 10(2), 31–52. http://jacr.iausari.ac.ir/article_663924.html
- Hammond, S. M. (2015). An Overview of MicroRNAs. *Advanced Drug Delivery Reviews*.
<https://doi.org/10.1016/j.addr.2015.05.001>
- Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
- Han, Y., Yang, Y., Yan, Y., Ma, Z., Sebe, N., & Zhou, X. (2014). Semisupervised feature selection via spline regression for video semantic recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 26(2), 252–264.
- Hancer, E., Xue, B., Karaboga, D., & Zhang, M. (2015). A binary ABC algorithm based on advanced similarity scheme for feature selection. *Applied Soft Computing*, 36, 334–348.
- He, W., Cheng, X., Hu, R., Zhu, Y., & Wen, G. (2017). Feature self-representation based hypergraph unsupervised feature selection via low-rank representation. *Neurocomputing*, 253, 127–134.

- He, X., Cai, D., & Niyogi, P. (2006). Laplacian score for feature selection. *Advances in Neural Information Processing Systems*, 507–514.
- Hira, Z. M., & Gillies, D. F. (2015). A review of feature selection and feature extraction methods applied on microarray data. *Advances in Bioinformatics*, 2015.
- Hou, C., Nie, F., Li, X., Yi, D., & Wu, Y. (2013). Joint embedding learning and sparse regression: A framework for unsupervised feature selection. *IEEE Transactions on Cybernetics*, 44(6), 793–804.
- Hu, R., Zhu, X., Cheng, D., He, W., Yan, Y., Song, J., & Zhang, S. (2017). Graph self-representation method for unsupervised feature selection. *Neurocomputing*, 220, 130–137.
- Huang, H.-L., & Chang, F.-L. (2007). ESVM: Evolutionary support vector machine for automatic feature selection and classification of microarray data. *Biosystems*, 90(2), 516–528.
- Huang, S. H. (2015). Supervised feature selection: A tutorial. *Artif. Intell. Research*, 4(2), 22–37.
- Inza, I., Sierra, B., Blanco, R., & Larrañaga, P. (2002). Gene selection by sequential search wrapper approaches in microarray cancer class prediction. *Journal of Intelligent & Fuzzy Systems*, 12(1), 25–33.
- Javed, K., Babri, H. A., & Saeed, M. (2010). Feature selection based on class-dependent densities for high-dimensional binary data. *IEEE Transactions on Knowledge and Data Engineering*, 24(3), 465–477.
- Kabir, M. M., Shahjahan, M., & Murase, K. (2012). A new hybrid ant colony optimization algorithm for feature selection. *Expert Systems with Applications*, 39(3), 3747–3763.
- Kar, S., Sharma, K. Das, & Maitra, M. (2015). Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive K-nearest neighborhood technique. *Expert Systems with Applications*, 42(1), 612–627.
- Khan, M. W., & Alam, M. (2012). A survey of application: Genomics and genetic programming, a new frontier. *Genomics*, 100(2), 65–71.
- Khodarev, N. N., Park, J., Kataoka, Y., Nodzenski, E., Hellman, S., Roizman, B., Weichselbaum, R. R., & Pelizzari, C. A. (2003). Receiver operating characteristic

- analysis: a general tool for DNA array data filtration and performance estimation. *Genomics*, 81(2), 202–209.
- Kim, J., Kwon, J., Kim, J. E., Koh, W. S., Chung, M., Yoon, S., Song, C. W., & Lee, M. (2005). Identification of potential biomarkers of genotoxicity and carcinogenicity in L5178Y mouse lymphoma cells by cDNA microarray analysis. *Environmental and Molecular Mutagenesis*, 45(1), 80–89.
- Kira, K., & Rendell, L. A. (1992a). A practical approach to feature selection. In *Machine Learning Proceedings 1992* (pp. 249–256). Elsevier.
- Kira, K., & Rendell, L. A. (1992b). The feature selection problem: Traditional methods and a new algorithm. *Aaii*, 2, 129–134.
- Komarasamy, G., & Wahi, A. (2012). An Optimized Kmeans Clustering Technique using Bat Algorithm. *European Journal of Scientific Research*, 84(2), 26–273.
- Kong, C. S., Yu, J., Minion, F. C., & Rajan, K. (2011). Identification of biologically significant genes from combinatorial microarray data. *ACS Combinatorial Science*, 13(5), 562–571.
- Kononenko, I. (1994). Estimating attributes: analysis and extensions of RELIEF. *European Conference on Machine Learning*, 171–182.
- Krzanowski, W. J. (1987). Selection of variables to preserve multivariate data structure, using principal components. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 36(1), 22–33.
- Lan, L., & Vucetic, S. (2011). Improving accuracy of microarray classification by a simple multi-task feature selection filter. *International Journal of Data Mining and Bioinformatics*, 5(2), 189–208.
- Lee, C.-P., & Leu, Y. (2011). A novel hybrid feature selection method for microarray data analysis. *Applied Soft Computing*, 11(1), 208–213.
- Leung, Y., & Hung, Y. (2008). A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(1), 108–117.
- Li, F., Piao, M., Piao, Y., Li, M., & Ryu, K. H. (2014). A New direction of cancer classification: positive effect of Low-ranking MicroRNAs. *Osong Public Health and Research Perspectives*, 5(5), 279–285.

- Li, H., Jiang, T., & Zhang, K. (2004). Efficient and robust feature extraction by maximum margin criterion. *Advances in Neural Information Processing Systems*, 97–104.
- Li, J., Chen, Z., Wei, L., Xu, W., & Kou, G. (2007). Feature selection via least squares support feature machine. *International Journal of Information Technology & Decision Making*, 6(04), 671–686.
- Li, L., Jiang, W., Li, X., Moser, K. L., Guo, Z., Du, L., Wang, Q., Topol, E. J., Wang, Q., & Rao, S. (2005). A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset. *Genomics*, 85(1), 16–23.
- Li, X., Peng, S., Chen, J., Lü, B., Zhang, H., & Lai, M. (2012). SVM-T-RFE: A novel gene selection algorithm for identifying metastasis-related genes in colorectal cancer using gene expression profiles. *Biochemical and Biophysical Research Communications*, 419(2), 148–153.
- Li, Z., Liu, J., Yang, Y., Zhou, X., & Lu, H. (2013). Clustering-guided sparse structural learning for unsupervised feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 26(9), 2138–2150.
- Liang, S., Ma, A., Yang, S., Wang, Y., & Ma, Q. (2018). A Review of Matched-pairs Feature Selection Methods for Gene Expression Data Analysis. *Computational and Structural Biotechnology Journal*, 16, 88–97.
<https://doi.org/10.1016/j.csbj.2018.02.005>
- Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, 2(3), 18–22.
- Lin, J. H., Chou, C. W., Yang, C. H., & Tsai, H. L. (2012). A Chaotic Levy Flight Bat Algorithm for Parameter Estimation in Nonlinear Dynamic Biological Systems. *Computer and Information Technology*, 2(2), 56–63.
- Lin, S.-W., & Chen, S.-C. (2012). Parameter determination and feature selection for C4. 5 algorithm using scatter search approach. *Soft Computing*, 16(1), 63–75.
- 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).
- Liu, H., Motoda, H., Setiono, R., & Zhao, Z. (2010). Feature selection: An ever evolving frontier in data mining. *Feature Selection in Data Mining*, 4–13.
- Liu, R., Rallo, R., & Cohen, Y. (2011). Unsupervised feature selection using incremental

- least squares. *International Journal of Information Technology & Decision Making*, 10(06), 967–987.
- Liu, Y., Nie, F., Wu, J., & Chen, L. (2013). Efficient semi-supervised feature selection with noise insensitive trace ratio criterion. *Neurocomputing*, 105, 12–18.
- Lønning, P. E., Sørbye, T., & Børresen-Dale, A.-L. (2005). Genomics in breast cancer—therapeutic implications. *Nature Clinical Practice Oncology*, 2(1), 26–33.
- Lovato, P., Bicego, M., Cristani, M., Jojic, N., & Perina, A. (2012). Feature selection using counting grids: application to microarray data. *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 629–637.
- Luo, T., Hou, C., Nie, F., Tao, H., & Yi, D. (2018). Semi-supervised feature selection via insensitive sparse regression with application to video semantic recognition. *IEEE Transactions on Knowledge and Data Engineering*, 30(10), 1943–1956.
- Luo, Y., Tao, D., Xu, C., Li, D., & Xu, C. (2013). Vector-valued multi-view semi-supervised learning for multi-label image classification. *Twenty-Seventh AAAI Conference on Artificial Intelligence*.
- Maldonado, S., & López, J. (2018). Dealing with high-dimensional class-imbalanced datasets: embedded feature selection for SVM classification. *Applied Soft Computing*, 67, 94–105.
- Maldonado, S., Weber, R., & Basak, J. (2011). Simultaneous feature selection and classification using kernel-penalized support vector machines. *Information Sciences*, 181(1), 115–128.
- Mao, Z., Cai, W., & Shao, X. (2013). Selecting significant genes by randomization test for cancer classification using gene expression data. *Journal of Biomedical Informatics*, 46(4), 594–601.
- Martens, H. (2001). Reliable and relevant modelling of real world data: a personal account of the development of PLS regression. *Chemometrics and Intelligent Laboratory Systems*, 58(2), 85–95.
- Martín-Merino, M., & De Las Rivas, J. (2009). Improving k-nn for human cancer classification using the gene expression profiles. *International Symposium on Intelligent Data Analysis*, 107–118.

- Metzner, W. (1991). Echolocation Behaviour in Bats. *Science Progress Edinburgh*, 75(298), 453–465. <http://files/27/ADABOOST> Ensemble Algorithms for Breast Cancer Classification.ris
- Meyer, P. E., Schretter, C., & Bontempi, G. (2008). Information-theoretic feature selection in microarray data using variable complementarity. *IEEE Journal of Selected Topics in Signal Processing*, 2(3), 261–274.
- Mishra, S., Shaw, K., & Mishra, D. (2012). A new meta-heuristic bat inspired classification approach for microarray data. *Procedia Technology*, 4, 802–806.
- Mitchell, T. M. (1997). *Machine learning*.
- Mohammadi, M., Noghabi, H. S., Hodtani, G. A., & Mashhadi, H. R. (2016). Robust and stable gene selection via maximum–minimum correntropy criterion. *Genomics*, 107(2–3), 83–87.
- Monobe, M. M. de S., & Silva, R. C. da. (2016). Gene expression: an overview of methods and applications for cancer research. *Veterinaria e Zootecnia*, 23(4), 532–546.
- Moody, J., & Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2), 281–294.
- 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 in Medicine Unlocked*, 9, 246–254.
- Mundra, P. A., & Rajapakse, J. C. (2009). SVM-RFE with MRMR filter for gene selection. *IEEE Transactions on Nanobioscience*, 9(1), 31–37.
- Musikapun, P., & Pongcharoen, P. (2012). Solving Multistage Multi-Machine Multi-Product Scheduling Problem using Bat Algorithm. *2nd International Conference on Management and Artificial Intelligence (IACSIT)*, 35, 98–102.
- Mutch, D. M., Berger, A., Mansourian, R., Rytz, A., & Roberts, M.-A. (2001). Microarray data analysis: a practical approach for selecting differentially expressed genes. *Genome Biology*, 2(12), preprint0009-1.
- Nakamura, R. Y. M., Pereira, L. A. M., Costa, K. A., Rodrigues, D., Papa, J. P., & Yang, X.-S. (2012). BBA: a binary bat algorithm for feature selection. *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, 291–297.
- Narayanan, A., Keedwell, E. C., & Olsson, B. (2002). Artificial intelligence techniques for

- bioinformatics. *Applied Bioinformatics*, 1, 191–222.
- Navarro, F. F. G., & Muñoz, L. A. B. (2009). Gene subset selection in microarray data using entropic filtering for cancer classification. *Expert Systems*, 26(1), 113–124.
- Nguyen, D. V., & Rocke, D. M. (2002a). Multi-class cancer classification via partial least squares with gene expression profiles. *Bioinformatics*, 18(9), 1216–1226.
- Nguyen, D. V., & Rocke, D. M. (2002b). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, 18(1), 39–50.
- Nie, F., Huang, H., Cai, X., & Ding, C. H. (2010). Efficient and robust feature selection via joint ℓ_2 , 1-norms minimization. *Advances in Neural Information Processing Systems*, 1813–1821.
- Ogundipe, S. (2017, February 6). WHO Urges Early Detection of Cancer, as 80,000 Die Yearly in Nigeria. *Vanguard News*. <https://www.vanguardngr.com/2017/02/urges-early-detection-cancer-80000-die-yearly-nigeria/>
- Panigrahi, P. P., & Singh, T. R. (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. *Journal of Theoretical Biology*, 334, 109–121.
- Panigrahi, R., & Borah, S. (2018). Rank allocation to J48 group of decision tree classifiers using binary and multiclass intrusion detection datasets. *Procedia Computer Science*, 132, 323–332.
- Passi, K., Nour, A., & Jain, C. K. (2017). Markov blanket: Efficient strategy for feature subset selection method for high dimensional microarray cancer datasets. *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 1864–1871.
- Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8), 1226–1238.
- Pihur, V., Datta, S., & Datta, S. (2008). Finding common genes in multiple cancer types through meta-analysis of microarray experiments: A rank aggregation approach. *Genomics*, 92(6), 400–403.
- Pirooznia, M., Yang, J. Y., Yang, M. Q., & Deng, Y. (2008). A comparative study of different machine learning methods on microarray gene expression data. *BMC*

- Genomics*, 9(S1), S13.
- 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 Systems with Applications*, 36(2), 1587–1592.
- Qi, Y., Sun, H., Sun, Q., & Pan, L. (2011). Ranking analysis for identifying differentially expressed genes. *Genomics*, 97(5), 326–329.
- Quackenbush, J. (2001). Computational analysis of microarray data. *Nature Reviews Genetics*, 2(6), 418–427.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106.
- Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.
- Rafii, F., Hassani, B. D. R., & Kbir, M. A. (2017). New Approach for Microarray Data Decision Making with Respect to Multiple Sources. *Proceedings of the 2nd International Conference on Big Data, Cloud and Applications*, 1–5.
- Raileanu, L. E., & Stoffel, K. (2004). Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1), 77–93.
- Rajeswari, P., & Reena, G. S. (2011). Human liver cancer classification using microarray gene expression data. *International Journal of Computer Applications*, 34(6), 25–37.
- Ramesh, B., Mohan, V. C. J., & Reddy, V. V. (2013). pplication of Bat Algorithm for Combined Economic Load and Emission Dispatch. *Int. Journal of Electrical Engineering and Telecommunications*, 2(1), 1–9.
- Rana, M. M., & Ahmed, K. (2020). Feature Selection and Biomedical Signal Classification Using Minimum Redundancy Maximum Relevance and Artificial Neural Network. *Proceedings of International Joint Conference on Computational Intelligence*, 207–214.
- Remeseiro López, B., & Bolon Canedo, V. (2019). A review of feature selection methods in medical applications. *Computers in Biology and Medicine*, 112.
- Revathy, N., & Amalraj, R. (2011). Accurate cancer classification using expressions of very few genes. *International Journal of Computer Applications*, 14(4), 19–22.
- Rodrigues, D., Pereira, L. A. M., Nakamura, R. Y. M., Costa, K. A. P., Yang, X.-S., Souza, A. N., & Papa, J. P. (2014). A wrapper approach for feature selection based on bat

- algorithm and optimum-path forest. *Expert Systems with Applications*, 41(5), 2250–2258.
- Ruiz, R., Riquelme, J. C., & Aguilar-Ruiz, J. S. (2006). Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition*, 39(12), 2383–2392.
- Saeid, M. M., Nossair, Z. B., & Saleh, M. A. (2020). A microarray cancer classification technique based on discrete wavelet transform for data reduction and genetic algorithm for feature selection. *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 857–861.
- Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19), 2507–2517.
- Sahu, B., & Mishra, D. (2012). A novel feature selection algorithm using particle swarm optimization for cancer microarray data. *Procedia Engineering*, 38, 27–31.
- Sánchez-Marono, N., Alonso-Betanzos, A., García-González, P., & Bolón-Canedo, V. (2010). Multiclass classifiers vs multiple binary classifiers using filters for feature selection. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 1–8.
- Santos, V., Datia, N., & Pato, M. P. M. (2014). Ensemble feature ranking applied to medical data. *Procedia Technology*, 17, 223–230.
- Saqib, P., Qamar, U., Khan, R. A., & Aslam, A. (2020). MF-GARF: Hybridizing Multiple Filters and GA Wrapper for Feature Selection of Microarray Cancer Datasets. *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, 517–524.
- Sathya, M., & Priya, S. M. (2020). A search space enhanced modified whale optimization algorithm for feature selection in large-scale microarray datasets. *Indian Journal of Science and Technology*, 13(42), 4396–4406.
- Schnitzler, H.-U., & Kalko, E. K. V. (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.
- Seera, M., & Lim, C. P. (2014). A hybrid intelligent system for medical data classification.

- Expert Systems with Applications*, 41(5), 2239–2249.
- Seijo-Pardo, B., Bolón-Canedo, V., & Alonso-Betanzos, A. (2019). On developing an automatic threshold applied to feature selection ensembles. *Information Fusion*, 45, 227–245.
- Selvaraj, S., & Natarajan, J. (2011). Microarray data analysis and mining tools. *Bioinformation*, 6(3), 95.
- Shafi, A. S. M., Molla, M. M. I., Jui, J. J., & Rahman, M. M. (2020). Detection of colon cancer based on microarray dataset using machine learning as a feature selection and classification techniques. *SN Applied Sciences*, 2(7), 1–8.
- Shah, M., Marchand, M., & Corbeil, J. (2011). Feature selection with conjunctions of decision stumps and learning from microarray data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1), 174–186.
- Sharbaf, F. V., Mosafer, S., & Moattar, M. H. (2016). A hybrid gene selection approach for microarray data classification using cellular learning automata and ant colony optimization. *Genomics*, 107(6), 231–238.
- Sharma, Alok, Imoto, S., & Miyano, S. (2011). A top-r feature selection algorithm for microarray gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(3), 754–764.
- Sharma, Anuj, & Dey, S. (2012). Performance investigation of feature selection methods and sentiment lexicons for sentiment analysis. *IJCA Special Issue on Advanced Computing and Communication Technologies for HPC Applications*, 3, 15–20.
- Shekar, B. H., & Dagneu, G. (2020). L1-Regulated Feature Selection and Classification of Microarray Cancer Data Using Deep Learning. *Proceedings of 3rd International Conference on Computer Vision and Image Processing*, 227–242.
- Shreem, S. S., Abdullah, S., & Nazri, M. Z. A. (2014). Hybridising harmony search with a Markov blanket for gene selection problems. *Information Sciences*, 258, 108–121.
- Shreem, S. S., Abdullah, S., Nazri, M. Z. A., & Alzaqebah, M. (2012). Hybridizing ReliefF, MRMR filters and GA wrapper approaches for gene selection. *J. Theor. Appl. Inf. Technol*, 46(2), 1034–1039.
- Song, L., Bedo, J., Borgwardt, K. M., Gretton, A., & Smola, A. (2007). Gene selection via the BAHSIC family of algorithms. *Bioinformatics*, 23(13), i490–i498.

- Song, L., Smola, A., Gretton, A., Bedo, J., & Borgwardt, K. (2012). Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(May), 1393–1434.
- Song, Q., Ni, J., & Wang, G. (2011). A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, 25(1), 1–14.
- Stoppiglia, H., Dreyfus, G., Dubois, R., & Oussar, Y. (2003). Ranking a random feature for variable and feature selection. *Journal of Machine Learning Research*, 3(Mar), 1399–1414.
- Student, S., & Fujarewicz, K. (2012). Stable feature selection and classification algorithms for multiclass microarray data. *Biology Direct*, 7(1), 33.
- Su, Y., Murali, T. M., Pavlovic, V., Schaffer, M., & Kasif, S. (2003). RankGene: identification of diagnostic genes based on expression data. *Bioinformatics*, 19(12), 1578–1579.
- Suresh, A., Udendhran, R., & Balamurgan, M. (2019). Hybridized neural network and decision tree based classifier for prognostic decision making in breast cancers. *Soft Computing*, 1–7.
- Svrakic, N. M., Nesic, O., Dasu, M. R. K., Herndon, D., & Perez-Polo, J. R. (2003). Statistical approach to DNA chip analysis. *Recent Progress in Hormone Research*, 58, 75–94.
- Swathi, S., Babu, G. A., Sendhilkumar, R., & Bhukya, S. N. (2012). Performance of ART1 Network in the Detection of Breast Cancer. *Proceedings of International Conference on Computer Design and Engineering (ICCDE 2012)*, 49, 100–105.
- Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., & Golub, T. R. (1999). Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences*, 96(6), 2907–2912.
- Tan, F., Fu, X., Zhang, Y., & Bourgeois, A. G. (2008). A genetic algorithm-based method for feature subset selection. *Soft Computing*, 12(2), 111–120.
- Tang, R., Fong, S., Yang, X.-S., & Deb, S. (2012). Integrating nature-inspired optimization algorithms to K-means clustering. *Seventh International Conference on Digital*

- Information Management (ICDIM 2012)*, 116–123.
- Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J., & Church, G. M. (1999). Systematic determination of genetic network architecture. *Nature Genetics*, 22(3), 281–285.
- Taylor, A., Steinberg, J., Andrews, T. S., & Webber, C. (2015). GeneNet Toolbox for MATLAB: a flexible platform for the analysis of gene connectivity in biological networks. *Bioinformatics*, 31(3), 442–444.
- Tong, D. L., & Schierz, A. C. (2011). Hybrid genetic algorithm-neural network: Feature extraction for unprocessed microarray data. *Artificial Intelligence in Medicine*, 53(1), 47–56.
- Tu, K., Yu, H., Guo, Z., & Li, X. (2004). Learnability-based further prediction of gene functions in Gene Ontology. *Genomics*, 84(6), 922–928.
- Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 417–424.
- Veerabhadrapa, & Rangarajan, L. (2010). Bi-level dimensionality reduction methods using feature selection and feature extraction. *International Journal of Computer Applications*, 4(2), 33–38.
- Ventimiglia, G., & Petralia, S. (2013). Recent advances in DNA microarray technology: an overview on production strategies and detection methods. *BioNanoScience*, 3(4), 428–450.
- Vukusic, I., Grellscheid, S. N., & Wiehe, T. (2007). Applying genetic programming to the prediction of alternative mRNA splice variants. *Genomics*, 89(4), 471–479.
- Wanderley, M. F. B., Gardeux, V., Natowicz, R., & de Pádua Braga, A. (2013). GA-KDE-Bayes: an evolutionary wrapper method based on non-parametric density estimation applied to bioinformatics problems. *ESANN*.
- Wang, Gaige, & Guo, L. (2013). A novel hybrid bat algorithm with harmony search for global numerical optimization. *Journal of Applied Mathematics*, 2013.
- Wang, Guangtao, Song, Q., Xu, B., & Zhou, Y. (2013). Selecting feature subset for high dimensional data via the propositional FOIL rules. *Pattern Recognition*, 46(1), 199–214.

- Wang, J., Bø, T. H., Jonassen, I., Myklebost, O., & Hovig, E. (2003). Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data. *BMC Bioinformatics*, 4(1), 60.
- Wang, L., Chu, F., & Xie, W. (2007). Accurate cancer classification using expressions of very few genes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(1), 40–53.
- Wang, S., Tang, J., & Liu, H. (2015). Embedded unsupervised feature selection. *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- Wang, Y., Chen, X., Jiang, W., Li, L., Li, W., Yang, L., Liao, M., Lian, B., Lv, Y., & Wang, S. (2011). Predicting human microRNA precursors based on an optimized feature subset generated by GA–SVM. *Genomics*, 98(2), 73–78.
- WHO. (2018). *Cancer Facts Sheet*. WHO. <http://www.who.int/mediacentre/factsheets/fs297/en/>
- Wisesty, U. N., Sthevanie, F., & Rismala, R. (2021). Momentum Backpropagation Optimization for Cancer Detection Based on DNA Microarray Data. *International Journal of Artificial Intelligence Research*, 4(2), 127. <https://doi.org/10.29099/ijair.v4i2.188>
- Xing, E. P., Jordan, M. I., & Karp, R. M. (2001). Feature selection for high-dimensional genomic microarray data. *Icml*, 1, 601–608.
- Xiong, W., Cai, Z., & Ma, J. (2008). A DSRPCL-SVM approach to informative gene analysis. *Genomics, Proteomics & Bioinformatics*, 6(2), 83–90.
- Xu, Z., King, I., Lyu, M. R.-T., & Jin, R. (2010). Discriminative semi-supervised feature selection via manifold regularization. *IEEE Transactions on Neural Networks*, 21(7), 1033–1047.
- Yahya, W. B., Ulm, K., Ludwig, F., & Hapflemeir, A. (2011). K-SS: A sequential feature selection and prediction method in microarray study. *International Journal of Artificial Intelligence*, 6(S11), 19–47.
- Yang, F., & Mao, K. Z. (2010). Robust feature selection for microarray data based on multicriterion fusion. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(4), 1080–1092.
- Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm. *Nature-Inspired*

- Cooperative Strategies for Optimization (NICSO 2010)*, 65–74.
- Yang, X., & Gandomi, A. H. (2012). Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*.
- Yang, Xin-She. (2010). *Nature-inspired metaheuristic algorithms*. Luniver press.
- Yang, Xin-She. (2011). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, 3(5), 267–274.
<https://doi.org/10.1504/IJBIC.2011.042259>
- Yang, Xin-She. (2013). Bat algorithm: literature review and applications. *ArXiv Preprint ArXiv:1308.3900*.
- Yang, Y., Shen, H. T., Ma, Z., Huang, Z., & Zhou, X. (2011). L2, 1-norm regularized discriminative feature selection for unsupervised. *Twenty-Second International Joint Conference on Artificial Intelligence*.
- Ye, Y., Wu, Q., Huang, J. Z., Ng, M. K., & Li, X. (2013). Stratified sampling for feature subspace selection in random forests for high dimensional data. *Pattern Recognition*, 46(3), 769–787.
- Yu, H., Gu, G., Liu, H., Shen, J., & Zhao, J. (2009). A modified ant colony optimization algorithm for tumor marker gene selection. *Genomics, Proteomics & Bioinformatics*, 7(4), 200–208.
- Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5(Oct), 1205–1224.
- Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 856–863.
- Zahiri, J., Yaghoubi, O., Mohammad-Noori, M., Ebrahimpour, R., & Masoudi-Nejad, A. (2013). PPlevo: Protein–protein interaction prediction from PSSM based evolutionary information. *Genomics*, 102(4), 237–242.
- Zhang, R., Nie, F., & Li, X. (2018). Feature selection under regularized orthogonal least square regression with optimal scaling. *Neurocomputing*, 273, 547–553.
- Zhang, R., Nie, F., Li, X., & Wei, X. (2019). Feature selection with multi-view data: A survey. *Information Fusion*, 50, 158–167.
- Zhang, Y., Gong, D., Hu, Y., & Zhang, W. (2015). Feature selection algorithm based on

- bare bones particle swarm optimization. *Neurocomputing*, 148, 150–157.
- Zhao, Z., & Liu, H. (2007). Searching for Interacting Features. *The 20th International Joint Conference on Artificial Intelligence*, 1156–1161.
- Zhao, Zheng, & Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. *Proceedings of the 24th International Conference on Machine Learning*, 1151–1157.
- Zhao, Zheng, Wang, L., & Liu, H. (2010). Efficient spectral feature selection with minimum redundancy. *Twenty-Fourth AAAI Conference on Artificial Intelligence*.
- Zhao, Zheng, Wang, L., Liu, H., & Ye, J. (2011). On similarity preserving feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 25(3), 619–632.
- Zheng, C.-H., Huang, D.-S., Kong, X.-Z., & Zhao, X.-M. (2008). Gene expression data classification using consensus independent component analysis. *Genomics, Proteomics & Bioinformatics*, 6(2), 74–82.
- Zhou, N., & Wang, L. (2007). A modified T-test feature selection method and its application on the HapMap genotype data. *Genomics, Proteomics & Bioinformatics*, 5(3–4), 242–249.
- Zhou, P., Hu, X., Li, P., & Wu, X. (2017). Online feature selection for high-dimensional class-imbalanced data. *Knowledge-Based Systems*, 136, 187–199.
- Zhu, Z., Ong, Y.-S., & Dash, M. (2007). Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognition*, 40(11), 3236–3248.
- Zibakhsh, A., & Abadeh, M. S. (2013). Gene selection for cancer tumor detection using a novel memetic algorithm with a multi-view fitness function. *Engineering Applications of Artificial Intelligence*, 26(4), 1274–1281.

APPENDIX

APPENDIX I

Statistic of Estimated Error Rate per Class for all the Datasets

CNS											
d $\geq$ 0.1				d $\geq$ 0.2				d $\geq$ 0.3			
Confusion matrix:				Confusion matrix:				Confusion matrix:			
0 1 class.error				0 1 class.error				0 1 class.error			
0	38	1	0.02564103	0	38	1	0.02564103	0	38	1	0.02564103
1	14	7	0.66666667	1	1	20	0.04761905	1	0	21	0.00000000

Ovarian											
d $\geq$ 0.4				d $\geq$ 0.6							
Confusion matrix:				Confusion matrix:							
Cancer Normal class.error				Cancer Normal class.error							
Cancer	162	0	0	Cancer	161	1	0.00617284				
Normal	0	91	0	Normal	1	90	0.01098901				

d $\geq$ 0.8											
Confusion matrix:											
Cancer Normal class.error											
				Cancer	161	1	0.00617284				
				Normal	0	91	0.00000000				

Leukemia_3c											
d $\geq$ 0. 4				d $\geq$ 0. 4							
Confusion matrix:				Confusion matrix:							
AML B-cell T-cell class.error				AML B-cell T-cell class.error							
AML	19	6	0	0.24000000	AML	17	8	0	0.3200000		
B-cell	2	36	0	0.05263158	B-cell	4	34	0	0.1052632		
T-cell	1	5	3	0.66666667	T-cell	0	9	0	1.0000000		

d ≥ 0.8				
Confusion matrix:				
AML B-cell T-cell class.error				
AML	18	7	0	0.2800000
B-cell	4	32	2	0.1578947
T-cell	1	6	2	0.7777778

Leukemia_4c												
d ≥ 0.4						d ≥ 0.6						
Confusion matrix:						Confusion matrix:						
B-cell BM PB T-cell class.error						B-cell BM PB T-cell class.error						
B-cell	37	1	0	0	0.02631579	B-cell	36	1	0	1		
	0.05263158											
BM	4	17	0	0	0.19047619	BM	7	14	0	0		
	0.33333333											
PB	0	4	0	0	1.00000000	PB	0	4	0	0		
	1.00000000											
T-cell	5	1	0	3	0.66666667	T-cell	7			1	0	1
	0.88888889											

d ≥ 0.8						
Confusion matrix:						
B-cell BM PB T-cell class.error						
B-cell	34	2	0	2	0.1052632	
BM	7	13	1	0	0.3809524	
PB	3	1	0	0	1.0000000	
T-cell	6	0	0	3	0.6666667	

Lung

$d \geq 0.4$								$d \geq 0.6$							
Confusion matrix:								Confusion matrix:							
1	2	3	4	5	class.error			1	2	3	4	5	class.error		
1	139	0	0	0	0	0.00000000		1	139	0	0	0	0	0.00000000	
2	1	16	0	0	0	0.05882353		2	1	16	0	0	0	0.05882353	
3	0	0	5	1	0	0.16666667		3	0	0	6	0	0	0.00000000	
4	0	0	0	21	0	0.00000000		4	0	0	0	21	0	0.00000000	
5	0	0	0	0	20	0.00000000		5	0	0	0	0	20	0.00000000	

$d \geq 0.8$

Confusion matrix:							
1	2	3	4	5	class.error		
1	139	0	0	0	0	0.00000000	
2	1	16	0	0	0	0.05882353	
3	0	1	5	0	0	0.16666667	
4	0	0	1	20	0	0.04761905	
5	0	0	0	1	19	0.05000000	

Lymphoma

$d \geq 0.4$					$d \geq 0.6$				
Confusion matrix:					Confusion matrix:				
CLL	DLBCL	FL	class.error		CLL	DLBCL	FL	class.error	
CLL	11	0	0	0.0000000	CLL	9	2	0	0.1818182
DLBCL	0	46	0	0.0000000	DLBCL	0	46	0	0.0000000
FL	0	4	5	0.4444444	FL	0	3	6	0.3333333

d ≥ 0.8				
Confusion matrix:				
	CLL	DLBCL	FL	class.error
CLL	11	0	0	0.0000000
DLBCL	0	46	0	0.0000000
FL	0	2	7	0.2222222

MLL

d ≥ 0.4					d ≥ 0.6				
Confusion matrix:					Confusion matrix:				
	ALL	AML	MLL	class.error		ALL	AML	MLL	class.error
ALL	20	0	4	0.16666667	ALL	22	0	2	0.08333333
AML	0	27	1	0.03571429	AML	0	27	1	0.03571429
MLL	4	2	14	0.30000000	MLL	0	2	18	0.10000000

d ≥ 0.8				
Confusion matrix:				
	ALL	AML	MLL	class.error
ALL	21	2	1	0.12500000
AML	2	26	0	0.07142857
MLL	3	1	16	0.20000000

SRBCT

d ≥ 0.4						d ≥ 0.6					
Confusion matrix:						Confusion matrix:					
	1	2	3	4	class.error		1	2	3	4	class.error
1	27	0	0	2	0.06896552	1	24	0	0	5	0.17241379
2	0	11	0	0	0.00000000	2	1	10	0	0	0.09090909
3	0	0	16	2	0.11111111	3	0	0	17	1	0.05555556
4	1	0	1	23	0.08000000	4	0	1	1	23	0.08000000

d ≥ 0.8					
Confusion matrix:					
	1	2	3	4	class.error
1	23	1	1	4	0.2068966
2	2	6	2	1	0.4545455
3	0	1	14	3	0.2222222
4	1	0	4	20	0.2000000

APPENDIX II

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

APPENDIX III

Detection Rate and Balanced Accuracy Details for All Datasets

Case 1: CNS Dataset

Algorithms	Detection Rate	Balanced Accuracy
C4.5	0.6111	1.0000
	0.6111	1.0000
	0.5556	1.0000
DECISION TREE	0.6111	1.0000
	0.6111	1.0000
	0.5556	1.0000
RANDOM FOREST	0.6111	1.0000
	0.6111	1.0000
	0.5556	1.0000
CART	0.6111	1.0000
	0.6111	1.0000
	0.5556	1.0000

Case 2: Leukemia_3c Dataset

Algorithms	Detection Rate	Balanced Accuracy
C4.5	0.3333	1
	0.254	0.7639
	0.3016	0.9511
DECISION TREE	0.2857	0.8944
	0.2699	0.8409
	0.2698	0.7094
RANDOM FOREST	0.3333	1
	0.3333	1
	0.3333	1
CART	0.254	0.6943
	0.3333	0.639
	0.3016	0.7881

Case 3: Leukemia_4c

Algorithms	Detection Rate	Balanced Accuracy
C4.5	0.25	1
	0.2381	0.9554
	0.25	1
DECISION TREE	0.1905	0.7419
	0.25	1
	0.1667	0.6315
RANDOM FOREST	0.25	1
	0.25	1
	0.25	1
CART	0.1905	0.7966

	0.1905	0.6339
	0.2024	0.7125

Case 4: Lung Dataset

Algorithms	Detection Rate	Balanced Accuracy
C4.5	0.2	1
	0.2	1
	0.2	1
DECISION TREE	0.19	0.8498
	0.2	1
	0.2	1
RANDOM FOREST	0.2	1
	0.2	1
	0.2	1
CART	0.2	0.9733
	0.2	1
	0.2	0.984

Case 5: Lymphoma Dataset

Algorithms	Detection Rate	Balanced Accuracy
C4.5	0.3333	1
	0.3334	1
	0.3333	1
DECISION TREE	0.3333	1
	0.3334	1
	0.3333	1

RANDOM FOREST	0.3333	1
	0.3334	1
	0.3333	1
CART	0.2807	0.8
	0.3334	1
	0.3333	1

Case 6: MLL Dataset

Algorithms	Detection Rate	Balanced Accuracy
C4.5	0.3334	1
	0.3333	1
	0.3333	1
DECISION TREE	0.2699	0.8775
	0.2857	0.8969
	0.3175	0.9583
RANDOM FOREST	0.3334	1
	0.3334	1
	0.3333	1
CART	0.3175	0.938
	0.3016	0.9239
	0.3333	0.9583

Case 7: Ovarian Dataset

Algorithms	Detection Rate	Balanced Accuracy
C4.5	0.6267	1.0000
	0.6533	1.0000
	0.6267	1.0000

DECISION TREE	0.6267	1.0000
	0.6533	1.0000
	0.6267	1.0000
RANDOM FOREST	0.6267	1.0000
	0.6533	1.0000
	0.6267	1.0000
CART	0.6267	1.0000
	0.6533	1.0000
	0.6267	1.0000

Case 8: SRBCT Dataset

Algorithms	Detection Rate	Balanced Accuracy
C4.5	0.25	1
	0.25	1
	0.25	1
DECISION TREE	0.1875	0.8696
	0.2292	0.9345
	0.2292	0.9451
RANDOM FOREST	0.25	1
	0.25	1
	0.25	1
CART	0.2188	0.9417
	0.2396	0.9622
	0.1875	0.801

APPENDIX IV

Sample of Raw and Reduced Datasets

The following are samples of original datasets used in this study showing their features and summary of the datasets.

Relation: CNS
Instances: 60
Attributes: 7131
Sum of weights: 60

Attributes

All None Invert Pattern

No.	Name
1	☐
2	☐ AFX-BioB-5_at
3	☐ AFX-BioB-M_at
4	☐ AFX-BioB-3_at
5	☐ AFX-BioC-5_at
6	☐ AFX-BioC-3_at
7	☐ AFX-BioDn-5_at
8	☐ AFX-BioDn-3_at
9	☐ AFX-CreX-5_at
10	☐ AFX-CreX-3_at
11	☐ AFX-BioB-5_st
12	☐ AFX-BioB-M_st
13	☐ AFX-BioB-3_st
14	☐ AFX-BioC-5_st
15	☐ AFX-BioC-3_st
16	☐ AFX-BioDn-5_st
17	☐ AFX-BioDn-3_st
18	☐ AFX-CreX-5_st
19	☐ AFX-CreX-3_st
20	☐ hum_alu_at
21	☐ AFX-DapX-5_at
22	☐ AFX-DapX-M_at
23	☐ AFX-DapX-3_at
24	☐ AFX-LysX-5_at
25	☐ AFX-LysX-M_at
26	☐ AFX-LysX-3_at
27	☐ AFX-PheX-5_at
28	☐ AFX-PheX-M_at
29	☐ AFX-PheX-3_at
30	☐ AFX-ThrX-5_at
31	☐ AFX-ThrX-M_at
32	☐ AFX-ThrX-3_at
33	☐ AFX-TrpnX-5_at
34	☐ AFX-TrpnX-M_at

Figure: Raw CNS Dataset

Relation: Leukemia_3c
Instances: 72
Attributes: 7131
Sum of weights: 72

Attributes

All None Invert Pattern

No.	Name
1	☐
2	☐ AFX-BioB-5_at
3	☐ AFX-BioB-M_at
4	☐ AFX-BioB-3_at
5	☐ AFX-BioC-5_at
6	☐ AFX-BioC-3_at
7	☐ AFX-BioDn-5_at
8	☐ AFX-BioDn-3_at
9	☐ AFX-CreX-5_at
10	☐ AFX-CreX-3_at
11	☐ AFX-BioB-5_st
12	☐ AFX-BioB-M_st
13	☐ AFX-BioB-3_st
14	☐ AFX-BioC-5_st
15	☐ AFX-BioC-3_st
16	☐ AFX-BioDn-5_st
17	☐ AFX-BioDn-3_st
18	☐ AFX-CreX-5_st
19	☐ AFX-CreX-3_st
20	☐ hum_alu_at
21	☐ AFX-DapX-5_at
22	☐ AFX-DapX-M_at
23	☐ AFX-DapX-3_at
24	☐ AFX-LysX-5_at
25	☐ AFX-LysX-M_at
26	☐ AFX-LysX-3_at
27	☐ AFX-PheX-5_at
28	☐ AFX-PheX-M_at
29	☐ AFX-PheX-3_at
30	☐ AFX-ThrX-5_at
31	☐ AFX-ThrX-M_at
32	☐ AFX-ThrX-3_at
33	☐ AFX-TrpnX-5_at
34	☐ AFX-TrpnX-M_at

Figure: Raw Leukemia_3c Dataset

Relation: Leukemia_4c
Instances: 72

Attributes: 7131
Sum of weights: 72

Attributes

AllNoneInvertPattern

No.	Name
1	☒
2	☐ AFX-BioB-5_at
3	☐ AFX-BioB-M_at
4	☐ AFX-BioB-3_at
5	☐ AFX-BioC-5_at
6	☐ AFX-BioC-3_at
7	☐ AFX-BioDn-5_at
8	☐ AFX-BioDn-3_at
9	☐ AFX-CreX-5_at
10	☐ AFX-CreX-3_at
11	☐ AFX-BioB-5_st
12	☐ AFX-BioB-M_st
13	☐ AFX-BioB-3_st
14	☐ AFX-BioC-5_st
15	☐ AFX-BioC-3_st
16	☐ AFX-BioDn-5_st
17	☐ AFX-BioDn-3_st
18	☐ AFX-CreX-5_st
19	☐ AFX-CreX-3_st
20	☐ hum_alu_at
21	☐ AFX-DapX-5_at
22	☐ AFX-DapX-M_at
23	☐ AFX-DapX-3_at
24	☐ AFX-LysX-5_at
25	☐ AFX-LysX-M_at
26	☐ AFX-LysX-3_at
27	☐ AFX-PheX-5_at
28	☐ AFX-PheX-M_at
29	☐ AFX-PheX-3_at
30	☐ AFX-ThrX-5_at
31	☐ AFX-ThrX-M_at
32	☐ AFX-ThrX-3_at
33	☐ AFX-TrpnX-5_at
34	☐ AFX-TrpnX-M_at

Figure: Raw Leukemia_4c

Relation: lung_cancer_5c_12600g
Instances: 203

Attributes: 12601
Sum of weights: 203

Attributes

AllNoneInvertPattern

No.	Name
1	☒ AFX-MurL2_at
2	☐ AFX-MurL10_at
3	☐ AFX-MurL4_at
4	☐ AFX-MurFAS_at
5	☐ AFX-BioB-5_at
6	☐ AFX-BioB-M_at
7	☐ AFX-BioB-3_at
8	☐ AFX-BioC-5_at
9	☐ AFX-BioC-3_at
10	☐ AFX-BioDn-5_at
11	☐ AFX-BioDn-3_at
12	☐ AFX-CreX-5_at
13	☐ AFX-CreX-3_at
14	☐ AFX-BioB-5_st
15	☐ AFX-BioB-M_st
16	☐ AFX-BioB-3_st
17	☐ AFX-BioC-5_st
18	☐ AFX-BioC-3_st
19	☐ AFX-BioDn-5_st
20	☐ AFX-BioDn-3_st
21	☐ AFX-CreX-5_st
22	☐ AFX-CreX-3_st
23	☐ AFX-hum_alu_at
24	☐ AFX-DapX-5_at
25	☐ AFX-DapX-M_at
26	☐ AFX-DapX-3_at
27	☐ AFX-LysX-5_at
28	☐ AFX-LysX-M_at
29	☐ AFX-LysX-3_at
30	☐ AFX-PheX-5_at
31	☐ AFX-PheX-M_at
32	☐ AFX-PheX-3_at
33	☐ AFX-ThrX-5_at
34	☐ AFX-ThrX-M_at

Figure: Raw Lung Dataset

Relation: Lymphoma66x4026_3c
Instances: 66

Attributes: 4027
Sum of weights: 66

Attributes

AllNoneInvertPattern

No.	Name
1	☒ GENE1835X
2	☐ GENE1836X
3	☐ GENE1865X
4	☐ GENE1380X
5	☐ GENE1933X
6	☐ GENE1932X
7	☐ GENE1931X
8	☐ GENE1930X
9	☐ GENE3129X
10	☐ GENE3126X
11	☐ GENE0X
12	☐ GENE3115X
13	☐ GENE3116X
14	☐ GENE3117X
15	☐ GENE3118X
16	☐ GENE3073X
17	☐ GENE3072X
18	☐ GENE3067X
19	☐ GENE3068X
20	☐ GENE3069X
21	☐ GENE2584X
22	☐ GENE3070X
23	☐ GENE1843X
24	☐ GENE3166X
25	☐ GENE3165X
26	☐ GENE1885X
27	☐ GENE3385X
28	☐ GENE3384X
29	☐ GENE1110X
30	☐ GENE1109X
31	☐ GENE1108X
32	☐ GENE1107X
33	☐ GENE4005X
34	☐ GENE4006X

Figure: Raw Lymphoma Dataset

Relation: MLL
Instances: 72

Attributes: 12583
Sum of weights: 72

Attributes

AllNoneInvertPattern

No.	Name
1	☒ AFFX-MurIL2_at
2	☐ AFFX-MurIL10_at
3	☐ AFFX-MurIL4_at
4	☐ AFFX-MurFAS_at
5	☐ AFFX-hum_alu_at
6	☐ AFFX-DapX-5_at
7	☐ AFFX-DapX-M_at
8	☐ AFFX-DapX-3_at
9	☐ AFFX-LysX-5_at
10	☐ AFFX-LysX-M_at
11	☐ AFFX-LysX-3_at
12	☐ AFFX-PheX-5_at
13	☐ AFFX-PheX-M_at
14	☐ AFFX-PheX-3_at
15	☐ AFFX-ThrX-5_at
16	☐ AFFX-ThrX-M_at
17	☐ AFFX-ThrX-3_at
18	☐ AFFX-TrpX-5_at
19	☐ AFFX-TrpX-M_at
20	☐ AFFX-TrpX-3_at
21	☐ AFFX-HUMISGF3A/M97935_5_at
22	☐ AFFX-HUMISGF3A/M97935_MA_at
23	☐ AFFX-HUMISGF3A/M97935_MB_at
24	☐ AFFX-HUMISGF3A/M97935_3_at
25	☐ AFFX-HUMRGE/M10098_5_at
26	☐ AFFX-HUMRGE/M10098_M_at
27	☐ AFFX-HUMRGE/M10098_3_at
28	☐ AFFX-HUMGAPDH/M33197_5_at
29	☐ AFFX-HUMGAPDH/M33197_M_at
30	☐ AFFX-HUMGAPDH/M33197_3_at
31	☐ AFFX-HSAC07/X00351_5_at
32	☐ AFFX-HSAC07/X00351_M_at
33	☐ AFFX-HSAC07/X00351_3_at
34	☐ AFFX-HUMTFRR/M11507_5_at

Figure: Raw MLL Dataset

Relation: Ovarian
Instances: 253

Attributes: 15155
Sum of weights: 253

Attributes

All None Invert Pattern

No.	Name
1	☒ MZ-7.86E-05
2	☐ MZ2.18E-07
3	☐ MZ9.60E-05
4	☐ MZ0.000366014
5	☐ MZ0.000810195
6	☐ MZ0.001428564
7	☐ MZ0.002221123
8	☐ MZ0.003187869
9	☐ MZ0.004328805
10	☐ MZ0.005643929
11	☐ MZ0.007133241
12	☐ MZ0.008796743
13	☐ MZ0.010634432
14	☐ MZ0.012646311
15	☐ MZ0.014832378
16	☐ MZ0.017192634
17	☐ MZ0.019727078
18	☐ MZ0.022435711
19	☐ MZ0.025318532
20	☐ MZ0.028375542
21	☐ MZ0.031606741
22	☐ MZ0.035012128
23	☐ MZ0.038591704
24	☐ MZ0.042345469
25	☐ MZ0.046273422
26	☐ MZ0.050375564
27	☐ MZ0.054651894
28	☐ MZ0.059102413
29	☐ MZ0.063727121
30	☐ MZ0.068526017
31	☐ MZ0.073499102
32	☐ MZ0.078646375
33	☐ MZ0.083967837
34	☐ MZ0.089463488

Figure: Raw Ovarian Dataset

Relation: SRBCT
Instances: 83

Attributes: 2309
Sum of weights: 83

Attributes

All None Invert Pattern

No.	Name
1	☒ gene1
2	☐ gene2
3	☐ gene3
4	☐ gene4
5	☐ gene5
6	☐ gene6
7	☐ gene7
8	☐ gene8
9	☐ gene9
10	☐ gene10
11	☐ gene11
12	☐ gene12
13	☐ gene13
14	☐ gene14
15	☐ gene15
16	☐ gene16
17	☐ gene17
18	☐ gene18
19	☐ gene19
20	☐ gene20
21	☐ gene21
22	☐ gene22
23	☐ gene23
24	☐ gene24
25	☐ gene25
26	☐ gene26
27	☐ gene27
28	☐ gene28
29	☐ gene29
30	☐ gene30
31	☐ gene31
32	☐ gene32
33	☐ gene33
34	☐ gene34

Figure: Raw SRBCT Dataset

Sample of Reduced Dataset After reduced with InfoGain-MBB Algorithm

		CLASS	U25128_a	M55621_a	HG1071.H X	D28416_a	HG180.HT	AB002409	U18548_a	X51466_at	U34962_a	U52111_rr	U60325_a	M23668_a	U21556_at	
2	1	1	9	145	122	1	831	-37	126	222	10045	213	-713	130	41	-38
3	2	1	152	-115	187	2	987	53	-108	-529	4822	219	-739	186	181	-38
4	3	1	79	248	110	3	1235	15	-168	443	9859	423	-1249	483	-201	-28
5	4	1	85	680	117	4	942	-183	-291	343	10981	413	-1573	218	-123	-345
5	5	1	273	-88	101	5	21452	-6	-44	1131	5204	-19	-2147	-76	400	31
7	6	1	17	143	55	6	1149	-124	-117	218	9029	211	-817	134	100	75
5	7	1	15	59	22	7	1174	4	-186	162	7143	188	-923	241	-55	-55
3	8	1	60	338	412	8	1771	-3	-281	942	7450	415	-1838	-90	-43	-875
0	9	1	49	-310	40	9	1743	-90	-64	697	10724	150	-762	242	-141	-116
1	10	1	-5	23	94	10	508	2	24	176	12327	185	-522	161	-47	-71
2	11	1	26	30	152	11	1330	10	-41	391	10240	276	-1319	293	-108	-150
3	12	1	74	687	96	12	2452	-7	32	444	11176	216	-336	442	-39	-62
4	13	1	-69	207	173	13	831	-61	-138	458	21262	245	-2160	63	-162	157
5	14	1	94	-21	-127	14	5108	-59	-22	1088	12603	240	-408	280	80	60
6	15	1	87	-217	90	15	1348	-50	6	180	13189	291	-564	313	-45	-36
7	16	1	10	47	90	16	763	15	78	-3	11383	140	-416	272	-36	23
8	17	1	18	-418	129	17	1733	-104	-81	851	11668	254	-1179	150	-115	-165
9	18	1	66	181	300	18	1701	-34	213	869	42068	415	-707	92	-14	-101
0	19	1	117	-506	282	19	1677	-74	-92	344	17799	368	-940	384	5	-144
1	20	1	148	349	483	20	2016	-137	162	1075	6885	334	-1406	130	-232	497
2	21	1	32	293	251	21	1882	-11	-41	1802	7770	32	-843	233	-79	-11
3	22	0	27	624	-5	22	708	-16	52	301	12100	83	-325	299	26	9
4	23	0	133	102	215	23	1678	-199	-160	947	11724	202	-981	252	-106	-38
5	24	0	106	348	110	24	1210	-37	-290	599	13692	181	-1669	104	25	-156
6	25	0	9	-135	173	25	2030	76	100	1280	22605	482	-748	969	102	72
7	26	0	-25	657	381	26	3397	-336	-203	575	13880	270	-2161	123	31	-266
8	27	0	117	370	163	27	946	-58	209	1054	19891	373	-923	261	-85	-176
9	28	0	-20	516	242	28	1379	68	5	-22	15258	148	-1070	265	5	10
0	29	0	230	279	274	29	2274	-131	-142	1485	7925	339	-1468	164	-54	27
1	30	0	106	348	110	30	1210	-37	-290	599	13692	181	-1669	104	25	-156
Binhat Reduced InfoGainCNS02																

Figure: Reduced CNS Dataset

	CLASS	U93553_a	J04130_s	L34060_at	M16404_a	U43328_a	S81944_at	U04313_a	U09413_a	U37139_a	M86406_a	U28727_a	X	U42390_a	U47634_a	X74764_at	U11037_a	M16038_at
1	B-cell	21	709	514	55	41	-120	2	38	45	-166	-719	1	-325	468	1170	564	164
2	B-cell	-153	486	209	-30	-32	-109	131	-43	367	-180	-1086	2	43	562	197	232	797
3	B-cell	-15	97	162	-7	-21	-72	-31	-14	322	-175	-269	3	-67	424	505	14	491
4	B-cell	99	420	566	-45	26	-37	-92	15	228	-301	-584	4	-82	716	172	67	265
5	B-cell	-84	278	185	9	-26	-122	103	19	360	-218	-430	5	-163	295	733	115	223
6	B-cell	-121	176	137	-151	-44	-51	-38	-23	85	-100	-444	6	25	388	544	-83	252
7	B-cell	-80	146	375	-245	-31	-18	-71	-25	108	142	-508	7	-138	212	258	13	87
8	B-cell	15	119	225	70	31	-77	24	77	109	-114	-189	8	46	414	363	-40	173
9	B-cell	92	359	193	-63	-159	25	109	0	159	-33	-357	9	-132	468	407	208	313
10	B-cell	38	647	248	-66	-30	-48	-187	120	484	23	-803	10	-152	529	767	121	178
11	B-cell	-5	955	1133	-89	21	-196	-179	-52	531	327	-1061	11	-263	877	1269	331	390
12	BM	-59	1323	294	36	-5	-69	-110	-3	206	-346	-767	12	-256	990	315	115	2735
13	BM	58	504	273	-102	-8	-58	-132	-82	203	10	-679	13	-211	442	923	225	2089
14	PB	96	616	571	-52	-41	-108	-44	-62	188	-80	-407	14	-294	630	229	46	1744
15	BM	25	816	737	-87	-11	-70	-141	29	310	154	-564	15	-160	860	1255	553	1522
16	BM	132	397	163	82	49	20	49	60	74	144	-444	16	-230	929	919	45	855
17	B-cell	-24	509	352	-77	47	-79	-62	-79	237	99	-441	17	-207	309	746	37	217
18	B-cell	61	762	229	-177	-20	-146	-110	25	167	-54	-502	18	-254	651	1143	136	185
19	BM	-98	304	66	-64	39	-29	-49	-37	220	51	-414	19	-261	418	595	64	444
20	BM	64	887	287	-191	-43	-253	-71	-86	170	117	-958	20	-77	676	1335	34	1020
21	B-cell	162	348	177	-27	7	-40	25	40	213	188	-305	21	-94	381	686	0	147
22	BM	107	283	540	-84	95	-60	-191	-90	209	139	-375	22	-84	716	935	-41	385
23	BM	73	632	849	-41	4	-62	-197	-49	208	67	-544	23	-186	422	951	236	489
24	PB	-16	644	375	-16	16	-34	-173	-18	280	275	-861	24	62	604	590	169	1837
25	PB	63	1436	876	-36	-35	-161	-175	9	316	227	-790	25	-331	585	572	386	2686
26	PB	279	902	406	-103	46	-260	-51	-59	601	113	-1016	26	-211	1553	1032	661	1499
27	BM	236	801	252	48	46	-157	-132	1	199	-308	-475	27	-343	523	526	-62	716
28	BM	-52	221	165	-32	47	-104	-23	36	143	82	-332	28	-232	56	399	17	460
29	T-cell	119	314	222	-62	5	-119	-98	-1	111	143	-302	29	-99	277	491	-40	1448

Figure: Reduced Leukemia_3c Dataset

	CLASS	35269_at
1	1	30.78
2	1	-20.61
3	1	26.01
4	1	-2.685
5	1	24.35
6	1	-3.7
7	1	29.1
8	1	5.63
9	1	10.79
10	1	-4.65
11	1	9.4
12	1	20.82
13	1	4.46
14	1	35.105
15	1	26.64
16	1	-4.52
17	1	-18.63
18	1	33.46
19	1	8.165
20	1	-11.555
21	1	10.89
22	1	29.423
23	1	22.79
24	1	84.85
25	1	0.7
26	1	14.305
27	1	54.985
28	1	-6.64
29	1	37.17

Figure: Reduced Lung Dataset

CLASS	GENE2419	GENE2302	GENE1525	GENE1740	GENE472X	GENE3459	GENE1618	GENE1582	GENE1732	GENE384X	GENE1752	GENE1710	GENE3588	GENE3964	GENE1785	GENE2479	GENE1747	GENE3971	GENE1576	GENE1628	GENE1589G
1 DLBCL	-1	-0.57	-0.22	0.08	0.53	-1.23	-2.32	-0.47	0.92	1.74	0	-0.49	1.66	-1.29	-0.04	-0.28	-0.44	-0.47	0.11	-0.23	-1.53
2 DLBCL	0.49	-0.44	-0.13	0.62	0.5	0.09	0.89	0.52	-0.09	1.77	-0.52	-1.13	0.9	0	0.15	0.08	0.09	-0.79	0.14	-0.38	-1.76
3 DLBCL	-1.33	0.87	-0.14	-0.42	0.87	-0.34	0.08	0.74	0.61	2	-0.34	-0.59	1.15	0.29	-0.61	-0.16	-0.38	0.04	-0.71	-0.13	0.69
4 DLBCL	0.23	0.13	-0.45	0.32	0.34	0.28	-0.74	-0.03	0.5	0.95	0.25	0.05	0.98	0.63	0.2	0.05	-0.35	-0.09	-0.12	0.16	-0.91
5 DLBCL	0.72	0	0.08	0.88	0.03	0.06	1.01	0.47	0.06	1.98	-0.87	0.09	0.48	0.85	-0.36	0.04	-0.43	-0.09	0.26	0.12	0.29
6 DLBCL	1.7	-0.54	0.11	0.258276	1.16	-0.57	0.22	0.76	-0.4	0.12	0.13	0.26	0.9	0.23	-0.62	-0.41	0.45	0.61	-0.99	0.94	0.27
7 DLBCL	0.64	-0.24	-0.16	0.38	0.4	0	0.12	0.39	0.86	1.91	-0.17	0.5	0.46	0.1	0.49	-0.2	0.85	0.12	0.31	1.13	0.27
8 DLBCL	-0.6	0.05	-0.16	0.258276	0.93	-0.01	0.86	0.71	0.09	1.17	0.41	0.33	0.81	0	0.89	-0.31	0.17	-0.47	0.24	1.06	0.56
9 DLBCL	-0.03	-0.6	0.08	0.44	0.18	0.19	0.81	-0.52	-0.37	0.1	-0.42	0.05	0	-0.52	0.02	0.13	-0.04	-0.1	-0.62	0.75	0.29
10 DLBCL	0.95	-0.65	0.49	-1.65	-0.07	-0.25	-2.76	-1.04	-1.04	2.44	-0.16	-1.4	0.4	-4.72	-0.94	0.71	-0.28	-0.65	-0.69	-0.31	-1.72
11 DLBCL	-1.62	-0.81	0.03	-0.87	-0.13	0.32	-0.28	-0.39	-0.01	-1.23	0.44	-0.36	0.49	-1.36	1.07	-0.92	0.08	0.05	0.36	0.67	-0.18
12 DLBCL	-2.3	-1.07	0.01	-0.96	-0.49	0	-0.81	-0.2	-0.66	-0.21	0.17	-0.01	0.17	-2.52	0.3	1.11	0.55	-0.21	0	0.29	-0.8
13 DLBCL	-1.14	-0.73	-0.4	0.03	-0.46	0.56	0.25	-0.68	2.63	-0.73	-0.36	0.05	-0.14	-1.49	-0.44	0.12	-0.16	-0.34	0.05	0.26	0.35
14 DLBCL	-1.39	-0.28	-0.02	-0.27	1.69	-0.14	0	-0.87	0.33	-1.26	-0.89	-0.05	-0.18	-1.71	-0.13	0.85	-0.22	-0.64	-0.01	-0.29	-0.23
15 DLBCL	-0.6	0.12	0.19	1.93	-0.04	-0.14	1.81	0.5	0.98	1.27	0.43	1.03	-0.65	-0.5	-0.4	0.11	-0.07	0.19	0.58	1.23	1.63
16 DLBCL	0.47	-0.47	-0.03	0.81	-0.22	-0.33	1.58	-0.09	0.21	0.58	-0.46	0.22	0.28	-1.1	0	-0.17	-0.05	-0.28	0.17	0.38	0.8
17 DLBCL	-1.45	-0.26	0	0.71	0.82	-0.26	2.13	0.22	0.45	0.81	0.33	0.44	-0.35	0.88	0.19	0	-0.15	0.04	0.9	1.48	1.85
18 DLBCL	0.39	0.28	-0.16	0.09	0.91	0.08	1.15	0.87	0.23	0.83	0.55	0.34	0.3	1.65	1.53	-0.23	0.51	0.73	0.74	0.22	0.32
19 DLBCL	1.52	0.85	0.39	1.63	1.08	0.1	2.37	0.15	1.32	1.54	-0.06	0.45	0.28	1.23	1.11	-0.48	0.72	-0.43	0.16	0.18	1.22
20 DLBCL	0.124063	0.49	0	0.22	0.38	0.28	1.53	0.25	0.66	0	0.68	0.43	-0.52	0.13	0.55	-0.54	0	0.38	0.15	1.02	1.54
21 DLBCL	-1.17	0.11	0.2	-0.01	0.78	0.16	1.47	-0.12	0.21	-0.1	0.1	0.37	0.21	0.29	1.28	-0.75	1.25	0.51	0.13	1.03	0.47
22 DLBCL	-0.42	-0.2	0.53	0.82	-0.48	-0.54	1.35	0.32	1.11	0.08	0.19	0.15	0.06	1.3	-0.15	-0.92	1.01	-0.13	0.44	0.2	0.48
23 DLBCL	-0.51	-0.9	0.05	-0.42	0.79	0.47	-0.09	0.09	0.52	-0.05	0.28	-0.22	0.59	0.91	0.88	-0.56	0.47	-0.18	0.41	1.07	-0.75
24 DLBCL	-0.76	-0.75	0.82	0.14	-0.08	0.38	1.46	0.75	0.55	0.07	0.03	-0.06	-0.18	0.73	0.92	0.63	1.56	0.1	0.19	1.52	0.37
25 DLBCL	-0.09	-0.94	0.11	-0.43	0	0.24	3.55	-0.21	0.56	-0.14	0.99	0.07	-0.3	-0.52	0.78	-0.69	1.27	0.58	-0.47	1.2	1.46
26 DLBCL	-0.27	0.12	0.3	0.54	0.15	0.61	2.12	0.4	1.26	0.73	0.96	0.14	-0.21	0.3	0.38	-0.39	1.17	-0.05	0.28	0.18	0.97
27 DLBCL	-0.62	-0.46	0.07	1.47	-0.17	0.014746	1.42	0.31	0.99	0.34	-0.08	0.2	-0.47	0.53	-0.35	0.38	0.69	0.84	0.26	0.75	1.58
28 DLBCL	-0.19	0.1	-0.02683	1.31	-0.08	0.6	2.14	1.11	0.78	1.48	0.96	0.43	-0.46	0.72	-0.37	-0.02	0.71	0.84	0.38	1.36	0.64
29 DLBCL	0.05	-0.83	-0.27	1.21	0.13	0.44	1.71	0.8	0.48	0.61	0.41	-0.25	0.02	-0.37	-0.16	-0.61	0.66	0.02	0.01	0.71	0.68

Figure: Reduced Lymphoma Dataset

CLASS	X40929_at	X37624_at	X34265_at	X39113_at	X33490_at	X41827_f	X31677_f	X34775_at	X33682_at	X34653_at	X36225_s	X32768_at	X31607_at	X41830_at	X292_s	X37096_at	X36817_at	X1767_s	X36038_r	X38062_at	X33330_at	X4
1 ALL	-5	131	42	513	48	5763	100	348	238	57	1048	385	134	1660	7868	3915	-43	518	1731	763	-755	
2 ALL	13	120	109	1142	482	21257	-44	-71	276	185	1533	499	-74	3127	7766	-48	124	539	1064	1612	-807	
3 ALL	129	-1	149	613	46	7538	34	-403	387	122	601	-90	-152	2763	2808	-218	-34	494	1155	395	-1439	
4 ALL	38	3	108	51	145	1115	-6	-18	137	182	1223	78	-86	104	9821	70	24	401	1203	379	-807	
5 MLL	-161	-6	139	1193	220	7211	-127	418	261	-193	1823	311	-303	5276	10834	314	-151	1257	956	824	-697	
6 MLL	171	39	128	1278	136	16426	-60	105	387	665	1324	252	-70	4612	3100	-201	49	515	1343	1318	140	
7 MLL	-56	67	236	2369	196	860	66	585	341	251	2176	874	41	5656	13001	233	56	531	1653	1461	-492	
8 AML	-18	-15	73	727	2	657	-67	387	160	-72	421	63	-168	2137	622	87	31	256	1339	1087	-337	
9 AML	-137	116	91	582	183	13142	54	-861	369	1276	1371	-178	-226	1845	2980	3459	9	619	3731	941	-611	
10 AML	110	53	152	502	25	2445	55	-300	421	39	3931	-153	-166	2316	7352	28158	0	654	2254	1365	-825	
11 AML	184	147	190	551	68	932	50	-428	450	256	1879	-90	167	2122	4737	2438	-25	618	3086	1555	-2222	
12 AML	43	147	3	531	32	5428	-185	-503	383	240	994	286	-204	2547	1419	5400	27	228	746	763	-810	
13 AML	100	65	232	875	18	170	-27	-1262	517	836	1301	-319	-265	2701	2711	6582	50	674	549	1169	-430	
14 AML	-73	83	130	471	-6	10924	-214	618	535	24	1281	510	-33	2603	706	46665	-21	402	1462	1306	-1507	
15 AML	-324	-205	155	539	-43	36	-160	-861	334	150	1616	2	-304	1240	4804	7746	-57	393	1028	1637	-254	
16 ALL	213.7	-2.8	75.5	1322.4	5.5	4082.2	-61.5	144.1	296.5	-29.5	1667	284.7	-140.5	1796.8	10157.6	-43.4	68.1	337.4	635.3	292.2	-888.4	
17 ALL	-9	43	126	1139	13	7198	-27	309	273	169	1456	574	-28	4876	5994	-234	25	493	1409	1008	-766	
18 ALL	10	63	74	277	-22	2059	65	64	190	-343	2217	-33	189	1236	9820	-82	6	285	573	304	-1016	
19 ALL	-460	-14	66	1356	-4	1950	-96	-226	215	-98	985	586	-82	2310	11578	-175	-216	240	716	418	-456	
20 ALL	100	109	143	568	30	13669	20	-556	217	272	1108	62	-166	3178	4328	547	28	467	1187	249	-1074	
21 ALL	18	-21	113	938	-4	11907	57	51	306	70	1240	230	36	3591	3360	-156	7	567	1617	870	-1149	
22 ALL	84	0	145	769	80	12167	-10	257	282	100	1398	264	-67	4096	4041	54	35	297	740	986	-853	
23 ALL	112	113	117	1043	111	15440	213	-756	263	486	2113	302	49	4691	2402	229	84	568	2537	1248	-1331	
24 ALL	-3	183	88	907	102	18289	11	-408	188	331	2033	315	-32	2524	3636	-153	131	402	1133	1836	-350	
25 ALL	57	52	22	1342	219	9648	60	388	220	262	1927	247	-44	4126	3272	-62	14	756	1499	338	-632	
26 ALL	121	13	64	1387	20	11159	43	-857	463	309	1696	503	55	3947	10326	1428	-41	638	1249	1488	-830	
27 ALL	-44	17	72	1005	-24	2992	66	18	131	122	1863	470	-18	2286	7430	-102	-10	484	1370	1197	-453	
28 ALL	-43	-59	93	1402	227	16502	104	-569	186	-206	948	100	170	3354	4699	3624	49	503	2089	916	-1406	
29 ALL	-87	52	195	2660	53	21723	-42	527	344	97	2949	493	-61	5131	8284	-436	18	930	1488	2867	-184	

Figure: Reduced MLL Dataset

	CLASS	MZ4676.9918		
1	Normal	0.311355		
2	Normal	0.327725		
3	Normal	0.183448		
4	Normal	0.584513		
5	Normal	0.309536		
6	Normal	0.195271		
7	Normal	0.372808		
8	Normal	0.464207		
9	Normal	0.117253		
10	Normal	0.338963		
11	Normal	0.218397		
12	Normal	0.148824		
13	Normal	0.70482		
14	Normal	0.701962		
15	Normal	0.118683		
16	Normal	0.565285		
17	Normal	0.128037		
18	Normal	0.145641		
19	Normal	0.18897		
20	Normal	0.152722		
21	Normal	0.549045		
22	Normal	0.086202		
23	Normal	0.126413		
24	Normal	0.215279		
25	Normal	0.132064		
26	Normal	0.377615		
27	Normal	0.166104		
28	Normal	0.143562		
29	Normal	0.078992		
Binbat_Reduced_infoGainOvarian0				

Figure: Reduced Ovarian Dataset

	CLASS	gene1622	gene1765	gene2024	gene789	gene684	gene671	gene434	X
1	1	1.3191	3.4752	0.4049	0.7673	0.6502	0.6073	0.324	
2	1	1.5636	2.4122	0.2806	0.8718	0.3573	1.6609	0.6039	
3	1	0.8813	1.1355	0.237	0.5169	0.3636	4.1881	1.7175	
4	1	0.682	4.2045	0.488	0.3821	0.322	0.8708	2.6961	
5	1	1.5536	2.3602	0.3298	0.5574	1.3174	2.0349	3.2339	
6	1	1.109	2.2329	0.1962	0.4946	0.8214	2.634	0.6742	
7	1	0.8827	2.3618	0.2363	0.7457	0.5009	4.013	0.4604	
8	1	0.8352	2.5956	0.3091	0.6061	0.5736	1.2634	0.6048	
9	1	0.9049	2.6495	0.2414	0.7396	0.412	0.4739	0.2033	
10	1	0.6565	1.1585	0.2292	0.5819	0.7748	1.5092	3.7838	
11	1	0.9136	3.3155	0.1541	0.4859	0.3968	0.6391	0.3658	
12	1	0.7671	3.0333	0.1542	0.7099	0.9369	1.81	0.496	
13	1	1.5149	2.1284	0.2158	0.9004	0.5608	0.7339	1.0956	
14	1	0.9233	2.7098	0.2834	1.6987	0.8206	1.5707	0.6115	
15	1	0.8382	1.8453	0.633	0.5201	0.7621	0.115	0.2973	
16	1	0.5921	1.9551	0.4838	0.8768	0.6464	0.1453	0.3712	
17	1	0.9753	2.987	0.434	0.6658	0.8851	1.3551	0.4601	
18	1	1.2007	2.2593	0.1984	1.6135	1.088	0.622	0.7875	
19	1	1.0616	2.2056	0.2014	1.5721	0.946	0.5645	0.7137	
20	1	1.4478	6.0306	0.255	0.7116	0.6142	0.3483	0.6388	
21	1	0.5473	2.3806	0.3236	0.7038	0.5298	0.4488	0.3721	
22	1	0.6967	2.6718	0.2137	0.9447	0.554	0.4989	0.476	
23	1	0.7181	1.7148	0.171	0.5549	0.387	0.6471	0.2764	
24	2	1.4993	2.251	0.1441	0.7466	0.7688	0.299	0.4858	
25	2	1.0012	5.153	0.1855	0.411	0.7264	0.1665	0.3897	
26	2	0.9841	5.3647	0.1218	0.3294	0.6986	0.2189	0.2413	
27	2	1.0358	4.1163	0.1662	0.484	0.5607	0.1602	0.1644	
28	2	1.4792	5.1774	0.3221	0.2007	0.2531	0.1163	0.2443	
29	2	1.1129	1.4971	0.208	0.5197	0.3358	0.3577	0.2248	
Binbat_Reduced_infoGainSRBCT0.8 (+)									

Figure: Reduced SRBCT Dataset

APPENDIX V

Sample of Tree Diagrams

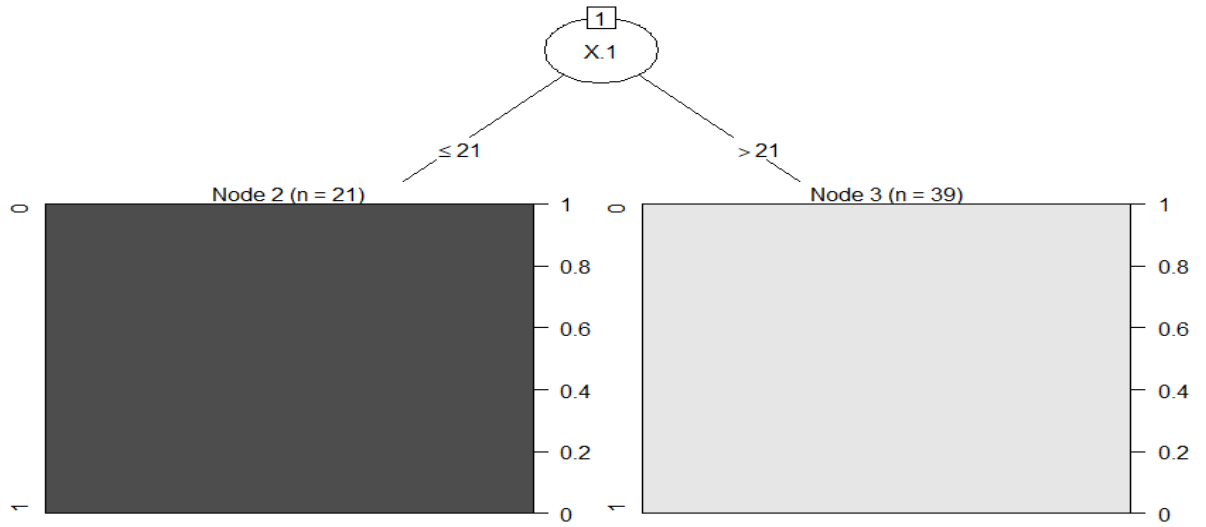


Figure: Tree Diagram of C4.5 for 1st Run of CNS Dataset

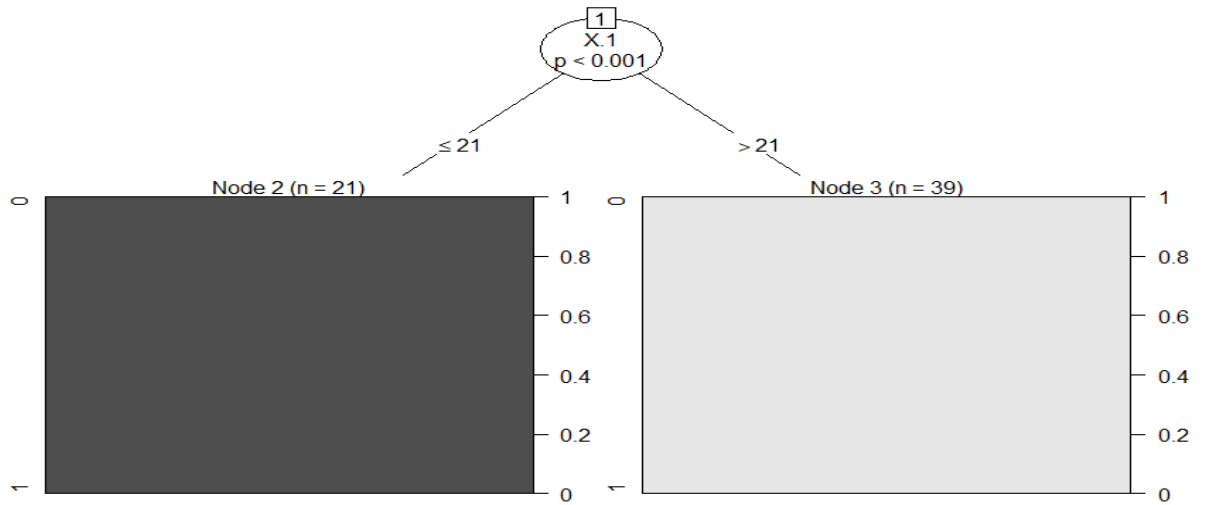


Figure: Tree Diagram of Decision Tree for 1st Run of CNS Dataset

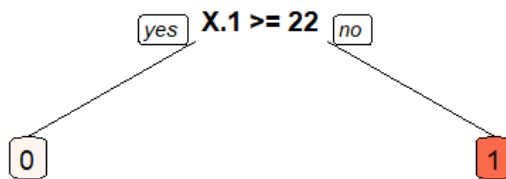


Figure: Tree Diagram of CART for 1st Run of CNS Dataset

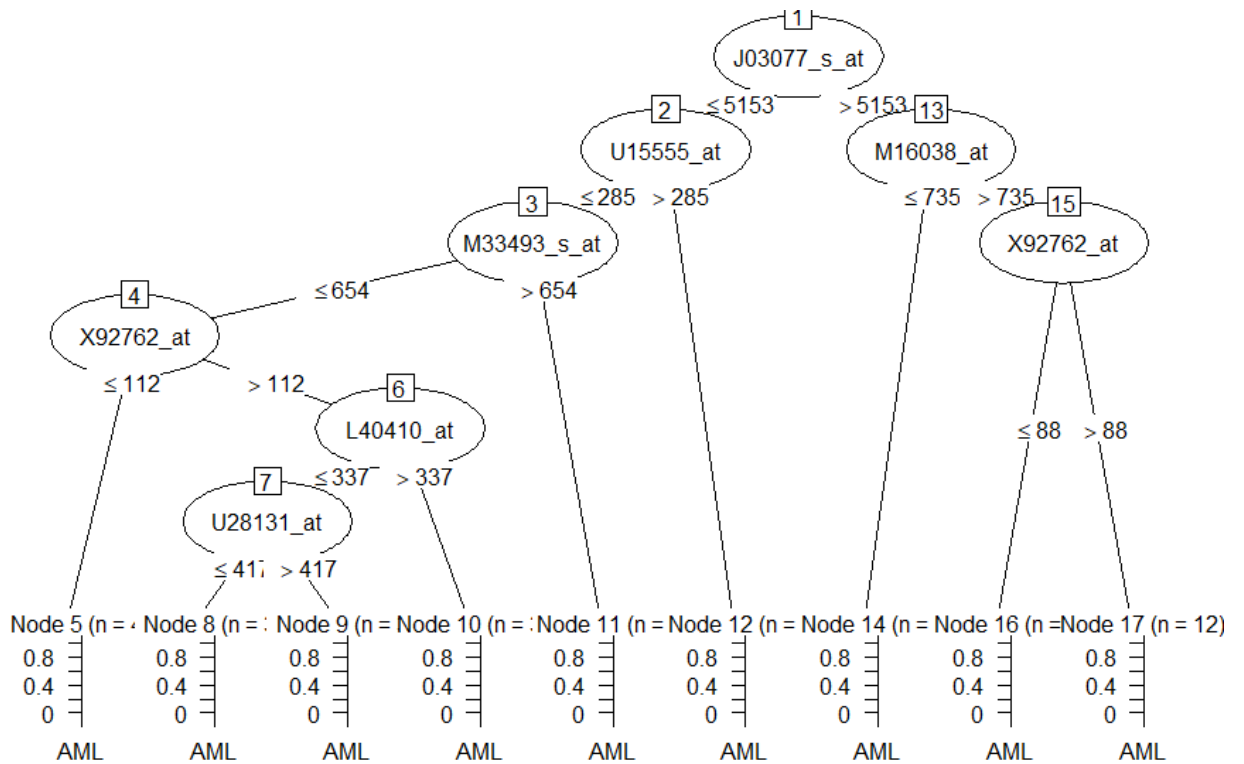


Figure: Tree Diagram of C4.5 for 1st Run of Leukemia_3c Dataset

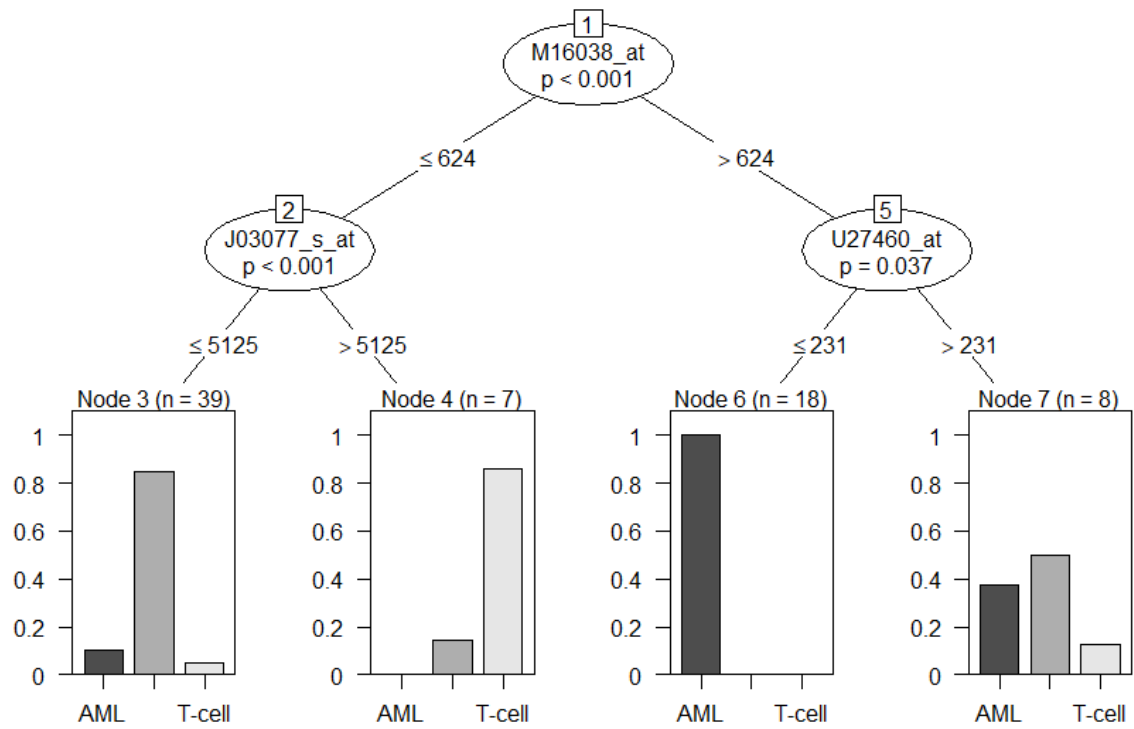


Figure: Tree Diagram of Decision Tree for 1st Run of Leukemia_3c Dataset

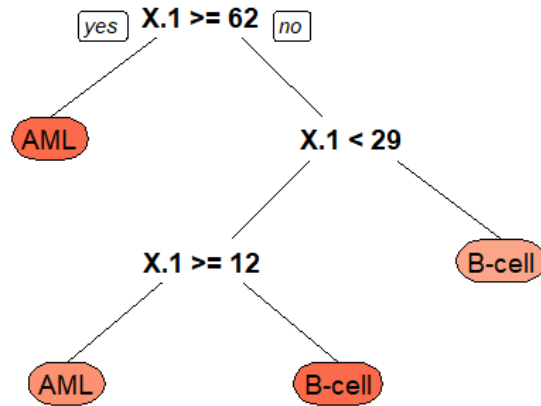


Figure: Tree Diagram of CART for 2nd Run of Leukemia_3c Dataset

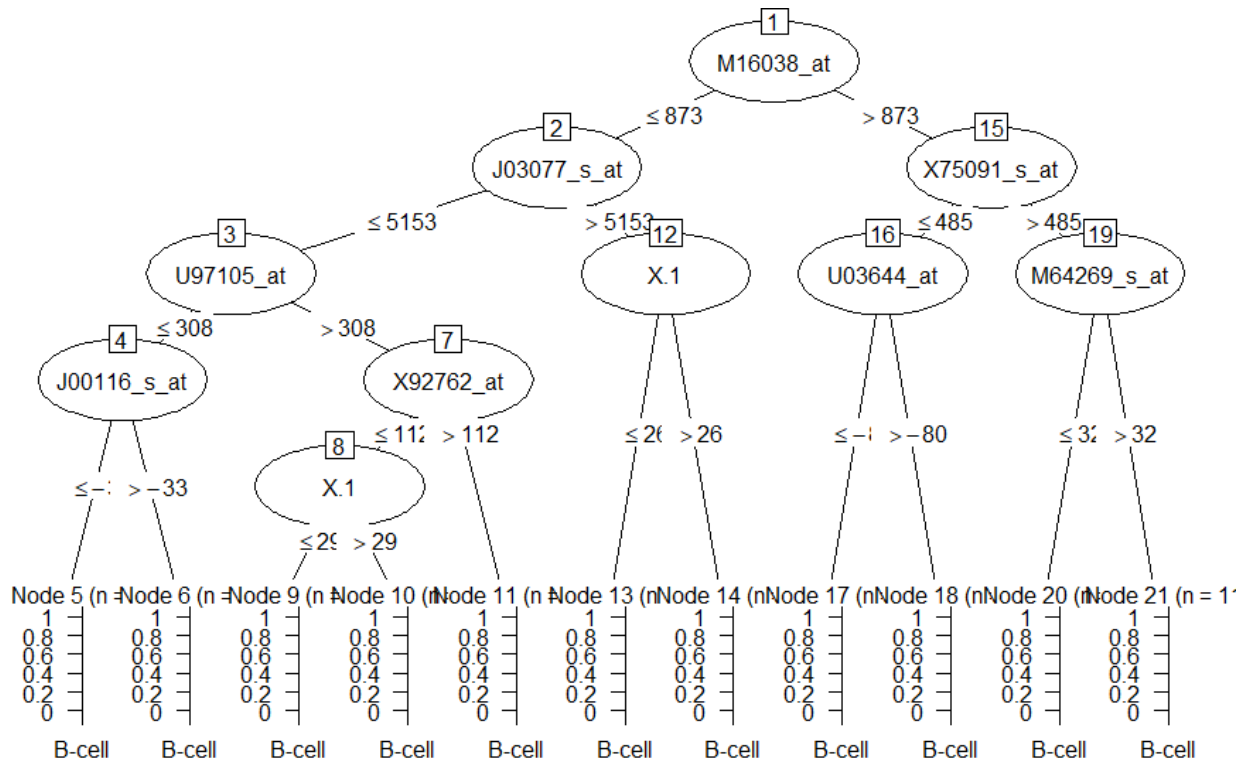


Figure: Tree Diagram of C4.5 for 1st Run of Leukemia_4c Dataset

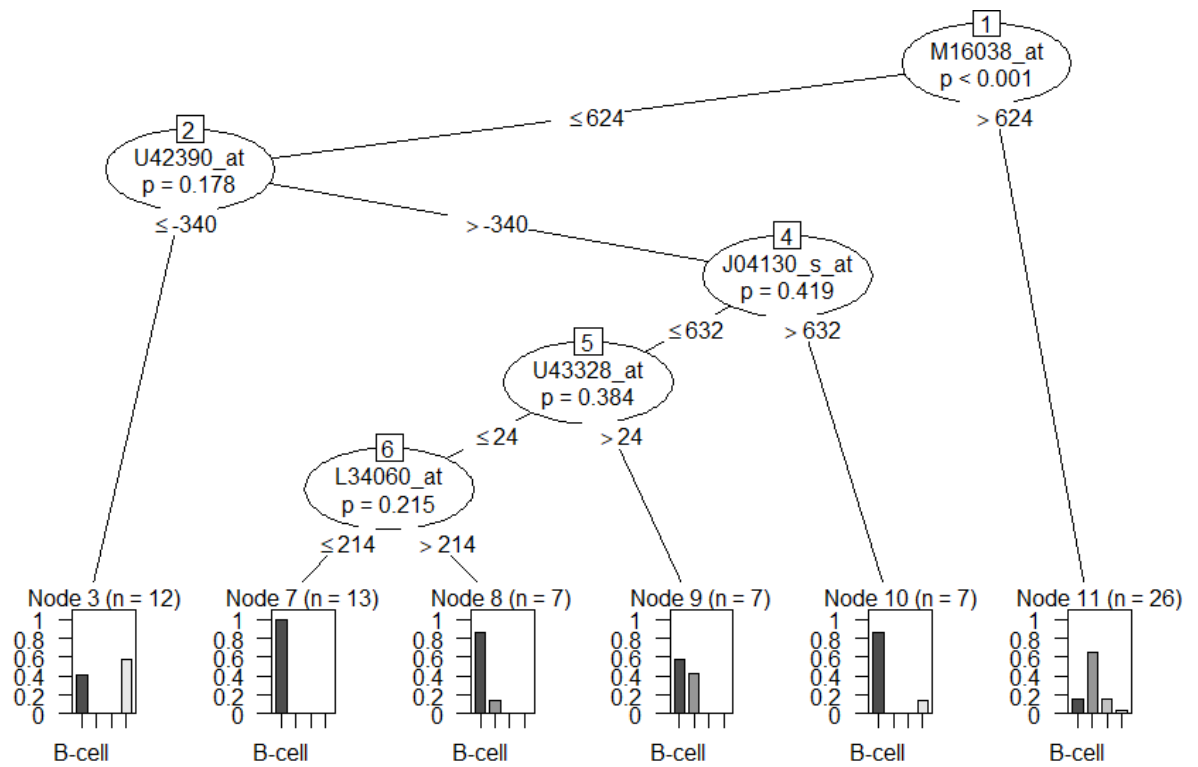


Figure: Tree Diagram of Decision Tree for 2nd Run of Leukemia_4c Dataset

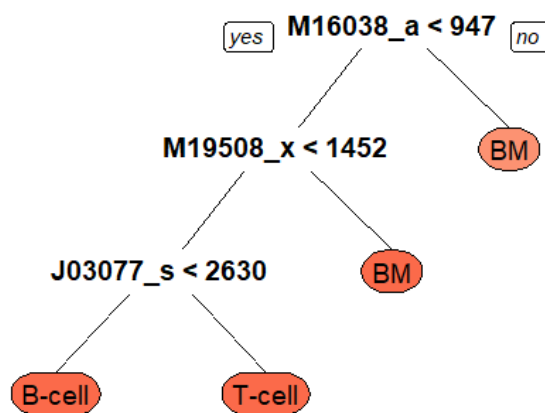


Figure: Tree Diagram of CART for 1st Run of Leukemia_4c Dataset

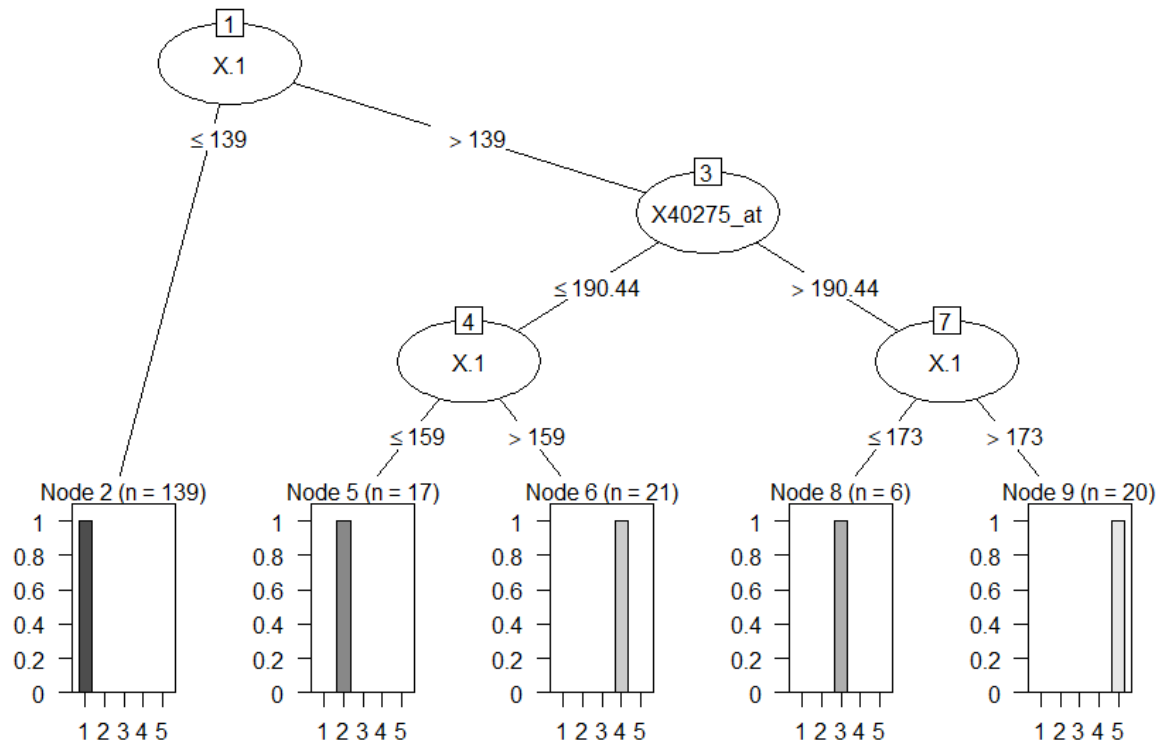


Figure: Tree Diagram of C4.5 for 1st Run of Lung Dataset

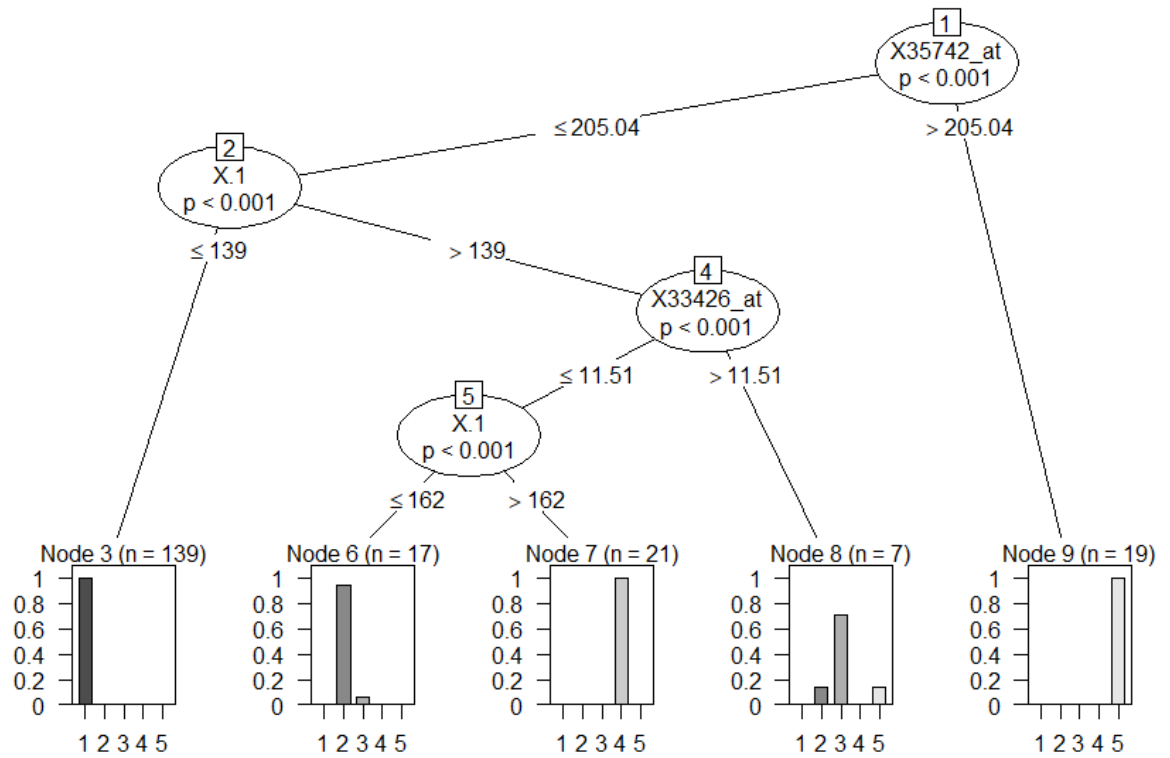


Figure: Tree Diagram of Decision Tree for 1st Run of Lung Dataset

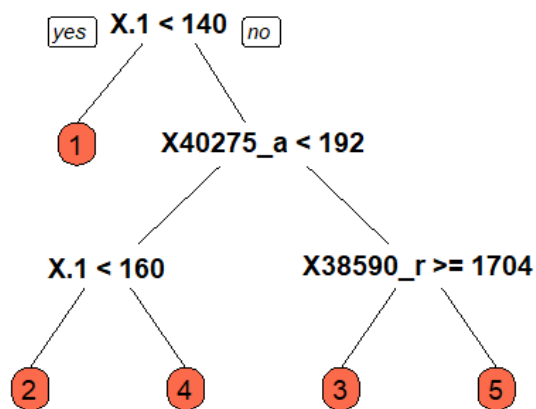


Figure: Tree Diagram of CART for 1st Run of Lung Dataset

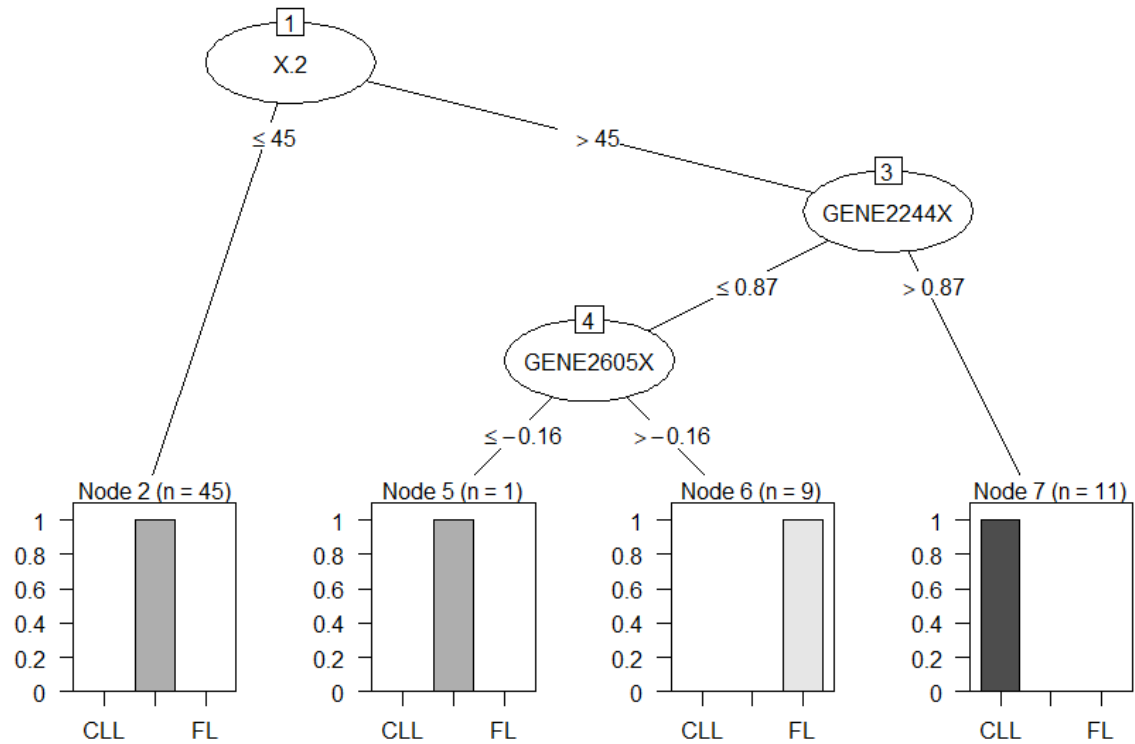


Figure: Tree Diagram of C4.5 for 2nd Run of Lymphoma Dataset

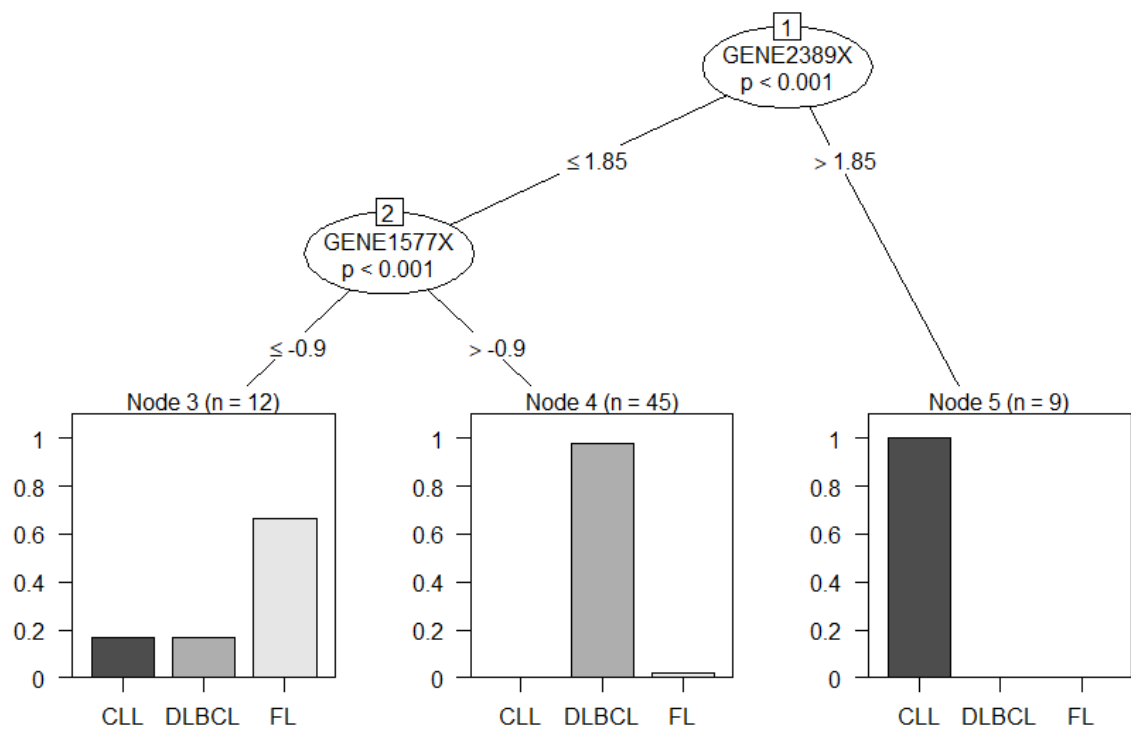


Figure: Tree Diagram of Decision Tree for 2nd Run of Lymphoma Dataset

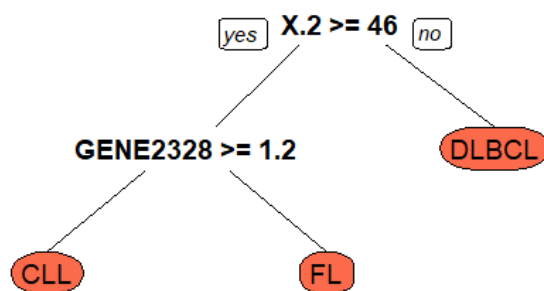


Figure: Tree Diagram of CART for 2nd Run of Lymphoma Dataset

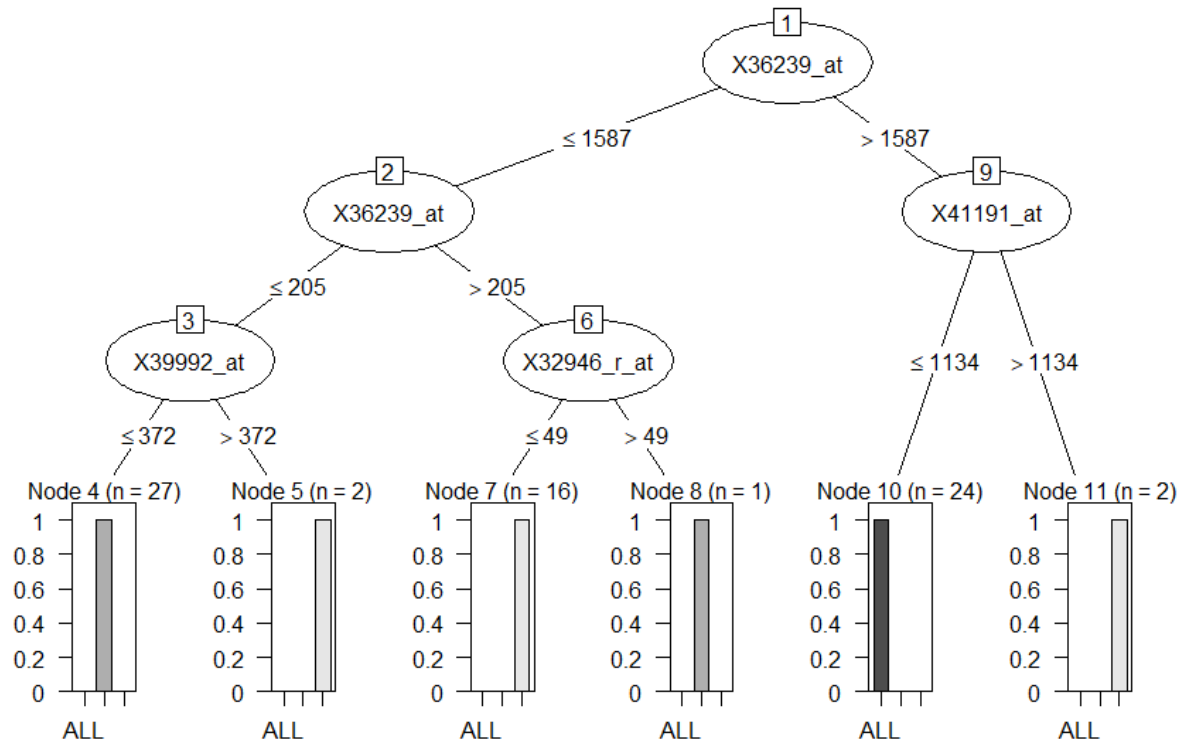


Figure: Tree Diagram of C4.5 for 1st Run of MLL Dataset

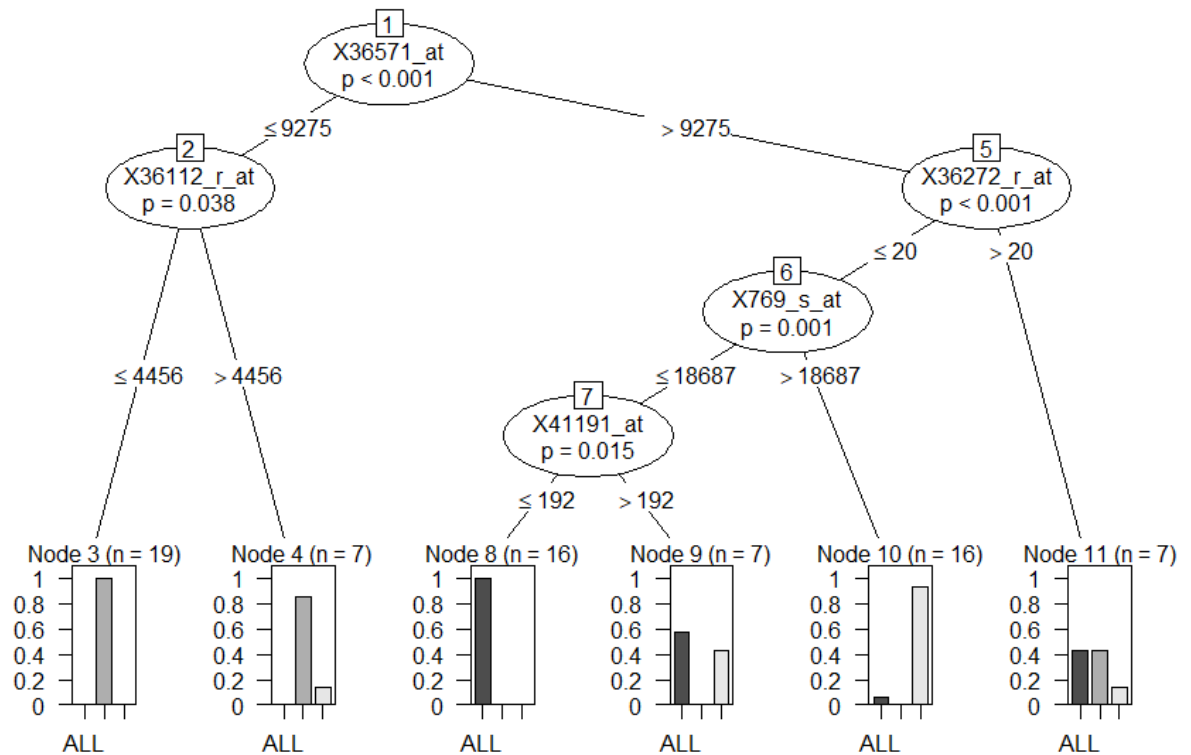


Figure: Tree Diagram of Decision Tree for 1st Run of MLL Dataset

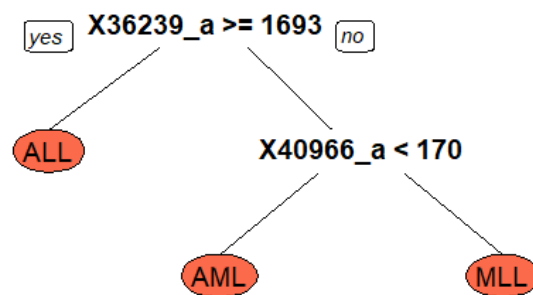


Figure: Tree Diagram of CART for 1st Run of MLL Dataset

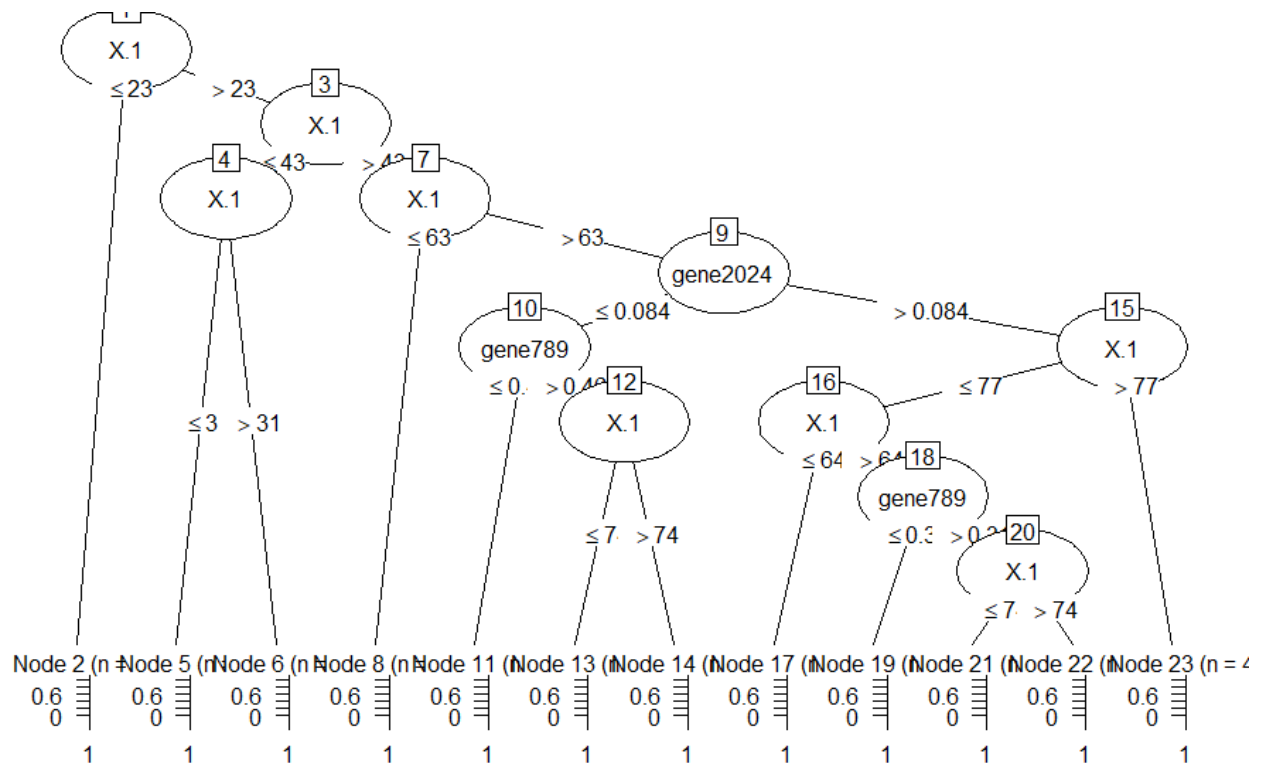


Figure: Tree Diagram of C4.5 for 3rd Run of SRBCT Dataset

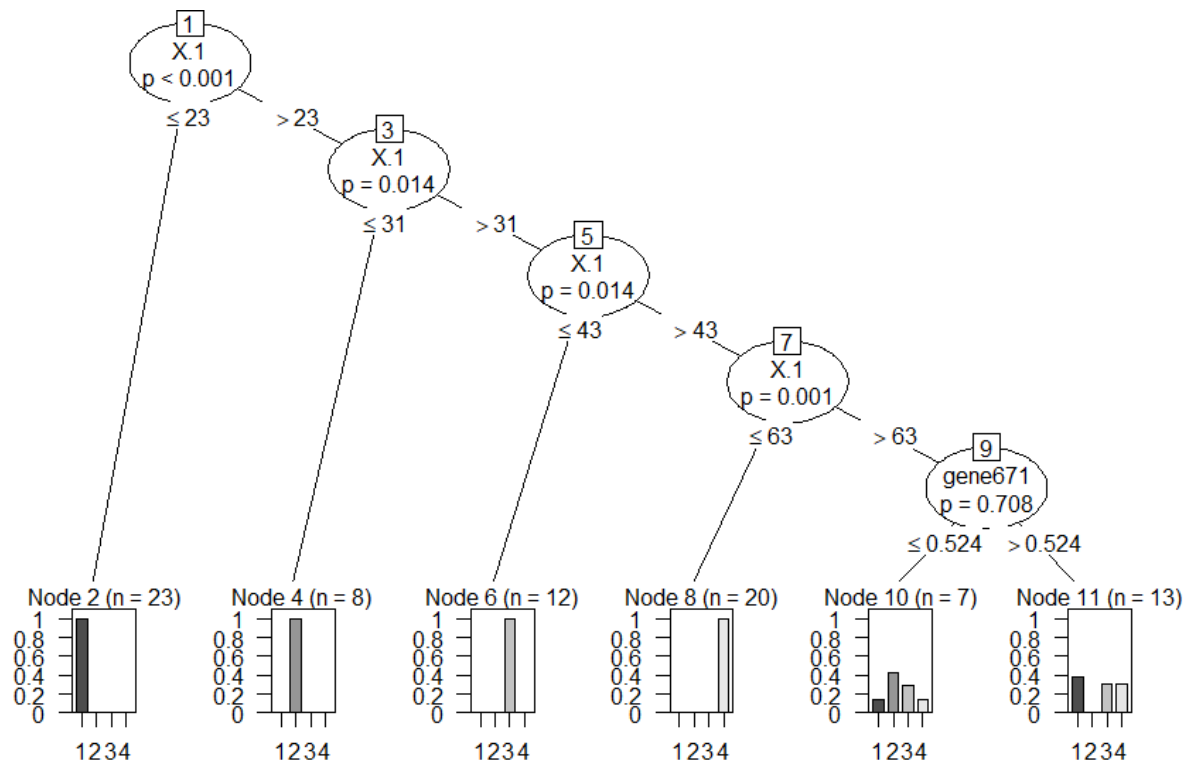


Figure: Tree Diagram of Decision Tree for 3rd Run of SRBCT Dataset

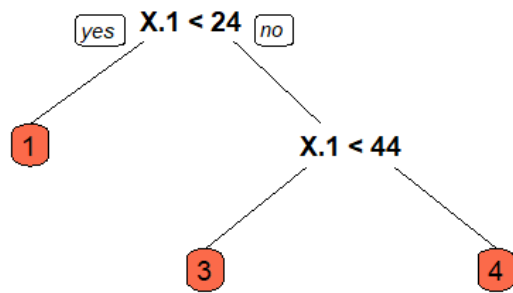


Figure: Tree Diagram of CART for 3rd Run of SRBCT Dataset

APPENDIX VI

Source Codes

A. Preprocess and missing value treatment:

```
#####  
## Treating Missing values  
library(Hmisc)  
d<-read.csv(file.choose(),header = T)  
  
cname<-colnames(d)  
View(cname)  
for(kk in cname){  
  im <- impute(d[,kk], mean)  
  d[,kk] =im  
}  
View(d)  
write.csv(d,"Lymphoma.csv")
```

B. Filter Algorithm (Ranking Approach)

```
rm(list = ls())  
library(ggplot2) # Data visualization  
library(readr) # CSV file I/O, e.g. the read_csv function  
library(ROCR) # Prediction Function  
library(e1071)  
library(pROC)  
library(plyr)  
library(FSelector)  
library(RWeka)  
library(caret)  
library(Biocomp)  
  
set.seed(12345)  
# Input data files are available in the "../input/" directory.  
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in  
# the input directory  
getwd()  
setwd("C:/PhD Project")  
list.files("C:/PhD Project")  
  
# Any results you write to the current directory are saved as output.  
ca.data2<-read.arff("CNS.arff")  
dim(ca.data2)
```

```

ca.subset<-subset(ca.data2,select=c(1:7130))
dim(ca.subset)
sapply(ca.subset,function(x) sum(is.na(x)))
sapply(ca.subset,function(x) length(unique(x)))
ca.subset$CLASS<-as.factor(ca.subset$CLASS)
is.factor(ca.subset$CLASS)

#ca.subset$CLASS<-revalue(ca.subset$CLASS,c("0"="B","1"="M"))
#Sample data into 80% train and 20% test dataset
indexes = sample(1:nrow(ca.subset), size=0.2*nrow(ca.subset))

# Split data
ca.test = ca.subset[indexes,]
dim(ca.test)
ca.train = ca.subset[-indexes,]
dim(ca.train)

#Information Gain Feature Selection
InfoGain_time<-system.time(InfoGain <- information.gain(CLASS~., ca.subset, unit =
"log2"))
print(InfoGain_time)
View(sort(InfoGain$attr_importance, decreasing = TRUE))
write.csv(sort(InfoGain$attr_importance, decreasing = TRUE), "infoGainLung.csv")

#Gain Ratio Feature Selection
GainR_time<-system.time(GainR <- gain.ratio(CLASS~., ca.subset))
print(GainR_time)
View(sort(GainR$attr_importance, decreasing = TRUE))
write.csv(sort(GainR$attr_importance,decreasing = TRUE),"gainR_featuresLung.csv")

# Chi-Squared Feature selection
chi_time<-system.time(weights <- chi.squared(CLASS~., ca.subset))
print(chi_time)
View(sort(weights$attr_importance, decreasing = TRUE))
write.csv(sort(weights$attr_importance, decreasing = TRUE), "chi_squaredLung.csv")

C. MBB ALGORITHM Subset Selection
rm(list = ls())
setwd("C:/PhD Project/BBA")
source("Wukari.R")
data<-read.csv(file.choose(),header = TRUE)

# Data preparation
y=as.factor(data$CLASS)
X<- data[-2]
Data<-cbind(y,X)

```

```

Data<- as.data.frame(Data)

#####
intervalStart <- Sys.time()

fitness <- function( ){
  ###Configure parallel processing
  #cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
  #   registerDoParallel(cluster)
  library(RRF)
  ###Develop training model
  fit<-RRF(y ~ ., data=Data, ntree=1000,keep.forest=FALSE, importance=TRUE)

  return(fit)
}
#####De-register parallel processing cluster
  #stopCluster(cluster)}

#####

nOfBat<- ncol(X)

binbat(nOfBat, 0.5, 0.5, 1, 1, 10, 50, 4, fitness)

selected_features<-imp

write.csv(selected_features,"binbat.csv")
RD<-read.csv(file.choose(),header = T)
selected_features<-RD[RD$value==1,'feature']
View(selected_features)
table(RD$value)
data_bin<-(data[,selected_features])
View(data_bin)
CLASS<-y
data_bin<-cbind(CLASS,data_bin)
View(data_bin)
write.csv(data_bin,"Reduced_binbat.csv")

#####

```

D. Fitness Function Computation

```

rm(list=ls())
# Load libraries
library(parallel)

```

```

library(doParallel)
library(caret)
library(randomForest)
library(RRF)
library(mlbench)
library(parallel)
library(doParallel)
library(foreach)
library(sigmoid)

set.seed(1958)
binbat <- function(nOfBat, gamma, alpha, zigma, beta, frequencyMin, frequencyMax,
BatSize, fitness)
{
  V=matrix(0,nOfBat,BatSize)#velocity
  Loudness_A=runif(nOfBat,1,2)#Loudness(A)
  PulseEmission_r=runif(nOfBat,0,1)# Pulse emission(r)
  fitnessValue=rep(1,nOfBat)*-Inf
  globalBestPosition=rep(0,BatSize)
  globalfitnessValue=-Inf
  itr=1

## Optimization

  foreach(i = 1:nOfBat,.combine=cbind)%dopar%{
##### fitness Calculation
    fit=fitness()
    imp=importance(fit,type=1)
    if (runif(1)< Loudness_A[i] && fitnessValue[i]<imp[,1])
      fitnessValue[i]=imp[i]
      Loudness_A[i]=alpha* Loudness_A[i]
      PulseEmission_r[i]=PulseEmission_r[i]*(1-exp(-gamma*itr))
  }
  bat<-imp

##### Get global fitness
  maxfit=max(imp[,1])
  maxind=min(imp[,1])+runif(1)
  if (globalfitnessValue<maxfit)
    globalfitnessValue=maxfit

#####**
# Position update phase1
  for(i in 1:nOfBat){
    bimp<- mean(imp)

```

```

    bat[i]<- bat[i]+sigma*Loudness_A[i]*(maxind)
    equalcheck=1/(1+(exp(-bat[i])))*bimp
    imp[imp<equalcheck]=0
    imp[imp>equalcheck]=1
  }
# Position update phase2
for(i in 1:nOfBat){
  beta=runif(1)
  rand<- runif(1)
  impbar=mean(imp)
  if (rand<Loudness_A[i] && fitnessValue[i]<globalfitnessValue)
    for(j in 1:length(bat[i])){
      fd=frequencyMin+(frequencyMax-frequencyMin)*beta
      V[i,j]=V[i,j]+(impbar-imp[i])*fd
      imp[i,]=imp[i]+V[i,j]
      bat[i,j]=bat[i,j]+(sigma*Loudness_A[i]*(maxind))
      randomNumber=imp[,1]
      br<-impbar
      equalcheck=1/(1+(exp(-bat[i,j])))*bimp
      imp[imp<equalcheck]=0
      imp[imp>equalcheck]=1
    }
  }
}

return(imp)

}

```

E. Classifiers

```

rm(list = ls())

# Load the party package. It will automatically load other dependent packages.
library(party)
library(partykit)
library(randomForest)
library(ROCR) # Prediction Function
library(rpart.plot)
library(RColorBrewer)
library(rattle)
library(ggplot2) # Data visualization
library(readr) # CSV file I/O, e.g. the read_csv function
library(e1071)
library(pROC)
library(plyr)
library(FSelector)
library(RWeka)

```

```

library(caret)
library(Biocomb)
library(mlbench)
library(Hmisc)

#Input
setwd("C:/PhD Project")
# Create the input data frame.
input.dat<-read.csv(file.choose(),header = TRUE)
#input.dat <- read.csv("Reduced_chi_squaredSRBCT0.8.csv")

# Any results you write to the current directory are saved as output.
dim(input.dat)
#ca.subset<-ca.data2
#dim(ca.subset)
#input.dat=na.omit(input.dat) # <- use with care...
#dim(input.dat)
sapply(input.dat,function(x) sum(is.na(x)))
sapply(input.dat,function(x) length(unique(x)))
input.dat$CLASS<-as.factor(input.dat$CLASS)
is.factor(input.dat$CLASS)

#Sample data into 70% train and 30% test dataset
indexes = sample(1:nrow(input.dat), size=0.3*nrow(input.dat))

# Split data
test = input.dat[indexes,]
dim(test)
train = input.dat[-indexes,]
dim(train)

#' Create fixed sampling scheme (10-folds) so we can compare the models later on.
set.seed(1958) # set a seed to get replicable results
train <- createFolds(input.dat$CLASS, k=10 )

#' ## C 4.5 Decision Tree
C45Fit <- train(CLASS ~., method = "J48", data = input.dat,
               tuneLength = 5,
               trControl = trainControl(
                 method="cv", indexOut=train))
C45Fit
C45Fit$finalModel
plot(C45Fit$finalModel)
c45_pred <- predict(object=C45Fit, newdata=test, type="raw")

```

```

table(c45_pred, test$CLASS)
c45_performance<-confusionMatrix(c45_pred,test$CLASS)
print(c45_performance)

#' ### Conditional Inference Tree (Decision Tree)
ctreeFit <- train(CLASS ~ ., method = "ctree", data = input.dat,
                 tuneLength = 5,
                 trControl = trainControl(
                   method = "cv", indexOut = train))
ctreeFit
plot(ctreeFit$finalModel)
ctree_pred <- predict(object=ctreeFit, newdata=test, type="raw")
table(ctree_pred, test$CLASS)
ctree_performance<-confusionMatrix(ctree_pred,test$CLASS)
print(ctree_performance)

#' ### Random Forest
randomForestFit <- train(CLASS ~ ., method = "rf", data = input.dat,
                       tuneLength = 5,
                       trControl = trainControl(
                         method = "cv", indexOut = train))
randomForestFit
randomForestFit$finalModel
plot(randomForestFit$finalModel)
#fancyRpartPlot(randomForestFit$finalModel)

randomForest_pred <- predict(object=randomForestFit, newdata=test, type="raw")
table(randomForest_pred, test$CLASS)
randomForest_performance<-confusionMatrix(randomForest_pred,test$CLASS)
print(randomForest_performance)

#' ### Classification and Regression Tree (CART)
trctrl <- trainControl(method = "cv", number = 10)
CRtreeFit <- train(CLASS ~ ., data = input.dat, method = "rpart",
                 parms = list(split = "information"),
                 trControl=trctrl,
                 tuneLength = 10)
CRtreeFit
prp(CRtreeFit$finalModel, box.palette = "Reds", tweak = 1.2)
predict(CRtreeFit, newdata = test)
test_pred <- predict(CRtreeFit, newdata = test)
confusionMatrix(test_pred, test$CLASS) #check accurac

```