UNIVERSITY OF NAIROBI

AN EXCITED CUCKOO SEARCH-GREY WOLF ADAPTIVE KERNEL SVM FOR EFFECTIVE PATTERN RECOGNITION IN DNA MICROARRAY CANCER CHIPS

BY

SEGERA RENE DAVIES

M.Sc, B.Sc (Eng) (UON)

F80/52397/2017

A Thesis Submitted for the Examination in the Fulfillment of the requirements for Award of the Degree of Doctor of Philosophy in Electrical and Electronic Engineering of the University of Nairobi

2021

DECLARATION

I declare that this thesis is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people's work, my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi's requirements.

SIGNATURE……... . DATE…04/09/2021**……………**

DAVIES RENE SEGERA

F80/52397/2017

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITY OF NAIROBI

This thesis is submitted for examination with our approval as research supervisors:

ENG. PROF. MWANGI MBUTHIA:

SIGNATURE DATE: 6th September, 2021

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITY OF NAIROBI
P.O. BOX 30197-00100
Nairobi Kenya
jmbuthia@uonbi.ac.ke

DR. ABRAHAM NYETE:

SIGNATURE……… …………...DATE………6th September 2021………

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITY OF NAIROBI
P.O. BOX 30197-00100
Nairobi Kenya
anyete@uonbi.ac.ke

DEDICATION

I would like to dedicate this project to the Almighty God for making it possible in bringing me this far in this project.

I also dedicate this project to my family and loved ones in tolerating my excuses throughout my entire lecture period.

ACKNOWLEDGEMENT

First of all, I am greatly indebted to my First Supervisor, Prof. Mwangi Mbuthia for his inspiring and timeless advice since I began my doctoral studies. Prof. Mbuthia has been instrumental in making my research possible in first place and has always been unfailingly supportive to me throughout my research so far. His critical, insightful and constructive comments have been invaluable to my research as much as to me as a person. I am forever grateful to Him.

Secondly, I would like to express my heartfelt gratitude to my Second Supervisor Dr. Abraham Nyete for giving me academic advice wherever I needed him and helping me develop this document.

Finally, I would like to express my special thanks to my parents, siblings and my family for their incredible love, encouragement and support since I began this research. May God bless you all.

ABSTRACT

The scarcity of patient samples, curse-of-dimensionality and class imbalance of the available DNA microarray chips remain big hindrances for researchers to accurately and reliably classify cancerous tissues without overfitting. Moreover, these challenges are magnified when resource (computational power and memory) constrained devices like smart phones, tablets, and personal digital assistants are used to mine these datasets, rendering effective *portable microarray data mining* a very difficult task to achieve. Thus, gene selection and classification have turned out to be the most researched topics in DNA microarray based cancer diagnosis. An effective gene selection phase derives an informative gene subset from otherwise a highly dimensional dataset to reduce noise, computational overheads and model overfitting. On the other hand, an enhanced learning and classification phase builds a model that accurately and reliably classify a given DNA patient sample. This research has formulated a novel memetic approach: Excited-(E)-Adaptive Cuckoo Search-(ACS)-Intensification Dedicated Grey Wolf (IDGWO), i.e. EACSIDGWO for optimal gene selection. EACSIDGWO is an algorithm where the step size of ACS and the nonlinear control strategy of parameter $a^{\rightarrow}$of the IDGWO are innovatively made adaptive via the concept of the complete voltage and current responses of a direct current (DC) excited resistor-capacitor (RC) circuit. Since the population has a higher diversity at early stages of the proposed EACSIDGWO algorithm, both the ACS and IDGWO are jointly involved in local exploitation. Furthermore, to enhance mature convergence at later stages of the proposed algorithm, the role of ACS is switched to global exploration while the IDGWO is still left conducting the local exploitation. The performance of EACSIDGWO as a gene selector is evaluated on six standard DNA microarray chips derived from Irvine (UCI) repository namely Ovarian Cancer(4000 genes), Central Nervous System Cancer (7129 genes), Colon Cancer (2000 genes), Breast Cancer Wisconsin(prognosis) (33 genes), Breast Cancer Wisconsin(diagnostic) (30 genes) and SPECTF Heart Cancer (44 genes). The EACSIDGWO achieved the most compact informative gene subsets along with the highest classification accuracies as follows: Ovarian Cancer (274 genes, 100%), Central Nervous System Cancer (1208 genes, 72%), Colon Cancer (538 genes, 91%), Breast Cancer Wisconsin (prognosis) (5 genes, 87%), Breast Cancer Wisconsin (diagnostic) (3 genes, 98%) and SPECTF Heart Cancer (4 genes, 88%). Extended Binary Cuckoo Search (EBCS), the second best state-of-the-art published algorithm, attained the following: Ovarian Cancer (1811 genes, 99%), Central Nervous System Cancer (3446 genes, 67%), Colon Cancer (988 genes, 89%), Breast Cancer Wisconsin (prognosis) (6 genes, 86%), Breast Cancer Wisconsin (diagnostic) (3 genes, 97%) and SPECTF Heart Cancer (6 genes, 86%). The results indicate that the proposed technique has comprehensive superiority in reducing the size of informative gene subsets as well as locating the most significant optimal gene subsets. To improve the performance of the classification phase (the last stage of the DNA microarray-based cancer analysis), another novel hybrid model is proposed. This model is based on particle swarm optimization (PSO), principal component analysis (PCA) and multiclass support vector machine (MCSVM) i.e. PSO-PCA-LGP-MCSVM. The MCSVM adopts a novel hybrid Linear-Gaussian-Polynomial (LGP) kernel formulated in this research. The hybrid LGP kernel innovatively combines the advantages of three standard kernels (Linear, Gaussian and Polynomial) in a novel manner, where a Gaussian kernel embedding a Polynomial kernel is linearly combined with a Linear kernel. To reveal the superior global gene extraction, prediction and learning ability of this model against three single kernel-based models: PSO-PCA-L-MCSVM (using a single Linear kernel), PSO-G-MCSVM (using a single Gaussian kernel) and PSO-P-MCSVM (using a single Polynomial kernel), four datasets: Colon cancer, Acute Lymphoblastic Leukemia-Acute myeloid Leukemia (ALL-AML), St. Jude Leukemia dataset and Lung cancer were used. Adopting three extended evaluation metrics (G-mean, Accuracy (Acc) and F-score) the proposed model achieved the following: Colon Cancer (G-mean: 0.88, Acc: 0.88, F-score: 0.87), ALL-AML (G-mean: 0.94, Acc: 0.94, F-score: 0.94), Lung Cancer (G-mean: 0.99, Acc: 0.97, F-score:

0.96) and St. Jude Leukemia dataset (G-mean: 0.97, Acc: 0.96, F-score: 0.90). The PSO-G-MCSVM, the second best published model, attained the following: Colon Cancer (G-mean: 0.82, Acc: 0.82, F-score: 0.82), ALL-AML (G-mean: 0.94, Acc: 0.94, F-score: 0.94), Lung Cancer (G-mean: 0.98, Acc: 0.96, F-score: 0.93) and St. Jude Leukemia dataset (G-mean: 0.97, Acc: 0.95, F-score: 0.85). Considering the reported compact informative gene subsets selection along with the very high classification accuracy, it is evident that the proposed models are promising DNA microarray data mining tools for both cost effective computers and online servers ,as well as resource constrained mobile devices.

TABLE OF CONTENTS

DECLARATION OF ORIGINALITY .... Error! Bookmark not defined.

DEDICATION .... **iii**

ACKNOWLEDGEMENT..**iv**

ABSTRACT...**v**

TABLE OF CONTENTS ....**vii**

LIST OF FIGURES..**xii**

LIST OF TABLES... **xiii**

LIST OF ACRONYMS ....**xv**

LIST OF SYMBOLS ....**xix**

CHAPTER ONE: INTRODUCTION...**1**

1.1 Background to the Study 1

1.1.1 DNA Microarray chips 2

1.1.2 Microarray experiment 2

1.1.3 Microarray gene expression matrix 4

1.1.4 Processing and analyzing microarray data 5

1.2 Motivation for this research 6

1.2.1 Class comparison..6

1.2.2 Class discovery..6

1.2.3 Class prediction 7

1.3 Research questions...7

1.4 Research objectives...8

1.5 Researcher's contribution..8

1.5.1 EBGWO: An Excited Binary Grey Wolf Optimizer for Feature Selection in Highly Dimensional Datasets 11

1.5.2 E-ACS-IDGWO: An Innovative Excited-ACS-IDGWO Algorithm for Optimal Biomedical Data Feature Selection 12

1.5.3 PSO-PCA-LGP-MCSVM: Particle Swarm Optimized Hybrid Kernel-Based Multiclass Support Vector Machine for Microarray Cancer Data Analysis 12

1.6 Organization of Dissertation 13

CHAPTER TWO: LITERATURE REVIEW..**16**

2.1 Introduction..16

2.2 Intrinsic features of highly dimensioned data 16

2.2.1 Many features 17

2.2.2 Limited sample size 17
2.2.3 Imbalance within dataset classes 18
2.2.4 Label noise/Mislabling 19
2.2.5 The intrinsic features of DNA microarray datasets 20
2.3 The Feature selection process 21
2.4 Filter Approaches 21
2.4.1 Approaches based on similarity 23
2.4.1.1 The Relief and extended relief (ReliefF) approaches 23
2.4.1.2 The Fisher's score technique 24
2.4.1.3 Laplacian score 24
2.4.2 Statistical techniques 25
2.4.2.1 Feature selection technique based on correlation (CFS) 25
2.4.2.2 Low Variance 25
2.4.2.3 The t-Score 26
2.4.3 Information theoretical-based approaches 26
2.4.3.1 FCBF (Fast correlation-based filter) 26
2.4.3.1 mRMR (Minimum redundancy-maximum relevance) filter 26
2.4.3.2 Information gain 27
2.5 Wrapper Approaches 27
2.5.1 ACO and $ABACO_H$ 28
2.5.2 PSO 29
2.5.3 IBGSA 30
2.5.4 BGWO and CBGWO 32
2.6 Hybrid techniques 35
2.7 Embedded techniques 36
2.8 Ensemble techniques 37
2.9 Analysis and discussion 41
2.9.1 Commonly utilized DNA microarray datasets 41
2.9.2 Commonly utilized criteria for performance evaluation 42
2.9.3 Normalization of data 43
2.9.4 Analysis of filter techniques 43
2.9.5 Analysis of hybrid techniques 44
2.9.6 Analysis of the hybrid-ensemble techniques 47

2.9.6.1 Hybrid-ensemble type 1 47

2.9.6.1.1 Selecting the most suitable filter and integration technique 48

2.9.6.2 Hybrid-ensemble type 2 52

2.10 Summary of the recently suggested attribute selection techniques for the DNA microarray chip analysis 54

2.11 Inferences drawn 54

CHAPTER THREE: AN EXCITED BINARY GREY WOLF OPTIMIZER FOR FEATURE SELECTION IN HIGHLY DIMENSIONAL DATASETS 58

3.1 Introduction 58

3.2 Grey Wolf Optimization (GWO) 59

3.3 Excited Grey Wolf Optimization (EGWO) 59

3.3.1 Nonlinearly controlling parameter $\boldsymbol{a}$ via the complete current response of the dc excited RC circuit 59

3.3.2 Socially Strengthened Hierarchy via a Fitness-value based position-updating criterion 61

3.4 Excited Binary Grey Wolf Optimization (EBGWO) 62

3.5 Experimental Results and Discussion 64

3.6 Dataset description 64

3.7 Parameter setting 65

3.8 Conclusion 70

CHAPTER FOUR: AN INNOVATIVE EXCITED-ACS-IDGWO ALGORITHM FOR OPTIMAL BIOMEDICAL DATA FEATURE SELECTION 72

4.1 Introduction 72

4.2 Related Works 74

4.2.1. Review of Hybridization of GWO with Other Search Algorithms 74

4.2.2. Review of Hybridization of CS with Other Search Algorithms. 75

4.3 Standard Cuckoo Search (CS) 76

4.3.1 Inspiration of CS 76

4.3.1.1 The Behavior of Cuckoo Birds 76

4.3.1.2 $\boldsymbol{Le'vy}$ Flights 77

4.3.2 Cuckoo Search (CS) Algorithm 77

4.3.2.1 Mathematical modelling of the standard CS 78

4.3.2.2 Intensification Using $\boldsymbol{Le'vy}$ Flight Random Walk (LFRW). 78

4.3.2.2 Diversification by the use of biased-selective random walk (BSRW) 79

4.3.2.3 Elitist scheme via greedy selection 80

4.4 Excited -Adaptive Cuckoo Search- Intensification dedicated Grey Wolf Optimization (EACSIDGWO) 80

4.4.1 Adaptive Cuckoo Search (ACS) 81

4.4.1.1 Adaptive step size via the complete voltage response of the dc excited RC circuit 81

4.4.2 Intensification dedicated grey wolf optimizer (IDGWO) 84

4.4.2.1 Nonlinearly controlling parameter $\boldsymbol{a}$ via the complete current response of the dc excited RC circuit 84

4.4.2.2 Enhanced mature convergence via a fitness value based position-updating criterion 85

4.4.3 Proposed EACSIDGWO (Continuous version) 86

4.4.4 Proposed EACSIDGWO (Binary version) 87

4.5 Experimental Methodology 90

4.5.1 Considered Biomedical Datasets 90

4.5.2 Evaluation Metrics 91

4.5.3 Evaluation of the classifier 92

4.5.4 Fitness function 94

4.5.5 Parameter setting for the considered feature selection techniques 95

4.6 Results 96

4.7 Discussion 99

4.7.1 Investigation of the obtained classification results 99

4.7.2 Statistical analysis 100

4.7.3 Ranking methods 101

4.8 Conclusion 102

CHAPTER FIVE: PARTICLE SWARM OPTIMIZED HYBRID KERNEL BASED MULTI-CLASS SUPPORT VECTOR MACHINE FOR MICROARRAY CANCER DATA ANALYSIS 104

5.1 Introduction 104

5.2 PSO-PCA-LGP-MCSVM PRINCIPLES 107

5.2.1. Normalization 107

5.2.2 Principal component analysis (PCA) 107

5.2.3 Multi-class support vector machine (MCSVM) 109

5.2.4 Linear-Gaussian-Polynomial MCSVM (LGP- MCSVM) 111

5.2.5 Particle swarm optimization (PSO) 113

5.2.6 PCA-PSO-LGP-MCSVM model........115

5.3 Performance evaluation........118

5.3.1 Microarray Datasets........118

5.3.2 Performance measures for imbalanced microarray datasets........120

5.4 Results and Discussions........122

5.5 Chapter Conclusion........124

CHAPTER SIX: CONCLUSION AND RECOMMENDATIONS........126

6.1 General Conclusions........126

6.2 Key Findings........132

6.3 Recommendation for Further Work........135

REFERENCES........136

LIST OF PUBLICATIONS........176

IMPLEMENTED SOURCE CODES........242

LIST OF FIGURES

Figure 1.1: Gene expression 3

Figure 1.2: Stages for microarray data preparation 3

Figure 1.3: Steps followed during a microarray experiment 4

Figure 1.4: Microarray image 5

Figure 1.5: Workflow for the conducted Research 10

Figure 2.1: Filter based feature selection process 23

Figure 2.2: Flowchart of Wrapper based feature selector..28

Figure 2.3: Flowchart of Hybrid based feature selector 35

Figure 2.4: Instance 1 of ensemble techniques (feature selection) 38

Figure 2.5: Instance 2 of ensemble techniques (classification) 38

Figure 2.6: Framework of hybrid-ensemble technique 40

Figure 2.7: Scheme for the hybrid-ensemble type 1. 48

Figure 2.8: Scheme for the hybrid-ensemble type 2. 52

Figure 2.9: Recent feature selection contributions to the DNA microarray data analysis according to the data collected in..54

Figure 5.1: Scheme of the proposed PSO-PCA-LGP-MCSVM algorithm 117

LIST OF TABLES

Table 2.1: Imbalance ratio of 10 standard cancer microarray datasets....................................18

Table 2.2: Hybrid techniques for feature selection in highly dimensional datasets 36

Table 2.3: Embedded techniques for feature selection in highly dimensional datasets...........37

Table 2.4: Ensemble-based techniques used for feature selection in highly dimensional datasets...41

Table 2.5: Commonly utilized DNA microarray datasets for benchmarking..........................42

Table 2.6: Experimental results for nine filters combine with the SVM classifier (10-fold cross validation)...44

Table 2.7: Experimental results for nine filters combine with the KNN classifier (10-fold cross validation)...44

Table 2.8: Performance of various hybrid techniques for feature selection and Cancer classification in DNA microarray data 46

Table 2.9: Experimental results for hybrid-ensemble techniques combined with KNN and utilizing the AND integrator 50

Table 2.10: Experimental results for hybrid-ensemble techniques combined with KNN and utilizing the OR integrator 51

Table 2.11: Experimental results for hybrid-ensemble techniques for different threshold values of the ReliefF filter approach...53

Table 3.1: Microarray datasets used in the experiments..65

Table 3.2: Parameter settings for each considered algorithm..65

Table 3.3: Experimental Results for Brain_Tumour1 datase...64

Table 3.4: Experimental Results for Brain_Tumour2 dataset 65

Table 3.5: Experimental Results for CNS dataset 65

Table 3.6: Experimental Results for DLBCL dataset..64

Table 3.7: Experimental Results for Leukemia dataset 68

Table 3.8: Experimental Results for Colon dataset 68

Table 3.9: Experimental Results for Lung Cancer dataset. 68

Table 4.1: Considered Biomedical Datasets 90

Table 4.2: Selection of suitable kernel functions...94

Table 4.3: Selection of parameter values for the considered approaches 95

Table 4.4: Experimental results for the ovarian cancer dataset 96

Table 4.5: Experimental results for the breast cancer Wisconsin (Diagnostic) dataset...........98

Table 4. 6: Experimental results for the breast cancer Wisconsin (Prognosis) dataset 98
Table 4. 7: Experimental results for the SPECTF Heart dataset...98
Table 4.8: Experimental results for the CNS dataset 99
Table 4.9: Experimental results for the colon dataset..99
Table 4.10: Using Wilcoxon's rank sum test at $p = 0.05$ to compare EACSIDGWO with other algorithms 101
Table 4. 11: Overall ranking of considered algorithms 102
Table 4.12: Overall ranking of considered algorithms 102
Table 5.1: Parameters and their respective ranges...114
Table 5.2: Initial PSO parameters setting 114
Table 5.3: The cancer microarray datasets utilized in this chapter..118
Table 5.4: Percentage proportion for calibration, validation and test sets.............................120
Table 5.5: Confusion matrix for a two-class problem 121
Table 5.6: Accuracy of all considered models on the four microarray datasets, where bold represents the best result and the italics denotes the worst in each column respectively 123
Table 5.7: F-Score of all considered models on the four microarray datasets, where bold represents the best result and the italics denotes the worst in each column respectively 123
Table 5.8: G-mean of all considered models on the four microarray datasets, where bold represents the best result and the italics denotes the worst in each column respectively 124
Table 6.1: Summary of research findings 132

LIST OF ACRONYMS

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ACC	Accuracy
ACS	Adaptive Cuckoo Search
ABACO$_H$	Advanced Binary Ant Colony Optimization
ACS	Adaptive Cuckoo Search
AMLALL	Acute Myeloctic Leukemia Acute Lymphocytic Leukemia
BACO	Binary Ant Colony Optimization
BA1	Bat Algorithm
BA2	Biogeography Algorithm
BC	Bagging Classifier
BDE	Binary Differential Evolution
BGA	Binary Genetic Algorithm
BGSA	Binary Gravitational Search Algorithm
BGWO	Binary Grey Wolf Optimization
BGWO1	Binary Grey Wolf Version 1
BGWO2	Binary grey Wolf version 2
BHA	Black Hole Algorithm
BPSO	Binary Particle Swarm Optimization
BSRW	Biased Selective Random Walk
CBGWO	Competitive Binary Grey Wolf Optimization
CCR	Correct Classification Rate
CFS	Correlation Feature Selection
CMA-CS	Covariance Matrix Adaptation Cuckoo Search
CNS	Central Nervous System
CS	Cuckoo Search
DA	Dragonfly Algorithm
DC	Direct Current
DE	Differential Evolution

DNA	Deoxyribonucleic Acid
DKD	Diabetic Kidney Disease
DLBCL	Diffuse Large B-Cell Lymphoma
E1-cp	Ensemble1-Cummulative Probability
E1-nk	Ensemble1-Naïve-Bayes K-Nearest Neighbor
E1-ns	Ensemble1-Naïve-Bayes Support Vector Machine
E2	Ensemble2
E-ACS-IDGWO	Excited Adaptive Cuckoo Search Intensification Dedicated Grey Wolf Optimization
EBGWO	Excited Binary Grey Wolf Optimization
EGWO	Excited grey Wolf Optimization
ER	Estrogen Receptor
FC	Fuzzy Classification
FCBF	Fast Correlation Based Filter
FCE	Feature Co-Association Ensemble
FN	False Negative
FP	False Positive
FSCBAS	Feature Selection Clustering Binary Ant System
GA	Genetic Algorithm
GADP	Genetic Algorithm Dynamic Parameter
GBC	Genetic Bee Colony
GI-FS	Gini Index Feature Selection
GM	Geometric Mean
GOA	Grasshopper Optimization Algorithm
GSA	Gravitational Search Algorithm
GWO	Grey Wolf Optimization
HM-ABACO	Hybrid Method Advanced Binary Ant Colony Optimization
IBGSA	Improved Binary Gravitation Search Algorithm
IDGWO	Intensification Dedicated Grey Wolf Optimization
IFP	Iterative Feature Selection
IG	Information Gain

KNN	K- Nearest Neighbor
KP-CSSV	Kernel-Penalized Cost Sensitive Support Vector
KP-SVM	Kernel-Penalized Support Vector Machine
LOOCV	Leave One Out Crosss-Validation
MCC	Mathew's Correlation Coefficient
mRMR	Minimum Redundancy Maximum Relevance
mRNA	Messenger Ribonucleic Acid
LFRW	Lev'y Flight Random Walk
LGP	Linear Gaussian Polynomial
MCSVM	Multi Class Support Vector Machine
MCF-RFE	Multi-Criteria Fusion Recursive Feature Elimination
MI	Mutual Information
MIM	Mutual Information Maximization
MIGRFE	Multilayer Genetic Recursive Feature Elimination
NB	Naïve Bayes
NFL	No Free Lunch
NGS	Next Generation Sequencing
NN	Neural Network
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
PSO-PCA-G-MCSVM	Particle Swarm Optimization Principle Component Analysis Gaussian Multi Class Support Vector Machine
PSO-PCA-L-MCSVM	Particle Swarm Optimization Principle Component Analysis Linear Multi Class Support Vector Machine
PSO-PCA-P-MCSVM	Particle Swarm Optimization Principle Component Analysis Polynomial Multi Class Support Vector Machine
PSO-PCA-LGP-MCSVM	Particle Swarm Optimization Principle Component Analysis Linear Gaussian Polynomial Multi Class Support Vector Machine
RC	Resistor Capacitor
RFACO-GS	Relief Algorithm Ant Colony Optimization Gene Selection
RFR	Random Forest Ranking

RNA-Seq	Ribonucleic Acid Sequencing
SA-EFS	Sort Aggregation Ensemble Feature Selection
SN	Sensitivity
SP	Specificity
SPECTF	Single Proton Emission Computer Tomography
SVM-RFE	Support Vector Machine Recursive Feature Elimination
SOMs	Self Organizing Maps
SRBCT	Small Round Blue Cell Tumors
SU	Symmetrical Uncertainty
SVM	Support Vector Machine
TP	True Positive
TN	True Negative
CFS-TGA	Correlation Feature Selection Taguchi-Genetic Algorithm
UCI	University of California at Irvine
V-WSP-PSO	Variable Wootton Sergent Phan-Tan-Luu's particle Swarm Optimization

LIST OF SYMBOLS

∑	Summation
#	Number
∪	Union
∈	Element of
∏	Product

CHAPTER ONE: INTRODUCTION

1.1 Background to the Study

The DNA microarrays (chips) have largely transformed the approach of conducting scientific research for genome analysis. Microarray slides have facilitated simultaneous recording of thousands of gene expression levels, which consequently has enabled many researchers acquire unprecedented insights of the living organism mechanism on a wider genome scale [1].

These microarray chips have facilitated parallel monitoring of thousands of gene expressions under distinct conditions like in a sample of a biological tissue or on an experimental time stamp. The genome, which is a group of genes belonging to an organism, influences every form of development, functioning and even susceptibility of an organism to certain disorders and diseases [2].

Genetic mutations are the major underlying causes of the many existing disorders and diseases. Thus, both gene activities and gene interactions need to be determined in order to examine the biochemical mechanism [3] [4].

The DNA chips are widely applied in various scientific disciplines of Medicine and Biology. Currently, the various studies utilizing microarray chips include gene co-regulation, clinical diagnosis, gene function discovery, differential gene expression and patterns of gene activity under different chemical treatment [1] [5].

Gene expression profiling entails taking note of which genes are unaltered under certain conditions. Though perceived as the most basic microarray application, it is one study that reveals useful biological insights about an organism's genome [3] [6].

A group of genes sharing related regulatory patterns under a given condition can also share related biological functions. Furthermore, a given gene expression profile can be key in determining diseased or abnormal cellular functions which makes it a necessary tool in the current clinical research, especially in cancer diagnosis [7] [8].

Globally, cancer has adversely affected the society. Cancer is defined as a cluster of around 100 distinct diseases that can attack any part of the body, and largely characterized by unrestricted rise in the number of abnormal cells. It is considered the leading cause of mortality and morbidity worldwide, more so in third-world countries like Kenya [9] [10].

A timely diagnosis combined with target specific therapies has proved to be effective in cancer treatment and thus increasing the survival rate of cancer patients [11]. Currently, many researchers are actively developing systems that can speed up the cancer diagnostic process by aiding in the medical investigations phase using the gene-based biomarkers [1].

DNA microarray chips provide deeper insights of a number of genetic alterations that are related to cancer [8].Moreover, these chips are widely being applied in toxicological prediction studies, gene mapping with their respective encoded proteins, drug response analysis, identification of molecular targets for drugs and pharmacogenomics applications [12] [13] [14].

The following are scientific objectives addressed by the current microarray research :

i) Detection of genes that are co-expressed.

ii) Mapping of active regions within a genome to facilitate testing of the internal metabolism of an organism.

iii) Determining of gene expression profiles that might be early biomarkers of a given disease.

iv) Classification of various types of tissues using the disease response i.e. early disease discovery.

v) Determining gene expression profiles that will aid in differentiating biological entities.

vi) Studying gene activity patterns in various different stress conditions.

Though DNA microarray chips have proved key in the cancer diseases diagnosis, they have an inherently enormous raw data whose handling and processing poses a great challenge to existing machine learning tools [1] [2].

A consideration of DNA microarray experiments and the various processes undertaken in generating the microarray data is outlined in the subsequent subsections.

1.1.1 DNA Microarray chips

DNA microarray chips is a technology initially developed by Patrick H. Brown Laboratory [15] i.e. cDNA Microarrays in 1995 and Affymetrix i.e. High-Density Oligonucleotide arrays in 1966 [16].

The DNA microarray chips have proved to be powerful tools in genomics. They have enabled scientific researchers to assess activities and interactions existing within tens of thousands of genes concurrently. This is a milestone in comparison to classical molecular biological tools, which facilitates an assessment of one or a small set of genes [17] [18]. Since the microarray chips facilitates an understanding of processes within living organisms at molecular level, it is a promising tool with many applications in the field of medicine and biology [19] [20].

Currently, microarray data analysis is one of the major research areas in bioinformatics. A DNA microarray chip enables scientists to conduct experiments on thousands of genes simultaneously with the aim of evaluating gene expression patterns. For the first time, these chips have revolutionized the study of human genomics by enabling scientists to monitor expressions of thousands of genes concurrently [21] [22].

1.1.2 Microarray experiment

A DNA microarray chip is a glass microscope slide with fixed spots of synthesized DNA molecular strands. One DNA chip consists of ten thousands of spots, where each spot corresponds to a single gene.

High-density oligonucleotide chips [23] and spotted arrays [24] are some of the variations of a microarray technologies that are in existence. Mostly, this microarray technology is used to compare two different samples i.e. an unknown sample and a control sample with the aim of determining the mRNA abundance [25] .

The DNA microarrays are used to monitor variations in gene expressions levels during upregulations or down regulations processes. The Expression of genetic information is described in two phases i.e. the transcription and the translation phases as depicted in Figure 1.1 [26].

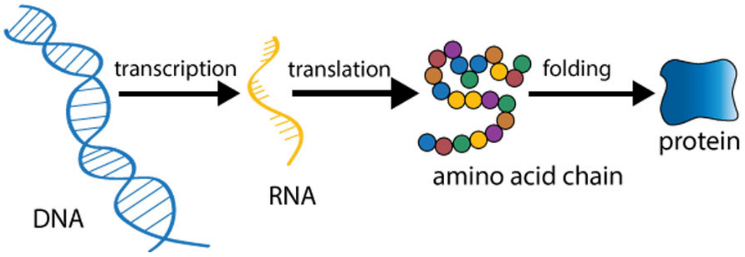


Figure 1.1: Gene expression[26]

The main steps undertaken in the preparation of microarray data are shown in Figure 1.2.

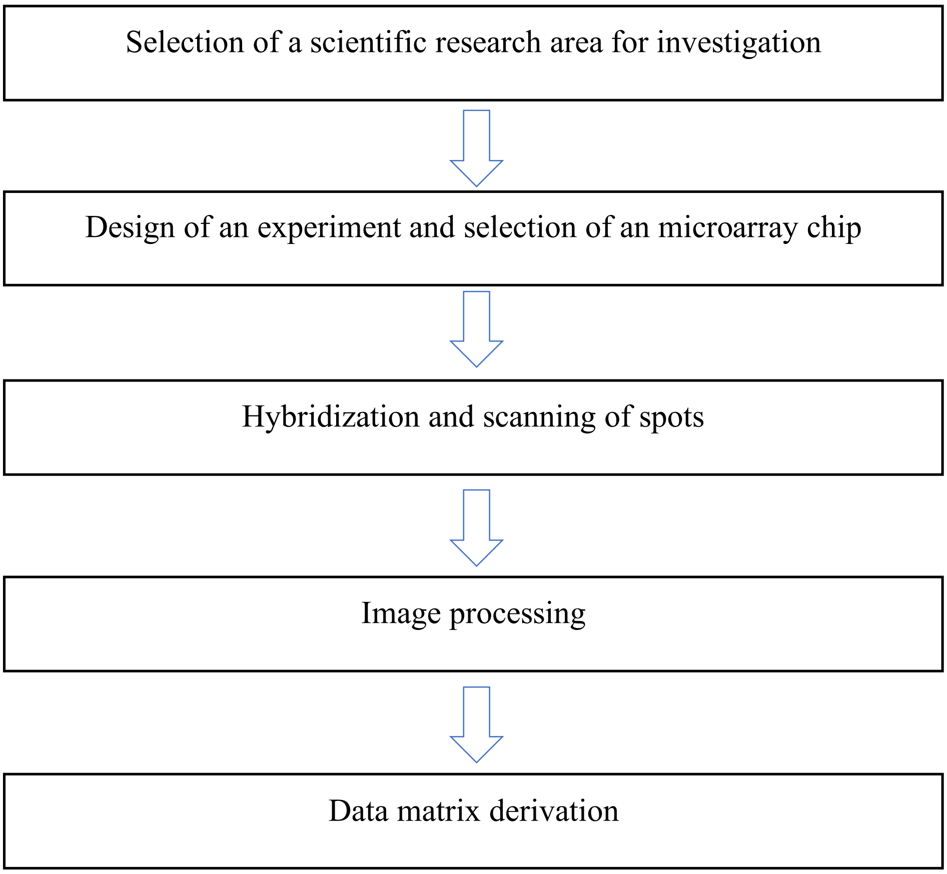

Selection of a scientific research area for investigation
Design of an experiment and selection of an microarray chip
Hybridization and scanning of spots
Image processing
Data matrix derivation

Figure 1.2: Stages for microarray data preparation

First and foremost, mRNA is extracted from a selected tissue or a given cell line. Furthermore, the extracted mRNA is utilized in the production of a sample, which is labelled with fluorescent nucleotides (which are red or green in color). The labelled sample is hybridized simultaneously to derive multiple DNA sequences that are affixed in an organized array structure mounted on a solid surface.

Further, the populated microarray is scanned using a microscope and the resulting fluorescent on every spot of the microarray is determined. The quantity of each mRNA in the originally selected sample is presumed to be directly proportional to the amount of the measured label.

Finally, an image analysis software is used to derive an image from the hybridized microarray surface. Digitization of the red-to-green fluorescence is carried out to determine the ratio output values that will indicate the gene expression values. Figure 1.3 [27] outlines the processes of a typical microarray experiment.

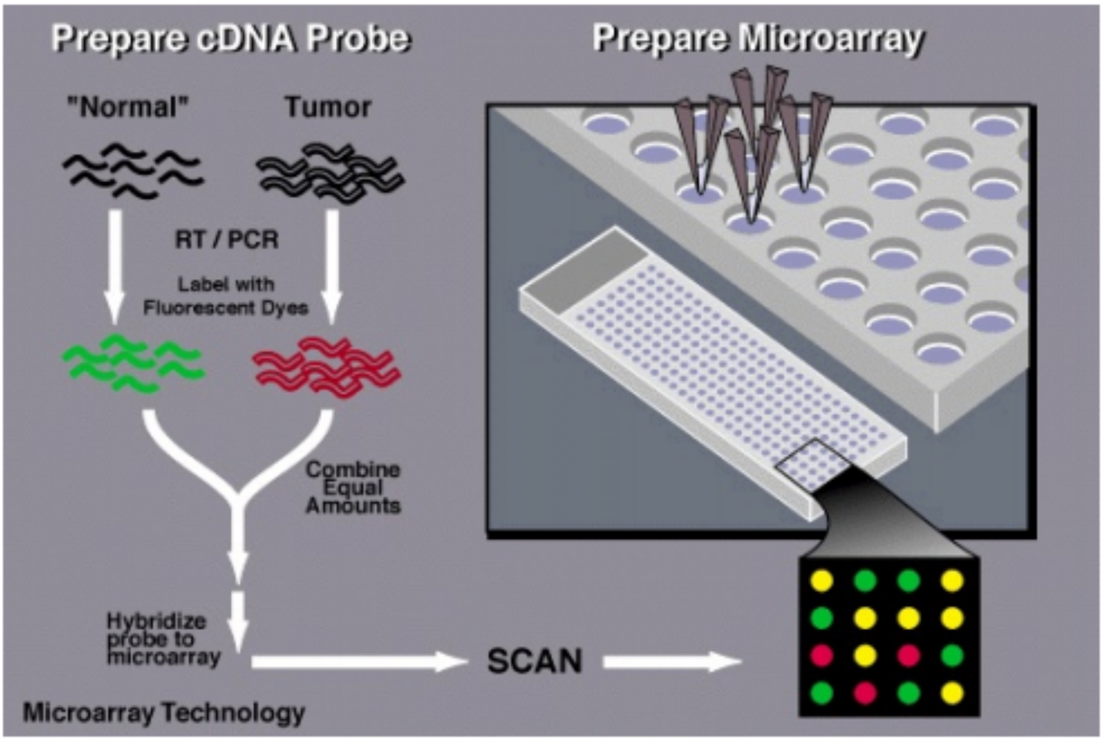


Figure 1.3: Steps followed during a microarray experiment [27]

1.1.3 Microarray gene expression matrix

The microarray data is normally depicted as a gene expression matrix. The rows of the matrix represent the genes while the columns outline different tissue samples, development phases or drug treatments. Each cell of the matrix contains a value corresponding to the gene expression level under a specific sample condition [28].

The gene expression matrix G can be described by an $r \times n$ matrix. The rows depict the expression patterns of genes i.e. $g_1, g_2, \dots, g_r$ while the columns depict the expression profile of considered samples i.e. $s_1, s_2, \dots, s_n$. The value e_{ab} is the measured expression level corresponding to gene a in sample b. Thus the definition of G can be represented by Equation 1.1

$$G = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{r1} & e_{r2} & \cdots & e_{rn} \end{bmatrix} \leftarrow \begin{matrix} g_a \; a = 1,2,\dots,r; \\ s_b \; b = 1,2,\dots,n \end{matrix} \qquad 1.1$$

1.1.4 Processing and analyzing microarray data

The microarray experiments normally generate raw data consisting of images derived from the hybridized arrays as depicted by Figure 1.4 [29].

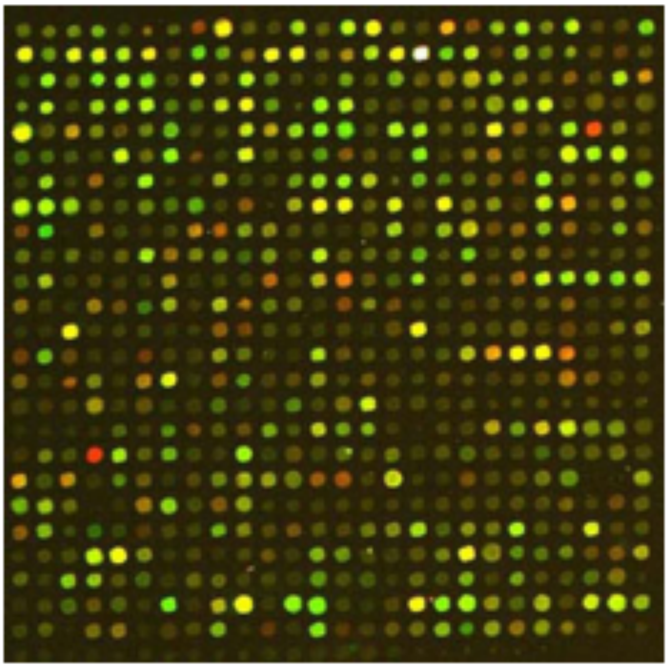


Figure 1.4: Microarray image[29]

To obtain numerical values that can constitute the gene expression matrix, the hybridized signals are transformed using salient data processing techniques [27]. Microarrays can reveal critical biological information such as the specific gene expression patterns associated with malignant tissues of many cancer disease types. However, interpretation of the derived data is key in microarray data analysis. Though very critical, interpretation of microarray data is normally affected by a number of factors not limited to statistical variations encountered during gene value estimation, management of huge microarray data and lack of sufficient knowledge on gene functions and their interactions [30].

1.2 Motivation for this research

By facilitating a concurrent measurement of tens of thousands of genes, the DNA microarray chips have paved way for the discovery of useful biological information regarding interactions at molecular level within an organism [1]. Moreover, these chips have facilitated the development of diverse and attractive clinical tools for disease diagnosis.

Clinical diagnosis entails the discovery of disease informative genes and the selection of the right diagnosis based on the discovered informative genes. Currently, the main research objective for the DNA microarray data analysis is the formulation of generic techniques for cancer disease classification [19]. This is mainly because, the microarray chips generate unprecedented biological information regarding the pathology and disease progression, molecular level changes, resistance and response to specific administered therapies [3].

Accurate and timely cancer disease diagnosis and the classification of its subtypes are the main challenges facing the medical field today [11]. Cancer diagnosis is mainly based on morphological properties of diseased/malignant tissues. However, classifications that solely rely on these morphological features have proved to be insufficient [18].This is because tumors that are morphologically similar can be categorized into distinct classes using the gene expression profiles [14].

It is a well-established fact that systematic pattern deviations of gene expressions within a specific type of cell is directly correlated with some biological variations belonging to a specific cancer type. However, to make use of this well-defined fact in the fight against cancer, selection of a small number of predictive genes from the highly dimensional microarray data is paramount [16].

Currently, there are three major research areas considered in the microarray data analysis as outlined below.

1.2.1 Class comparison

This research area considers the development of various techniques to analyze the DNA microarray chips with aim of determining differentially expressed genes among selected tissue samples. It is an area mainly focusing on the establishment of the upregulated or downregulated genes [31] [32]. The outcome of this research mainly helps the scientific and the medical world to interpret the genetic differences existing between diseases and their subtypes. Researchers in this field mainly use statistical analysis techniques like the ANOVA, z- and t-tests [33] [34].

1.2.2 Class discovery

Researchers in this area mainly focuss on developing approaches that can identify similar gene expression patterns and group these genes into functionally related gene expression classes [35] [36]. Many unsupervised approaches like self-organizing maps (SOMs) hierarchical clustering have been utilized in this research area [37]. However, since the considered microarray chips in this research are inherently highly dimensional with correlated genes, in many cases the derived clusters may not necessarily reveal all the needed information for biological discrimination [37].

1.2.3 Class prediction

Researchers in this area develop enhanced prognostic and diagnostic tools that incorporate supervised machine learning and soft computing techniques for the management of various diseases [8].

Some of the recent studies in this area [1] have revealed that the DNA microarray chips are promising tools for the cancer disease diagnosis. These studies strongly suggested that the gene expression profiles could be utilized in the classification of tissue samples. These studies pointed out that the disease identification especially cancer disease classification is the most considered area of application for the DNA microarray chips.

However, these high-throughput functional genomic tools i.e. the DNA chips are highly dimensional (contain a vast number of genes in the range of tens of thousands) with limited sample space (in the range of hundreds) which has demanded development of techniques which are low-demanding in terms of computing resources, fast and efficient in dimensionality reduction prior to cancer classification.

Though a number of soft computing and pattern recognition tools have been reported to handle this challenge, a number of them suffer from the risk of overfitting and are computationally expensive. Thus, the main objective of this research is to develop versatile hybrid soft computing and pattern recognition tools to aid in predicting the class of unknown samples quickly and accurately.

Soft computing tools are proficient in utilizing artificial intelligence to handle the uncertainty and precision posed by the highly dimensional and unsymmetrically structured DNA microarray chips. Though promising, a good number of the existing soft computing tools are inherently slow in computation, converge prematurely and suffer from a large and complex search space resulting from the highly dimensional DNA data [38].

To overcome the aforementioned challenges and still apply these promising tools in the DNA microarray data analysis for cancer disease diagnosis, this research considers formulation of novel hybridization techniques for existing computing and pattern recognition. Efficient hybridization will not only overcome the individual shortcomings of the combined soft computing tools but also portray superior diversification and intensification capabilities while handling the DNA microarray gene data.

1.3 Research questions

This research tries to answer the following research questions:

i) How can the concept of a complete current response of DC excited RC circuit be adopted to overcome the local optimal trapping and strike a better balance between exploration and exploitation in grey wolf optimization based feature selection, for effective gene selection and improved classification accuracy in DNA microarray based cancer data?

ii) How can the step-size of the cuckoo search algorithm be made adaptive via the concept of a complete voltage response of DC excited RC circuit in order to improve both its convergence speed and search ability?

iii) How can the EBGWO be hybridized with the ACS in order to strike an optimal balance between the EBGWO's exploitation and exploration capabilities?

iv) How can an adaptive hybrid kernel using three standard kernels (linear, Gaussian and Polynomial kernels) be formulated for the multi-class SVM classifier inorder to effectively classify DNA microarray based cancer data?

1.4 Research objectives

The main objective of this research is to develop versatile and suitable decision models to effectively analyze DNA microarrays that are; imbalanced, highly dimensional with low sparcity and genes that are directly or indirectly correlated. Specifically, this research is focused on the following objectives:

i) To improve the stability, diversity and robustness of the existing binary grey wolf optimization gene selector in highly dimensional DNA microarray based cancer data via the concept of the complete current response of DC excited RC circuit (i.e. EBGWO).

ii) To make the step-size of the cuckoo-search algorithm adaptive (i.e. ACS) via the concept of a complete voltage response of DC excited RC circuit with the aim of improving both its convergence speed and search ability in DNA microarray based cancer disease classification.

iii) To optimally strike a balance between the EBGWO's exploitation and exploration capabilities by hybridizing it with the ACS algorithm.

iv) To improve the SVM's (a commonly utilized classifier in the DNA microarray based cancer classification) learning and classification capability by innovatively hybridizing its three standard kernels (i.e. linear, Gaussian and polynomial kernels).

1.5 Researcher's contribution

From the literature review, a number of researchers are actively involved in the analysis of DNA microarray cancer chips. This is because DNA microarray chips have portrayed huge potential in the fight against cancer.

The Microarray experiments are generating voluminous amount of data that is directly linked to genetic mutations resulting from a specific cancer disease. Thus, it is desirable to build reliable diagnostic tools that are based on these potential chips in order to speed up the cancer diagnosis and classification process.

Though microarray chips are promising in the fight against cancer, they are associated with a number of computational issues that need to be addressed for the whole process to be deemed successful. These challenges include the identification of genes that are differentially expressed for a specific cancer classification, efficient classification of these imbalanced microarray data and determining the existing relationships among the gene expression profiles.

Dimensionality reduction has been identified as a key issue in designing reliable cancer diagnostic tools. So far, a number of gene selection techniques have been reported to tackle this issue. Majority of these approaches fall under the filter and wrapper categories. Though filter approaches are computationally efficient and fast compared to wrappers, they are classifier independent and ignore possible interactions among microarray genes making them inefficient for the microarray classification task.

On the other hand, wrappers provide attractive classification accuracies compared to filters, but they have a high computation cost resulting to slow convergence. This renders wrappers too inefficient for microarray classification task.

Thus, it is evident that neither a single filter approach nor wrapper approach can offer optimal dimensionality reduction for the DNA microarray chips. Both approaches need to be either improved or hybridized in order for them to be effective in the dimensionality reduction of microarray datasets.

The workflow adopted for this research is outlined in Figure 1.5. The colored lines depict the roadmap followed in this research. From Figure 1.5, this research made three notable contributions, which are summarized in the following subsections.

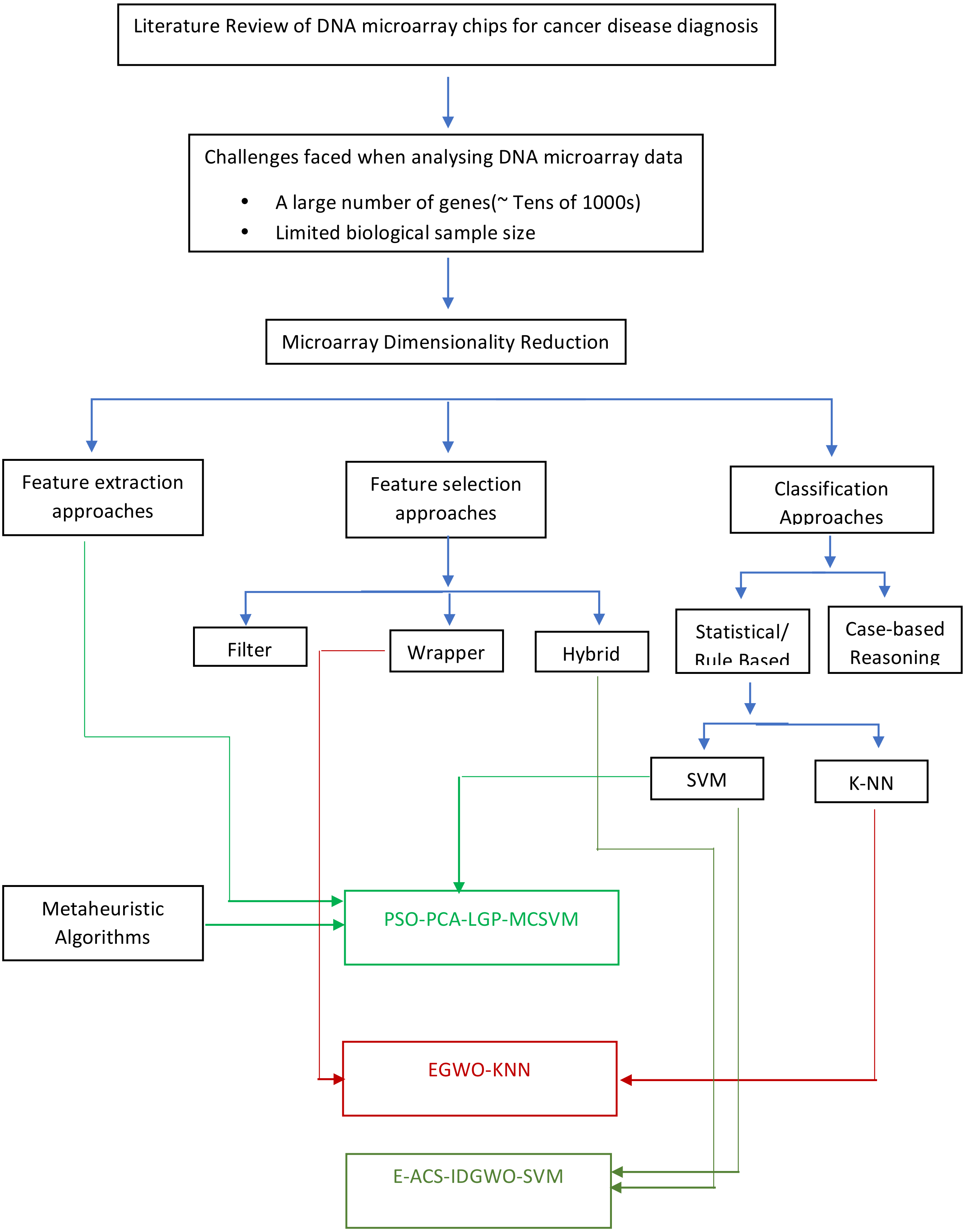

Literature Review of DNA microarray chips for cancer disease diagnosis
Challenges faced when analysing DNA microarray data
A large number of genes(~ Tens of 1000s)
Limited biological sample size
Microarray Dimensionality Reduction
Feature extraction approaches
Feature selection approaches
Classification Approaches
Filter
Wrapper
Hybrid
Statistical/ Rule Based
Case-based Reasoning
SVM
K-NN
Metaheuristic Algorithms
PSO-PCA-LGP-MCSVM
EGWO-KNN
E-ACS-IDGWO-SVM

Figure 1.5: Workflow for the conducted Research

The mains contributions made in this research are summarized in subsections 1.5.1, 1.5.2 and 1.5.3 respectively.

1.5.1 EBGWO: An Excited Binary Grey Wolf Optimizer for Feature Selection in Highly Dimensional Datasets

To select a subset of informative genes from the highly dimensional DNA microarray chips, a novel excited binary grey wolf optimization (EBGWO) based wrapper utilizing the K-NN classifier is presented in Chapter 3. To overcome the local minima trapping of the existing BGWO that normally results into semi-optimal solutions, in the proposed EBGWO, a new position-updating criterion is formulated. The new position updating criterion utilizes the fitness values of vectors $X_1^{\rightarrow}$, $X_2^{\rightarrow}$and $X_3^{\rightarrow}$ to determine the new candidate individuals. These vectors are derived from the union of scalars X_1, X_2 and X_3 respectively of the existing BGWO. Moreover, to make full use of and strike a better balance between exploration and exploitation, which is also a challenge in the BGWO, a novel nonlinear control strategy is formulated. This non-linear strategy innovatively decreases parameter $a^{\rightarrow}$ via the concept of the complete current response of a direct current (DC) excited resistor-capacitor (RC) circuit. One induction algorithm i.e. the K-Nearest Neighbor (K-NN) is utilized in the proposed wrapper approach to evaluate the classification performance of subset of genes selected by the EBGWO, using 5-fold cross-validation technique.

The performance of EBGWO as a gene selector is evaluated on 7 standard DNA microarray chips derived from Irvine (UCI) repository namely Brain Tumour1 (5920 genes), Brain Tumour2 (30367 genes), Central Nervous System Cancer (7129 genes), Diffuse Large B-Cell Lymphoma (DLBL) (5469 genes), Leukemia (7129 genes), Colon Cancer (2000 genes) and Lung Cancer(12600). The EBGWO achieved the most compact informative gene subsets along with the highest classification accuracies as follows: Brain Tumour1 (501 genes, 92%), Brain Tumour2 (1151 genes, 88%), Central Nervous System Cancer (710 genes, 83%), DLBL (426 genes, 100%), Leukemia (649 genes, 90%), Colon Cancer (143 genes, 92%) and Lung Cancer(1005 genes, 98%). Binary Grey Wolf Optimization 2 (BGWO2), the second best state-of-the-art published algorithm, attained the following: Brain Tumour1 (1343 genes, 89%), Brain Tumour2 (3083 genes, 85%), Central Nervous System Cancer (2175 genes, 78%), DLBL (1408 genes, 98%), Leukemia (1805 genes, 87%), Colon Cancer (455 genes, 90%) and Lung Cancer(2413 genes, 97%).On average, the proposed EBGWO algorithm attained a reduced informative gene subset with 655 genes along with a classification accuracy of 92%. On the other hand, on average the BGWO2 (second best algorithm) attained a reduced informative gene subset with 1812 genes along with a classification accuracy of 89%. **Thus in comparison with BGWO2 (the current best gene selector that is based on the GWO algorithm), on average the proposed EBGWO algorithm reduced the number of selected genes from 1812 to 655 (i.e. a further reduction of 1157 genes) while improving the classification accuracy from 89% to 92% (i.e. an improvement by 3%)**.

1.5.2 E-ACS-IDGWO: An Innovative Excited-ACS-IDGWO Algorithm for Optimal Biomedical Data Feature Selection

Though the proposed EBGWO wrapper has proved attractive in selecting informative genes from the highly dimensioned DNA microarray datasets due to its enhanced stability and diversity capabilities, it does not strike an optimal balance between exploitation and exploration during the search process. This is because exploitation and exploration are two contradicting principles, which must be balanced efficiently in order to achieve an improved performance of a metaheuristic. Moreover, attaining an optimal balance between these antagonist principles is difficult with a single metaheurist. To attain the required optimal balance between exploitation and exploration, another innovative excited-ACS-IDGWO complementary hybrid model comprising of two improved wrappers i.e. adaptive cuckoo search algorithm (ACS) and intensification dedicated grey wolf optimizer (IDGWO) (a variant of the EBGWO wrapper presented in Chapter 3) and using the SVM classifier is presented in Chapter 4. The proposed model innovatively adopts the concept of the complete voltage and current responses of a direct current (DC) excited resistor-capacitor (RC) circuit to nonlinearly control parameter $\boldsymbol{a}^{\rightarrow}$ of IDGWO and the step size of ACS. To handle the higher diversity of the search space during the early stages, both the ACS and IDGWO are jointly involved in the local exploitation. Conversely, to promote mature convergence during later stages of the search space, the role of ACS is shifted to global exploration while the IDGWO is left carrying out local exploitation. The performance of the proposed model is compared with those of four state-of-art wrappers. The proposed technique emerged to be superior in attaining a good learning from a few samples and optimally deriving a reduced feature subset from the information-rich datasets.

The superiority of the proposed E-ACS-IDGWO is further proved via a number of statistical approaches like ranking techniques and statistical analysis.The performance of EACSIDGWO as a gene selector is evaluated on six standard DNA microarray chips derived from Irvine (UCI) repository namely Ovarian Cancer (4000 genes), Central Nervous System Cancer (7129 genes), Colon Cancer (2000 genes), Breast Cancer Wisconsin (prognosis) (33 genes), Breast Cancer

Wisconsin (diagnostic) (30 genes) and SPECTF Heart Cancer (44 genes). The EACSIDGWO achieved the most compact informative gene subsets along with the highest classification accuracies as follows: Ovarian Cancer (274 genes, 100%), Central Nervous System Cancer (1208 genes, 72%), Colon Cancer (538 genes, 91%), Breast Cancer Wisconsin (prognosis) (5 genes, 87%), Breast Cancer Wisconsin (diagnostic) (3 genes, 98%) and SPECTF Heart Cancer (4 genes, 88%). Extended Binary Cuckoo Search (EBCS), the second best state-of-the-art published algorithm, attained the following: Ovarian Cancer (1811 genes, 99%), Central Nervous System Cancer (3446 genes, 67%), Colon Cancer (988 genes, 89%), Breast Cancer Wisconsin (prognosis) (6 genes, 86%), Breast Cancer Wisconsin (diagnostic) (3 genes, 97%) and SPECTF Heart Cancer (6 genes, 86%). On average, the proposed EACSIDGWO algorithm attained a reduced informative gene subset with 339 genes along with a classification accuracy of 89%. On the other hand, on average the EBCS (second best algorithm) attained a reduced informative gene subset with 1043 genes along with a classification accuracy of 87%. **Thus in comparison with EBCS (the current best improved version of the Binary Cuckoo Search algorithm), on average the proposed EBGWO algorithm reduced the number of selected genes from 1043 to 339 (i.e. a further reduction of 704 genes) while improving the classification accuracy from 87% to 89% (i.e. an improvement by 2%).**

1.5.3 PSO-PCA-LGP-MCSVM: Particle Swarm Optimized Hybrid Kernel-Based Multiclass Support Vector Machine for Microarray Cancer Data Analysis

From the results presented in section 1.4.2, the proposed hybrid EACSIDGWO algorithm achieved an optimal balance between exploitation and exploration during the search thus overcoming EBGWO’s shortcoming. However, this wrapper adopted the SVM classifier (a commonly utilized classifier in DNA microarray based cancer classification) whose performance is largely dependent on the kernel adopted for it as well as tuning of the kernel parameters. Moreover, utilizing a single kernel function based MCSVM classifier in a given application such as gene expression data does not attain both a good learning ability, proper global feature extraction ability and a better generalization capability. Thus, to enhance both the learning and classification ability of the SVM classifier a particle swarm optimized hybrid kernel-based multi-class support vector machine i.e. PSO-PCA-LPG-MCSVM is presented in Chapter 5. In this model, particle swarm optimization (PSO) algorithm, principal component algorithm (a gene extractor) and multiclass support vector machine (MCSVM) that is based on a hybrid kernel i.e. linear-gaussian-polynomial (LGP) are combined. The major contribution of this work is the novel hybrid kernel i.e. LGP that combines the advantages of three standard kernels (linear, Gaussian and polynomial) in a novel manner; where the linear kernel is linearly combined with a Gaussian kernel that is embedding a polynomial kernel. Further, the validity of the proposed kernel is proved.

The effectiveness of the proposed model is revealed by carrying out a number of experiments and obtained results compared with those of three single kernel-based models i.e. PSO-PCA-L-MCSVM, PSO-PCA-G-MCSVM and PSO-PCA-P-MCSVM that utilize the standard alone linear, polynomial and Gaussian kernels respectively. Two dual and two multiclass imbalanced DNA microarray datasets that are publicly available were utilized. The obtained experimental results in terms of three extended evaluation metrics i.e. G-mean, F-score and accuracy reveal how superior the proposed model is in terms of global feature extraction, learning and prediction , compared to the other standalone kernel-based models.

To reveal the superior global gene extraction, prediction and learning ability of this model against three single kernel-based models: PSO-PCA-L-MCSVM (using a single Linear kernel), PSO-G-MCSVM (using a single Gaussian kernel) and PSO-P-MCSVM (using a single Polynomial kernel), four datasets: Colon cancer (2000 genes), Acute Lymphoblastic Leukemia-Acute myeloid Leukemia (ALL-AML) (7129 genes), St. Jude Leukemia dataset (12558 genes) and Lung cancer(3312 genes) were used. Adopting three extended evaluation metrics (G-mean, Accuracy (Acc) and F-score) the proposed model achieved the following: Colon Cancer (G-mean: 0.88, Acc: 0.88, F-score: 0.87), ALL-AML (G-mean: 0.94, Acc: 0.94, F-score: 0.94), Lung Cancer (G-mean: 0.99, Acc: 0.97, F-score: 0.96) and St. Jude Leukemia dataset (G-mean: 0.97, Acc: 0.96, F-score: 0.90). The PSO-G-MCSVM, the second best published model, attained the following: Colon Cancer (G-mean: 0.82, Acc: 0.82, F-score: 0.82), ALL-AML (G-mean: 0.94, Acc: 0.94, F-score: 0.94), Lung Cancer (G-mean: 0.98, Acc: 0.96, F-score: 0.93) and St. Jude Leukemia dataset (G-mean: 0.97, Acc: 0.95, F-score: 0.85). On average, the proposed PSO-PCA-LPG-MCSVM algorithm attained the following for the four datasets: G-mean: 0.95, Acc: 0.94 and F-score: 0.92. On the other hand, on average the PSO-G-MCSVM (second best published model) attained the following for the four datasets: G-mean: 0.93, Acc: 0.92 and F-score: 0.89. **Thus in comparison with PSO-G-MCSVM (the second best published model), on average the proposed PSO-PCA-LPG-MCSVM model improved both the G-mean and Acc by 0.02 (2%) and F-score by 0.03 (3%).**

1.6 Organization of Dissertation

Chapter One: Gives an introduction of the DNA microarray chips and the biological of DNA microarray data analysis is presented. The necessity of applying DNA microarray data analysis in both biomedical and clinical research with the aim of developing robust techniques for cancer disease diagnosis is articulated. The contribution of this research is also outlined in this section.

Chapter Two: A detailed literature review covering the various dimensionality reduction techniques developed for DNA microarray data analysis is presented. The motivation for the proposed research is also given. The research gaps identified in the literature are also discussed

Chapter Three: Presents a detailed account of the excited binary grey wolf optimizer and its application as an efficient feature selector in the highly dimensional DNA microarray datasets.

Chapter Four: Presents a detailed account of the excited-ACS-IDGWO algorithm and its application as a feature selector in various biomedical datasets.

Chapter Five: Presents a detailed account of the particle swarm optimized hybrid kernel-based multiclass support vector machine and its application in the DNA microarray data analysis.

Chapter Six: The significant contributions and the main findings of this work are highlighted in this chapter. Moreover, the scope for further work in this area of research is also presented.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

Due to the ever-increasing data dimensions experienced in the world today, many contexts such as medicine, machine learning and informatics are facing a number of challenges. However, dimensionality reduction can be regarded a basic technique to these feature-rich data, since by deriving and utilizing informative features only, processing of these data with existing tools will be facilitated.

DNA microarray chips are acquired from cells and tissues by taking into consideration the existing differences among the genes, which has proved to be useful in the diagnosis of diseases as well as tumors. However, due to a vast number of genes with a few samples existing in these DNA chips, selection of the most informative genes is still a difficult but important task[39].

Among the various existing machine learning approaches, feature/attribute selection as well as data classification are two essential tasks, which play a critical role in promoting human health; ranging from the detection of both voice and emotion to illness detection[40] [41].

In the medical field, effective informative genes selection can to a large extent enhance the prediction of the cancer disease as well as its diagnosis. After a successful gene selection, a specific classifier is utilized to discriminate healthy people from people suffering from cancer based on the expression levels of the selected genes.

So far, a number of researchers have proposed feature selection techniques to handle these feature-rich microarray chips.These techniques can be grouped into: filters, wrappers and hybrid techniques. Furthermore, reseachers have recently proposed new approaches like ensemble techniques to enhance both the process of gene selection as well as cancer disease classification.

In this chapter is a detailed account of the most utilized approaches that handle the DNA microarray datasets. The chapter starts by outlining a broad overview of the highly dimensional microarray data and attribute selection (Section 2.2 and Section 2.3). In section 2.4, a review of the state-of-art techniques falling within the filter category is given. In the three subsequent sections i.e. 2.5, 2.6 and 2.7 a description of the wrappers, hybrid and embedded techniques is presented. In each of these sections, several works reported on these approaches is also presented. In section 2.8, a review of the ensemble approaches recently reported in literature is outlined. Finally, in section 2.9 inferences drawn and the identification of the contribution of this research are presented.

2.2 Intrinsic features of highly dimensioned data

In this era, data is a critical resource in both industrial production and scientific research. Data is generated in various ways and at different costs. In fact, determining the most suitable and efficient data extraction technique is key to providing a solution. Each existing technique has its own merits and demerits. Additive noise and elevated costs are two demerits that these techniques may possess[38].

Moreover, since data quality as well its inherent features may affect the outcome of classification, it is important to fully understand the data under investigation. The common characteristics of highly dimensional data are presented in the subsequent subsections.

2.2.1 Many features

To recognize patterns, some quantitative features are derived from real world patterns and utilized in describing these patterns digitally. Though measurements of a vast number of these features can be derived from the existing patterns in the real world, only a few of them are documented taking into consideration their associated necessity, storage devices as well as the accessibility of resources required for extraction.

The main objective of feature/attribute selection is to derive a subset of informative features/attributes from all the recorded features. The selected features should enable researchers to fully describe and optimally classify the considered patterns.

Theoretically, a higher classification rate can be attained by using a vast number of attributes/features [42]. However, in practice, with the limited training samples, using a vast number of attributes can slow the learning phase as well as elevate the computational burden of this problem. This is because the presence of redundant or irrelevant attributes confuse the learning algorithm [42].

In this era, the number of attributes in the generated data has increased considerably. For instance, the prostate dataset; a DNA chip with only two classes has 10,509 genes. On the other hand, the 11_tumour dataset has 11 classes and 12,533 genes.

Many researchers have pointed out that despite the existence of vast number of attributes, a large portion of these attributes are either not relevant or are redundant to the classes under consideration. Thus, it is possible to attain an effective learning process by utilizing the only informative features.

Moreover, an excessively complex model results when a dataset contains a large number of features compared to samples. This in return leads to overfitting. In scenarios where it is not practical to increase the size of the training subset, it is key to reduce the number of features. This will considerably improve the classifier's overall performance.

Thus feature selection entails shrinking the dimensions of a dataset by identifying a subset of informative features (features required by the classifier) from the whole original feature set contained in the dataset [43]. The identified informative feature subset should optimally describe the dataset being processed. This implies that only redundant, irrelevant and noisy features are eliminated.

Generally, the possible number of informative attribute subsets will increase exponentially (2^N, whereby N is the number of attributes originally contained in the dataset) with the dimension of the dataset [44]. Thus, determining the optimal subset of informative attributes is normally a difficult task; a reason why the problem of feature selection is considered NP-hard [45].

2.2.2 Limited sample size

According to the research findings in [46], to accomplish a classification task with C classes and N features effectively, at least $10 \times N \times C$ samples are required for training. For example, 40000 training samples are needed for the DNA microarray colon dataset which has only two classes ($C = 2$) and 2000 genes ($N = 2000$). However, in total this dataset has only 62 observations. Thus, the limited number of observations (samples) is the most significant challenge concerning highly dimensional DNA chips [47].

2.2.3 Imbalance within dataset classes

Class imbalance normally occurs if in a given DNA chip, the number of available samples in a given class greatly supersedes the available sample of the other classes. In such a dataset, the class with the least sample size is termed the minority class. For a dataset with two classes, the negative class is termed as the majority class while positive class is deemed minority.

In most classification tasks, the classifiers assume that the training samples contain equal number of classes. Hence, when these classifiers are subjected to imbalanced data, they will be trained based on the majority class samples. This will consequently lead to a poor prediction performance of the minority class samples due to improper training [38].

The ratio of the total number of samples in the majority class of the dataset, Num_{major}, to the total number of samples in the minority class of the dataset, Num_{minor} is referred to as the ratio of imbalance and is defined by Equation 2.1 [46].

$$IR = \frac{Num_{major}}{Num_{minor}} \tag{2.1}$$

To identify the samples of the minority class is still a big challenge. The majority class normally have a greater influence and the improper classification of samples within the minority class leads to elevated risks [46].

For example, the sample size of the positive class in the cancer disease diagnosis is relatively smaller compared to that of the negative class. It is important to point out that the identification of the positive class samples is of great importance to researchers [47]–[50].Table 2.1 presents the imbalance ratio of ten standard cancer microarray datasets.

Table 2.1: Imbalance ratio of 10 standard cancer microarray datasets

DATASET	IMBALANCE RATIO
Brain_Tumors	15.00
Breast_Cancer	1.1
CNS	1.857
Colon	1.818
Leukemia	1.780
Prostate_Cancer	1.04
SRBCT	2.64
Gli	2.27
Lung_Cancer	23.17
Ovarian	1.78

In the literature, a number of techniques have been suggested to try and improve the classification rate of existing classifiers that are utilized in tackling the problem of class imbalance. The proposed approaches can be broadly categorized as follows [38], [46]:

a) Classifier-independent preprocessing techniques

In this category, two techniques are employed to rebalance the distribution of class sample sizes within the training set. The first technique is termed as *over-sampling* where researchers try adding more samples to the class with few samples. Another technique is *under-sampling* where researchers try to remove some samples from the class with a higher number of samples. It is evident that both techniques try to introduce some balance to the classes[46].

In [38], HTSS (an under-sampling approach) is proposed to derive suitable training subsets for datasets with imbalanced classes.

b) Algorithm modifications

Utilizing solutions that are specific, these approaches attempt to enhance the performance of classifiers with the aim of matching it with imbalanced datasets. In [51], an ensemble approach is proposed to improve the classifier performance when dealing with imbalanced datasets.

c) Ensemble learning techniques

These techniques utilize multiple classifiers' results. For instance, in [52] a technique is proposed where by imbalanced data is first rebalanced and then multiple classifiers are utilized for classification. Finally a combination of the results of the individual classifiers is done.

2.2.4 Label noise/Mislabling

The data retrieved from practical real-world applications is full of noise. This noise can arise due to the use of defective appliances for measurement or sometimes irregularities during the transmission. Noisy data can adversely affect classifier's performance. This implies that the classifier's performance largely depends on the quality of the training data [38].

Foremost, label noise or mislabeling can arise due to insufficient data for labelling [51], [53], [54]. An example of a significantly low quality training data is reported in [55], [56]. Normally, after data collection, an expert carries out the labelling. The labelling phase can be prone to human error since it is solely dependent on the opinion of the engaged expert; sometimes two experts can allocate different labels to the same sample.

Mislabling normally alters the number of samples within a given class. This challenge is common in the medical field. This is because determining the incident rate of a given disease in a given population is a great objective in the medical field. Moreover, due to the limited sample sizes experienced in medical research, a slight change in the number of observations will lead to biased measurements[38].

Feature selection based on ranking techniques are among the approaches that are negatively affected by label noise. This is because rankers can either overlook the significance of a given attribute or try to select an irrelevant attribute as appropriate.

In literature, a number of researchers have proposed various techniques to handle the label noise. The proposed approaches can be categorized as follows[57]:

a) Label noise robust approaches

These techniques attempt to tackle the noise label problem by reducing overfitting. Some examples of such techniques are the bagging and boosting approaches [58].

b) Data cleansing techniques

These techniques try to eliminate samples suspected to be mislabeled. Various techniques have been proposed to spot and filter the samples that have been mislabeled. For instance, the technique of detecting outliers and the isolation of classified data points are some of these techniques [38].

c) Label-noise tolerant learning techniques

These approaches cope with the noise through a modelling step. In [59], for example, a modification of the loss function is proposed to handle the noise.

2.2.5 The intrinsic features of DNA microarray datasets

A Gene expression profile is a fundamental concept in genetics. Genes are atomic genetic inheritance units within a genome. They hold information pertaining to the corporal features of an individual [30].

The DNA gene expression can either transfer a given property or reject it for an individual. In the field of bioinformatics, research on gene expression profile is key to effective prediction and diagnosis of a number of diseases like cancer.

The DNA microarray data is widely utilized for cancer detection. Taking into consideration the gene variations that may be useful for tumors and disease diagnosis, the microarray data is extracted from cells and tissues [13]. However, these DNA chips normally contain a vast number of genes and a limited sample size. These characteristics hinder an effective gene selection process from these data. Moreover, these data are prone to overfitting due to their inherent few samples. For instance, the DNA breast cancer dataset has 24,481 genes from just 60 samples[38].

The *curse of dimensionality* is another challenge attributed to the microarray datasets. This challenge arises because the extent of the vector of these genes is extremely large, which in most cases confuses the classifier during the learning process.

Like most of the highly dimensional datasets, the DNA microarray datasets have the class imbalance problem. As already mentioned, the standard classification algorithms assume balanced classes within a dataset. Consequently, these learning algorithms yield deceptive classification results when subjected to imbalanced datasets such as the DNA data. This is because the training being carried out will be skewed to the majority class, which will subsequently make the trained classifiers largely classify the samples of the minority class as samples contained in the majority class.

Considering the suggested machine learning approaches, attribute selection process is key to the successful classification of highly dimensioned datasets. For instance, in medicine the utilization of a proper feature selection process can to a large extent improve both the prediction as well as the diagnosis of the cancer disease[38]. After carrying out an efficient gene selection process, an identified classifier is employed to differentiate between healthy and unhealthy (cancerous) tissues using the selected profiles of gene expression. The importance of selecting genes comes with an extra effort of deriving an informative subset of genes which can be representative to the original DNA microarray dataset .

In bioinformatics, researchers are mainly involved in developing and testing optimal and efficient gene selection techniques with a minimal computational burden (complexity). An optimal gene selection technique not only derives a smaller set of representative genes from the whole DNA microarray data but also improves the results and performance of the subsequent classification stage [60].

2.3 The Feature selection process

Currently, feature selection has become a fundamental research area in the context of highly dimensional datasets. In the past, various feature selection techniques were proposed to handle the classical data. However, with the ever-increasing dimensions of datasets the existing feature selection approaches become inappropriate.

The traditional datasets with a few 10s of attributes are rapidly being replaced by big datasets with tens of 1000s of features. These highly dimensioned datasets are evident in bioinformatics, text processing and combinatorial chemistry.

Among the tens of thousands of features attributed to these datasets, a number of them are not relevant (unrelated) to the labels within the dataset classes. Thus, data preprocessing is a very crucial stage in attaining accurate and reliable classification when handling these big datasets[61]. Selecting informative features by eliminating those that are redundant and irrelevant is a difficult but important steps towards obtaining an appropriate classifier[57].

In bioinformatics, a number of recent studies have pointed out that majority of the measured genes within a given DNA chip experiment are normally not directly related to the classification validity of the classes of that dataset. To prevent this curse of dimensionality, it is important to remove genes that are both not relevant as well as redundant prior to the classification phase.

In this regard, feature selection has been termed as the most important preprocessing phase in medicine and bioinformatics. For instance, determining the risk factors of cancer deaths and optimally selecting informative features for diagonising the cancer disease has proved to be a major application of the selection of representative genes in the field of medicine. An informative feature subset is strongly correlated with the class labels while at the same time uncorrelated to other features[55].

In literature, various feature selection techniques have been reported to preprocess the highly dimensional datasets. Normally, the relation existing between the utilized function and the utilized classifier can be broadly categorized as follows: filters, wrappers, embedded techniques and the hybrid versions [59].

In the subsequent sections and subsections, a detailed presentation of the proposed methods is presented.

2.4 Filter Approaches

To select an informative feature subset, these approaches employ evaluation metrics that are based on independent and statistical methods. Without the utilization of data mining techniques, these approaches only utilize the inherent features of the data to select relevant and informative features. In other words, filters do not require a feedback from learning algorithms [38].

Figure 2.1 depicts the flowchart of a filter based feature selection process.

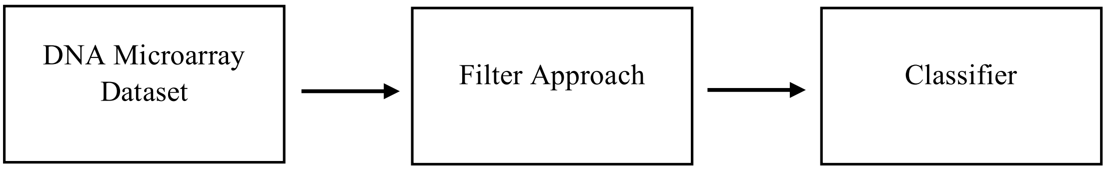


Figure 2.1: Filter based feature selection process

Filter techniques are fast; making them suitable for highly dimensioned datasets. However, since they are classifier independent the classification accuracy yielded by their informative features is low[62] [63].

A filter technique can either be univariate or *multivariate*. Univariate techniques use one evaluation metric in determining the relevance of a given attribute. Those attributes with the highest rank values become members of the informative subset. On the other hand, multivariate techniques utilize the relation among dataset features to select informative features [61].

In general, multivariate filter techniques are slower compared to their univariate counterparts. F-Score(FS) [64] and Information Gain (IG) [65] are two widely utilized univariate filter approaches. On the other hand ReliefF [66], mRmR[67] and FCBF[68] are the three mostly used multivariate filter approaches.

Moreover, the filters can also be categorized as follows: *statistical based, similarity based and those based on information theoretical.*

2.4.1 Approaches based on similarity

These approaches evaluate the significance of an attribute by considering its ability in maintaining the data similarity. However, most of these techniques are not able to tackle attribute redundancy because they normally determine the relevance of attributes on an individual basis[69].

For a supervised attribute selection process, similarity of samples can be obtained from data labels while the attribute selection that is not supervised, various distance metrics are used [69].

2.4.1.1 The Relief and extended relief (ReliefF) approaches

Relief [66], a similarity based technique, is widely applied to numerical and nominal attributes. This approach uses searches for attributes that are correlated to a given class. According to relief technique, an attribute is informative if it has a bigger difference among observations of different classes and a similar value among observations of the same class [66].

The relief approach starts by choosing a random observation and then uses the Euclidean distance to determine both the "*near miss*" and "*near hit*". The *near hit* are observations with minimal Euclidean distances among observations of the same class while the *near miss* are observations with minimal Euclidean distances among observations of different classes [70].

Initially weights of all features are set to zero but during each execution of the algorithm are updated using Equation 2.2[71].

$$Weight_i = Weight_{i-1} - \left(Feat_i - Near_{Hit_i}\right)^2 + \left(Feat_i - Near_{Miss_i}\right)^2 \qquad 2.2$$

The weight associated with each attribute will increase if its Euclidean distance from near observations within the same class is less in comparison with the Euclidean distance from near observations within a different class of the same dataset and vice versa.

m vectors of relevance are generated by m iterations of the relief technique using m random observations and dividing the $Weight$ component by m, which is the evaluation metric used to determine informative attributes. Thus, attributes whose vector of relevance is higher than the set threshold become members of the subset of informative features.

However, the relief approach cannot handle noisy and incomplete data. Moreover, this technique was formulated for two-class datasets and thus cannot handle multi-class datasets [71]. In trying to overcome these shortcomings, the reliefF which is an extension of the relief technique was formulated [66].

2.4.1.2 The Fisher's score technique

This approach selects a subset of attributes whose data points have very large distances in unrelated classes and smaller distances within the same class [72].

Consider an attribute A_i of an m-class dataset. If an observation set of attributes i is in the k^{th} class ${A_i}^k$ and $|{A_i}^k| = \gamma_k$. Where $k = 1,2,3 \ldots.,m$ and $\bar{A}_i^{\,k}$ and $\bar{A}_i$ are mean of the ${A_i}^k$ and A_i. Equation 2.3 then defines the F-score,F of a given attribute [72].

$$F(A_i) = \frac{\sum_{k=1}^{m} \gamma_k (\bar{A}_i^{\,k} - \bar{A}_i)^2}{\sum_{k=1}^{m} \sum_{x \in {A_i}^k} (x - \bar{A}_i^{\,k})^2} \qquad 2.3$$

The numerator in Equation 2.3 indicates the discrimination between two classes while the denominator depicts the scattering in each class. If the F-score of an attribute is higher, then its discrimination power will be higher. Finally, features that attain F-scores that are higher than the set threshold become members of the informative subset.

2.4.1.3 Laplacian score

This is an approach,which selects attributes that preserve the structure of the manifold [73]. To utilize this approach, the following three steps are required:

i) Construction of affinity matrix

The construction of the affinity matrix is conducted as follows:

$$S(a,b) = \begin{cases} exp^{\frac{\|c_a - c_b\|^2}{t}} & , if\ c_a\ is\ a\ member\ of\ the\ p \\ 0, & otherwise \end{cases} \\ - nearest\ neighbour\ of\ c_b \qquad 2.4$$

Where t is a desirable constant.

ii) diagonal matrix

Equation 2.5 depicts the definition of the diagonal matrix

$$D_m(a,b) = \sum_{j=1}^{n} S(a,b) \qquad 2.5$$

Then Equation 2.6 gives the expression for matrix L

$$Matrix_L = D - S \qquad 2.6$$

iii) Computation of the Laplacian score of each feature, $feat_i$

$$Laplacian_{Score(\,feat_i)} = \frac{\widetilde{feat_i}'.Matrix_L.\widetilde{feat_i}}{\widetilde{feat_i}'_{D\widetilde{feat_i}}} \qquad 2.7$$

Where $\widetilde{feat_i}$ is defined by Equation 2.8

$$\widetilde{feat_i} \; = feat_i - \frac{feat_i{}'D_1}{1'D_1}.1 \qquad 2.8$$

Since the Laplacian score is considered to be a ranking technique, the top k attributes with least scores become members of the informative subset.

2.4.2 Statistical techniques

These approaches utilize various statistical criteria to select informative attributes. Majority of these techniques are based on preset statistical criteria to remove uninformative attributes. Due to their low computational cost, they are suitable for pre-processing highly dimensional datasets. However, like similarity-based techniques they are unable to tackle redundant features [38].

2.4.2.1 Feature selection technique based on correlation (CFS)

CFS, a multivariate selection technique, was proposed in [74]. CFS evaluates attributes as per a correlation measure that is based on a given heuristic evaluation criterion that favors attributes with a higher correlation within a class.

The heuristic “merit” for an informative feature subset S containing d attributes is defined by Equation 2.9

$$CFS_{score}(\mathrm{S}) \; = \frac{d.\overline{r_{cf}}}{\sqrt{d + d(d-1)}\,.\overline{r_{ff}}} \qquad 2.9$$

Where $\overline{r_{ff}}$ is the average attribute-attribute correlation while $\overline{r_{cf}}$ is the mean attribute class correlation. To compute the $\overline{r_{cf}}$ and $\overline{r_{ff}}$, CFS utilizes symmetrical uncertainty [75].

2.4.2.2 Low Variance

This technique eliminates attributes whose variance is below a set threshold. With low variance, an attribute whose value is constant for all the observations, is deemed non-informative and its variance is normally zero[38].

2.4.2.3 The t-Score

This approach is determined by taking into consideration the mean, standard deviation and the sample values of attributes for each class as defined by Equation 2.10.

$$T_{score}(f_i) = \frac{|\alpha_i{}^{+} - \alpha_i{}^{-}|}{\sqrt{\frac{\beta_i{}^{+}(\sigma_i{}^{+})^2 + \beta_i{}^{-}(\sigma_i{}^{-})^2}{\beta_i{}^{+} + \beta_i{}^{-}}}} \qquad 2.10$$

Where β^{+} and β^{-} depict the sample size in the positive and negative classes respectively. $\alpha_i{}^{+}$ and $\alpha_i{}^{-}$ represent the means of the class labels. $\sigma_i{}^{+}$ is the positive class' standard deviation, while $\sigma_i{}^{-}$ is the negative class' standard deviation.

This technique is only applicable to two-class datasets i.e. binary classification. It is important to point out that this technique is categoized as a ranker; thus features with higher T-score values are conidered informative [76].

2.4.3 Information theoretical-based approaches

These approaches evaluate the significance of each attribute by using various heuristic filter techniques. They mainly maximize attribute relevance while minimizing attribute redundancy. Because most of these approaches are only applicable to discrete data, a discretization phase is required if data values are continuous [77].

Unlike statistical-based and similarity-based filter techniques, which were insufficient when handling datasets full of redundant features, information theoretical-based techniques have the ability to tackle the redundancy problem.

2.4.3.1 FCBF (Fast correlation-based filter)

FCBF, a multivariate based feature approach, incorporates mutual information to handle highly dimensioned datasets [69].FCBF employs the symmetrical uncertainty (SU) measure (refer to Equation 2.11 below) to detect redundant as well as irrelevant attributes, and in evaluating the correlation between attribute-attribute and attribute-class [68].

$$SU(H,I) = 2\left[\frac{IG(H,I)}{F(H) + F(I)}\right] \qquad 2.11$$

Where $F(H)$ and $F(I)$ depict the entropy of two attributes and $IG(H,I)$ represent the information gain.

Foremost, this technique picks a subset of attributes with a high correlation with class by the utilization of the SU. Then, after eliminating redundant attributes, relevant attributes to the class are retained.

2.4.3.1 mRMR (Minimum redundancy-maximum relevance) filter

This is another multivariate filter technique utilizing mutual information (MI) in evaluating the level of correlation existing between attribute-attribute and attribute-class. This technique retains features whose relevance is maximum with respect to a given class and their redundancy is minimum in regards to other features[67].

The score for an unselected attribute $feat_k$ is given by Equation 2.12.

$$Score_{mRMR}(feat_k) = I(feat_k; Y) - \frac{1}{|S|} \sum_{feat_j \in S} I(feat_k; feat_j) \qquad 2.12$$

Where attribute relevance is given by $I(feat_k; Y)$ and $I(feat_k; feat_j)$ depicts the mutual information between feature $feat_k$ and feature $feat_j$.

2.4.3.2 Information gain

This univariate technique utilizes information to evaluate attributes. The entropy concept, from information theory, is used to derive the information [38]. In other words, IG for an attribute $feat_i$ in set S_k is defined by Equation 2.13.

$$IG(S_k, feat_i) = H - \sum_{\mu = values(feat_i)}^{\frac{\left|S_{feat_i}=0\right|}{|S_k|}} H(S_{feat_i} = \mu) \qquad 2.13$$

Where $values(feat_i)$ is a set of values that can be allocated to $feat_i$.

Entropy $H(S_{feat_i} = \mu)$ is defined by Equation 2.14.

$$H(S) = -(\beta_+)log_2(\beta_+) - (\beta_-)log_2(\beta_-) \qquad 2.14$$

Where β_+ depicts the ratio of positive class observations to the dataset's total observations and β_- is the ratio of negative class observations to the sample size of the dataset.

After computing the IG of each attribute, attributes are sorted based on their ranks. Finally, features that meet a given set threshold are selected.

2.5 Wrapper Approaches

These techniques select desired attributes by utilizing both the results and performance of a classifier in evaluating the significance of the attribute subsets.

They utilize a search algorithm to derive the optimal subset of informative features from among the possible subsets. The random (stochastic) search and the greedy search are the commonly utilized search mechanisms [78].

The employed classifier evaluates each possible subset proposed by the utilized search technique. The accuracy rate is considered the fitness index of this subset of features.

The greedy search techniques are single-track based approaches that are highly prone to the local optima trapping. Both the sequential forward selection (SFS) technique as well as the sequential backward selection (SBS) approach are two main greedy search techniques. However, the stochastic search techniques select suitable features randomly.

The flowchart of attribute selection using wrappers is depicted by Figure 2.2.

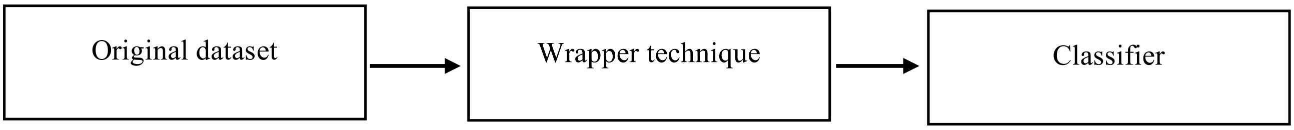


Figure 2.2: Flowchart of Wrapper based feature selector

Metaheuristics are the main stochastic search techniques. The commonly utilized metaheuristics in the selection of features include grey wolf optimization (GWO), particle swarm optimization (PSO), ant colony optimization (ACO), gravitational search algorithm (GSA) and genetic algorithm (GA).

Since wrappers utilize a classifier's accuracy in evaluation, they normally attain accuracy rates relatively higher compared to the filters [79].However, a number of reseachers have pointed out that wrappers are slow and have a high computational burden when subjected to highly dimensional datasets [80] [81].

In the subsequent subsections, a discussion of the aforementioned metaheuristics is presented.

2.5.1 ACO and ABACO$_H$

ACO is motivated by the traits exhibited by ants that are looking for food [82]. Though dumb and blind, ants are capable of determining a path whose distance is shorter from their nest to the source of food. This is possible because they can track the remaining pheromone by liasing with each other and sharing the information of the identified path. Thus, the intensity of the pheromone and its associated evaporation in the rarely used paths can enable these ants select the path with the least distance.

To utilize ACO as an attribute selector, the feature selection problem must be formulated graphically whereby nodes that depict the attributes used are on the graph. Foremost, the ants' initial position is randomly picked on this graph. Next, the subsequent node for each ant is computed by Equation 2.14 [83].

$$P_{ab}{}^{c}(t) = \begin{cases} \dfrac{\tau_{ab}{}^{\alpha}\,\eta_{ab}{}^{\beta}}{\sum_j \tau_{aj}{}^{\alpha}\,\eta_{aj}{}^{\beta}} & if\ a\ and\ b\ are\ nodes\ that\ are\ admissible \\ 0 \quad otherwise & \end{cases} \qquad 2.15$$

If the c^{th} ant is in position a during time t, it might be in position b at time $(t+1)$ with a probability $P_{ab}{}^{c}$. τ_{ab} is intensity of the pheromone at the edge between node a and node b. η_{ab} depicts the cost incurred in moving from a to b. β and α are parameters which govern the significance of the trace against vision.

The trace added to edge (ab) by the c^{th} ant is computed by Equation 2.16 [83]

$$\Delta\tau_{ab}{}^{c} = \begin{cases} \dfrac{Q}{F_c} & if c^{th}\ \text{ant traverses edge}\ (ab)\ \text{in}\ T_c \\ 0 \quad otherwise & \end{cases} \qquad 2.16$$

F_c denotes the cost of taking the path passed by the c^{th} ant and branch (ab) belongs to that path. T_c is the tour for the c^{th} ant. Thus, the trace all the ants add to edge (ab) is given by Equation 2.17 [83].

$$\Delta\tau_{ab}{}^{c} = \sum_{1}^{n} \Delta\tau_{ab}{}^{c} \qquad 2.17$$

The size of the swarm of ants is represented by n .

Taking into consideration all these relations, another pheromone intensity generated on the edges between a and b can be computed by Equation 2.18 [83].

$$\tau_{ab}(new) = (1-\gamma)\tau_{ab}(old) + \Delta\tau_{ab}{}^{c} \qquad 2.18$$

γ is an evaporation coefficient for the pheromone intensity that guards against excessive accumulation of the trace.

If a vast number of ants use a given path, the trace of that given path is incremented while if a limited number ants traverse a path, the trace will evaporate gradually[83].

In the recent past, a number of techniques based on the ACO have been proposed for the feature selection problem. The proposed binary version of ACO i.e. BACO has reported an attractive classification rate and relatively higher convergence speed.

However, it has been reported too that the BACO has a limitation when handling the feature selection problem. This is because, each ant located at a can only determine a subsequent attribute. In addition, if this ant is ignored or is unable to select this attribute, it cannot be able to investigate the same attribute in subsequent nodes [83].

To address these shortcomings, ABACO$_{\mathrm{H}}$ was proposed in 2013[83].This technique combines BACO and discrete ACO. In the ABACO$_{\mathrm{H}}$, $P_{ab}{}^{c}$ is redefined as depicted by Equation 2.19 [83].

$$P_{a\_x,b\_y}{}^{c} = \begin{cases} \dfrac{\tau_{a\_x,b\_y}{}^{\alpha}\,\eta_{a_x,b\_y}{}^{\beta}}{\sum_j \tau_{a\_x,j\_0}{}^{\alpha}\,\eta_{a\_xj\_0}{}^{\beta} + \tau_{a\_x,j\_1}{}^{\alpha}\,\eta_{a\_xj\_1}{}^{\beta}} & if\ j\epsilon admissible \\ 0 \quad otherwise & \end{cases} \qquad 2.19$$

Equation 2.19 gives a specification of the probability required to pick a bit whose value $y \in \{0,1\}$ within the subsequent point for the c^{th} ant during timestamp t and located at $x \in \{0,1\}$ of point a. Moreover, τ_{a_0,b_1} ,τ_{a_0,b_0}, τ_{a_1,b_1} and τ_{a_1,b_0} denote the intensity of pheromone available in the paths that connect nodes a and b on (0 to 1), (0 to 0), (1 to 1) and (1 to 0) edges[75].

The ABACO$_{\mathrm{H}}$ enables a given ant to search among all attributes thus resolving the major challenge of ACO.

2.5.2 PSO

The particle swarm optimization approach i.e. (PSO) [84] is motivated by the social traits exhibited by birds. It is one of the commonly adopted optimization algorithms due to its superior global search ability, relatively cheap computational complexity and few parameters to set.

With PSO, each possible outcome is a given particle within the selected swarm and whose position within the identified search space is depicted by a given vector x_j:

$$x_j = (x^1{}_j, x^2{}_j, \dots, x^d{}_j) \qquad 2.20$$

Where d depicts the extent of this search space.

The particles within the swarm traverses this search space in search of the best result and Equation 2.21 [84] represents their velocity.

$$v_j = (v^1{}_j, v^2{}_j, \dots, v^d{}_j) \qquad 2.21$$

Taking into consideration both the experiences of a given swarm particle and those of its neighbours, then the velocity and location of that particle are updated as follows [84]:

$$x^d{}_j(t+1) = x^d{}_j(t) + v^d{}_j(t+1) \qquad 2.22$$

$$v^d{}_j(t+1) = \omega * v^d{}_j(t) + \gamma_1 * rand_1\left(pbest^d{}_j - x^d{}_j(t)\right) + \gamma_2 * rand_2\left(gbest^d{}_j - x^d{}_j(t)\right) \qquad 2.23$$

Where t and d denotes the t^{th} generation and the d^{th} extent of this search space respectively. ω denotes the inertia weighting factor dedicated to controlling the influence of the previous velocity on the next velocity. γ_1 as well as γ_2 depict the speedup constraints. $rand_1$ and $rand_2$ are values randomly generated and whose distribution is uniform in [0,1]. $pbest^d{}_j$ represent the best outcome attained by particle j within the d^{th} dimension, and $gbest^d{}_j$ is the optimal result attained by this whole swarm in the d^{th} dimension.

The PSO algorithm stops when a desired outcome is attained, or the sum of generations reaches a predefined value.

2.5.3 IBGSA

The gravitational search algorithm (GSA) is motivated by gravity and mass [85].

According to the law proposed by Newton, every particle existing within the universe exerts a force on adjacent particles. This force is commensulate to product of the particles' masses and indirectly commensulate to the square root of their respective distances [86].

Recently, the GSA has gained attention because it has an attractive efficiency when tackling a number of optimization problems. Its binary version i.e. BGSA was suggested in 2010 [87].

To ensure that BGSA is not trapped in the local optimum while tackling the feature selection problem, IBGSA (an enhanced BGSA)was suggested in 2014[88].

For a system with n particles, the position of the j^{th} agent in the IBGSA is formulated in Equation 2.24 [87].

$$x_j = \left(x^1{}_j, x^2{}_j, x^r{}_j \dots, x^d{}_j\right) \quad j = 1,2,3,\dots,n \qquad 2.24$$

Where $x^r{}_j$ denotes the position of the dimension r belonging to mass j. d indicates the search space dimension.

After computing the fitness of the current population, the mass of each agent can be determined as per Equation 2.25 [87].

$$M_j(t) = \frac{fit_{val\_j}(t) - worst_{val}(t)}{\sum_{a=1}^{n} fit_{val\_j}(t) - worst_{val}(t)} \qquad 2.25$$

Where $M_j(t)$ and $fit_{val\_j}(t)$ denotes the mass and the fitness value of the j^{th} agent during time t. The $worst_{val}(t)$ is given by Equation 2.26 [87].

$$worst_{val}(t) = \max fit_{val_j}(t) \quad j \in \{1,2,3,\dots,n\} \qquad 2.26$$

Utilizing the gravity law, resultant forces exerted on the j^{th} agent by heavier agents are computed as per Equation 2.27 [87]

$$F^d{}_j(t) = \sum_{i \in kbest, i \neq j} rand_i G(t) \frac{M_j(t) M_i(t)}{R_{ji}(t) + \omega} (x^d{}_i(t) - x^d{}_j(t)) \qquad 2.27$$

$kbest$ comprises of k superior agents with better fitness values. The fitness function is a function of time starting at $k0$ and its value reduces with time.

The law defining the accelerating movement of the agent is computed using Equation 2.28 [87].

$$a^d{}_j(t) = \frac{F^d{}_j(t)}{M_j(t)} = \sum_{i \in kbest, i \neq j} rand_i G(t) \frac{M_i(t)}{R_{ji}(t) + \omega} (x^d{}_i(t) - x^d{}_j(t)) \qquad 2.28$$

Finally, Equation 2.29 [87] is used to update the speed of each agent.

$$v^d{}_j(t+1) = rand_j * v^d{}_j(t) + a^d{}_j(t) \qquad 2.29$$

$rand_j$ and $rand_i$ are numbers randomly generated and whose distribution is uniform in the span [0,1] and ω represents a small value. The hamming distance associated with agents j and i is represented by $R_{ji}(t)$ and is calculated as per Equation 2.30 [87].

$$R_{ji}(t) = \frac{1}{n} \sum_{d=1}^{n} \left| (x^d{}_i(t) - x^d{}_j(t)) \right| \qquad 2.30$$

$G(t)$ is a function of time termed as gravitational constant. Its initial value is $G(0)$ and it normally decays with time.

The agents' position varies in accordance with some probability i.e. the transfer function represented by Equation 2.31 [87]

$$T_F\left(v^d{}_j(t)\right) = B + (1 - B) * \left|\tanh\, v^d{}_j(t)\right| \qquad 2.31$$

In Equation 2.30, B is computed using Equation 2.32.

$$B = g_1(1 - e^{(\frac{F_c}{g2})}) \qquad 2.32$$

Where g_1 is constant and g_2 is time constant whose definition is dependent on the application of this algorithm. F_c denotes the failure counter. This failure is experienced when a given monitored result remains constant after a generation.

The agents traverse the search space according to Equation 2.33.

$$x^d{}_j(t+1) = \begin{cases} complement\ x^d{}_j(t) & if\ rand < T_F\left(v^d{}_j(t)\right) \\ x^d{}_j(t) & otherwise \end{cases} \quad 2.33$$

One of the notable differences between IBGSA and BGSA is the attractive elitism trait where by the position of an agent is altered only if then newer position possesses a fitter or an equal value of the previous location.

Equation 2.34 defines this elitism property.

$$M_j(t+1) = \begin{cases} M_j(t+1) & if\ fit\_val(M_j(t+1)) \leq fit\_val(M_j(t)) \\ M_j(t) & otherwise \end{cases} \quad 2.34$$

IBGSA algorithm is halted when a number of measures are taken into consideration.

2.5.4 BGWO and CBGWO

The grey wolf optimizer (GWO) is among the new memetic approaches formulated by Mirjalili [89]. This algorithm is motivated by social ranking and hunting traits portrayed by a pack of between 5 to 12 grey wolves.

With the GWO algorithm, the pack is categorized as follows: alpha (α) which is the overall leader of the pack, beta (β) which is the second leader in command, delta (δ) which is the third leader in command and the remaining wolves of the pack are termed as omega (ω). The α wolf is mainly involved in decision-making. The β wolf in most cases assists the α wolf in the decision-making process or other critical activities. The δ wolf is normally engagd in guiding the remaining ω wolves.

In formulating the GWO algorithm, the best three solutions attained are termed as the α, $\beta$and δ respectively, while the remaining solutions are regarded as ω. Moreover, the whole process of searching for the prey as well as hunting is advanced by the three leaders (α, β and δ) and the ω follow them.

The pack's encircling behavior in hunting a prey is expressed by Equation 2.35.

$$X(t+1) = X_p(t) - A.D \quad 2.35$$

Where X_p denotes the location of the prey while A is termed as a coefficient vector . The value D is determined using Equation 2.36.

$$D = \left|C.X_p(t) - X(t)\right| \quad 2.36$$

C is another coefficient vector . The location of the grey wolf is denoted by X and the number of generations is given by t.

A and C are determined using Equations 2.37 and 2.38 respectively.

$$A = 2.a.r_1 - a \quad 2.37$$

$$C = 2.r_2 \quad 2.38$$

The two independent random values i.e. r_1 and r_2 have a uniform distribution in the range [0, 1].

The encircling coefficient a, balances the trade-off between diversification and intensification. In this algorithm, the coefficient a linearly decays from value two to value 0 as per Equation 2.39.

$$a = 2(1 - \left(\frac{t}{T}\right)) \tag{2.39}$$

Where t is the current generation while T is the total number of generations.

The three leaders i.e. α, β and δ are deemed to be aware of the probable location of the prey. Thus, these leaders lead the ω wolves to the optimal solution.

Equation 2.40 expresses the mathematical formulation of the new position of the wolf.

$$X(t+1) = (\frac{X_1 + X_2 + X_3}{3}) \tag{2.40}$$

Where X_1, X_2 and X_3 are formulated in Equations 2.41, 2.42 and 2.43

$$X_1 = |X_\alpha - A_1.D_\alpha| \tag{2.41}$$

$$X_2 = |X_\beta - A_2.D_\beta| \tag{2.42}$$

$$X_3 = |X_\delta - A_3.D_\delta| \tag{2.43}$$

Where X_α, X_β and X_δ are the position of the α, β and δ wolves during iteration $t.$ A_1, A_2 and A_3 are computed as per Equation 2.35 ,and D_α, D_β and D_δ are expressed by Equations 2.44, 2.45 and 2.46.

$$D_\alpha = |C_1.X_\alpha - X| \tag{2.44}$$

$$D_\beta = |C_2.X_\beta - X| \tag{2.45}$$

$$D_\delta = |C_3.X_\delta - X| \tag{2.46}$$

The three coefficients C_1, C_2 and C_3 are determined using Equation 2.38.

To tackle optimization problems such as feature selection which are binary in nature, Emary et al [90] formulated 2 binary versions of GWO i.e. the first one BGWO1 and the second BGWO2. Generally, BGWO is simple, flexible, has few parameters that require setting. Moreover, in comparison to other binary optimization approaches it is more adaptable to various problems.

However, it has a major shortcoming in that the wolves in most cases get held-up in the local optimum. This is largely attributed to the tendency of all the ω wolves trying to advance towards the locations of the α, β and δ leaders. This normally leads to insufficient diversity and hinders mature convergence [91].

In trying to overcome these challenges, in 2018 Jingwei et al [91] suggested a new competitive BGWO i.e. CBGWO. The main idea of CBGWO is motivated by the concept of competiveness nature among couples within the pack of wolves. To implement this competition concept, a random pairwise selection of wolves from the pack is conducted. For instance, a pack containing N wolves, will be randomly divided into $^N/_2$ wolves. Next, a competition between the two wolves in each of the derived couples is carried out. This implies that each wolf will participate once in this competition. The competition will yield two categories of wolves i.e. *winners* and l*osers*. *Winners* are those wolves with better values in comparison to their counter parts in each couple. On the contrary, the l*osers* have worse values in comparison to their counter parts in their respective couples.

These *winners* automatically become candidates of the subsequent generation without any position update. However, the respective positions of losers are updated by learning from their counterpart winners. Consequently, only $^N/_2$ wolves in the pack that will be updated.

In the CBGWO, the modified Equations 2.42-2.44 are expressed in Equations 2.47-2.49.

$$\bar{D}_\alpha = |C_1.X_\alpha - (X_w - X_l)| \tag{2.47}$$

$$\bar{D}_\beta = |C_1.X_\beta - (X_w - X_l)| \tag{2.48}$$

$$\bar{D}_\delta = |C_1.X_\delta - (X_w - X_l)| \tag{2.49}$$

Where X_w is the wolf deemed to be the winner *while* X_l is the loser.

From Equations 2.47-2.49 the positions of wolves deemed to be *losers* are updated by learning from their respective *winners*. Consequently, losers not only take guidance from the overall leaders of the pack i.e. α, β and δ but also from the couple winners in their decision to move towards the prey.Thus, the CBGWO approach exhibits a better search within the search space.

Leader enhancement

The three leaders (α, β and δ) play a critical role in the CBGWO. They guide the rest of the pack in search for the prey. To ensure that the CBGWO is not stuck in the local optimum like the BGWO, the α, β and δ wolves update their positions as per the enhancement strategy depicted by Equation 2.50.

$$L^d = \begin{cases} rand(0,1), if\ \theta \geq rand \\ \quad X^d{}_L, otherwise \end{cases} \tag{2.50}$$

Where θ is the change rate, $rand(0,1)$ is a number randomly generated which is either a 0 or 1, $rand$ is a number randomly generated and is normally distributed uniformly in the range [0,1] and X_L is the leader which is either α, β or δ.

In the CBGWO, the θ linearly decreases from 0.9 to 0 as expressed by Equation 2.51.

$$\theta = 0.9 - 0.9(\frac{t}{T}) \tag{2.51}$$

t is the current generation while T is the total number of generations.

From Equation 2.51 a bigger θ at the start of the search process will facilitate adequate changes within the positions thus enhancing diversification. However during higher iteration values, when θ is small, exploitation is enhanced.

Since there are only three leaders i.e. α, β and δ in the CBGWO, implying only three wolves get updated during each iteration using Equation 2.50. This approach tries to maintain a relatively low computational cost.

Finally, if the newly generated leader is established to be fitter than the current leader, the current leader is replaced. Otherwise, the current leader is immediately transfered to the next iteration.

2.6 Hybrid techniques

These approaches combine filters and wrappers. Foremost, the size of the original feature set is reduced by a filter technique, after which a wrapper is employed on this reduced feature set. The accuracy achieved by the hybrid approaches is higher than that of filters [92]. In addition, these techniques have a higher speed with lower computational complexity in comparison to wrappers; thus making them suitable candidates for selecting informative features in highly dimensioned datasets[93].

Figure 2.3 depicts the flowchart of attribute selection using the hybrid techniques.

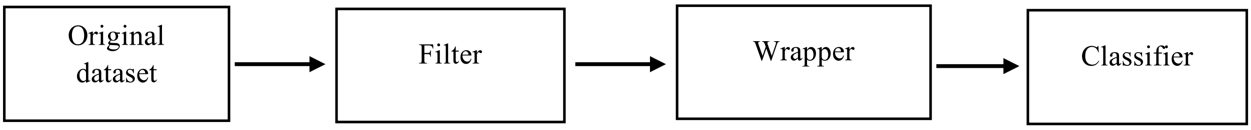


Figure 2.3: Common flowchart in Hybrid feature selectors

An approach hybridizing SVM-RFE with mRMR is proposed by authors in [94]. The results of this technique proved superior to SVM-RFE, mRMR and a couple of techniques they used. In 2011, another hybrid approach for text categorization was proposed [95]. This approach combined the information gain (IG) and genetic algorithm (GA).

Chuang et al [96] hybridized CFS with the novel TGA i.e, Taguchi-genetic algorithm for selecting features using 11 DNA microarray chips. The performance of the proposed CFS-TGA algorithm was attractive in terms of computational complexity and classification rate.

In [97], a hybrid of GA adopting dynamic setting of parameters i.e. GADP and χ^2 as a feature selector is proposed. Foremost, the GADP is employed to generate various feature subsets and then χ^2 is employed to select the final features. This approach was used for gene selection in DNA microarray datasets.

Shreem et al [98] hybridized two filters i.e. ReliefF and mRMR with the genetic algorithm (GA). They termed the technique as R-m-GA. In 2015, authors in [99] hybridized the information gain (IG) with a binary version of differential algorithm (DE).

In 2019, authors in [100] combined MI i.e. mutual information filter approach and RFE i.e. recursive feature elimination approach for feature selection of three standard datasets obtained from the UCI dataset repository.

A hybrid approach termed as FSCBAS was proposed in [101]. This technique combined the clustering approach with a modified binary ant system (BAS).

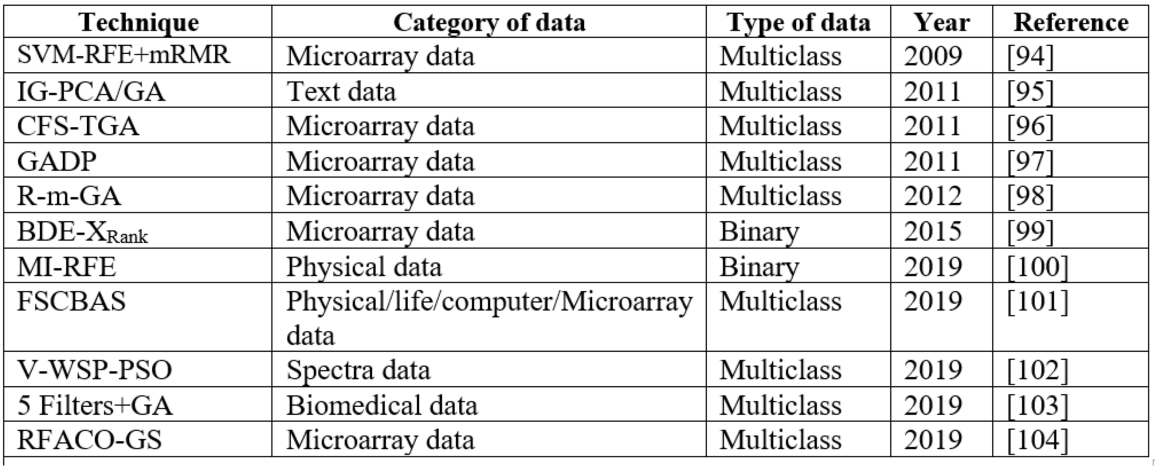
Another hybrid approach was proposed in [102]. Foremost, a filter approach based on V-WSP is used to derive the top attributes. Then PSO wrapper approach is utilized in the selection of the final informative features.

In [103], a new hybrid approach combining five filters i.e. mRMR, IG, CFS, corrFeatureEval and oneRFeatureEval with genetic algorithm is proposed for selecting informative feature using three biomedical datasets.

A memetic comprising of the relief technique and ACO i.e. ant colony optimization wrapper is proposed by authors in [104]. The proposed approach was used as a feature selector in DNA chip data as well as classifying tumor data.

A summary of the described hybrid approaches along with the category of data, the type of data used and the original reference is presented in Table 2.2.

Table 2.2: Hybrid techniques for feature selection in highly dimensional datasets



Technique	Category of data	Type of data	Year	Reference
SVM-RFE+mRMR	Microarray data	Multiclass	2009	[94]
IG-PCA/GA	Text data	Multiclass	2011	[95]
CFS-TGA	Microarray data	Multiclass	2011	[96]
GADP	Microarray data	Multiclass	2011	[97]
R-m-GA	Microarray data	Multiclass	2012	[98]
BDE-X_{Rank}	Microarray data	Binary	2015	[99]
MI-RFE	Physical data	Binary	2019	[100]
FSCBAS	Physical/life/computer/Microarray data	Multiclass	2019	[101]
V-WSP-PSO	Spectra data	Multiclass	2019	[102]
5 Filters+GA	Biomedical data	Multiclass	2019	[103]
RFACO-GS	Microarray data	Multiclass	2019	[104]

2.7 Embedded techniques

With these approaches, the process of selecting features is embedded within the machine learning technique [105], i.e. the learning phase and the feature selection phase are 2 inseparable processes. These approaches are faster compared to wrappers. However, their associated computational burden is larger than those of filters but lower than those of wrappers [106].

In [107], an embedded based feature selector is proposed for highly dimensional cancer datasets. Though it achieved an attractive performance, it requires repeated training of the SVM i.e. support vector machine.

In 2010, a kernel-penalized support vector machine (KP-SVM) was proposed by [108].This approach carries out feature selection by penalizing the use attribute in the SVM’s dual formula.

In the year 2012, an iterative feature perturbation (IFP) technique was proposed for feature selection in [109]. The IFP adopts backward elimination and a given metric to determine non-informative features. It also takes into consideration the effect of every attribute on the performance of a classifier in a noisy environment.

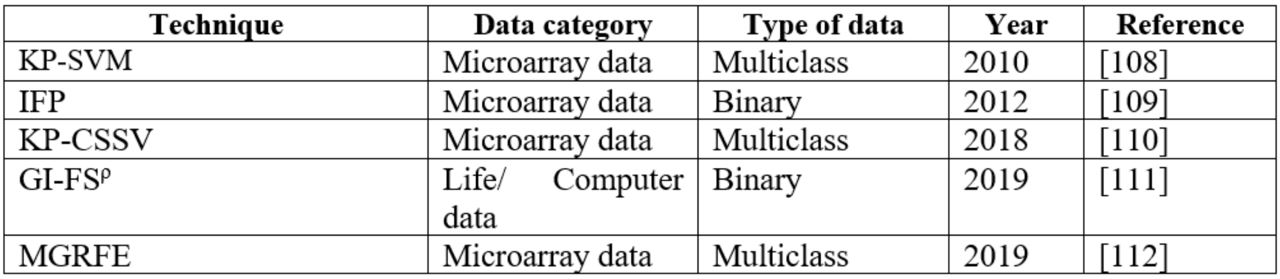
In 2018, authors in [110] proposed KP-CSSV feature selector. This approach is inspired by the kernel-penalized support vector machine (KP-SVM) to tackle the challenge of class imbalance in DNA microarray datasets.

In the year 2019, an embedded based feature selector was proposed in [111]. This technique features the weighted Gini index technique to handle the class imbalance challenge in classifications tasks.

A MGRFE feature selection technique was proposed in [112]. This technique is based on a novel embedded integer-coded GA technique to derive informative genes in DNA chip data.

A summary of the embedded based feature selection techniques discussed above is presented in Table 2.3.

Table 2.3: Embedded techniques for selection of features in highly dimensional datasets



Technique	Data category	Type of data	Year	Reference
KP-SVM	Microarray data	Multiclass	2010	[108]
IFP	Microarray data	Binary	2012	[109]
KP-CSSV	Microarray data	Multiclass	2018	[110]
GI-FS$^{\rho}$	Life/ Computer data	Binary	2019	[111]
MGRFE	Microarray data	Multiclass	2019	[112]

2.8 Ensemble techniques

Highly dimensional data may not only contain an immense number of data as well as attributes, but also face challenges such as feature redundancy, nonlinearities and noise. Due to this, one technique that achieves superior performance in one dataset cannot be deemed efficient in all highly dimensioned datasets. A number of techniques need to work together [38].

Thus, researchers have also been attracted towards developing ensemble feature selection/classification techniques. By utilizing ensemble approaches, chances of settling on the wrong solution are minimized and learning techniques that get trapped in the local minima can achieve better approximations[113].

In ensemble approaches, instead of taking into consideration the outcomes of a single approach as final, a number of approaches are applied to these data and then their results are combined.

Figures 2.4 and 2.5 depict the commonly utilized ensemble frameworks in feature selection for highly dimensional datasets.

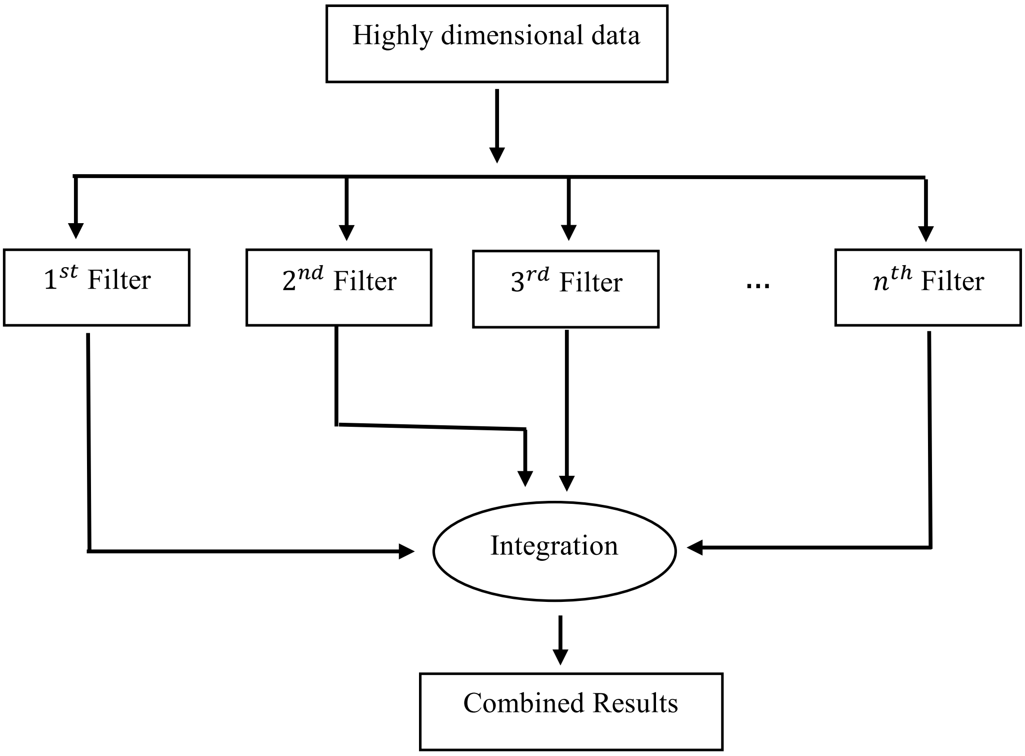


Figure 2.4: Instance one of ensemble approaches (feature selection)

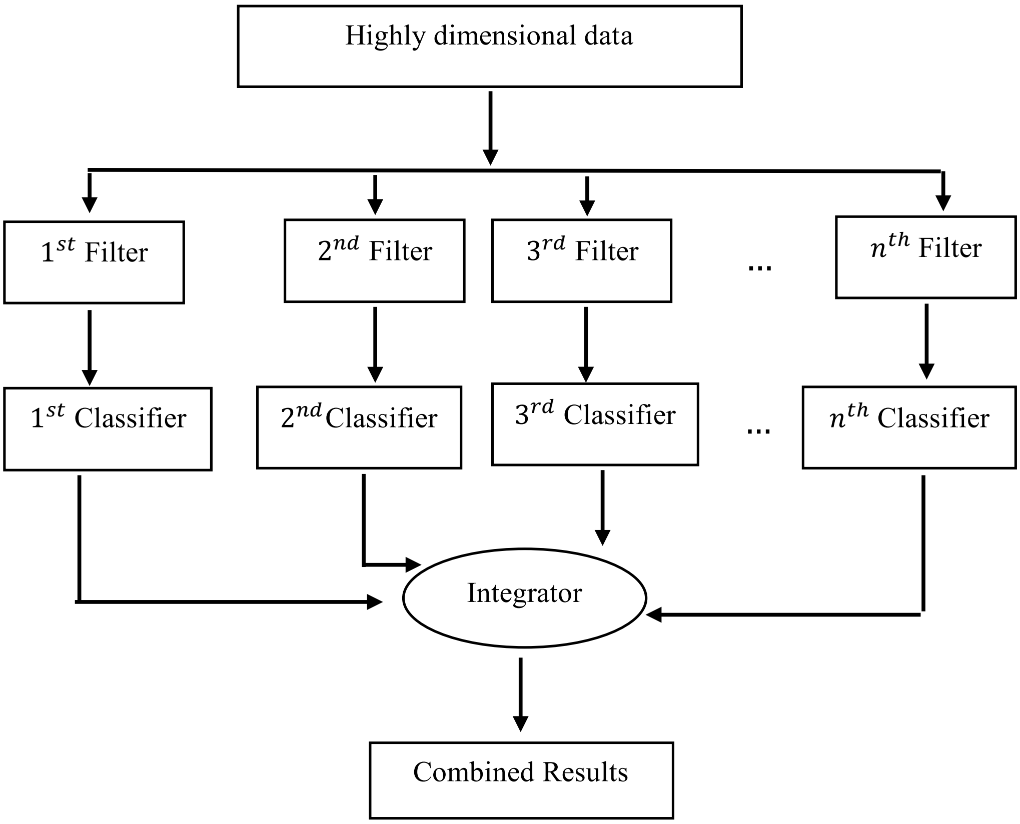


Figure 2.5: Instance two of ensemble approaches (classification)

In Figure 2.4, the results of a number of filter techniques on highly dimensional datasets are integrated together in various ways to derive the final informative feature subset. This approach requires an integration criterion to merge the features selected by every filter approach considered.

In Figure 2.5, a number of filters are applied independently to the highly dimensional datasets. Then the feature subset selected by each filter is fed to a classifier. Eventually, after the classification phase, an integrator is applied to combine the outcomes of every classifier considered[114].

Yang et al [115] formulated an ensemble approach termed as MCF-RFE i.e. multi-criteria fusion (MCF) combined with recursive feature elimination (RFE) to tackle the microarray attribute selection challenge. The motivation of the approach comes from the combination of RFE search technique and various principles.

A filter ensemble combined with another classifier ensemble was proposed by Bolon et al [116]. The filter ensemble comprises of several filters such as INTERACT, information gain (IG) and CFS etc.

In [71], a hybrid-ensemble technique to select informative genes in DNA microarray chips is proposed. Foremost, three filters i.e. IG, F-score and relief are employed to select gene subsets individually. Then these subsets are combined prior to feeding the result to IBGSA for the final informative gene selection.

In 2014, authors in [114], proposed four ensemble approaches i.e E1-nk, E1-ns, E1-cp and E2. These approaches are attractive because they incorporate a number of responses from filter techniques.

In [113], an ensemble approach comprising of IG, reliefF and CFS is proposed. This approach achieved a desirable classification accuracy in comparison to two other ensemble approaches.

A hybrid ensemble approach termed as HM-ABACO$_H$ is proposed in [70].Foremost, the results of reliefF, IG and FCBF are integrated before feeding the combined results to the ABACO$_H$, a wrapper for the final selection of features. The performance of this approach was evaluated on 7 DNA microarray chips.

In [114], four filters i.e. Cons, relief, CFS and IG are combined with various classifiers using ensemble approaches. To evaluate their performance evaluation, a number of DNA microarray datasets are used.

In 2017, a new type of hybrid-ensemble approach was proposed by [59]. In this approach, FCBF technique (a filter approach) was used at the initial stage to reduce the dimension of highly dimensional dataset. Then two wrappers i.e. ABACO and IBGSA are independently applied for further reduction of the selected features. Finally, the features selected by these two wrappers are combined to derive the informative feature subset.

Authors in [79] proposed another hybrid-ensemble framework whereby each filter approach generates its selected features. Then each of the reported attribute subset is provided to a number of wrappers for further feature reduction. Finally, the outputs of the wrappers are combined to derive the informative feature subset. Figure 2.6 depicts the flowchart of this model.

In 2018, an ensemble approach utilizing the t-test as well as nested GA technique was proposed to select attributes in highly dimensioned datasets [117].

In 2019, authors of [118] having examined various feature selection approaches, concluded that ensemble approaches are more robust in comparison to single approaches in feature selection of highly dimensional datasets.

Another ensemble-based technique utilizing bits from the k-mean approaches was proposed by [119]. This approach is termed as the feature co-association ensemble (FCE) and was used to select informative attributes from the UCI repository datasets.

In [120], an ensemble approach combining 3 attribute selection techniques i.e. XGBoost, chi-square and maximum information coefficient is proposed for attribute selection in two-class highly dimensioned datasets.

Authors of [121] proposed an ensemble feature selection approach that combines four filters to identify the robust risk factors for the diabetic kidney disease (DKD). This was achieved by striking a balance between the predictability and the stability of the system.

A summary of the ensemble techniques utilized as feature selectors in highly dimensional datasets is presented in Table 2.4

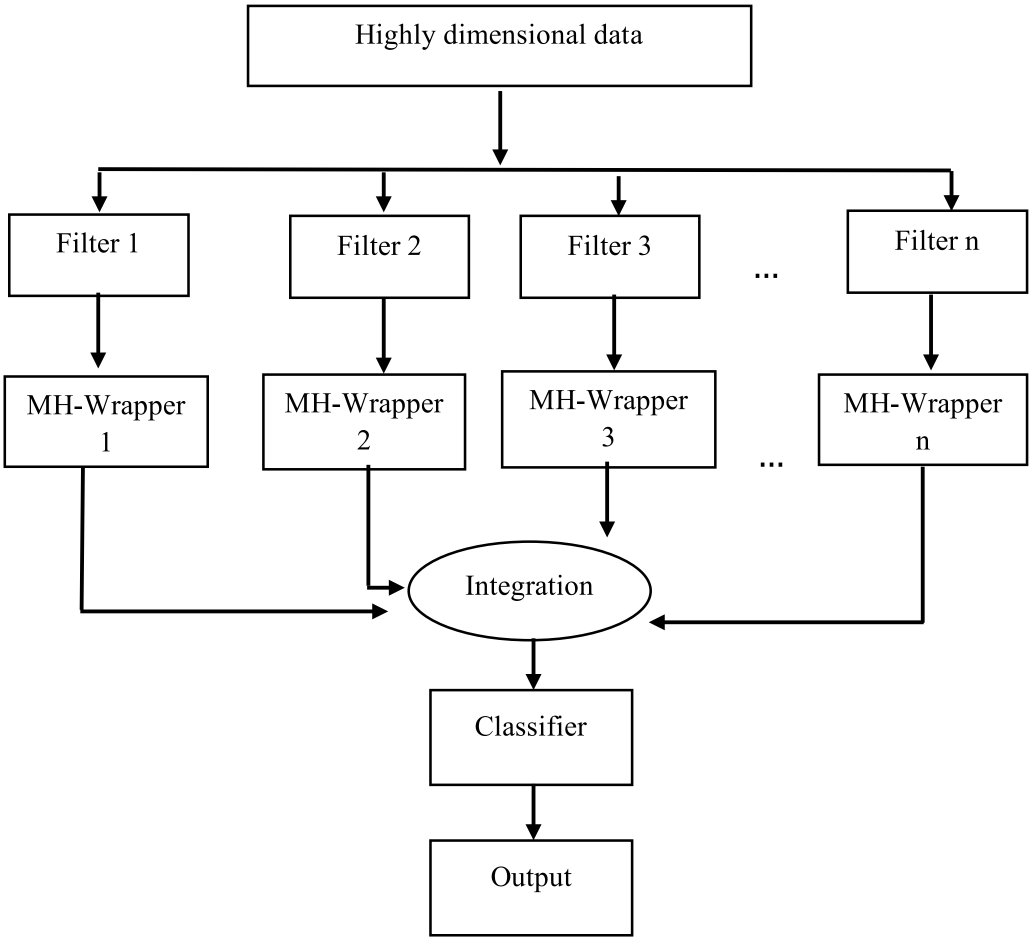
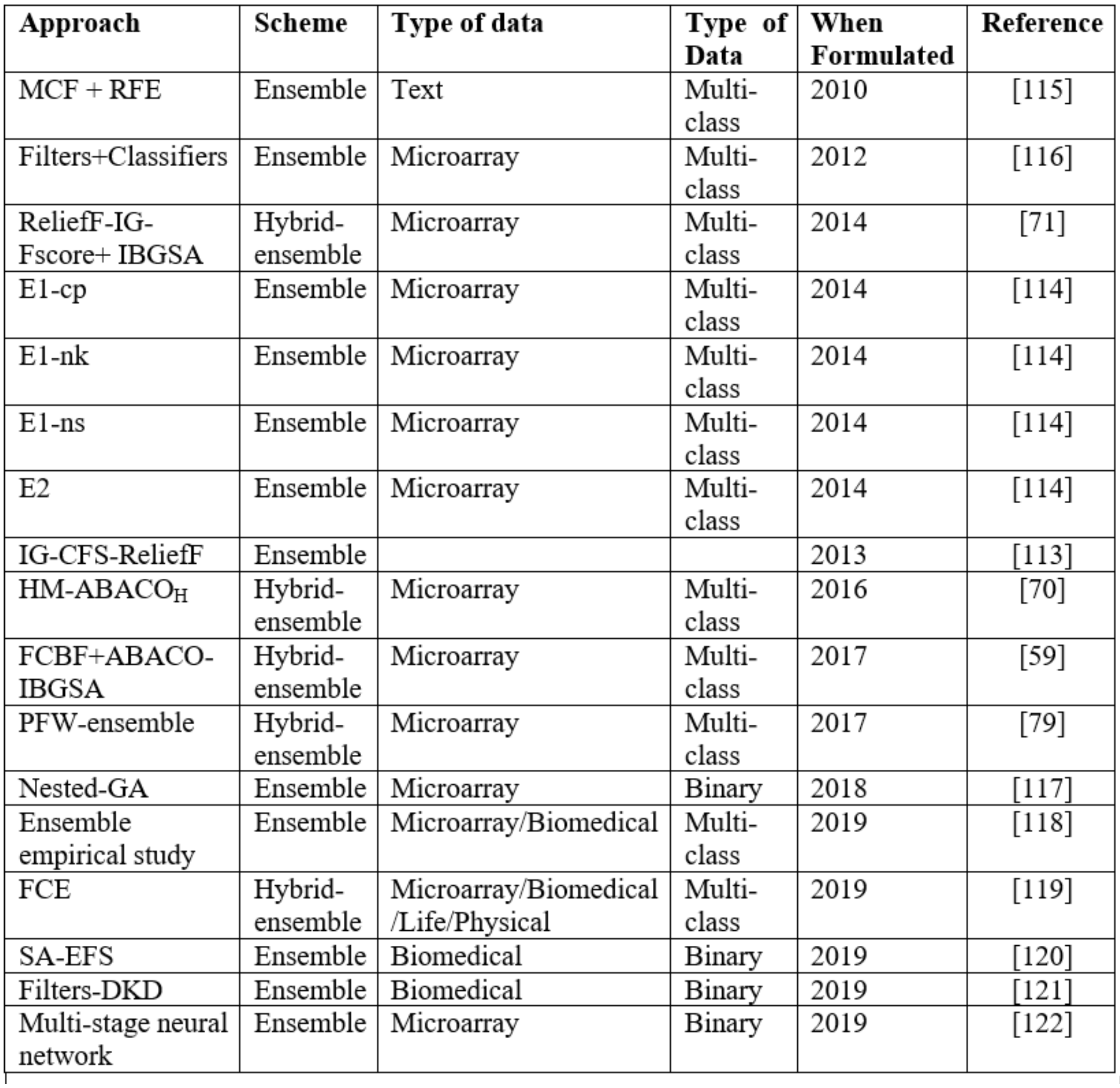


Figure 2.6: Framework of hybrid-ensemble technique

Table 2.4: Ensemble-based techniques used for attribute selection in highly dimensional datasets



Approach	Scheme	Type of data	Type of Data	When Formulated	Reference
MCF + RFE	Ensemble	Text	Multi-class	2010	[115]
Filters+Classifiers	Ensemble	Microarray	Multi-class	2012	[116]
ReliefF-IG-Fscore+ IBGSA	Hybrid-ensemble	Microarray	Multi-class	2014	[71]
E1-cp	Ensemble	Microarray	Multi-class	2014	[114]
E1-nk	Ensemble	Microarray	Multi-class	2014	[114]
E1-ns	Ensemble	Microarray	Multi-class	2014	[114]
E2	Ensemble	Microarray	Multi-class	2014	[114]
IG-CFS-ReliefF	Ensemble			2013	[113]
HM-ABACO$_H$	Hybrid-ensemble	Microarray	Multi-class	2016	[70]
FCBF+ABACO-IBGSA	Hybrid-ensemble	Microarray	Multi-class	2017	[59]
PFW-ensemble	Hybrid-ensemble	Microarray	Multi-class	2017	[79]
Nested-GA	Ensemble	Microarray	Binary	2018	[117]
Ensemble empirical study	Ensemble	Microarray/Biomedical	Multi-class	2019	[118]
FCE	Hybrid-ensemble	Microarray/Biomedical /Life/Physical	Multi-class	2019	[119]
SA-EFS	Ensemble	Biomedical	Binary	2019	[120]
Filters-DKD	Ensemble	Biomedical	Binary	2019	[121]
Multi-stage neural network	Ensemble	Microarray	Binary	2019	[122]

2.9 Analysis and discussion

2.9.1 Commonly utilized DNA microarray datasets

Table 2.5 represents ten commonly utilized DNA microarray datasets for feature selection tasks. From the table, it is evident that all the datasets are richly endowed with genes but their sample sizes is limited. Moreover, the Colon dataset has the least number of genes with sixty-two observations. On the other hand, the Breast Cancer dataset has largest number of genes with 97 samples.

Table 2.5: Commonly utilized DNA chips for benchmarking

Name of Chip	Number of genes	Sample size	Number of classes
Brain Tumours1	5920	90	5
CNS	7129	60	2
Breast cancer	24481	97	2
Colon	2000	62	2
Prostate	10509	102	2
Leukemia	7129	72	2
Prostate cancer	12600	21	2
Lung cancer	12533	181	2
SRBCT	2308	83	4
Ovarian	15154	253	2

2.9.2 Commonly utilized criteria for performance evaluation

The seven commonly utilized evaluation criteria in feature selection for highly dimensional datasets are discussed below.

Correct classification rate(CCR): is the ratio of test observations correctly classified to the total number of available test observations. It is expressed by Equation 2.52.

$$CCR = \frac{\#\ of\ correctly\ classified\ test\ samples}{\#\ of\ available\ test\ samples} \qquad 2.52$$

The higher the CCR, the more significance the selected feature subset is in the CCR; thus the subset will be considered as informative subset.

Sensitivity and *specificity* are two criteria used in evaluating the performance of binomial (two-class) classifications. Taking into consideration a two-class dataset whose classes are labelled as positive and negative, then *TP,FP, TN* and *FN* can be defined as follows:

TP- test observations classified correctly as positive.

FP- test observations incorrectly classified as positive.

TN- test observations correctly classified as negative.

FN- test observations incorrectly classified as negative.

Utilizing *TP, FP, TN* and *FN,* then geometric mean, specificity, sensitivity and Mathew's correlation coefficient can be defined as follows:

$$Sensitivity = \frac{TP}{TP + TN} \qquad 2.53$$

$$Specificity = \frac{TP}{TP + TN} \qquad 2.54$$

$$Geometric\ mean = \sqrt{Sensitivity \times Specificity} \tag{2.55}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{2.56}$$

The feature reduction rate (F_r) is another commonly utilized evaluation criterion defined by Equation 2.57.

$$F_r = \frac{Num_{Features} - Num_{Selected\_Features}}{Num_{Features}} \tag{2.57}$$

Where $Num_{Features}$ is the total sum of attributes in a dataset and $Num_{Selected\_Features}$ is the sum of selected attributes. In reference to Equation 2.55, the closer F_r is to one, the more suitable is the corresponding feature selection algorithm for that dataset. Moreover, the bigger the value of F_r, the lesser the computational burden of the corresponding feature selection algorithm.

A number of authors have pointed out since the F_r alone cannot clearly portray either the strength or weakness of a given feature selection technique, an alternative criterion is required. Thus, the geometric mean utilizing ACC and F_r was formulated and is expressed in Equation 2.58.

$$GM = \sqrt{ACC \times F_r} \tag{2.58}$$

2.9.3 Normalization of data

Normalization of data is a preprocessing stage where data values within a given dataset are assigned values in the interval [0, 1]. With this technique; all attributes within the dataset are assigned one weight when computing the distance existing between datasets.

2.9.4 Analysis of filter techniques

Tables 2.6 and 2.7 depict the average classification accuracy over 10 different runs of nine filters on eight commonly utilized microarray datasets. Table 2.6 show the results when the SVM classifier is utilized while Table 2.7 present the results when the KNN classifier is utilized.

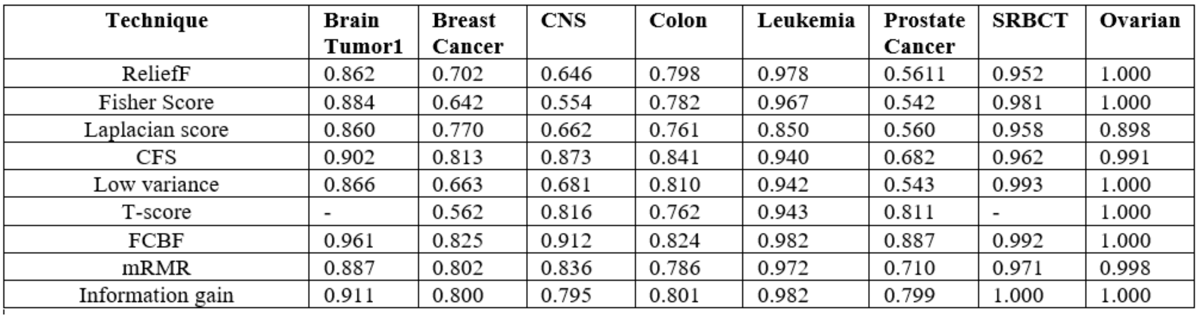
From Tables 2.6 and 2.7, it is evident that Prostate microarray dataset is the most challenging dataset in comparison to the other seven datasets. This is because this dataset has a vast number of genes (i.e. 12600) and a few samples (i.e. 21). Moreover, the test data for this dataset was derived from a number of different datasets making it to have a shift problem . However, from the two Tables (i.e. 2.6 and 2.7), the FCBF and the T-score filters achieved acceptable results on this dataset.

It is evident that the SVM classifier achieved higher classification accuracies in comparison to the KNN counterpart. From the two Tables, the FCBF filter (a theoretical-based filter technique), achieved higher classification accuracies in comparison to the other seven filter approaches.

It is important to point out the filters utilized in Tables 2.6 and 2.7 are rankers except the FCBF and the CFS, which assign a value to each attribute that represent the importance of the attribute to the classes of the dataset. Thus for all the considered filters, top 100 features were selected as informative features. The considered filter approaches might achieve better classification accuracies if this threshold value (i.e. 100) is changed. Generally, determining the optimal threshold value for effective feature selection is still a challenge for ranking based feature selection.

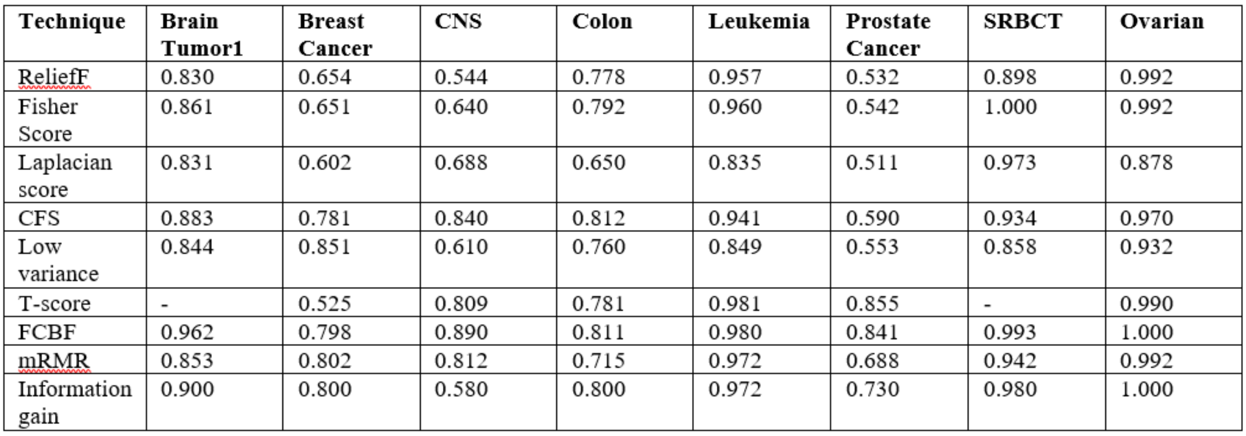
Thus, achieving optimal filter-based feature selection solely depends on filter technique adopted, the threshold value set for the ranking techniques and the classifier algorithm utilized.

Table 2.6: Experimental results for nine filters combine with the SVM classifier (10-fold cross validation)



Technique	Brain Tumor1	Breast Cancer	CNS	Colon	Leukemia	Prostate Cancer	SRBCT	Ovarian
ReliefF	0.862	0.702	0.646	0.798	0.978	0.5611	0.952	1.000
Fisher Score	0.884	0.642	0.554	0.782	0.967	0.542	0.981	1.000
Laplacian score	0.860	0.770	0.662	0.761	0.850	0.560	0.958	0.898
CFS	0.902	0.813	0.873	0.841	0.940	0.682	0.962	0.991
Low variance	0.866	0.663	0.681	0.810	0.942	0.543	0.993	1.000
T-score	-	0.562	0.816	0.762	0.943	0.811	-	1.000
FCBF	0.961	0.825	0.912	0.824	0.982	0.887	0.992	1.000
mRMR	0.887	0.802	0.836	0.786	0.972	0.710	0.971	0.998
Information gain	0.911	0.800	0.795	0.801	0.982	0.799	1.000	1.000

Table 2.7: Experimental results for nine filters combine with the KNN classifier (10-fold cross validation)



Technique	Brain Tumor1	Breast Cancer	CNS	Colon	Leukemia	Prostate Cancer	SRBCT	Ovarian
ReliefF	0.830	0.654	0.544	0.778	0.957	0.532	0.898	0.992
Fisher Score	0.861	0.651	0.640	0.792	0.960	0.542	1.000	0.992
Laplacian score	0.831	0.602	0.688	0.650	0.835	0.511	0.973	0.878
CFS	0.883	0.781	0.840	0.812	0.941	0.590	0.934	0.970
Low variance	0.844	0.851	0.610	0.760	0.849	0.553	0.858	0.932
T-score	-	0.525	0.809	0.781	0.981	0.855	-	0.990
FCBF	0.962	0.798	0.890	0.811	0.980	0.841	0.993	1.000
mRMR	0.853	0.802	0.812	0.715	0.972	0.688	0.942	0.992
Information gain	0.900	0.800	0.580	0.800	0.972	0.730	0.980	1.000

2.9.5 Analysis of hybrid techniques

Table 2.8 presents the performance of a number of hybrid techniques that have been previously utilized in selecting genes and classifying various types of Cancer in microarray chips.

From the table it is evident that hybrid techniques are superior in terms of the classification accuracy rate and the quantity of genes selected. It has been established that hybrid approaches tackle the overfitting as well as the curse of dimensionality well by foremost utilizing the filter techniques to shrink the dimensionality of these chips in the preprocessing phase.

To tackle the overfitting problem in supervised machine learning, application of the LOOCV i.e. leave-one-out cross-validation is highly recommended. With the LOOCV, a part of the considered dataset is held out as a test bench. This test set is set aside for the final validation, but a validation set is not required when cross-validation is adopted.

It is important to point out that fitting of parameters for feature selection and Cancer classification problem is still challenging. This is because, it is dependent on the considered DNA microarray dataset, adopted feature selection technique and the classifier. Thus, different DNA microarray datasets will have different parameter values, which are not universal to all algorithms.

In trying to tackle the parameter fitting challenge, majority of the researchers manually fine-tune the parameter values. A number of various values are tried until accepted results are reached. This approach is normally time-consuming and in many cases sub-optimal results are arrived at.

From Table 2.8, a hybrid of CFS and GA selected the highest number of genes i.e. 195 for the Lung dataset, while a combination of IG and GA attained 100% accuracy with the least number of genes i.e. 9 for the same dataset.

A hybrid of FCBF, GA and PSO selected a subset with the largest count of genes for the DLBCL dataset (i.e. 3204) in comparison with the other considered approaches.

For the Colon dataset, a combination of χ^2 −test and GA attained the highest rate of classification (i.e. 100%) and a set with the least count of genes (i.e. 8).

A hybrid comprising of mRMR, GBC and GA reported the highest accuracy rate (i.e. 100%) with the least number of genes (i.e. 6) for the SRBCT microarray dataset.

Moreover, a probabilistic random function combined with PSO reported the least accuracy value and a set with the highest count of genes for the Colon, Leukemia 1 and Lymphoma datasets.

It is evident that GA is mostly utilized wrapper technique in literature. Among the utilized wrappers, GA attained the most attractive classification value and sets with the least count of selected genes. From Table 2.8, GA attained 100% accuracy on most DNA microarray datasets in five out of six reported hybrid approaches.

All techniques employing the ACO wrapper attained classification accuracies greater than 90% with selected genes less than 15 in number.

The ABC achieved a classification accuracy of more than 98% and sets whose gene count is less than 15 in all the reported techniques. However, a combination of ACO and the SVM classifier achieved 100% accuracy. PSO reported an accuracy of 100% in 2 out of 4 reported approaches. Nevertheless, this technique reported sets with a relatively higher count of genes in comparison to other suggested wrapper approaches.

Though the Firefly, Cuckoo Search and Grey Wolf algorithms have been reported to perform incredibly in optimization tasks, they have not been utilized as wrappers in selecting informative genes for the DNA microarray chip data classification.

Table 2.8: Performance evaluation of suggested hybrid techniques for selecting genes and classifying types of cancer in DNA chips

Filter	Wrapper	Classifier	Datasets	Accuracy	Number of genes selected	Year	Reference
MI Maximization	GA	SVM	Colon	83.41	202	2017	[123]
χ^2 −test	GA	SVM	Colon	100	8	2011	[97]
			DLBCL	100	6		
			SRBCT	100	8		
			Leukemia1	100	5		
CFS	GA	KNN	SRBCT	100	29	2011	[96]
			Prostate	99.22	24		
			Lung	98.42	195		
Laplacian and Fisher score	GA	SVM	SRBCT	100	18	2017	[124]
			Leukemia1	100	15		
			Prostate	96.3	14		
			Breast	100	2		
			DLBCL	100	9		
		KNN	SRBCT	91.6	NAN		
			Leukemia1	97.2	NAN		
			Prostate	95.6	NAN		
			Breast	95.5	NAN		
			DLBCL	97.9	NAN		
		NB	SRBCT	98.2	NAN		
			Leukemia1	93.1	NAN		
			Prostate	93.4	NAN		
			Breast	100	NAN		
			DLBCL	95.8	NAN		
Fisher criteria	ACO	SVM	Leukemia1	95.95	3	2016	[125]
			Prostate	98.35	14		
		KNN	Leukemia1	94.30	3		
			Prostate	99.25	15		
		NB	Leukemia1	95.95	4		
			Prostate	99.40	10		
MI	ACO	FC	Colon	100	NAN	2018	[126]
			Leukemia 1	100	NAN		
			Prostate	90.85	NAN		
Fisher criterion	BA1	SVM	SRBCT	85	6	2018	[127]
			Prostate	94.1	6		
		KNN	SRBCT	100	6		
			Prostate	97.1	6		
		NB	SRBCT	100	6		
			Prostate	97.1	6		
ICA	ABC	NB	Colon	98.14	16	2017	[128]
			Leukemia1	98.68	12		
			Leukemia2	97.33	15		
			Lung	92.45	24		
mRMR	ABC	SVM	Colon	96.77	15	2018	[129]
			SRBCT	100	10		
			Leukemia1	100	14		
			Leukemia2	100	20		
			Lung	100	8		
			Lymphoma	100	5		
CFS	PSO	NB	Colon	94.89	4	2018	[130]
			SRBCT	100	34		
			Leukemia1	100	4		
			Leukemia2	100	6		
			Lymphoma	100	24		
			MILL	100	30		

			Breast	100	10		
Probabilistic random	PSO	KNN	Colon	84.38	60	2016	[131]
			Leukemia1	89.29	100		
			Lymphoma	87.71	50		
RFR	BHA	BC	Colon	91.93	3	2016	[132]
			MILL	98.61	5		
Fisher-Markov Selector	BA2	SVM	SRBCT	100	6	2013	[133]
			Prostate	98.3	12		
			Lung	98.4	16		
Symmetrical Uncertainty	HSA	NB	Colon	87.53	9	2016	[134]
			SRBCT	99.89	37		
			Leukemia1	100	26		
			Leukemia2	100	24		
			Lymphoma	100	10		
			MILL	98.97	10		
Logarithmic transformation	GOA	NN	Colon	95	NAN	2017	[135]
			Leukemia1	94	NAN		
FCBF	PSO+GA	SVM	Colon	96.3	1000	2017	[136]
			DLBCL	100	3204		
mRMR	GBC+GA	SVM	Colon	98.38	10	2015	[137]
			SRBCT	100	6		
			Leukemia1	100	4		
			Leukemia2	100	8		
			Lung	100	4		
			Lymphoma	100	4		

2.9.6 Analysis of the hybrid-ensemble techniques

As already, mentioned, with ensemble approaches, the results of various techniques are combined and the result of each constituent approach affects the final result.

2.9.6.1 Hybrid-ensemble type 1

The hybrid-ensemble approach proposed in [92] is one of the ensemble techniques whose performance is attractive when dealing with highly dimensional datasets. In this ensemble approach, feature reduction is foremost carried out by two filter approaches i.e. ReliefF and FCBF, then two wrappers are employed independently on the already dimensionally reduced data to select informative genes. Finally, the two subsets of informative genes derived by these wrappers are integrated together as shown in Figure 2.13.

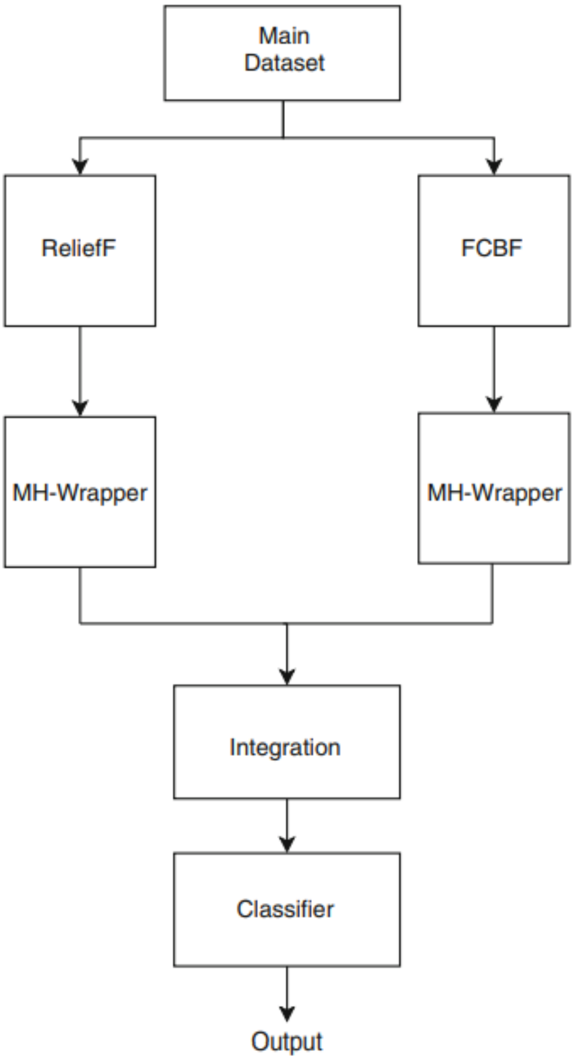


Figure 2.7: Scheme of hybrid-ensemble type 1

In this study, IBGSA as well as ABACO$_H$ techniques which attained attractive performances when handling data with large dimensions in [70] and [71] were utilized as the two wrappers in the scheme presented in Figure 2.7. In this study, the results of a number of various filter techniques have been compared to determine the most suitable filter approach. Moreover, a comparison of the results using the OR and AND operators as the integrators has been carried out to determine the most suitable integration technique.

2.9.6.1.1 Selecting the most suitable filter and integration technique

To identify the most suitable filter approach and integration technique, the outcome of identifying a number of various filter-based techniques and two integration operators i.e. OR and AND were considered. The outcome of this comparison is given in Tables 2.8 and 2.9 for the OR and the AND integrators respectively. In this experiment, the validation technique was utilized whereby 2/3 of the data and the remaining 1/3 were adopted for the training and testing phases respectively.

All the filter approaches except FCBF utilized in Tables 2.9 and 2.10 are rankers. Foremost, rankers assign a value (i.e. a rank) to every attribute and then sort all the features as per the assigned values. Consequently, these approaches need a threshold value to select informative features. In the experiment carried, a threshold value of 0.004 was adopted for all the filter techniques. Moreover, all the reported results were evaluated using the KNN classifier whereby k was set to 1.

From Tables 2.9 and 2.10, for the Colon dataset (with 62 samples and 2000 genes) the HMEBO-FCBF ensemble achieved the highest values for the ACC, MCC, GM and SP. The HMEBA-IG technique attained the top values for both the GMEAN as well as SN with this dataset.

For the Leukemia dataset with 72 samples and 7129 genes, the proposed HEMO-F-score ensemble technique attained the top values for both ACC, GM and MCC. However, the HEMO-FCBF approach attained the highest values for the SP and SN.

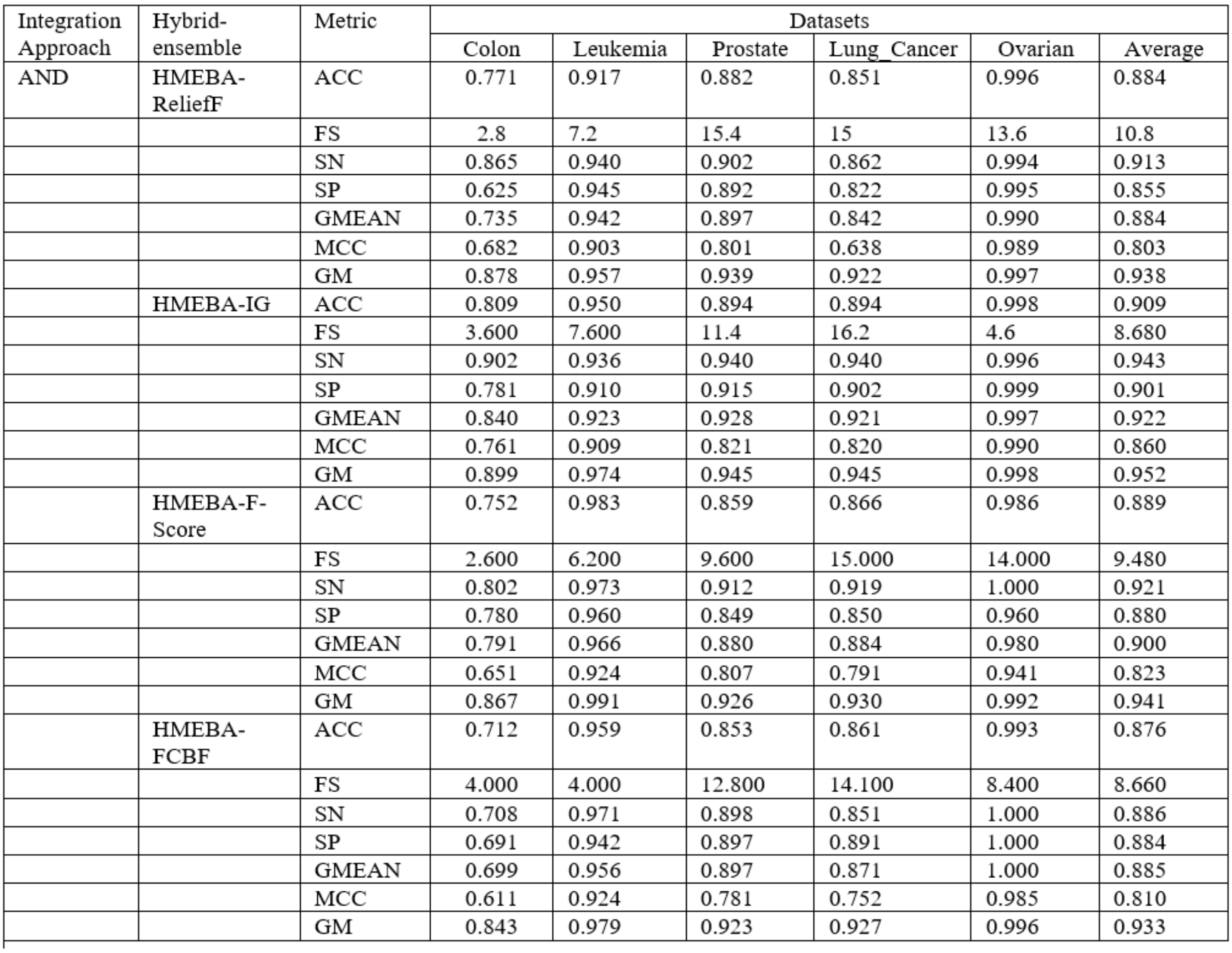
The Prostate cancer dataset with 10509 genes and 102 observations is deemed to be challenging. This is because it is derived from a number of different tests and has the data drift problem [71]. Though challenging, the HMEBO-FCBF ensemble technique attained the highest ACC, MCC, SP, GMEAN and GM for this data. However, for the SN parameter the HEMO-Relief technique reported the top value.

For the considered Lung cancer dataset containing 12,533 genes and sample size of 181, the HMEBO-FCBF technique attained the best values for the ACC, SN, SP, GMEAN and GM. However, for the MCC parameter, the HMEBO-F-score attained the most attractive value.

For the Ovarian microarray dataset with15, 154 and 253 samples, a dataset with the highest count of genes among all the considered datasets, the proposed HEMO-FCBF technique attained a score of 1.00 for the ACC, SN, SP, GMEAN and MCC. For the MC parameter, this ensemble approach attained a score of 0.999, which is still the best result in comparison to those reported by other techniques.

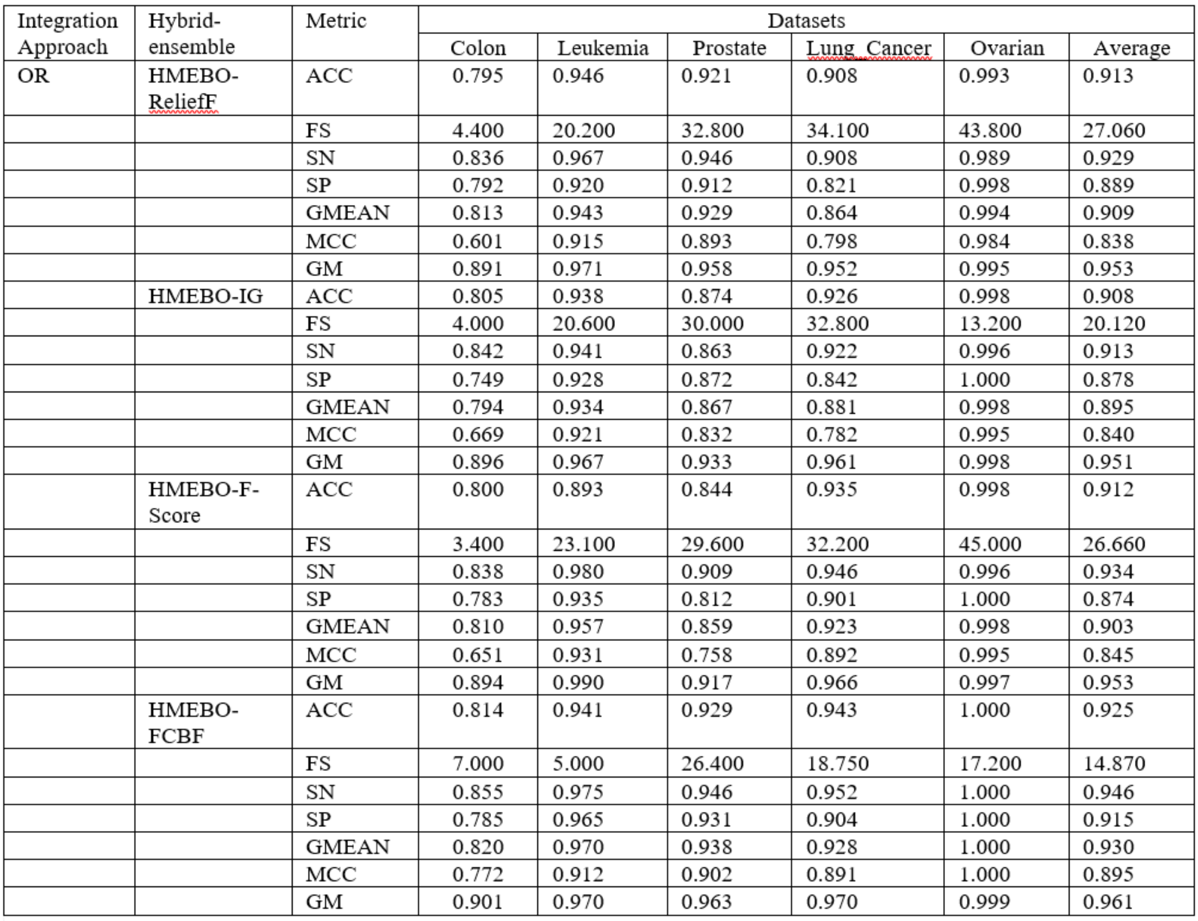
Considering the average values attained for all the five evaluation metrics (see Tables 2.9 and 2.10), the HMEBO-FCBF ensemble technique has proved to be superior compared to the other techniques for the five DNA microarray datasets. Moreover, the FCBF filter technique with the OR integrator attained the best results.

Table 2.9: Experimental results for hybrid-ensemble techniques combined with KNN and utilizing the AND integrator



Integration Approach	Hybrid-ensemble	Metric	Datasets					
			Colon	Leukemia	Prostate	Lung_Cancer	Ovarian	Average
AND	HMEBA-ReliefF	ACC	0.771	0.917	0.882	0.851	0.996	0.884
		FS	2.8	7.2	15.4	15	13.6	10.8
		SN	0.865	0.940	0.902	0.862	0.994	0.913
		SP	0.625	0.945	0.892	0.822	0.995	0.855
		GMEAN	0.735	0.942	0.897	0.842	0.990	0.884
		MCC	0.682	0.903	0.801	0.638	0.989	0.803
		GM	0.878	0.957	0.939	0.922	0.997	0.938
	HMEBA-IG	ACC	0.809	0.950	0.894	0.894	0.998	0.909
		FS	3.600	7.600	11.4	16.2	4.6	8.680
		SN	0.902	0.936	0.940	0.940	0.996	0.943
		SP	0.781	0.910	0.915	0.902	0.999	0.901
		GMEAN	0.840	0.923	0.928	0.921	0.997	0.922
		MCC	0.761	0.909	0.821	0.820	0.990	0.860
		GM	0.899	0.974	0.945	0.945	0.998	0.952
	HMEBA-F-Score	ACC	0.752	0.983	0.859	0.866	0.986	0.889
		FS	2.600	6.200	9.600	15.000	14.000	9.480
		SN	0.802	0.973	0.912	0.919	1.000	0.921
		SP	0.780	0.960	0.849	0.850	0.960	0.880
		GMEAN	0.791	0.966	0.880	0.884	0.980	0.900
		MCC	0.651	0.924	0.807	0.791	0.941	0.823
		GM	0.867	0.991	0.926	0.930	0.992	0.941
	HMEBA-FCBF	ACC	0.712	0.959	0.853	0.861	0.993	0.876
		FS	4.000	4.000	12.800	14.100	8.400	8.660
		SN	0.708	0.971	0.898	0.851	1.000	0.886
		SP	0.691	0.942	0.897	0.891	1.000	0.884
		GMEAN	0.699	0.956	0.897	0.871	1.000	0.885
		MCC	0.611	0.924	0.781	0.752	0.985	0.810
		GM	0.843	0.979	0.923	0.927	0.996	0.933

Table 2.10: Experimental results for hybrid-ensemble techniques combined with KNN and utilizing the OR integrator



Integration Approach	Hybrid-ensemble	Metric	Datasets					
			Colon	Leukemia	Prostate	Lung Cancer	Ovarian	Average
OR	HMEBO-ReliefF	ACC	0.795	0.946	0.921	0.908	0.993	0.913
		FS	4.400	20.200	32.800	34.100	43.800	27.060
		SN	0.836	0.967	0.946	0.908	0.989	0.929
		SP	0.792	0.920	0.912	0.821	0.998	0.889
		GMEAN	0.813	0.943	0.929	0.864	0.994	0.909
		MCC	0.601	0.915	0.893	0.798	0.984	0.838
		GM	0.891	0.971	0.958	0.952	0.995	0.953
	HMEBO-IG	ACC	0.805	0.938	0.874	0.926	0.998	0.908
		FS	4.000	20.600	30.000	32.800	13.200	20.120
		SN	0.842	0.941	0.863	0.922	0.996	0.913
		SP	0.749	0.928	0.872	0.842	1.000	0.878
		GMEAN	0.794	0.934	0.867	0.881	0.998	0.895
		MCC	0.669	0.921	0.832	0.782	0.995	0.840
		GM	0.896	0.967	0.933	0.961	0.998	0.951
	HMEBO-F-Score	ACC	0.800	0.893	0.844	0.935	0.998	0.912
		FS	3.400	23.100	29.600	32.200	45.000	26.660
		SN	0.838	0.980	0.909	0.946	0.996	0.934
		SP	0.783	0.935	0.812	0.901	1.000	0.874
		GMEAN	0.810	0.957	0.859	0.923	0.998	0.903
		MCC	0.651	0.931	0.758	0.892	0.995	0.845
		GM	0.894	0.990	0.917	0.966	0.997	0.953
	HMEBO-FCBF	ACC	0.814	0.941	0.929	0.943	1.000	0.925
		FS	7.000	5.000	26.400	18.750	17.200	14.870
		SN	0.855	0.975	0.946	0.952	1.000	0.946
		SP	0.785	0.965	0.931	0.904	1.000	0.915
		GMEAN	0.820	0.970	0.938	0.928	1.000	0.930
		MCC	0.772	0.912	0.902	0.891	1.000	0.895
		GM	0.901	0.970	0.963	0.970	0.999	0.961

2.9.6.2 Hybrid-ensemble type 2

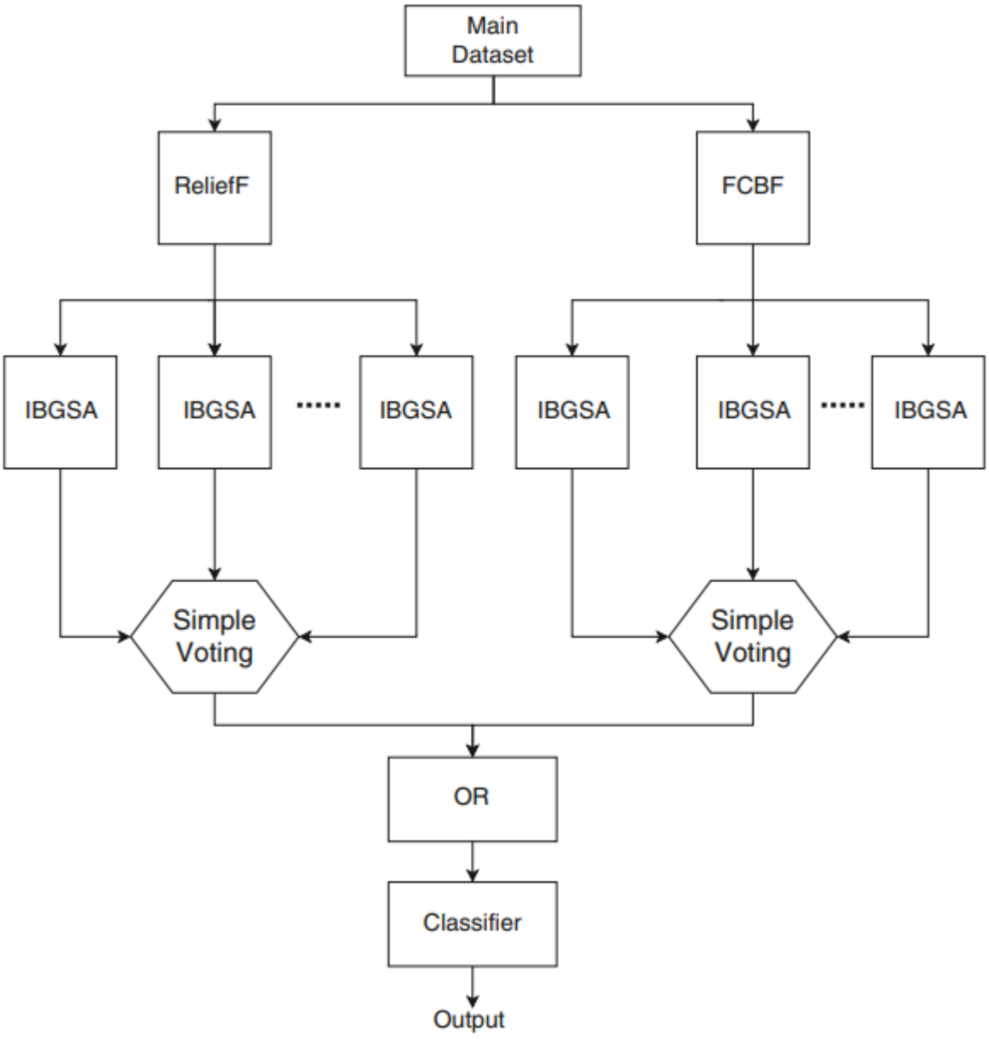


Figure 2.8: Scheme of hybrid-ensemble type 2

Reference [79] has tackled the feature selection problem in highly dimensional datasets by proposing a framework derived from the hybrid-ensemble techniques. This authors' scheme is presented in Figure 2.8.

In this framework, each line tries to reduce the dimensions of these highly dimensioned datasets. Foremost, each filter approach outputs a subset of features, in which the dimension of a dataset is largely reduced. Then, the informative features associated with the highest reported accuracy are selected by metaheuristic approaches. Thirdly, the outcome of the attribute selection process of every line are merged using various integrators like the AND and OR logic operators. Eventually, the accuracy of this scheme is computed using a specific classifier.

After considered a number of filter approaches, the authors settled on the FCBF and reliefF filter approaches for their scheme. Moreover, they utilized the IBGSA approach as the wrapper in this framework.

To select the informative features, a simple voting technique was adopted. With this technique a wrapper repeats for g times, and if the total occurrences in which a given attribute is selected as informative is more than the total occurrences is not picked, the attribute will be picked; otherwise it will not.

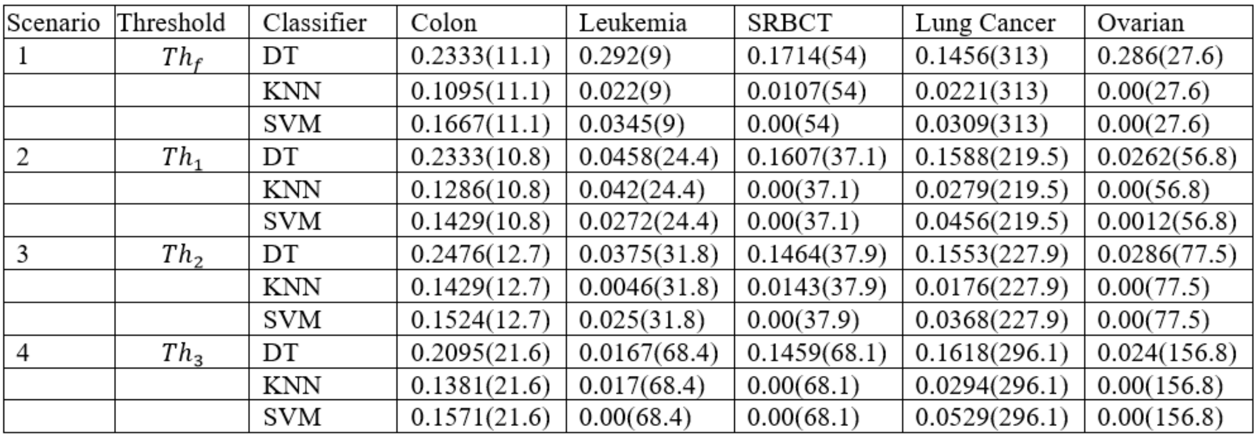
As already pointed out, ranking based filter approaches such as the ReliefF technique require one to supply a threshold value for them to select of features. Table 2.11 presents the experimental results achieved by setting different threshold values for the ReliefF technique. Three different classifiers i.e. SVM, k-nearest neighbor (KNN) and the decision tree (DT) were utilized.

The values for Th_1, Th_2 and Th_2 were set as 0.0066, 0.009 and 0.02 respectively. The value for Th_f was set equal to the count of attributes returned by the adopted FCBF technique.

Moreover, Table 2.11 presents the error rate of three considered classifiers i.e. DT, SVM and KNN and the total count of selected attributes by each of the proposed hybrid-ensemble technique (in brackets). From the results presented, KNN yielded the most attractive results in comparison to the other classifiers. Moreover, scenarios 1 and 2 reported the least count of selected attributes and classification error rate in comparison to scenarios 3 and 4.

It is evident that an increase in the threshold value reduces the classification accuracy of the classifiers. This is because this increase facilitates the selection of redundant and irrelevant features in highly dimensional datasets.

Table 2.11: Experimental results for hybrid-ensemble techniques for different threshold values of the ReliefF filter approach



Scenario	Threshold	Classifier	Colon	Leukemia	SRBCT	Lung Cancer	Ovarian
1	Th_f	DT	0.2333(11.1)	0.292(9)	0.1714(54)	0.1456(313)	0.286(27.6)
		KNN	0.1095(11.1)	0.022(9)	0.0107(54)	0.0221(313)	0.00(27.6)
		SVM	0.1667(11.1)	0.0345(9)	0.00(54)	0.0309(313)	0.00(27.6)
2	Th_1	DT	0.2333(10.8)	0.0458(24.4)	0.1607(37.1)	0.1588(219.5)	0.0262(56.8)
		KNN	0.1286(10.8)	0.042(24.4)	0.00(37.1)	0.0279(219.5)	0.00(56.8)
		SVM	0.1429(10.8)	0.0272(24.4)	0.00(37.1)	0.0456(219.5)	0.0012(56.8)
3	Th_2	DT	0.2476(12.7)	0.0375(31.8)	0.1464(37.9)	0.1553(227.9)	0.0286(77.5)
		KNN	0.1429(12.7)	0.0046(31.8)	0.0143(37.9)	0.0176(227.9)	0.00(77.5)
		SVM	0.1524(12.7)	0.025(31.8)	0.00(37.9)	0.0368(227.9)	0.00(77.5)
4	Th_3	DT	0.2095(21.6)	0.0167(68.4)	0.1459(68.1)	0.1618(296.1)	0.024(156.8)
		KNN	0.1381(21.6)	0.017(68.4)	0.00(68.1)	0.0294(296.1)	0.00(156.8)
		SVM	0.1571(21.6)	0.00(68.4)	0.00(68.1)	0.0529(296.1)	0.00(156.8)

2.10 Summary of the recently suggested attribute selection techniques for the DNA microarray chip analysis

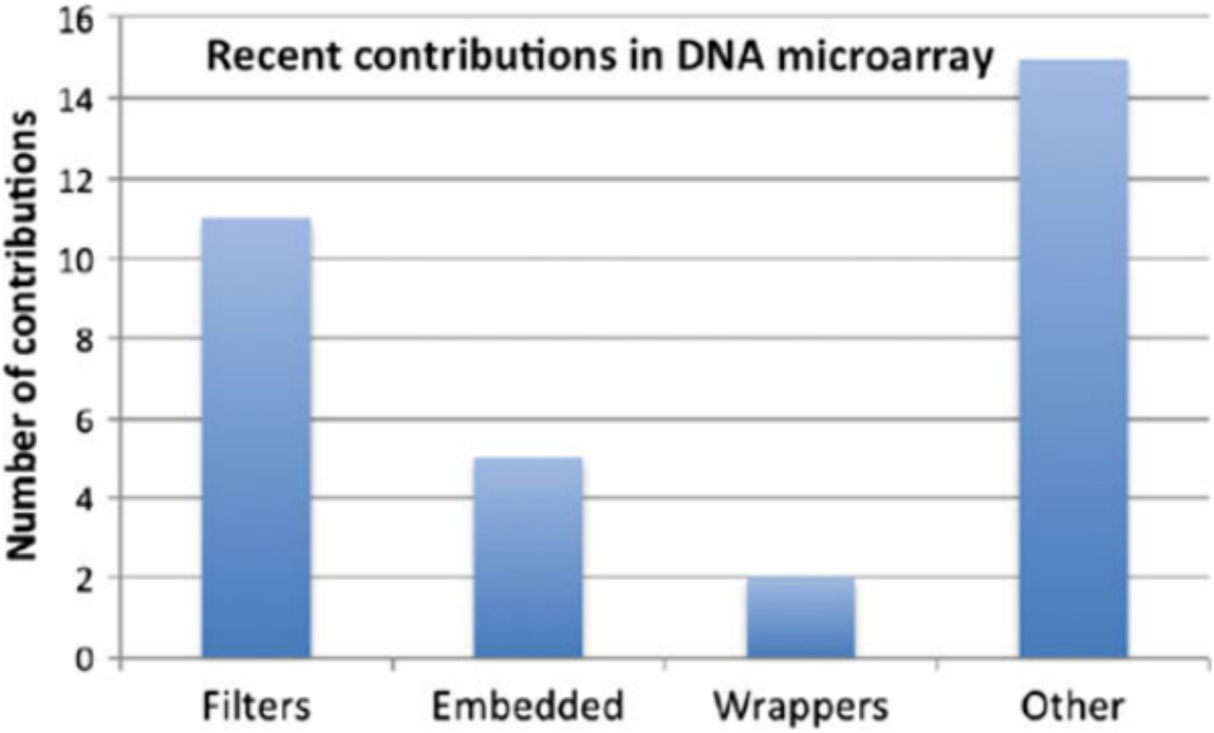


Figure 2.9: Recently suggested attribute selection techniques for the DNA microarray chip analysis

A complete assessment of the state-of-the-art attribute selection techniques for the DNA microarray data can be found in [138]. In this review, since 2008 many contributions fall under the category of filters (see Figure 2.9). The wrappers have been largely avoided due to their large computational cost and high chances of overfitting. Though embedded techniques were not largely utilized during the infant stages of classifying DNA chips, a number of contributions have been made in the recent past.

Thus, it is important to note that the recent review reveals a trend to combine techniques in the ensemble or hybrid approaches (depicted by "Other" in Figure 2.9).

2.11 Inferences drawn

To date, optimal gene selection and accurate classification of a given patient sample are the most sought topics in a DNA microarray based cancer disease diagnosis.This is because an effective gene selection phase derives a reduced informative gene subset from the gene-rich DNA microarray datasets which subsequently minimizes noise, computational overheads as well as model overfitting. On the other hand, an improved learning and classification stage builds an effective classifier that achieves a reliable and accurate classification of a DNA patient sample.

Optimal gene selection requires a stable, diverse and robust gene selector. This can only be achieved by a wrapper that maturely converges during the search process and thus ensuring an exhaustive search of the whole population of DNA microarray genes. On the other hand, mature convergence demands striking of a proper and optimal balance between exploitation and exploration in the design of a metaheuristic. Exploitation and exploration are two must attain antagonistic principles that pose a big challenge in striking a proper balance between them in the design metaheuristics. A reason why utilizing single-based metaheuristic wrappers have proved inadequate in solving the feature selection problem in DNA microarray based cancer disease diagnosis. Thus, researchers are keen on new unions of existing feature selection approaches such as hybrid or ensemble techniques. This is because the hybrid or ensemble techniques enhance adequately the robustness of the final informative gene subset, which is also a trending research topic in this area.

It is desirable that the techniques selected to form the ensemble algorithm are diverse, i.e. the consituent algorithms of the ensemble should be able to return outputs that are different and enough when handling the sample of data. Nevertheless, if this sample of data is changed, it is preferable that the considered approaches attain similar outputs i.e. an attribute regarded as stability. Thus, research on the stability and diversity of ensemble attribute selection need to be carried out. Moreover, new demands are emerging in society for instance in the area of real-time processing and distributed learning, where a critical gap that needs to be researched upon is developing.

Designing an efficient gene selector without enhancing both the learning and classification phase will still render the DNA microarray based cancer classification pipeline incomplete.Though currently the SVM is a promising classifier in DNA microarray data classification, its performance largely depends on the kernel adopted for this classifier as well as tuning of the kernel parameters. The linear, polynomial and Gaussian kernels are the three standard kernels commonly adopted by a large number of researchers for this classifier. The linear kernel function has a better extraction of global features from samples, the polynomial kernel has good generalization ability and the gaussian kernel (the most widely used kernel) has a good learning ability among all the single kernel functions. Thus, it is evident that utilizing a single kernel function based MCSVM classifier in a given application such as gene expression data may neither attain good learning ability, proper global feature extraction ability and a better generalization capability.To date, this has necessitated a combinination of two or more of these standard kernel functions.

In trying to address the issue of stability and diversity in feature selection using wrappers, an excited binary grey wolf optimizer (EBGWO) based wrapper approach is proposed in Chapter 3 for the selection of informative genes and cancer type classification using the highly dimensional DNA datasets. To make full use of/ and strike an effective balance between diversification and intensification of the existing BGWO, a novel electrically inspired nonlinear strategy for the control parameter $a^{\rightarrow}$ of the BGWO is proposed. In this strategy, the value of $a^{\rightarrow}$ is decreased via the concept of the complete current response of the direct current (DC) excited resistor-capacitor (RC) circuit. Since the proposed strategy allocates a large number of generations to diversification in comparison to intensification, the convergence speed of the EBGWO algorithm is heightened while reducing the local optimal trapping effects. To enhance diversity and improve the quality of the reported solutions a weighting scheme utilizing the fitness values of the three leaders of the pack (alpha(α), beta(β) and delta(δ)), that of the currently considered wolf and that worst wolf is adopted. Finally, to maintain and strengthen the social hierarchy of the pack, a fitness-value based position-updating criterion is used.

Although the proposed EBGWO is able to report a subset with the least number of features while maintaining an attractive classification accuracy, it does not attain an optimal balance between exploitation and exploration. This is because exploration of search domain and exploitation of optimal solutions are two conflicting principles that difficult to attain in single-metaheuristic based wrappers. In trying to achieve the required optimal balance between the two, a new memetic excited (E) -adaptive cuckoo search (ACS)-intensification dedicated grey wolf optimizer (IDGWO) i.e. EACSIDGWO algorithm is proposed in chapter 4. The EACSIDGWO algorithm hybridizes IDGWO (a variant of the EBGWO) and another new improved cuckoo search algorithm i.e. ACS.The step size of ACS is also innovatively made adaptive via the concept of complete voltage response of the direct current (DC) excited resistor-capacitor (RC) circuit. Since the diversity of the population is higher during the early stages of proposed EACSIDGWO algorithm, both the ACS and IDGWO jointly carry out local exploitation during these stages. However, to enhance mature convergence during later stages of the proposed algorithm, the role of ACS is switched to global exploration while the IDGWO is still left carrying out local exploitation.

Finally, to enhance the performance of the classification phase (the last stage of the DNA microarray-based cancer analysis), a novel hybrid linear-gaussian-polynomial (LGP) kernel-based multiclass support vector machine i.e. LGP-MCSVM is proposed in chapter five. The hybrid LGP kernel innovatively combines the advantages of three standard kernels (linear, gaussian and polynomial); where the linear kernel is linearly combined with a gaussian kernel embedding the polynomial kernel.

CHAPTER THREE: AN EXCITED BINARY GREY WOLF OPTIMIZER FOR FEATURE SELECTION IN HIGHLY DIMENSIONAL DATASETS

3.1 Introduction

The major challenge in analysing big data is the elevated count of features. Out of the many features, only a small subset is useful in distinguishing observations belonging to different classes while majority of the features will be either noise, irrelevant or redundant. Foremost, features that are irrelevant result into noise generation in the analysis of these big data. In addition, they normally lead to elevated dataset dimensions and a further computational burden in both the classification as well as the clustering operations. Consequently, all these attributes hinder attaining a higher classification accuracy. Thus, superior approaches are needed to identify diverse features, compute the relationship between the features and optimally select informative attributes from these highly dimensioned datasets [60].

For a given dataset with N attributes, there exists 2^N possible candidate subsets. The main objective of formulating various attribute selectors is to be able to determine a shrinked and optimal subset which can attain the highest precision among all the possible candidate subsets.

Since the scope of possible results is wide and the size of the set of responses is on the increase due to the ever-increasing count of features, determining the optimal subset of N informative features is extremely difficult and costly [140].

Attribute selection techniques can be broadly categorized into two i.e. filters and wrappers. Filter approaches utilizes the distance, dependency, information theory and mutual information in carrying out attribute selection [141]. Unlike filters, wrappers utilize classifiers as the learning technique in optimizing the classification outcome by selecting the informative attributes. In most cases, filter techniques are often faster compared to wrappers, which is largely attributed to their reduced computational complexity [142]. Nevertheless, wrapper techniques can usually offer better performances compared to filters [143]. Wrappers apply metaheuristic optimization approaches, such as binary genetic algorithm (BGA) [144], binary version of grey wolf optimization (BGWO) [90], binary ant colony optimization (BACO) [145], binary version of particle swarm optimization (BPSO) [146], to select the optimal informative feature subsets.

BGWO is among the recently suggested attribute selection approaches. This technique usually attains an attractive performance compared to other existing conventional approaches[90] . Nontheless, the wolves' new locations are solely depend on the their leaders' experience i.e. delta, alpha and beta, which normally leads to ill-timed convergence. Moreover, an absolute balance between the diversification and intensification is still a big problem with the BGWO [91].

This chapter proposes a new excited binary version grey wolf optimizer (EBGWO) whose main objective is to improve the performance of existing BGWO [90] in selecting informative features in highly dimensioned microarray datasets. Foremost, to overcome the insufficiency of the existing BGWO in regard to the criterion used to update the wolves' positions, which is good at intensification but poor at diversification, a new position-updated equation utilizing the fitness values of vectors $\vec{X}_1, \vec{X}_2$ and $\vec{X}_3$ is proposed to determine new candidate individuals. Moreover, inspired by the concept of the complete current response of a direct current (DC) excited resistor-capacitor (RC) circuit, another new nonlinear criterion to control parameter $\vec{a}$

is introduced to ensure full use of and balance the diversification and intensification of the existing BGWO technique.

The performance of EBGWO is tested using seven standard gene expression datasets. To assess the appropriateness of suggested method, the performance of EBGWO is contrasted with those of five state-of-the-art binary metaheuristic algorithms i.e. BPSO, BGWO1, BDE, BGWO2 and BGA. It is evident from the achieved results that the proposed technique has a lower computational burden while maintaining a comparative performance in selecting features.

The rest of this chapter is arranged as follows. A summary of the GWO algorithm is presented in section 3.2. The proposed excited grey wolf optimizer (EGWO) is presented in section 3.3. The binary version of EGWO i.e. Excited binary grey wolf optimizer (EBGWO) is presented in section 3.4. Section 3.5 reports the experimental setting and a discussion of the obtained results. Finally, a conclusion and further works are given in section 3.6.

3.2 Grey Wolf Optimization (GWO)

GWO is a recently proposed optimization technique [89]. In nature, grey wolves live in groups ranging between 5 to 12. GWO mimics the behaviour portrayed by these grey wolves while hunting and searching of a prey. In GWO, to simulate the leadership hierarchy in a pack, the population is divided into 4 classes of wolves i.e. Beta (β), omega (ω), alpha (α), and delta (δ). The α wolf is the overall pack leader and is largely engaged in decision-making. The β wolf is the second in command and it usually assists the α wolf in planning the various pack endevours. The δ wolf, the third in command, dominates the ω wolves. The 3 leaders i.e. α, β and δ guide the hunting (optimization) while the remaining omega wolves (ω) follow them [147].

3.3 Excited Grey Wolf Optimization (EGWO)

3.3.1 Nonlinearly controlling parameter a via the complete current response of the dc excited RC circuit

It is a well-established fact that for population-based metaheuristics, both exploration (diversification) and exploitation (intensification) are conducted concurrently.

Exploration is termed as the ability of a population-based metaheuristic to examine new areas within the defined search space with the aim of determining the global optima. On the hand, exploitation is the ability to utilize the information of already identified individuals in deriving better individuals [148], [149].

In every population-based metaheuristic, both exploration (diversification) and exploitation (intensification) abilities are attained by applying specific operators.

In the conventional GWO algorithm, parameter a plays a critical role in striking a balance between diversification and intensification of an individual candidate search [148]. A big value of a enhances global diversification, on the other hand its smaller value promotes local intensification. Thus, selection of a suitable control strategy for parameter a is critical in attaining an effective balance between local exploitation and global exploration. From literature, one proved way to achieve the required balance is critically studying the control of parameter a. To date, various approaches have been proposed to control the conventional GWO's parameter a [148], [149], [150].

However, in the conventional GWO, a linearly decreases from 2 to 0 using Equation (2.39). Since GWO incorporates a highly complicated nonlinear search process, the utilized linear control of parameter a doesn't clearly portray the real search process [148]. In addition, [150] suggested that the performance of GWO would improve if parameter a is nonlinearly controlled.

Motivated by both the above consideration and the complete current response of a direct current (DC) excited resistor-capacitor (RC) circuit[151], a novel nonlinear control strategy for parameter a is proposed in this paper.

The complete current response of the RC circuit to a sudden application of a dc voltage source, with the assumption that the capacitor is initially not charged is given in Equation 3.1.

$$i(t) = \frac{V_s}{R}((\frac{1}{e^t})^\tau) \qquad 3.1$$

Where $\tau = R\times C$ is the time constant that expresses the rapidity with which the value if i decreases from the initial value $\frac{V_s}{R}$ to zero over time. V_s is value of a constant DC voltage while R and C are the resistor and capacitor values of the circuit.

We adopt this concept i.e. the exponential decay of i over time to develop a new nonlinear control strategy of parameter a (refer to Equation 2.39) as presented in Equation 3.2.

$$a_{i,t} = a_{initial}\times(\frac{MaxIter\ - t}{MaxIter})^{\tau_{i,t}} \qquad 3.2$$

Where $a_{i,t}$ is the computed value of the a assigned to grey wolf i during iteration t. $MaxIter$ indicates the total count of generations and $a_{initial}$ is the initial value of the control parameter a. $\tau_{i,t}$ is a nonlinear modulation index assigned to the grey wolf i during iteration t.

In ensuring that $a_{i,t}$ is proportional to the fitness value of grey wolf i during iteration t, a new formulation of the value of the nonlinear modulation index $\tau_{i,t}$ is given in Equation 3.3.

$$\tau_{i,t} = \left|\frac{(\frac{F\alpha_t +\ F\beta_t + F\delta_t}{3}) - FX_t}{(\frac{F\alpha_t +\ F\beta_t + F\delta_t}{3}) - Fw_t}\right| \qquad 3.3$$

Where $F\alpha_t$, $F\beta_t$ and $F\delta_t$ are fitness values of α, β and δ wolves (the 3 leaders) respectively during the current iteration t. FX_t is the fitness value of grey wolf i during iteration t and finally Fw_t is the worst fitness value among the omega (ω) wolves during iteration t .

Consequently, A_1, A_2 and A_3 are determined using Equation 3.4 which is a variant of Equation 2.37.

$$A = 2.a_{i,t}.r_1 - a_{i,t} \quad 3.4$$

From the literature of conventional GWO [89], when A is less than 1 the wolves are compelled to attack the current prey (intensification) and when A is greater than 1 the wolves are compelled to move away from the current prey with the hope of finding another fitter prey. This implies that a smaller value of a advances local intensification while a larger value enhances global diversification.

According to Equation 2.39 of the conventional GWO algorithm, it is evident that $1/_2$ of the iterations are committed to diversification and the other $1/_2$ to intensification. This strategy fails to consider the effect of effective balancing between these 2 conflicting milestones in ensuring accurate approximation of the global optimum.

The nonlinear control strategy of parameter a proposed in Equation 3.2, tries to overcome this challenge by adopting a variant of decay function to facilitate a proper balance between diversification and intensification. Since this strategy allocates a large proportion of the iterations to global exploration compared to local exploitation, the convergence speed of the proposed EGWO algorithm is enhanced while minimizing the local minima trapping effect.

Moreover, since the proposed scheme is correlated to the fitness values of the each grey wolve in the search space and the current count of generations, diversity and the quality of the solutions is enhanced.

3.3.2 Socially Strengthened Hierarchy via a Fitness-value based position-updating criterion

In the conventional GWO, social order is the cornerstone in both the internal governance as well as the hunting patterns of the pack [152]. All the wolves within the pack conduct hunting under the close guidance of the α, β and δ wolves. An assumption that these 3 leaders have a better understanding of the prey's position is made. Consequently, the omega (ω) wolves update their locations with the help of these three leaders during the hunting process. This implies that the conditions of the α, β and δ wolves are key in updating the whole pack. Meanwhile, the higher the rank a wolf attains during the search, the closer it gets to the global optimum.

In addition, all the wolves including the three leaders utilize Equation 2.40 to update their positions. That is to say the α wolf will utilize the lowly ranked β and δ wolves to update its position. Likewise, β wolf will utilize the lowly ranked δ wolf to update itself. Since the conditions of the β and δ wolves are worse compared to that of the α wolf, there are higher chances that the two wolves will compel the α wolf to move away from the global optimum. Likewise, β wolf may also be misled by the δ wolf. Ultimately, the accumulative error will have an adverse effect on updating the positions of all the wolves in the pack and the convergence efficiency of the GWO will drastically reduce[152].

On the other hand, since all the omega (ω) wolves are attracted towards the α, β and δ wolves, they may prematurely converge due to limited exploration within the search space. Thus, the conventional GWO is good at intensification but poor at diversification.

Thus, to overcome the GWO's premature converge and still maintain the social hierarchy of the pack, a different scheme for updating both the dominant (α, β and δ) and the omega (ω) wolves is needed. To attain this, a new position-updated equation utilizing the fitness values of vectors $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ is utilized in determining new candidate individuals.

Foremost, for each wolf in the pack, vectors X_{vec1}, X_{vec2} and X_{vec3} are computed using Equations 3.5 to 3.7

$$X_{vec1} = \bigcup_{j=1}^{d} X_1(j) \qquad 3.5$$

$$X_{vec2} = \bigcup_{j=1}^{d} X_2(j) \qquad 3.6$$

$$X_{vec3} = \bigcup_{j=1}^{d} X_3(j) \qquad 3.7$$

Where d is the dimension of the search space and $X_1(j), X_2(j)$ and $X_3(j)$ are determined using Equations 2.41 to 2.43 respectively.

Next, the fitness values FX_{vec1}, FX_{vec2} and FX_{vec3} for vectorsX_{vec1}, X_{vec2} and X_{vec3} respectively are determined and the one with the best fitness forms the new position as depicted by Equations 3.8 to 3.9.

$$[fittest, Pos] = min\left(\bigcup_{a=1}^{3} FX_{vec(a)}\right) \qquad 3.8$$

$$X(t+1) = \left(\bigcup_{a=1}^{3} X_{vec(a)}\right)_{Pos} \qquad 3.9$$

3.4 Excited Binary Grey Wolf Optimization (EBGWO)

The selection of features (FS) is a significant challenge in machine learning as well as pattern recognition areas. Guided by a given evaluation criterion, FS aims at deriving a subset with least count of the most informative features [153], [152].

Thus, FS is a broad-based optimization challenge that is characterized by huge computations.

Since the FS problem utilizes a binary search space, conversion of the proposed continuous EGWO to binary i.e. EBGWO is required. One of the commonly adopted approach for this transformation is the utilization of transfer functions[152], [153].

In these experiments, the transfer function utilized in converting the real values of each solution to binary is depicted by Equation 3.10.

$$X^j(t+1) = \begin{cases} 1, if\ S\left(X^j(t+1)\right) > \rho, \\ 0, \quad otherwise \end{cases} \tag{3.10}$$

Where $\rho \in [0,1]$ depict a random threshold and S is the considered sigmoid function as expressed by Equation 3.11.

$$S(x) = \frac{1}{1 + \exp\left(-10(x - 0.5)\right)} \tag{3.11}$$

$X^j(t+1)$=1 imply that the j^{th} element of $X(t+1)$ is selected as an informative attribute while $X^j(t+1)$=0 imply that the corresponding j^{th} element is ignored.

For instance, if $X(t+1) = [0.55, 0.21, 0.35, 0.8]$ and $\rho = 0.5$, the output of Equation 3.21 becomes $X(t+1) = [1, 0, 0, 1]$ which imply that the 1^{st} and 4^{th} features be selected while the 2^{nd} and 3^{rd} features will be ignored.

By doing so, the number of features is greatly reduced without adversely affecting the performance in terms of classification accuracy.

Because FS task aims at attaining better classification accuracy with the utilization of fewer attributes, the objective function Fit utilized in this paper is given by Equation 3.12 [153].

$$Fit = \varepsilon * \frac{|S|}{|N|} - ((1 - \varepsilon) * Acc) \tag{3.12}$$

Where Acc is indicates the accuracy of a given classifier, $|S|$ is the count of features in the derived subset and $|N|$ is the total count of features within the dataset. Thus, FS is turned into a problem of determining the least value of Equation 3.12.

Herein, ε and $(1 - \varepsilon)$ are weights corresponding to the feature subset size and average accuracy respectively. The parameter of ε in Equation 3.12 is set 0.2 [153].

The pseudocode of the proposed excited binary grey wolf optimizer (EBGWO) algorithm is presented in Algorithm 3.1.

Algorithm 1: Pseudo-code for the EBGWO

```
Input: labelled gene dataset $D$, Total number of iterations $MaxIter$, Population size $N$, Initial value of the control parameter $a_{initial}$
Output: Optimal Individual's position $X_{\alpha}$ , Best fitness value $Fit\ (X_{\alpha})$
1. Randomly initialize $N$ individuals' positions to establish a population
2. Using Equation (3.12), evaluate the fitness of all wolves, $Fit\ (X)$
3.      $[\sim, Index] = Sort\ (Fit\ (X), 'Ascend')$
4.      $F\alpha = Fit\ (X)_{Index(1)}$
5.      $F\beta = Fit\ (X)_{Index(2)}$
6.      $F\delta = Fit\ (X)_{Index(3)}$
7.      $Fw = Fit\ (X)_{Index(N)}$
8.      $X_{\alpha} = X(Index(1))$
9.      $X_{\beta} = X(Index(2))$
10.     $X_{\delta} = X(Index(3))$
11.     For $t=1$ To $MaxIter$
12.       For $i=1$ To $N$
13.           Determine $a_{i,t}$ using Equation (3.2)
14.           Compute $X_{vec1}, X_{vec2}$ and $X_{vec3}$ using
              Equations (3.5)-(3.7)
15.           Generate $X_{vec1}^{new}$, $X_{vec2}^{new}$ and
              $X_{vec3}^{new}$ using Equation (3.10)
16.           Evaluate the fitness values $FX_{vec1}$, $FX_{vec2}$ and $FX_{vec3}$ of the binary
        vectors$X_{vec1}^{new}$ $X_{vec2}^{new}$ and $X_{vec3}^{new}$ respectively using Equation (3.12)
17.         Determine the minimum value(fittest) of the
            three evaluated fitness values and its Index
            using Equations (3.8)
18.         If (fittest<$Fit\ (X)_{Index(i)}$) Then
19.             $Fit\ (X)_{Index(i)}$= fittest
20.             Update $X_{Index(i)}$ using Equation (3.9)
            End If
21.     Next i
22.           Repeat steps 3 to 10
23.     Next t
```

3.5 Experimental Results and Discussion

All the computations were conducted on a Windows 10 Home Single Language 64-bit operating system; processor Intel(R) Core (TM) i7-3770CPU processor speed of 3.4GHZ; 12GB of RAM. All the considered approaches were implemented and executed using MATLAB 2017 environment.

3.6 Dataset description

In order to evaluate the effectiveness of the proposed technique, seven standard DNA chips derived from Irvine (UCI) repository were utilized. The datasets were selected to have a variety of observations (sample-size), genes and classes as prototypical of various issues. Table 3.1 outlines the detailed distribution of instances, genes and classes for each considered dataset.

Table 3.1: Microarray datasets used in the experiments

Dataset	No. of Instances	No. of Genes	No. of Classes
Brain_Tumour1	90	5920	5
Brain_Tumour2	50	10367	4
CNS	60	7129	2
DLBCL	77	5469	2
Leukemia	72	7129	2
Colon	62	2000	2
Lung Cancer	203	12600	4

3.7 Parameter setting

The proposed EBGWO is benchmarked with 2 novel versions of BGWO i.e. BGWO2 and BGWO1 [90], BPSO [146], BDE and BGA [146]. The optimizer-specific settings of the considered algorithms are presented in Table 3.2.

Table 3.2: Parameter settings for each considered algorithm

Algorithm	Year	Parameter settings
EBGWO	New	N=10, $MaxIter = 100$, $a_{initial} = 2$
BGWO1 [90]	2016	N=10, $MaxIter = 100$, $a_{initial} = 2$
BGWO2 [90]	2016	N=10, $MaxIter = 100$, $a_{initial} = 2$
BPSO [146]	2019	N=10, $MaxIter = 100$, $C_1 = C_2 = 2$, $V_{max} = 6$, $W_{max} = 0.9$, , $W_{min} = 0.4$
BDE [146]	2019	N=10, $MaxIter = 100$, $CR = 0.9$
BGA [146]	2019	N=10, $MaxIter = 100$, $CR = 0.8$, MR=0.01

Additionally, all the considered algorithms are repeated over 10 noncorrelated runs (i.e. N) to ensure statistical significance and stability of the achieved results. Furthermore, the commonly utilized 10-fold cross validation scheme is used to split the considered DNA chips into training and testing [154]. $MaxIter$ indicates the maximum number of iterations, $a_{initial}$ depicts the initial value of factor a that controls the balancing between exploitation and exploration. W_{max} and , W_{min} are the maximum and minimum inertia weights respectively.The inertia weight also strikes a balance between exploitation and exploration in the BPSO algorithm. C_1 is a cognitive acceleration constant while C_2 is a social acceleration constant. C_1 allows the definition of the ability of the group to be influenced by the best personal solutions attained over the iterations. On the other hand, C_2 allows the definition of the group to be influenced by the best global solution attained over the iterations. V_{max} is the maximum velocity that each particle can stochastically be accelerated towards its previous best position(personal best) and towards the best solution of the group (global best). CR is the crossover probability that controls the diversity of the BDE algorithm i.e. it controls the number of elements that can change in the algorithm. In the BGA algorithm, CR and MR are the crossover and mutation rates respectively. The crossover rate controls the swapping of solutions with others within chromosomes while the mutation rate controls the change of parts of one solution randomly, which increases the diversity of the population and thus providing a mechanism of avoiding trapping in the local optimum.

A wrapper technique based on the K-Nearest Neighbour (K-NN) classifier [90], [153] is used in selecting genes in this chapter. The K-NN classifier (whereby k is set to 5) is adopted to obtain the classification accuracy of the solutions.

Tables 3.3 to 3.9 presents the achieved results of all the techniques considered for the attribute selection task using the gene expression datasets whose details are presented in Table 3.1.

The following information is presented in each column of Tables 3.3 to 3.9:

i) Algorithm: presents the abbreviations of the considered techniques i.e. Excited Binary Grey Wolf Optimizer (EBGWO), Binary Grey Wolf Optimizer 1(BGWO1), and Binary Grey Wolf Optimizer 2 (BGWO2)

ii) Max_Acc: Maximum Accuracy value obtained when a given algorithm is repeated for 10 independent runs.

iii) Min_Acc: Minimum Accuracy value obtained when a given algorithm is repeated for 10 independent runs.

iv) Avg_Acc: Is the average of all the accuracy values obtained when a given algorithm is repeated for 10 independent runs.

v) Max_Nfeat: Is the largest count of attributes reported by a given technique during the 10 independent runs.

vi) Min_Nfeat: Is the largest count of features reported by a given technique during the 10 independent runs.

vii) Avg_Nfeat: Is the average of all the count of attributes reported by a given technique during the 10 independent runs.

viii) Dataset: Captures the datasets utilized in conducting the experiments as articulated in Table 3.1.

Table 3.3: Experimental Results for Brain_Tumour1 dataset

Algorithm	**Accuracy**			**Number of Genes**			**Dataset**
	Max_Acc	Min_Acc	Avg_Acc	Max_Nfeat	Min_Nfeat	Avg_Nfeat	
EBGWO (New)	**0.933**	**0.911**	**0.919**	**673**	**440**	**501.9**	Brain_Tumour1

BGWO1 [90]	0.889	0.856	0.871	*3831*	*2952*	*3356.9*	
BGWO2 [90]	0.911	0.878	0.894	1656	1094	1343.3	
BPSO [156]	*0.854*	*0.823*	*0.843*	2972	2763	2863.9	
BDE [156]	0.864	0.834	0.854	3017	2737	2937.6	
BGA [156]	0.869	0.844	0.859	2950	2840	2889.4	

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 3.4: Experimental Results for Brain_Tumour2 dataset

Algorithm	Accuracy			Number of Genes			Dataset
	Max_Acc	Min_Acc	Avg_Acc	Max_Nfeat	Min_Nfeat	Avg_Nfeat	
EBGWO (New)	**0.920**	**0.84**	**0.884**	**2811**	**712**	**1151.5**	Brain_Tumour2
BGWO1 [90]	0.840	0.820	0.838	*7415*	*6103*	*6813.4*	
BGWO2 [90]	0.880	0.820	0.846	4019	2528	3083.8	
BPSO [156]	0.800	0.780	0.798	5126	5090	5122.4	
BDE [156]	*0.728*	*0.713*	*0.714*	5198	5076	5172.3	
BGA [156]	0.767	0.753	0.752	5139	5039	5089.5	

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 3.5: Experimental Results for CNS dataset

Algorithm	Accuracy			Number of Genes			Dataset
	Max_Acc	Min_Acc	Avg_Acc	Max_Nfeat	Min_Nfeat	Avg_Nfeat	

EBGWO (New)	**0.85**	**0.8**	**0.827**	**1020**	**564**	**710.3**	CNS
BGWO1 [90]	0.783	0.750	0.760	*4942*	*4217*	*4606.4*	
BGWO2 [90]	0.800	0.750	0.780	2502	1842	2175.8	
BPSO [156]	0.767	0.733	0.737	3502	3486	3487.6	
BDE [156]	*0.693*	*0.663*	*0.683*	3530	3478	3521.9	
BGA [156]	0.727	0.707	0.717	3528	3428	3501.7	

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 3.6: Experimental Results for DLBCL dataset

Algorithm	**Accuracy**			**Number of Genes**			**Dataset**
	Max_Acc	Min_Acc	Avg_Acc	Max_Nfeat	Min_Nfeat	Avg_Nfeat	
EBGWO (New)	**1.000**	**0.987**	**0.997**	**534**	**333**	**426.7**	DLBCL
BGWO1 [90]	0.987	0.961	0.971	*3706*	*2826*	*3343.4*	
BGWO2 [90]	**1.000**	0.948	0.986	1700	1002	1408.3	
BPSO [156]	0.919	0.891	0.901	2703	2672	2675.1	
BDE [156]	*0.885*	*0.869*	*0.882*	2732	2687	2721.4	
BGA [156]	0.906	0.883	0.896	2709	2699	2685.1	

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 3.7: Experimental Results for Leukemia dataset

Algorithm	Accuracy			Number of Genes			Dataset
	Max_Acc	Min_Acc	Avg_Acc	Max_Nfeat	Min_Nfeat	Avg_Nfeat	
EBGWO (New)	**0.931**	**0.889**	**0.903**	**913**	**524**	**649.8**	Leukemia
BGWO1 [90]	0.861	0.833	0.849	*5065*	*3897*	*4428.5*	
BGWO2 [90]	0.889	0.847	0.874	2141	1618	1805.5	
BPSO [156]	0.828	0.809	0.814	3516	3505	3514.9	
BDE [156]	*0.782*	*0.751*	*0.784*	3537	3527	3531.2	
BGA [156]	0.801	0.782	0.792	3501	3461	3481.8	

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 3.8: Experimental Results for Colon dataset

Algorithm	Accuracy			Number of Genes			Dataset
	Max_Acc	Min_Acc	Avg_Acc	Max_Nfeat	Min_Nfeat	Avg_Nfeat	
EBGWO (New)	**0.935**	**0.903**	**0.919**	**220**	**103**	**143.4**	Colon
BGWO1 [90]	0.887	0.855	0.865	*1316*	*1096*	*1189.4*	
BGWO2 [90]	0.919	0.871	0.900	622	351	455.2	
BPSO [156]	0.849	0.829	0.839	986	931	936.5	
BDE [156]	*0.810*	*0.780*	*0.794*	995	955	965.3	
BGA [156]	0.881	0.875	0.878	990	984	987.3	

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 3.9: Experimental Results for Lung Cancer dataset

Algorithm	Accuracy			Number of Genes			Dataset
	Max_Acc	Min_Acc	Avg_Acc	Max_Nfeat	Min_Nfeat	Avg_Nfeat	
EBGWO (New)	**0.985**	**0.970**	**0.977**	**1148**	**781**	**1005.5**	Lung Cancer
BGWO1 [90]	0.966	0.941	0.951	*7598*	*6621*	*7211*	
BGWO2 [90]	0.975	0.956	0.966	2672	2167	2413.2	
BPSO [156]	0.936	0.931	0.935	6196	6179	6180.7	
BDE [156]	*0.931*	*0.921*	*0.924*	6256	6218	6226.8	
BGA [156]	0.952	0.939	0.945	6235	6214	6218.2	

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

3.8 Conclusion

In this chapter, an excited binary grey wolf optimizer (EBGWO) is proposed to solve the feature selection problem in the gene-rich DNA microarray datasets. In the proposed algorithm, the concept of the complete current response of a direct current (DC) excited resistor capacitor (RC) circuit are innovatively utilized to make the non-linear control strategy of parameter a of the GWO adaptive. Since this scheme allocates a large proportion of the number of iterations to global exploration compared to local exploitation, the convergence speed of the proposed EGWO algorithm is enhanced while minimizing the local minima trapping effect. Moreover, since the proposed scheme assigns each wolf a value of parameter a that is proportional its fitness values in both the search space and the current iteration (generation), diversity and the quality of the solutions is improved as well.

To overcome premature converge (a limitation of existing versions of GWO algorithms) and still maintain the social hierarchy of the pack, a new position-updated equation utilizing the fitness values of vectors $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ is proposed in determining new candidate individuals. As a feature selector, EBGWO is compared with five metaheuristic algorithms i.e. BGWO1, BGWO2, BPSO, BDE and BGA that are in existence. The obtained experimental results revealed that EBGWO yielded the best performance and overtook the other algorithms. EBGWO not only attained the highest classification accuracy, but also selected subsets with the least number of informative features (genes). In conclusion, the proposed EBGWO is successful, and more appropriate to be used as a feature selector in highly dimensional datasets. For further works, various chaotic maps can be adopted in fine-tuning the parameters of the EBGWO. Utilizing EBGWO as a hybrid filter-wrapper for selecting features seeking to evaluate the generality of the attributes selected will be another useful contribution. Moreover, EGWO can be adopted in other areas that require optimization like knapsack, training a neural network and numerical problems.

Though the proposed EBGWO wrapper has proved attractive in selecting informative genes from the highly dimensioned DNA microarray datasets, it does not attain an optimal balance between exploitation and exploration during the search process.This is because exploitation and exploration are two contradicting principles, which must be balanced efficiently in order to achieve an improved performance of a metaheuristic. Moreover, attaining an optimal balance between these antagonist principles is difficult with a single metaheuristic [162]. In tying to attain the required optimal balance between exploitation and exploration, another novel hybrid wrapper combining a variant of the proposed EBGWO and adaptive cuckoo search is proposed in the next chapter.

CHAPTER FOUR: AN INNOVATIVE EXCITED-ACS-IDGWO ALGORITHM FOR OPTIMAL BIOMEDICAL DATA FEATURE SELECTION

4.1 Introduction

Currently, there is a growing research interest in developing and deploying population-based metaheuristics to tackle combinatorial optimization challenges. This is because they are simple, flexible with an inexpensive computational cost, and gradient-free[155]. Many researchers have applied these optimization algorithms in various research domains because of their ability to achieve best solutions.

The optimization challenge grows bigger when tackling highly dimensioned datasets. This is because these datasets have a vast feature space with many classes. Due to the presence of redundant and non-informative attributes within these datasets, the process of effective machine learning greatly hindered. Thus, the construction of efficient classifiers with high predictive power largely depends on selection of informative features[44].

Feature selection (FS) is one of the main steps in data preprocessing that aims at selecting a subset of attributes out of the whole dataset resulting into removal of noisy non-informative and redundant features. This in turn increases the accuracy of a considered classifier or clustering model [156].

FS algorithms can be broadly categorized into two classes: filter and wrapper techniques[157],[158]. Filters include techniques independent of classifiers and work directly on presented data. Moreover, these methods in many situations determine the correlations between features. On the contrary, wrapper approaches engage classifiers and mainly determine interactions between dataset features. From literature, wrapper approaches have proved to be superior compared to filters for classification algorithms[159],[160].

To utilize wrapper-based techniques, three key factors need to be outlined: considered classifiers (i.e. (KNN), SVM), evaluation criteria for the identified feature subset, and a search technique utilized in determining a subset of optimal features [161].

Many researchers have pointed out that determining an optimal subset of attributes is not only challenging but computationally expensive as well. Though, in the recent past, metaheuristics have proved to be reliable and efficient tools in tackling many optimization tasks (e.g., engineering designs problems, machine learning, feature selection, and data mining), they are not efficient in solving problems with high computational burden[158],[162], [163],[164] .

In the recent past, various metaheuristic search techniques have been adopted for FS using highly dimensioned datasets. Some of these metaheuristics are the GWO [90], [165], GA [166], PSO[164], ACO[83], DEA [143], CSA[153], and DA[167]. Though, many of these algorithms have already made an important contribution in the field of feature selection, in many cases, they offer acceptable solutions without a guarantee of determining optimal solutions since they do not explore the entire search space[164].

Some of the new modifications that have been suggested to improve the performance of these metaheuristics include chaotic maps[168], evolutionary methods [169], sine cosine algorithms [170], biogeography-based optimization, and local searches [171].

While designing or utilizing a metaheuristic, it should be noted that diversification and intensification are two contradicting principles that must be balanced efficiently in order to achieve an improved performance of the metaheuristic [162].

With regard to this, one promising option is developing a memetic approach whereby an integration of (at least) 2 techniques is done with the objective of enhancing the overall performance.

Motivated by this, various hybrid algorithms have been suggested in the recent past to solve a variety of optimizations and feature selection problems [60]. However, to enhance diversification and intensification of these hybrid algorithms, exploration and fine-tuning within their basic constituent algorithms is needed prior to hybridization [172].This emphasizes, too, that there are a number of techniques lying within these memetic algorithms that are yet to be investigated.

Firstly, the technique of combining one or more nature-inspired algorithms (NIAs) needs to be determined. Secondly, the criterion of determining how many NIAs need to be combined within the search space has to be accomplished. Thirdly, the method of determining the application area upon which the proposed memetic algorithm will be applied has to be done. Finally, the criterion of applying the memetic algorithm in a specific domain has to be accomplished [24].

Although the proposed EBGWO in chapter 3 is able to report a subset with the least number of features while maintaining an attractive classification accuracy, it does not attain an optimal balance between exploitation and exploration. This is because exploration of search domain and exploitation of optimal solutions are two conflicting principles that must be considered when modelling metaheuristics.Inspired by this, this chapter proposes a new hybrid algorithm called Excited- (E-) Adaptive Cuckoo Search- (ACS-) Intensification Dedicated Grey Wolf Optimization (IDGWO), i.e., EACSIDGWO algorithm to solve the feature selection problem in biomedical science. In the proposed algorithm, the concept of the complete voltage and current responses of a direct current (DC) excited resistor capacitor (RC) circuit is innovatively utilized to make the step size of ACS and the nonlinear control strategy of parameter $\vec{a}$ of the IDGWO adaptive. Since the population has a higher diversity during early stages of the proposed algorithm, both the ACS and IDGWO are jointly utilized to attain accelerated convergence. However, to enhance mature convergence while striking an effective balance between exploitation and exploration in latter stages, the role of ACS is switched to global exploration while the IDGWO is still left conducting the local exploitation.

The remainder of this chapter is organized as follows: Section 4.2 discusses the existing literature within the same research domain. Section 4.3 presents the background information of the CS and the GWO, respectively, where their inspirations and mathematical models are given emphasis. The continuous version of the proposed EACSIDGWO algorithm is presented in Section 4.4 while the details of its binary version are given in Section 4.5. The experimental methodology considered in this chapter is presented in Section 4.6 while the results on feature selection are discussed in Section 4.7. Finally, the conclusion is given in Section 4.8.

4.2 Related Works

4.2.1. Review of Hybridization of GWO with Other Search Algorithms

Combining two or more metaheuristics to attain better solutions is currently a new insight in the area of optimization. In the literature, many researchers have utilized GWO in the field of hybrid metaheuristics. For instance, in [173], a hybrid of GWO and ABC is proposed to improve performance of a complex system. In [174], GWO is hybridized with ant lion optimizer (ALO) for wrapper feature selection. Alomoush et al. [175] proposed a hybrid of GWO and harmony search (HS). In this memetic, GWO updates the bandwidth and pitch adjustment rate in HS, which in return improves the global optimization abilities of the hybrid algorithm. In [176], Arora et al. combined GWO with the crow search algorithm (CSA). The performance of the derived memetic as a feature selector is evaluated using 21 datasets. The obtained results reveal that the combined algorithm is superior in solving complex optimization algorithms. In [177], a novel combination between GWO and PSO is utilized as a load-balancing technique in the cloud-computing arena. The conclusions point out that the hybrid algorithm improved both the convergence speed and the simplicity in comparison with other algorithms. Zhu et al. [178] hybridized GWO with differential evolution (DE). The hybrid algorithm was tested on 23 different functions and a nondeterministic polynomial hard problem. The obtained results indicate that this combination achieved superior exploration. In [179], a new memetic combining the exploration ability of the fireworks algorithm (FWA) with the exploitation ability of GWO is proposed. Utilizing 16 benchmark functions with varied dimensions and complexities, the experimental results indicate that the hybrid algorithm attained attractive global search abilities and convergence speeds.

4.2.2. Review of Hybridization of CS with Other Search Algorithms.

Utilizing the concept of random and best agents within a population, Cheng et al. [180] developed an ensemble cuckoo search variant combining three different CS approaches that coexist within the entire search domain. These CS variants actively compete to derive superior generations for numerical optimization. To maintain population diversity, he introduced an external archive. The statistical results obtained reveal that the ensemble CS attained attractive converge speeds as well as robustness. In [181], GWO is hybridized with CS, i.e., GWOCS for the extraction of parameters for different PV cell models situated in different conditions. Zhang et al. [182] developed an ensemble CS algorithm that foremost divides a population into two smaller groups and then utilizes CS and differential evolution (DE) on the derived subgroups independently. The subgroups are free to share useful information by division. Further, the CS and DE algorithms can freely utilize each other’s merits to complement their weaknesses. This approach proved to balance the quality of solutions and the computation consumption. In [182], CS is hybridized with a covariance matrix adaptation evolution approach, i.e., CMA-CS to improve the performance of CS in different optimization problems.

Despite the advantages portrayed by the aforementioned hybrid GWO and CS metaheuristics for optimization and feature selection, superior hybrid approaches can be achieved if the single

GWO and CS algorithms are improved prior to hybridization. Furthermore, the no-free-lunch (NFL) theorem has logically proved that there has been, is, and will be no single metaheuristic capable of solving all optimization and feature selection problems[181]. While a given metaheuristic can show an attractive performance on specific datasets, its performance might degrade when applied to similar or different types of datasets [182]. Thus, there is still a dire need to improve existing algorithms or develop new ones to solve problems that require function optimization as well as selection of features efficiently.

4.3 Standard Cuckoo Search (CS)

4.3.1 Inspiration of CS

The following subsections describe the inspiration of the CS algorithm.

4.3.1.1 The Behavior of Cuckoo Birds

To date, more than a thousand different species of birds are in existence in nature [183]. For most of these species, the female birds lay eggs in nests they have built themselves [184]. However, there exist some types of birds that do not build nests of their own, but instead lay their eggs in other different species' nests, leaving the responsibility of taking care of their eggs to the host birds. The cuckoos are the most famous of these brood parasites [185].

There are three types of brood parasites: intraspecific brood parasites, cooperative breeding, and nest takeover [38].The cuckoo strategy is full of amazing traits; foremost, it replaces one host egg with its own to increase the chances of its egg being hatched by the host bird. Next, it tries to mimic the pattern and color(s) of this host eggs with the aim of reducing the chances of its egg being noticed and discarded by the host bird. It is also important to point out that the timing of laying its egg is amazing since it cleverly selects a nest where a host bird has just laid eggs, implying that the cuckoo's egg will hatch prior to the host eggs. The first action taken by the hatched cuckoo is evicting the host eggs that are yet to hatch out of the nest by blind propelling in order to increase its chances of being fed well by the host bird [185]. In addition, this young cuckoo mimics the call of host chicks thus enhancing more access to the food provided by the host bird [186].

However, if this host bird is able to identify the cuckoo's egg, it can either discard it from the nest or quit this nest to build a completely new nest in a different location.

4.3.1.2 $Le'vy$ Flights

From literature, many researchers have shown that the behavior of many flying animals, birds, and insects can be demonstrated by a $Le'vy$ flight [187],[188],[189],[190]. $Le'vy$ flights are evident when some birds, insects, and animals follow a long path with sudden turns in combination with random-short moves [190].These $Le'vy$ flights have been successfully applied in optimization [188],[190],[191],[192]. A $Le'vy$ flight is a random walk characterized with step lengths whose distribution is according to a heavy-tailed probability distribution.

4.3.2 Cuckoo Search (CS) Algorithm

CS is a metaheuristic swarm-based global optimization based on cuckoos that was proposed by Yang and Deb in 2009.The CS combines the obligate brood parasitic nature of cuckoos with the $Le'vy$ flight existing in fruit flies and some birds [193]. There are three basic idealized rules for the CS, namely:

(i) A female cuckoo lays one egg at a time and puts it in a randomly chosen nest.

(ii) The best nests with high-quality eggs (highest fitness/solutions) will carry over to the next generations.

(iii) The number of available host nests is kept fixed, and the host bird can discover the egg laid by the female cuckoo (alien egg) with a probability $P_a \in [0,1]$. Depending on the value of P_a, the host bird can either throw away the alien egg or abandon the nest. An assumption that only a fraction of P_a nests are replaced by new ones.

Based on the above rules, an illustration of the CS is shown in Algorithm 1.

Algorithm 1: Pseudo-code for the standard CS

```
Begin:
 Initialize $P_a = 0.25$
 Define objective function$f(x), x = (x_1, x_2, \dots, x_d)$, where $d$ is the number of dimensions
Generate initial population of $n$ host bird nests,$X_i (i = 1,2, \dots, n)$
while $g \leq g_{max}$ or any other stopping criteria
Generate a new cuckoo (solution) randomly via $Le'vy$ flight according to Equation (4.1)
Evaluate the fitness of the new cuckoo,$F_i$
```

```
Randomly choose a nest from among the host nests $n$ (For example $j$)
if $F_i > F_j$ then
    Replace nest j by the new cuckoo $i$
end
Abandon a fraction of $P_a$ worst nests and generate new ones according to Equation (4.6)
Keep best solutions(or those nests with quality solutions)
Rank these solutions, then keep the current best
end while
Report the final best
end
```

4.3.2.1 Mathematical modelling of the standard CS

Considering Algorithm 1, the standard CS has three major steps [194], [195], [196]:

i) Exploitation (intensification) by the use of $Le'vy$ flight random walk (LFRW)

ii) Exploration (diversification) using biased selective random walk (BSRW)

iii) Elitist scheme via greedy selection

4.3.2.2 Intensification Using $Le'vy$ Flight Random Walk (LFRW).

In this phase, new solutions are generated around the current best solution, which in return enhances the speed of the local search. This phase is achieved via the LFRW that is generally presented in Equation 4.1 where the step size is derived from the $Le'vy$ distribution.

$$X_{i,gen+1} = X_{i,gen} + \alpha \oplus \text{Le'vy}(\lambda) \qquad 4.1$$

Where $X_{i,gen}$ is the i^{th} nest in the gen^{th} generation and $X_{i,gen+1}$ is a new nest generated by the $Le'vy$ flight. $\oplus$ implies entry-wise multiplications and α is the step size where $\alpha > 0$ and is formulated in Equation 4.2.The formula in Equation 4.2 ensures that a new solution will be close to the current best-solution.

$$\alpha = \alpha_0 \times (X_{i,gen} - X_{best}) \qquad 4.2$$

Where X_{best} is the current solution and α_0 is a scaler that is set to 0.01 in the standard CSA [38, 49]. $Le'vy$ (λ) is a random number derived from the $Le'vy$ distribution and is formulated in Equation 4.3.

$$\text{Le'vy}(\lambda) \sim \frac{\partial \times \varepsilon}{|\varphi|^{\frac{1}{\lambda}}} \tag{4.3}$$

Where λ is a constant whose value is 1.5 as suggested by Yang is in the standard CS [39]. ε and φ are random numbers derived from a normal distribution whose mean and standard deviation is 1. ∂ is a parameter computed in Equation 4.4.

$$\partial = \left(\frac{\lceil(1+\lambda)\times\sin\left(\frac{\pi\times\lambda}{2}\right)}{\lceil\left(\frac{1+\lambda}{2}\times\lambda\times2^{\frac{\lambda-1}{2}}\right)} \right)^{\frac{1}{\lambda}} \tag{4.4}$$

Where ⌈ is a gamma function. The final form of $Le'vy$ (λ) flight random walk (LFRW) is a combination of equations 4.1 to 4.4 as presented in equation 4.5.

$$X_{i,gen+1} = X_{i,gen} + \alpha_0 \frac{\partial \times \varepsilon}{|\varphi|^{\frac{1}{\lambda}}} (X_{i,gen} - X_{best}) \tag{4.5}$$

4.3.2.2 Diversification by the use of biased-selective random walk (BSRW)

In this phase, new solutions are randomly generated in locations far from the current best solution. An approach that ensures that the CSA is not trapped in the local optimum thus enhancing suitable diversity and exploration of the entire search space [196]. This phase of the CSA is achieved by utilizing the BSRW which is efficient in exploring the entire search space especially when it is large since the step-size in the $Le'vy$ flight is much longer in the long run [194],[196].

To find new solutions that are far from the current best solution, foremost, a trial solution is obtained by using a mutation of the current best solution and a differential step size from two solutions selected randomly. Then a new solution is derived from a crossover operator between the current best solution and the two trial solutions [196].The formulation of the BSRW is given in Equation 4.6 [195].

$$X_{i,gen+1} = \begin{cases} X_{i,gen} + s\times\left(x_{a,j,gen} - x_{b,j,gen}\right) \; with \; P_a \\ \quad X_{i,gen} \; with \; the \; remaining \; P_a \end{cases} \qquad 4.6$$

Where a and b are two random indexes, s is a random number in the range [0, 1] and P_a is the probability discovery whose best value is 0.25 [193], [196] .

4.3.2.3 Elitist scheme via greedy selection

After each random walk process, the cuckoo search algorithm utilizes the greedy strategy to select solutions with better fitness values that will be passed to the next generation. This facilitates maintenance of good solutions [196].

4.4 Excited -Adaptive Cuckoo Search- Intensification dedicated Grey Wolf Optimization (EACSIDGWO)

In general, effective balancing between diversification (global search) and intensification (local search) in a metaheuristic plays a beneficial and crucial role in achieving excellent performance of an algorithm [199], [200], [201]. However, it is difficult to achieve this balance with a single metaheuristic (for example either using CSA or GWO) [199], [200]. For instance, CSA is efficient at exploring the promising area of the whole search space (diversification) but ineffective at fine-tuning the end of the search space (exploitation/intensification) [202], [203]. On the other hand, GWO is good at intensification (exploitation) but inefficient at diversification (exploration) [180], [148].

For this reason, to enhance mature convergence while ensuring that the required effective balance between diversification and intensification is met, a hybrid algorithm called Excited-Adaptive Cuckoo Search-Intensification Dedicated Grey Wolf Optimization (EACSIDGWO) utilizing the strengths of each algorithm (i.e. CSA's diversification and GWO's intensification abilities) is proposed in this thesis. Moreover, the adaptability of the proposed EACSIDGWO is guided innovatively by the complete voltage and current responses of a dc excited RC circuit (whose analysis results in first order differential equations) that find continual applications in electronics, communications and control systems [151].

4.4.1 Adaptive Cuckoo Search (ACS)

4.4.1.1 Adaptive step size via the complete voltage response of the dc excited RC circuit

From the details of the standard CS algorithm presented in section 4.3, it is evident that the algorithm lacks a criterion to control its step size through the iteration process. Control of the step size is key in guiding the CS algorithm to reach either its global maxima or minima [196], [204].

Inspired by the complete voltage response of a direct current (dc) excited RC circuit which increases with time, a novel mechanism to control the step size is proposed. Contrary to prior research [196], [204]where the step size decays with generations, in this research the step size grows with generations with the aim of strengthening the diversification (exploration) ability of the CS, which is a component of the proposed EACSIDGWO algorithm.

The solution to first order differential equation of the direct current excited RC circuit motivated the formulation of a new variant of ACS in this chapter.

The complete voltage response of the RC circuit to a sudden application of a dc voltage source, with the assumption that the capacitor is initially not charged is given in Equation 4.7.

$$v(t) = \begin{cases} 0, & t < 0 \\ V_s\left(1 - e^{-t/\tau}\right), & t > 0 \end{cases} \qquad 4.7$$

Where $\tau = R * C$ is the time constant, which expresses the rapidity with which the voltage $v(t)$ rises to the value of V_s which is a constant dc voltage source. R and C are the equivalent resistance and capacitance in the circuit.

Considering the situation when $t > 0$, Equation 4.7 can be rewritten as presented in Equation 4.9

$$v(t) = V_s(1 - (e^{-t})^{\tau}) \qquad 4.8$$

$$v(t) = V_s(1 - (\frac{1}{e^{t}})^{\tau}) \qquad 4.9$$

As $t \to \infty$, the component $\frac{1}{e^t} \to 0$ forcing $v(t \to \infty) \to V_s$. We adopt this concept i.e. the exponential growth of $v(t)$ to control the step size of the cuckoo search algorithm by introducing the proposed Equation 4.10.

$$step_{gen+1} = step_{Max} \times (1 - (\frac{gen_{Max} - gen}{gen_{Max}})^{\tau}) \qquad 4.10$$

Where gen is the current generation (iteration), $step_{Max}$ is the upper bound of the step size $step$ and gen_{Max} is the maximum number of generations (iterations).

To ensure that the $step_{gen+1}$ is proportional to the fitness of a given individual nest within the search space in the current generation, the non-linear modulation index τ is formulated in Equation 4.11.

$$\tau_{i,gen} = \left| \frac{\left(\frac{\alpha_{nestf_{gen}} + \beta_{nestf_{gen}} + \delta_{nestf_{gen}}}{3}\right) - \mathrm{i}_{nestf_{gen}}}{\left(\frac{\alpha_{nestf_{gen}} + \beta_{nestf_{gen}} + \delta_{nestf_{gen}}}{3}\right) - \mathrm{worst}_{nestf_{gen}}} \right| \qquad 4.11$$

Where $\tau_{i,gen}$ is the the non-linear modulation index for i^{th} nest in generation gen, $\alpha_{nestf_{gen}}$is the fitness value of the alpha(α) nest (overall best nest) in generation gen, $\beta_{nestf_{gen}}$ is the fitness value of the beta (β) nest (2nd best nest) in generation gen, $\delta_{nestf_{gen}}$ is the fitness value of the delta (δ) nest (3rd best nest) in generation gen, $\mathrm{i}_{nestf_{gen}}$ is the fitness value of the i^{th} nest in generation gen and $\mathrm{worst}_{nestf_{gen}}$ is the fitness value of the worst nest among the remaining omega(ω) nests (i.e. nests whose fitness values do not feature among the top three fitness values).

Thus, Equation 4.10 is further modified as Equation 4.12.

$$step_{i,gen+1} = step_{Max} \times (1 - (\frac{gen_{Max} - gen}{gen_{Max}})^{\tau_{i,gen}}) \qquad 4.12$$

Where $step_{i,gen+1}$ is the step size for the for i^{th} nest in generation $gen + 1$.

From Equation 4.12, the step size $step_{i,gen+1}$ is non-linearly increasing from relatively small values to values close to $step_{Max}$. The reason for proposing a non-linearly increasing strategy are as follows. Foremost, at the early stages of the proposed EACSIDGWO algorithm, whereby ACS is a component, the population has a higher diversity. A higher diversity imply a stronger ability to explore the global space. Our aim at this point is to accelerate convergence. Therefore, the value of the step size $step_{i\ ,gen+1}$ is set to a smaller value.

It is important to point out that the anticipated accelerated convergence is a joint effort attained by foremost setting the $step_{i\ ,gen+1}$ of the ACS to a small value at early stages, and utilizing the IDGWO (a variant of the EGWO algorithm whose details are presented in section 3.3) whose core task is exploitation.

On the other hand, since the proposed EACSIDGWO algorithm is a hybrid algorithm where the ACS cooperatively works with the *IDGWO,* all the nests will be attracted to the global optima i.e. the alpha (α) nest at the later stage. This will compel them to converge prematurely without being given enough room to explore the search space. Such a situation will lead the nests away from a local optimum, and encourage diversification. For this reason, the value of the step size $step_{i\ ,gen+1}$ is set to a larger value i.e. $step_{Max}$. In this thesis the $step_{Max}$ is set to 1.

In other words, the main reason for proposing a non-linearly increasing step size $step_{i\ ,gen+1}$ is that its small values at the initial stages of the proposed EACSIDGWO algorithm facilitates "local exploitation" while its larger values in the later stages will facilitate "global exploration".

The ACS can then be modeled as presented in Equation 4.13.

$$X_{i,gen+1} = X_{i,gen} + randn\times\ step_{i,gen+1} \qquad 4.13$$

Equation 4.13 is a formulation of the new search space for the ACS from the current solution.

Moreover, if this step size is considered proportional to the global best solution, then Equation 4.13 can be formulated as given in Equation 4.14.

$$X_{i,gen+1} = X_{i,gen} + randn\times\ step_{i,gen+1}{*}(X_{i,gen} - X_{gbest,gen}) \qquad 4.14$$

Where $X_{gbest,gen}$ is the global best solution among all X_i for $i = 1,2, \dots, n$ at generation gen ,and n is the number of host bird nests.

Thus, from Equations 4.10 – 4.14 it is evident that the diversification ability of the ACS is heightened as the number of generations (gen) approach the maximum number of generations

(gen_{Max}). This is because the value of the step size rapidly increases towards the set maximum value of step ($step_{Max}$).

4.4.2 Intensification dedicated grey wolf optimizer (IDGWO)

4.4.2.1 Nonlinearly controlling parameter $\vec{a}$ via the complete current response of the dc excited RC circuit

It is evident from sub-section 2.4.4 that parameter $\vec{a}$ plays a critical role in balancing the diversification (exploration) and the intensification (exploitation) of a search agent.

A large value of control parameter $\vec{a}$ facilitates diversification while a smaller value of this parameter facilitates intensification. Thus, a suitable selection of the control parameter $\vec{a}$ can enhance a good balance between global diversification (exploration) and local intensification (exploitation).

In the original GWO (described in section 2.4.4), the value of $\vec{a}$ linearly decreases from 2 to 0.(refer to Equation 2.39). However, the search process of the GWO algorithm is both non-linear and complicated, which cannot be truly reflected by the linear control strategy of $\vec{a}$ presented in Equation 2.39.

In addition, Mittal [150] proposed that an attractive performance can be attained if parameter $\vec{a}$ is non-linearly decreased rather than decreased linearly.

Inspired by the complete current response of a direct current (dc) excited RC circuit which decreases with time, a novel nonlinear adjustment mechanism of control parameter $\vec{a}$ is formulated in this chapter.

The complete current response of the RC circuit to a sudden application of a dc voltage source, with the assumption that the capacitor is initially not charged is given in Equation 4.15.

$$i(t) = \frac{V_s}{R}((\frac{1}{e^t})^{\tau}) \qquad 4.15$$

As $t \rightarrow \infty$, the component $\frac{1}{e^t} \rightarrow 0$ forcing $i(t \rightarrow \infty) \rightarrow 0$. Using this concept i.e. the exponential decay of $i(t)$ to formulate a novel improved strategy i.e. Equation 4.16 to generate the values for control parameter $\vec{a}$.

$$\vec{a}_{i,gen}=a_o\times(\frac{gen_{Max}-gen}{gen_{Max}})^{\tau_{i,gen}} \qquad 4.16$$

Where gen is the current generation (iteration), a_o is the initial higher value of parameter a and gen_{Max} is the maximum number of generations (iterations). $\tau_{i,gen}$ is the non-linear modulation index described earlier by Equation 4.11.

Consequently, vector $\vec{A}$ is computed as given in Equation 4.17.

$$\vec{A}=2\vec{a}_{i,gen}.\vec{r}_1-\vec{a}_{i,gen} \qquad 4.17$$

Equation 4.16 is a non-linear decreasing control parameter for $\vec{a}_{i,gen}$ whose initial upper limit is equal to the value a_o while its final lower limit is zero.

From the original literature of GWO, the value $|\vec{A}|<1$ compels the grey wolves to move towards the prey (exploitation) while $|\vec{A}|>1$ compels them to move away from the prey in search of a fitter prey (exploration).Thus, setting a_o to 1 will always force the wolves to move to the prey which will enable us dedicate modified GWO algorithm, a component of proposed EACSIDGWO, for intensification.

4.4.2.2 Enhanced mature convergence via a fitness value based position-updating criterion

Both diversification and intensification are crucial for population-based optimization algorithms [150]. However, from the detailed account of the conventional GWO (refer to section 2.4.4), it is evident that all the other wolves are attracted towards the three leaders α, β and δ , a scenario that will force the algorithm to converge prematurely without attaining sufficient diversification of the search space. In other words, the conventional GWO is prone to pre-mature convergence.

In reference to the position-updated criterion of GWO described by Equation 4.11, a new candidate individual is obtained by moving the old individual towards the best leader ($\alpha\ wolf$), the second best leader (β $wolf$) and the third best leader (δ $wolf$). This approach will force all the other grey wolves to crowd in a reduced section of the search space that might be different from the optimal region, and without giving them a leeway to escape from such a region. In an effort to overcome this major drawback, in this chapter a scheme that promotes mature convergence is devised.

Instead of averaging values of vectors $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ (a form of recombining them) as a mechanism of updating the wolves' positions (refer to Equation 2.40), this chapter makes full use of information of their fitness values as a criteria of arriving at new positions for the wolves.

Foremost the search agents of the populations $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ are computed as given in Equations 4.18 - 4.20.

$$\vec{X}_1(i,j) = \vec{X}_\alpha(j) - \vec{A}_1.\vec{D}_\alpha \qquad 4.18$$

$$\vec{X}_2(i,j) = \vec{X}_\beta(j) - \vec{A}_2.\vec{D}_\beta \qquad 4.19$$

$$\vec{X}_3(i,j) = \vec{X}_\delta(j) - \vec{A}_3.\vec{D}_\delta \qquad 4.20$$

Where $i = 1,2,\dots,n$ and $j = 1,2,\dots,d$. n is the population size while d is the dimension of the search space.

Next, the fitness value for each search agent in each of the derived populations i.e. $\vec{X}_1, \vec{X}_2$ and $\vec{X}_3$ is evaluated. Further a new population with the fittest values is derived from these three populations i.e. $\vec{X}_1, \vec{X}_2$ and $\vec{X}_3$.

Equations 4.21-4.22 represents the process undertaken to derive this new population.

$$[Fit_{max}, Index] = \max\left(\bigcup_{j=1}^{3} X_j f_{i,gen}\right) \qquad 4.21$$

$$X_{i,gen+1} = \bigcup_{j=1}^{3} \vec{X}_{j\ i,gen_{Index}} \qquad 4.22$$

Where $\vec{X}_{j\ i,gen}$ is vector j computed using search agent i during iteration gen, $X_j f_{i,gen}$ is the fitness value of vector $\vec{X}_{j\ i,gen}$.

4.4.3 Proposed EACSIDGWO (Continuous version)

The Excited-ACSIDGWO cooperatively combines the adaptive cuckoo search (ACS) and the intensification-dedicated grey wolf optimization (IDGWO). In the EACSIDGWO algorithm, the ACS is actively involved in intensification (exploitation) during the early stage when the population has higher diversity and diversification at later stages. On the other hand, the IDGWO is only actively involved in intensification in all the stages of the proposed algorithm. By doing so, an effective balance between diversification and intensification is achieved. In addition, mature convergence is enhanced which in the end leads to high quality solutions.

4.4.4 Proposed EACSIDGWO (Binary version)

Selection of features is binary by nature [61]. Therefore, the proposed EACSIDGWO algorithm cannot be utilized in selection of features without further modifications.

In the proposed EACSIDGWO algorithm, the new positions of the search agents will have continuous solutions, which must be converted into corresponding binary values.

In this chapter, this conversion is achieved by foremost applying squashing of the continuous solutions in each dimension using a sigmoid (S-shaped) transfer function [205].This compels the search agents to move into a binary search space as depicted by Equation 4.23.

$$S = \frac{1}{1 + e^{-10({X^d}_{i,gen} - 0.5)}} \qquad 4.23$$

Where ${X^d}_{i,gen}$ is a continuous-valued position of the i^{th} search agent in the d^{th} dimension during generation gen.

The output S of the sigmoid transfer function is still a continuous value and thus it has to be the threshold to reach the binary-value one. Normally, the sigmoid function maps smoothly the infinite input to a finite output [205].To arrive at the binary solution when a sigmoid function is used, the commonly stochastic threshold is applied as presented in Equation 4.24.

$${y^d}_{i,gen} = \begin{cases} 0 & if\ rand < S \\ 1 & if\ rand \geq S \end{cases} \qquad 4.24$$

$$Y_{i,gen} = \bigcup_{i=1}^{n} {y^d}_{i,gen} \qquad 4.25$$

Where ${y^d}_{i,gen}$ is the binary updated position at generation gen in the d^{th} dimension and $rand$ is a random number drawn from a uniform distribution ∈ [0,1]. $Y_{i,gen}$ is the equivalent binary vector of the i^{th} search agent at generation gen.

Using this approach, the original solutions remain in the continuous domain of the proposed EACSIDGWO algorithm and can be converted to binary when need arises.

The pseudocode of the binary version of the proposed EACSIDGWO algorithm is presented in Algorithm 3.

Algorithm 3: Pseudo-code for the EACSIDGWO (Binary Version)

```
Input: labelled biomedical dataset D, MaxIter, ACS and IDGWO parameters value, number of host bird nests ($n$), number of dimensions (features) $d$, Lower bound ($L_b$) and Upper bound ($U_b$)
Output: Best Fitness , Best Search Agent
for each nest i (i =1, 2...n) do
  for each dimension j(j=1,2,…,d) do
    ${X^j}_{i,0}$=random number drawn from $[L_b, U_b]$
  end
  Convert continuous values of $X_{i,0}$ to binary using Eq. 4.23, 4.24 and 4.25
  Train a classifier to evaluate the accuracy of the equivalent binary vector of $X_{i,0}$ and store the value in $Xf_{i,0}$
end
[~, Index]=Sort ( $Xf_0$,'descend')
$\alpha_{nestf_0}$= $Xf_0(Index(1))$
$\beta_{nestf_0}$= $Xf_0(Index(2))$
$\delta_{nestf_0}$= $Xf_0(Index(3))$
$\text{worst}_{nestf_0}$= $Xf_0(Index(n))$
$\alpha_{nest_0}$= $X_0(Index(1))$
```

```
14 $\beta_{nest_0} = X_0(Index(2))$
15 $\delta_{nest_0} = X_0(Index(3))$
16 While ($gen \le MaxIter$)
17   for each nest i (i =1, 2...n) do
18     Calculate $\tau_{i,gen}$ and $step_{i,gen+1}$ using
       Eq. 4.11 and 4.12 respectively
19     Generate a new cuckoo nest $X_{i,gen+1}$
       using Eq. 4.14
20 Convert continuous values of $X_{i,gen+1}$ to binary using Eq. 4.23, 4.24 and 4.25
21 Train a classifier to evaluate the accuracy of the equivalent binary vector of $X_{i,gen+1}$ and store
   the value in $Xf_{i,gen+1}$
22 if( $Xf_{i,gen+1} > Xf_{i,0}$) then
23   $Xf_{i,0} = Xf_{i,gen+1}$
24   $X_{i,0} = X_{i,gen+1}$
25 end
26  end
27 Repeat step 8 to 15
28 for each nest i (i =1, 2...n) do
29 Calculate $\tau_{i,gen}$ and $a_{i,gen}$ using  Eq. 4.11 and 4.16 respectively
30  for each dimension j(j=1,2,...,d) do
31 Calculate coefficients $A$ and $C$ as shown in Eq. 4.16 and Eq. 4.10 respectively
```

```
32 Compute vectors$\vec{X}_{1_{i,gen}}(j)$, $\vec{X}_{2_{i,gen}}(j)$ and $\vec{X}_{3_{i,gen}}(j)$ using Eq. 4.18, 4.19 and 4.20 respectively.
33 end
34 Convert continuous values of $\vec{X}_{1_{i,gen}}$, $\vec{X}_{2_{i,gen}}$ and $\vec{X}_{3_{i,gen}}$ to binary using Eq. 4.23, 4.24 and 4.25
35 Consecutively, train a classifier to evaluate the accuracies of the equivalent binary vectors of
   $\vec{X}_{1_{i,gen}}$, $\vec{X}_{2_{i,gen}}$ and $\vec{X}_{3_{i,gen}}$ and store the value in $X_1f_{i,gen}$, $X_2f_{i,gen}$ and $X_3f_{i,gen}$ respectively.
36 Determine $\vec{X}_{i,gen+1}$ using equations 4.21 and 4.22 respectively
37 end
38 Repeat step 8 to 15
39 Abandon a fraction of $P_a$ worst nests and generate new ones according to Equation (4.6)
40 Keep best solutions(or those nests with quality solutions)
41 Repeat step 8 to 15
   end
42 Best Search Agent=$\alpha_{nest_0}$
43 Best Fitness=$\alpha_{nestf_0}$
```

4.5 Experimental Methodology

In this section, detailed accounts of the biomedical datasets, evaluation metrics, proposed fitness function and the parameter setting for the considered metaheuristic algorithms are outlined.

4.5.1 Considered Biomedical Datasets

To validate the performance of the considered metaheuristic algorithms, six benchmark biomedical datasets extracted from the UCI Irvine Machine [206] were utilized. Each dataset has two classes and the performance of each of these algorithms is evaluated based on its ability to classify these classes correctly. Details of these datasets are given in Table 4.1.

Table 4.1: Considered Biomedical Datasets

Dataset	Number of Features	Number of Cases
Breast Cancer Wisconsin (Prognosis)	33	198
Breast Cancer Wisconsin (Diagnostic)	30	569
SPECTF Heart	44	267
Ovarian Cancer	4000	216
CNS	7129	60
Colon	2000	62

4.5.2 Evaluation Metrics

For the considered feature selection problem, the following evaluation metrics were utilized to compare the performance of each considered feature selection technique.

Average Accuracy (Avg_Acc) : It is one of the commonly used classification metric that represents the number of correctly classified instances by using a particular feature set. The mathematical formulation of this metric is given in Equation 4.26.

$$Avg\_Acc = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{k}\sum_{j=1}^{k}Acc_j \qquad (4.26)$$

Where N is the number of times (runs) a given metaheuristic algorithm is run, k represents the number folds utilized and Acc_j is the accuracy reported during fold j. Acc_j is defined in Equation 4.27.

$$Acc_j = \frac{TP_j + TN_j}{TP_j + TN_j + FP_j + FN_j} \qquad 4.27$$

Where TP and FN denote the number of positive samples in fold j , that are accurately and falsely predicted, respectively, and TN and FP represent the number of negative samples in the same fold that are predicted accurately and wrongly, respectively [207].

Average Feature Length (Avg_NFeat)-This metric characterizes the average length of selected features to the total number of features in the dataset. Equation 4.28 gives its mathematical formulation.

$$Avg\_NFeat = \frac{1}{N}\sum_{i=1}^{N}Sel\_Feat_i \qquad 4.28$$

Where Sel_Feat_i is the number of selected features in the testing dataset during run i.

Minimum Accuracy (Min_Acc) - Is the least value of accuracy reported during N runs. Equation 4.29 depicts its formulation.

$$Min\_Acc = \min\left(\bigcup_{j=1}^{N} Avg\_crossAcc_j\right) \qquad 4.29$$

Where $Avg\_crossAcc_i$ is given by Equation 4.30

$$Avg\_crossAcc_i = \frac{1}{k}\sum_{j=1}^{k} Acc_j \qquad 4.30$$

Maximum Accuracy (Max_Acc) - Is the largest value of accuracy reported during N runs. Its mathematical formulation is given by Equation 4.31.

$$Max\_Acc = \max\left(\bigcup_{j=1}^{N} Avg\_crossAcc_j\right) \qquad 4.31$$

Maximum Features Selected (Max_NFeat) - Is the largest number of selected features during N runs. Equation 4.32 gives its mathematical formulation.

$$Max\_NFeat = \max\left(\bigcup_{i=1}^{N} Sel\_Feat_i\right) \qquad 4.32$$

Minimum Features Selected (Min_NFeat) - Is the least number of selected features during N runs. Equation 4.33 gives its mathematical formulation.

$$Min\_NFeat = \min\left(\bigcup_{i=1}^{N} Sel\_Feat_i\right) \qquad 4.33$$

4.5.3 Evaluation of the classifier

Since the Support Vector machine classifier has already made immense contributions in the field of microarray-based cancer classification [207], it was adopted in this paper to evaluate the classification accuracy using the selected subset of features returned by the various considered metaheuristic feature selection approaches. The Matlab fitcsvm function that trains and cross-validates an SVM model was adopted in this thesis. The specified kernel scale parameter is set to "auto" to allow the function select the appropriate scale factor using a heuristic search.

With the SVM classifier, the data items are mapped points in an n −dimensional feature space (i.e. n=number of features) and the each feature's value is a value of a given coordinate. The final output of this classifier is an optimal hyperplane which can be used to classify new cases [153], [207].

However, the performance of the SVM classifier is highly dependent on the selection of its kernel function [153], [207].A reason why experiments were conducted using various kernels in this thesis.

Selecting a suitable kernel is both dataset and problem specific and selected experimentally [153], [207]. Based on the conducted experiments, suitable kernel functions were selected for the considered datasets. The considered datasets and their suitable kernel functions are presented in Table 4.2. More information of selecting suitable SVM kernel functions is presented in [207].

Table 4.2: Selection of suitable kernel functions

Dataset	Kernel function
Breast Cancer Wisconsin (Prognosis)	Radial Basis Function (RBF)
Breast Cancer Wisconsin (Diagnostic)	Radial Basis Function (RBF)
SPECTF Heart	Radial Basis Function (RBF)
Ovarian Cancer	Linear Function
CNS	Linear Function
Colon	Linear Function

4.5.4 Fitness function

The main aim of a feature selection exercise is to discover a subset of features from the whole set of existing features in a given dataset such that, the considered optimization algorithm is able to achieve the highest possible accuracy using that subset. For instance, in datasets with many features (attributes), the objective is to minimize the number of selected features while improving the classification accuracy of the feature selection approach.

In classifications tasks, there exists higher chances that two feature subsets containing different number of features will have the same accuracy [153].However, if a subset with a large number of features is discovered earlier by a given optimization algorithm, it is likely that the one with least features will be ignored [153].

In trying to overcome this challenge, a fitness function proposed in [153] to evaluate the classification performance of optimization algorithms for feature selection tasks is adopted. This fitness function is given in Equation 4.34.

$$Fit = \alpha * \frac{|R|}{|N|} - \beta * Avg\_crossAcc_i \qquad 4.34$$

Where $|N|$ represents the total number of features within a given dataset, $|R|$ represents the number of selected features during run i and $Avg\_crossAcc_i$ is the average crossvalidation accuracy reported during run i (refer to Equation 4.30). β and α are two weights corresponding to the significance of the classification quality and the subset length respectively. In this paper, β is set to 0.8 and $\alpha = 0.2$ as adopted from [153].

It is important to point out that both terms are normalized by dividing by their largest possible values i.e. the number of selected features $|R|$ is divided by the total number of features $|N|$, and average accuracy $Avg\_crossAcc_i$ is divided by the value 1.

4.5.5 Parameter setting for the considered feature selection techniques

The performance of the proposed EACSIDGWO algorithm was compared to those of Extended Binary Cuckoo Search (EBCS), Binary Anti-Colony Optimization (BACO), Binary Genetic Algorithm (BGA) and Binary Particle Swarm Optimization (BPSO) that were reported earlier in [153].

Table 4.3 indicates the selected parameter values for both the proposed BEACSIDGWO algorithm and each of other algorithms as reported in [153]. In this chapter, all the experiments were conducted using Matlab 2017 running on Windows 10 operating system on a HP desktop with Intel(R) Core (TM) i7-3770CPU @ 3.4GHZ with 12.0GB of RAM.

Table 4.3: Selection of parameter values for the considered approaches

Algorithm	**Parameter values**
EACSIDGWO (New)	$step_{Max} = 1, a_o = 1, P_a$=0.25
EBCS [153]	$N_{mut} = 10, \lambda = 1, \alpha = 1, P_a$=0.4
BACO [153]	$\Gamma_{initial} = 0.1, \alpha = 1, p = 0.1$
BGA [153]	$M_r = 0.1, C_r = 0.1$
BPSO [153]	$C_1 = 1, C_2 = 2,\ \omega_{initial} = 0.9,$ $\omega_{vary-for} = 0.9$

$step_{Max}$ is the maximum value set for the random step size in this research. The random step size determines how far a random waker can go for a given number of iterations.Thus; this parameter controls the balancing between exploitation and exploration. For the Cuckoo Search algorithm, the number of host nests available is fixed. Thus, the host bird can discover the alien egg with a probability P_a whose value is set to 0.25 in this research. N_{mut} is the number of mutations . λ is the mean or expectation of an occurrence of a given event during a unit interval. α is the step size which is normally related to the scales of the problem at hand. For the problem at hand, the probability of ants choosing a given path is proportional to the pheromone concentration on that path i.e $\alpha = 1$. For the BACO algorithm, p is the rate of pheromone evaporation and $\Gamma_{initial}$ is the initial cost of the ant tour length. For the BGA and BPSO parameters, refer to an explanation given for Table 3.2.

To be consistent with the setup proposed in [18], the population size for the proposed EACSIDGWO was set to 30. Then the algorithm was run 10 times to perform the feature selection task for each considered dataset. In addition, each run terminated when 10000 fitness function evaluations were attained. This approach, allowed the proposed algorithm to utilize the fitness function at an equal number of times.

4.6 Results

To examine the diversification and intensification of the proposed EACSIDGWOA, detailed comparative study is presented in this section. The efficiency and the optimization performance of the proposed algorithm has been verified by comparing and analyzing its results with those of four other state-of-the-art optimization algorithms. The experimental classification results have been probed through statistical tests, comparative analysis and ranking methods.

Tables 4.4-4.9 provides the performance of all the considered optimizations approaches for feature selection using the datasets described in subsection 4.5.1. It is important to point out that the best result achieved in each column for all the considered biomedical datasets is highlighted in bold while the worst is italicized.

To prove that the proposed EACSIDGWO is superior over the other four-optimization algorithms, Wilcoxon rank-sum test i.e. a non-parametric statistical test is also performed. The statistical results for the p, h and z values obtained from the pairwise comparisons of the four groups are tabulated in Table 4.10. Tables 4.11-4.12 present a comparison of the overall ranking of the results obtained from the considered algorithms.

Table 4.4: Experimental results for the ovarian cancer dataset

Algorithm	Accuracy			Number Of Features		
	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat

EACSIDGWO (New)	**1.000**	**1.000**	**1.000**	**292**	**264**	**274.8**
EBCS [153]	*0.991*	0.991	0.991	1855	1747	1811.6
BACO [153]	*0.991*	*0.986*	*0.990*	*1971*	*1912*	*1945.7*
BGA [153]	*0.991*	0.991	0.991	1830	1755	1887.3
BPSO [153]	*0.991*	*0.986*	*0.990*	1913	1777	1857

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 4.5: Experimental results for the breast cancer Wisconsin (Diagnostic) dataset

Algorithm	**Accuracy**			**Number Of Features**		
	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat
EACSIDGWO (New)	0.977	**0.974**	**0.975**	**3**	**3**	**3**
EBCS [153]	**0.981**	**0.974**	0.973	4	**3**	3.1
BACO [153]	*0.972*	*0.960*	*0.969*	*8*	*6*	*7*
BGA [153]	0.975	0.965	0.972	6	**3**	3.6
BPSO [153]	**0.981**	0.963	0.974	*8*	**3**	5.4

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 4. 6: Experimental results for the breast cancer Wisconsin (Prognosis) dataset

Algorithm	**Accuracy**			**Number Of Features**		
	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat
EACSIDGWO (New)	**0.879**	**0.864**	**0.873**	**7**	**3**	**5.6**
EBCS [153]	0.874	0.828	0.856	8	4	6.2
BACO [153]	*0.818*	*0.768*	*0.794*	*12*	*5*	*8.4*
BGA [153]	0.874	0.793	0.843	10	4	6.5
BPSO [153]	*0.848*	*0.798*	0.821	11	4	8.3

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 4. 7: Experimental results for the SPECTF Heart dataset

Algorithm	**Accuracy**			**Number Of Features**		
	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat
EACSIDGWO (New)	**0.884**	**0.861**	**0.875**	**6**	**3**	**4.5**
EBCS [153]	0.873	0.846	0.861	8	5	6.2
BACO [153]	*0.846*	*0.813*	*0.831*	*15*	*10*	*12.1*
BGA [153]	**0.884**	0.846	0.866	11	4	8.4
BPSO [153]	0.865	0.846	0.854	*15*	9	10.9

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 4.8: Experimental results for the CNS dataset

Algorithm	Accuracy			Number Of Features		
	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat
EACSIDGWO (New)	**0.767**	**0.700**	**0.718**	**1623**	**807**	**1208.1**
EBCS [153]	*0.667*	0.667	0.667	3490	3391	3446.7
BACO [153]	*0.667*	*0.650*	*0.660*	*3589*	3432	*3522.9*
BGA [153]	0.683	0.667	0.668	3566	*3438*	3489.7
BPSO [153]	*0.667*	0.667	0.667	3547	3359	3474.3

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 4.9: Experimental results for the colon dataset

Algorithm	Accuracy			Number Of Features		
	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat
EACSIDGWO (New)	**0.919**	**0.887**	**0.905**	**637**	**397**	**538.5**
EBCS [153]	0.903	0.871	0.887	*1016*	*961*	*988.7*
BACO [153]	0.903	0.871	*0.881*	1002	932	976
BGA [153]	*0.887*	0.871	0.882	1003	944	962.8
BPSO [153]	*0.887*	*0.855*	0.879	1003	933	971.2

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

4.7 Discussion

4.7.1 Investigation of the obtained classification results

From Tables 4.4-4.9, the following observations can be made.

i) The proposed EACSIDGWO algorithm outperformed all the other considered algorithms in terms of classification accuracy for all the utilized datasets. It recorded the highest classification accuracy on the three highly dimensioned datasets (i.e. Ovarian, CNS and Colon) as well as the remaining three small sample sized datasets. This promising performance is largely attributed to the cooperative exploitation conducted by ACS and IDGWO components of the proposed algorithm during the early generations, as well as the single-handedly exploitation and exploration by IDGWO and ACS respectively at later generations.

ii) For four datasets i.e. Ovarian, Heart, CNS and Colon, the proposed algorithm attained a value for Avg_Acc that is larger than the value for Max_Acc attained by the EBCS. EBCS is a variant of Cuckoo Search, which is a component of the proposed EACSIDGWO algorithm. This superior performance proves the competency of the proposed approach to efficiently determine the optima within the search space.

iii) With regard to the average feature length (Avg_NFeat), the proposed B-EACSIDGWO algorithm demonstrated a superior performance by selecting the least number of features compared to the other algorithms. According to the results reported in Tables 4.4-4.9, the proposed algorithm performed better on all the considered datasets.

iv) In comparison with the original number of features in the considered datasets, there is a notable reduction in the number selected features by the proposed approach. For instance, the actual number of features in ovarian cancer, CNS and Colon cancer datasets is 4000, 7129 and 2000 respectively, whereas the number of selected features by the proposed EACSIDGWO is 274.8, 1208.1 and 538.5 respectively.

This clearly indicates the proposed algorithm is able to reduce the number of features as well as locate the most significant optimal feature subsets. The strength of the proposed EACSIDGWO lies in its well-formulated algorithm (refer to section 4.5) that enhances both its diversification and intensification capabilities which enables it to eliminate redundant (non-informative) attributes and then actively search within the high-performance regions of the feature space.

4.7.2 Statistical analysis

The superiority of the proposed EACSIDGWO algorithm has been verified via Wilcoxon rank-sum test i.e. a non-parametric test with a significance level of 5%. The results obtained for the pairwise comparison of the four groups are presented in Table 4.10. Observations from Table 4.10 reveal the statistical significance of the obtained experimental results for all the considered datasets. This clearly indicates that the proposed approach has an attractive performance in relation to the other four approaches. Thus, the overall statistical results by the new algorithm are highly significant when compared to the results of the four algorithms for all the considered datasets.

4.7.3 Ranking methods

Tables 4.10 - 4.12 outlines a detailed ranking of all the considered algorithms with their respective comparative analysis. The ranking is based on maximum accuracy (Max_Acc), minimum accuracy (Min_Acc), average accuracy (Avg_Acc), maximum number of selected features (Max_NFeat), minimum number of selected features (Min_NFeat), and average number of selected features (Avg_NFeat). From the ranking, it is evident that the proposed EACSIDGWO algorithm obtained the best values in all these measures for all the datasets. Considering the final ranks, the proposed algorithm attained an attractive performance whose overall rank value is 37.This clearly reveals the superiority of EACSIDGWO algorithm in relation to the four state-of-the-art algorithms.

Table 4.10: Using Wilcoxon's rank sum test at $p = 0.05$ to compare EACSIDGWO with other algorithms

Dataset	Wilcoxon's rank sum test	EBCS Vs EACSIDGWO (New)	BACO Vs EACSIDGWO (New)	BGA Vs EACSIDGWO (New)	BPSO Vs EACSIDGWO (New)
Ovarian Cancer	p value	0.000181651	0.000181651	0.000182672	0.000181651
	h value	1.000000000	1.000000000	1.000000000	1.000000000
	z value	3.743255786	3.743255786	3.741848283	3.743255786
Breast Cancer Wisconsin (Diagnostic)	p value	0.022591996	0.000146767	0.017044126	0.000582314
	h value	1.000000000	1.000000000	1.000000000	1.000000000
	z value	2.28026466	3.796476695	2.38575448	3.439721266
Breast Cancer Wisconsin (Prognosis)	p value	0.000730466	0.0001707	0.00073729	0.000174624
	h value	1.000000000	1.0000000	1.00000000	1.000000000
	z value	3.377881495	3.758843896	3.375323463	3.753152986
SPECTF Heart	p value	0.000321376	0.000176611	0.000176611	0.000177611
	h value	1.000000000	1.000000000	1.000000000	1.000000000
	z value	3.597430949	3.750317207	3.750317207	3.748901726
CNS	p value	0.000182672	0.000182672	0.000182672	0.000182672
	h value	1.000000000	1.000000000	1.000000000	1.000000000
	z value	3.741848283	3.741848283	3.741848283	3.741848283
COLON	p value	0.000182672	0.000182672	0.000182672	0.000181651
	h value	1.000000000	1.000000000	1.000000000	1.000000000
	z value	3.741848283	3.741848283	3.741848283	3.743255786

Table 4. 11: Overall ranking of considered algorithms

Algorithm	Measures	Datasets									
		Ovaria n Cancer	Breast Cancer Wisconsin (Diagnostic)	Breast Cancer Wisconsin (Prognosis)	SPECTF Heart	CN S	Colon	Sum of ranks	Overall rank	Total sum	Final ranks
EACSIDGWO (New)	Max_Acc	1	2	1	1	1	1	7	1	37	1
	Min_Acc	1	1	1	1	1	1	6	1		
	Avg_Acc	1	1	1	1	1	1	6	1		
	Max_NFeat	1	1	1	1	1	1	6	1		
	Min_NFeat	1	1	1	1	1	1	6	1		
	Avg_NFeat	1	1	1	1	1	1	6	1		
EBCS	Max_Acc	2	1	2	3	3	2	13	2	84	2
	Min_Acc	2	1	2	2	2	2	11	2		
	Avg_Acc	2	2	2	3	3	2	14	2		
	Max_NFeat	3	2	2	2	2	4	15	2		
	Min_NFeat	2	1	2	3	3	5	16	2		
	Avg_NFeat	2	2	2	2	2	5	15	2		
BACO	Max_Acc	2	4	4	4	2	2	18	4	138	5
	Min_Acc	3	4	5	3	3	2	20	4		
	Avg_Acc	3	5	5	5	4	4	26	5		
	Max_NFeat	5	4	5	4	5	2	25	5		
	Min_NFeat	5	2	3	5	3	2	20	3		
	Avg_NFeat	5	5	5	5	5	4	29	5		

Table 4.12: Overall ranking of considered algorithms

Algorithm	Measures	Datasets									
		Ovarian Cancer	Breast Cancer Wisconsin (Diagnostic)	Breast Cancer Wisconsin (Prognosis)	SPECTF Heart	CNS	Colon	Sum of ranks	Overall rank	Total sum	Final ranks
BGA	Max_Acc	2	3	2	1	2	3	13	2	95	3
	Min_Acc	2	2	5	2	2	2	15	3		
	Avg_Acc	2	4	3	2	2	3	16	3		
	Max_NFeat	2	3	3	3	4	3	18	3		
	Min_NFeat	3	1	2	2	4	4	16	2		
	Avg_NFeat	3	3	3	3	3	2	17	3		
BPSO	Max_Acc	2	1	3	3	3	3	15	3	110	4
	Min_Acc	3	3	2	2	2	3	15	3		
	Avg_Acc	3	2	4	4	3	5	21	4		
	Max_NFeat	4	4	4	4	3	3	22	4		
	Min_NFeat	4	1	2	4	2	3	16	2		
	Avg_NFeat	3	4	4	4	3	3	21	4		

4.8 Conclusion

This chapter proposed a new hybrid Excited (E) - Adaptive Cuckoo Search (ACS)-Intensification Dedicated Grey Wolf Optimizer (IDGWO) i.e. EACSIDGWO algorithm which tries to overcome the challenge of semi-optimal balancing between exploitation and exploration depicted by the EBGWO algorithm proposed in chapter three in solving the feature selection problem in biomedical science. In the EACSIDGWO algorithm, the concept of the complete voltage and current responses of a direct current (DC) excited resistor capacitor (RC) circuit are innovatively utilized to make the step size of ACS and the non-linear control strategy of parameter $\vec{a}$ of the IDGWO (a variant of the EBGWO algorithm proposed in chapter 3) adaptive. Since the population has a higher diversity during early stages of the proposed algorithm, both the ACS and IDGWO are jointly utilized to attain accelerated convergence. However, to enhance mature convergence while striking an effective balance between exploitation and exploration in latter stages, the role of ACS is switched to global exploration while the IDGWO is still left conducting the local exploitation. In order to test the efficiency of the proposed EACSIDGWO as a feature selector, six standard biomedical datasets from the University of California at Irvine (UCI) repository were utilized. The experimental results obtained prove that the proposed algorithm is superior to the state-of-the-art feature selection techniques i.e. BACO, BGA, BPSO and EBCSA in attaining a good learning from fewer instances, and optimal feature selection from information-rich biomedical data, all these while maintaining a high classification accuracy of the utilized data. In future, utilizing this hybrid algorithm as a filter-feature selection approach seeking to evaluate the generality of the selected features will be a valuable contribution.

Though this chapter proposed a superior informative feature selector, which attains optimal balancing between exploitation and exploration in solving the feature selection challenge in highly dimensional DNA microarray datasets, still the performance of the classification phase need to be improved for the microarray based cancer disease classification pipeline to be complete.

Thus, to enhance both the learning and classification ability of the SVM classifier i.e. a commonly adopted classifier in microarray based cancer disease classification; the next chapter proposes a novel adaptive hybrid kernel for this classifier.

CHAPTER FIVE: PARTICLE SWARM OPTIMIZED HYBRID KERNEL BASED MULTI-CLASS SUPPORT VECTOR MACHINE FOR MICROARRAY CANCER DATA ANALYSIS

5.1 Introduction

Cancer is a disorder caused by excessive and uncontrolled cell division in a body. A total of 9.6 million people died of cancer in 2018 [208]. As a matter of fact, death due to cancer can be reduced to nearly half if the cancer types are detected early and the right treatment administered in time. However, it is still a challenge for researchers to effectively diagnose cancer on the basis of morphological structure since different cancer types exhibit thin differences [209].

This challenge encourages application of data mining techniques, especially the use of gene - expression data in determining the types of cancer cells. The level of gene expression can duly indicate the activity of a gene in a body cell based on the number of messenger ribonucleic acids (mRNAs). It is well known to contain information about the disease that may be in the gene sample, which may help experts in treating or preventing the disease [210].

Though next generation sequencing (NGS) especially RNA-sequencing (RNA-Seq) are slowly replacing microarrays when analyzing and identifying complex mechanism in gene expression e.g. in the gene-expression based cancer classification problem, they are relatively expensive compared to microarrays. Since microarrays have been used for a long time, there exists robust statistical and operational methods for their processing [57], [211]–[219].In addition, many significant microarray experiments have been conducted and are publicly available to the research community [60], [220]–[225]. For microarrays, there exists large and well-maintained repositories that have collected these type of data for long. While the pre-processing and analysis steps of microarray data are mostly standardized, the establishment of RNA-Seq data analysis techniques are still ongoing in the field of transcriptomics. Because of these reasons, to date microarrays are still utilized in many gene-expressions based cancer classification studies as presented in the most recent survey of hybrid feature selection methods in microarray gene expression for data for cancer classification [60], [226]–[228].

The DNA microarray technology has the capability of determining the expression level of thousands of genes concurrently in a given experiment, which so far has facilitated the development of cancer classification by the use of gene expression data [57], [211]–[219].

Clinical decision support is the most recent application of DNA microarrays in the medical domain. This support can take the form of disease diagnosis or predicting clinical outcomes in response to a treatment. Currently, the two major areas in medicine that are drawing much attention in this regard are management of cancer and other contagious diseases [229].

With the rapid development of artificial intelligence (AI), machine-learning algorithms such as artificial neural network (ANN), support vector machine (SVM), K-nearest neighbor (KNN) , many researchers have immensely applied them in the gene-expression base cancer diagnosis. For instance, the artificial neural networks (ANN) have been proposed for the microarray gene classification due to their superior ability to map input-output structured data. Khan and Meltzer utilized the ANN in analyzing microarray gene data from patients with small round blue-cell tumours [215]. Bevilacqua and Tommasi developed an accurate classifier model based on the feed-forward ANN for estrogen receptor (ER) +/- metastasis recurrence of breast cancer tumours [230]. Chen and Cheng [231] also modeled a classifier for microarray gene data using ANN ensembles that were based on filtering of samples. In all these studies attractive classification accuracies were obtained.

Furey proposed an SVM based on a simple kernel to carry out gene expression data analysis, which turned out to perform remarkably [232]. Vanitha utilized SVM alongside mutual information gained (MI-SVM) for feature selection [217]. In his research, he used various SVM models; linear SVM, radial basis function (RBF) SVM, Quadratic SVM and Polynomial SVM. He further compared the results obtained from the proposed scheme with the k-nearest neighbor (K-NN) and ANN classifier results. Based on the obtained result, utilization of the MI-SVM obtained better results compared to K-NN and ANN, and even in some datasets, 100% accuracy was achieved.

Based on these previous researches, it is evident that SVM has already made an important contribution in the field of microarray-based cancer classification. However, many researchers have pointed out that though the SVM is a promising classifier in microarray-based cancer classification, its performance solely depends on three aspects; the penalty parameter C of this classifier, the type of kernel utilized and its parameters [233]–[237].

To improve the classification accuracy of the SVM classifier, some techniques have been presented to search for the optimal model parameters, such as the grid-search and the gradient descent [208]. Although, these approaches have proven their effectiveness in the corresponding experiments, in most cases they fall into the local optimum point easily and have a defect of low efficiency [208], [224].

Recently, some meta-heuristic techniques, such as particle swarm optimization (PSO), genetic algorithm (GA), bat algorithm (BA) and dragonfly algorithm (DA) have attained promising results when utilized in tuning SVM classifier's parameters [41]. However, most of these research has not been applied to gene-expression based cancer analysis. In addition, they only focus on SVM with a single kernel function. Though some research [233] point out that combining multiple kernel functions can achieve better performance compared to a single kernel function, little research has provided an in-depth formulation and analysis of the performance of a multi-class support vector machine (MCSVM) with a combined kernel function. Thus, there would a definite need to systematically study the complex optimization problem in the MCSVM classifier with a combined kernel applicable to gene-expression based cancer classification.

Considering PSO is easy to implement, has a few parameters to adjust, is computationally efficient compared to other optimization techniques [238] ,and existence of few studies on MCSVM classifier with combined kernels in microarray-based cancer classification, this chapter proposes a novel gene-expression based cancer classification model i.e. PSO-PCA-LGP-MCSVM. This model is based on particle swarm optimization (PSO), principal component analysis (PCA) and multi-class support vector machine (MCSVM) with a novel hybrid kernel function i.e. linear-gaussian-polynomial (LGP) kernel.

The objective of this Chapter is to construct a MCSVM classifier with three different standard kernel functions (linear, gaussian and polynomial). Use PCA to reduce the dimensional complexity of the considered microarray datasets and optimize all the parameters of this model using PSO.

The overall structure of this chapter takes the form of five sections, inclusive of the introduction. The remaining part of this chapter proceeds as follows: A detailed presentation of the proposed model is presented in section 5.2. Section 5.3 deals with the considered cancer microarray datasets. Section 5.4 focusses on the experimental results and discussions. Finally, conclusions and recommendations are given in section 5.5.

5.2 PSO-PCA-LGP-MCSVM PRINCIPLES

5.2.1. Normalization

Microarray gene expressions can differ by an order of magnitude. Thus, it is necessary to normalize these data to improve the performance of subsequent microarray data analysis stages like gene selection/feature extraction, clustering, and classification [208].

In this chapter, the microarray gene expressions are linearly transformed from the interval $[X_{min}, X_{max}] \rightarrow [0,1]$ uniformly utilizing Equation 5.1 [208];

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad 5.1$$

Where, X' is the new normalized value of the gene expression level, X is the value of the gene expression level before normalization, while X_{max} and X_{min} respectively declare the largest and least values of all the data in an attribute (gene) to be normalized.

Since the min-max normalization has the advantage of preserving exactly all the relationships among the original gene data values and does not introduce any bias [208], it is considered in this chapter.

5.2.2 Principal component analysis (PCA)

One of the major challenges encountered in working with DNA microarray data is their high dimensionality that is coupled with a relatively small sample size. While there is a plethora of crucial information that can be derived from these large datasets, their high dimensional nature can often hide the critical information. Thus, a process that can reduce the dimensionality complexity of this type of data is required. In addition, a dimensionality reduction step will minimize errors obtained in the subsequent classification stage [208], [218], [238]–[240].

In this chapter, principal component analysis (PCA) that includes the calculation of variance of proportion for eigenvector is used. The steps of this algorithm are as follows:

a) Let X'(the normalized microarray gene expression data) be the input matrix for PCA. Each row vectors of X' represent the normalized expression gene values for each of the genes.

b) Compute the mean (centroid) $\overline{X}$ of each gene j using Equation 5.2 where the sum goes through all M samples (tissues):

$$\overline{X} = \frac{1}{M}\sum_{i=1}^{M} X'_{ij} \qquad 5.2$$

Where M is the number of tissues and $X'_{ij}$is gene j data.

c) Compute the covariances (degree to which the genes are linearly correlated) as per Equation 5.3:

$$C_{kj} = \frac{1}{M-1}\sum_{i=1}^{M}(X'_{ki} - \overline{X}_k)(X'_{ji} - \overline{X}_j) \qquad 5.3$$

Where, C_{kj} is the covariance of gene k and gene j, M is the number of samples(tissues), X'_{ki} is the expression level of gene k in sample i, X'_{ji} is the expression level of gene j in sample i, $\overline{X}_k$ is the mean of expression levels of gene k and $\overline{X}_j$ is the mean of expression levels of gene j

d) Form a covariance matrix C using the computed covariances and transform it into a diagonal matrix as depicted in Equation 5.4:

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{1M} \\ \vdots & \vdots & \vdots \\ C_{M1} & C_{M2} & C_{MM} \end{bmatrix} \rightarrow \begin{bmatrix} \partial_1 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \partial_M \end{bmatrix} \qquad 5.4$$

The diagonal elements of the transformed matrix are the eigenvalues $\partial_1, \partial_2, \dots.., \partial_M$ which denotes the amount of variability captured along a particular new dimension.

e) Calculate corresponding eigenvectors as $\rho_1, \rho_2, \dots\dots \rho_M$ using Equation 5.5:

$$\partial_k \rho_k = C\partial_k \qquad 5.5$$

f) Sort the eigenvalues in descending order i.e. $\partial_1 \geq \partial_2 \geq \partial_2, .. \partial_{M-1} \geq \partial_M$

g) The eigenvectors corresponding to the k largest eigenvalues (where $k < M$) are the first k *principal components*

h) Select the first k *eigenvectors* via the cumulative proportion of variance (eigenvalues). The proportion of variance (PPV) for each principal component is determined as follows:

$$PPV = \frac{\partial_i}{\sum_{i=1}^{M} \partial_i} \times 100\% \qquad 5.6$$

i) Form the principal component matrix P, a matrix consisting of selected k eigenvectors that correspond to the largest k eigenvalues. Where the k eigenvectors are derived from eigenvalues that meet the criterion in Equation 5.7;

$$\frac{\sum_{i=1}^{k} \partial_i}{\sum_{i=1}^{M} \partial_i} \times 100\% \geq 95\% \qquad 5.7$$

j) Compute dimensionally reduced microarray gene expression data X'_{DimRed} using Equation 5.8;

$$X'_{DimRed} = X' \times P \qquad 5.8$$

Hence, the analysis reduces the highly dimensioned original microarray datasets to P for each sample, which are the inputs for the multi-class support vector machine (MCSVM).

To be able to measure the generalization error for each considered model, per-fold PCA was adopted. This is achieved by first conducting a separate PCA on each calibration set and then applying this transformation on the validation set. This same transformation is achieved by first subtracting the means of the calibration set from the validation set and then projecting these, data onto the principal components of the training set achieved this. The underlying assumption is that the testing and training set should be derived from the same distribution, which justifies this process.

5.2.3 Multi-class support vector machine (MCSVM)

The MCSVM classifier is based on Vapnik Chervonenkis (VC) dimension of the statistical learning theory and the structural risk minimization [208], [212], [213], [217], [241].

The main objective of MCSVM is to map the preprocessed, on-linear inseparable microarray gene expression data into a linear highly dimensioned manifold θ by the use of a transformation $\emptyset: \mathrm{R}^{\mathrm{N}} \rightarrow \theta$, then obtaining the optimal hyper-plane $\Psi: \psi(\mathrm{x}) = (\omega.\phi(\mathrm{x}) + \mathrm{b})$ by solving the following optimization convex problem(the soft margin problem) [23]:

$$\min(\omega, \xi) = \frac{1}{2}\|\omega\|^2 + \beta \sum_{i=1}^{n} \xi_i \qquad 5.9$$

Subject to $y_i(\omega.\phi(x) + b) \geq 1 - \xi_i$ for all $1 \leq i \leq n$

Where ω is a coefficient vector of the hyper-plane in the manifold (feature space), b is the threshold value of the hyper-plane, ξ_i is a slack factor introduced for classification errors and β is a penalty factor for errors.

The parameter β controls the penalty of misclassification and its value is normally determined via cross-validation. Larger values of β normally leads to a small margin which minimizes classification errors while smaller values of β may produce a wider margin resulting to many misclassifications.

The feature space θ is highly dimensioned, so its direct computation can lead to "dimension disaster". However, since $\omega = \sum_{i=1}^{n} \delta_i y_i \emptyset(x_i)$, then all the operations of the support vector machine (MCSVM) in the feature space θ are only dot products. And since kernel functions i.e$G(x_i, x_{i'}) = \emptyset(x_i).\emptyset(x_{i'})$, are efficient at handling dot products, they were introduced into the SVM. This implies there is no need to know how to map the microarray gene expression data from its original space to the feature space θ. Thus, selection of a kernel and its coefficients are vital in the computational efficiency and accuracy of an MCSVM classifier model [233]–[237].

The common kernel functions that are utilized as continuous predictors include [1, 5, 22]:

1) Linear Kernel:

$$G(x_i, x_{i'}) = x_i . x_{i'} \qquad 5.10$$

2) Polynomial Kernel:

$$G(x_i, x_{i'}) = (\eta * (x_i . x_{i'}) + \delta)^d \qquad 5.11$$

Where $\eta > 0, \delta \in R$ and $d \in Z^+$

3) Gaussian kernel:

$$G(x_i, x_{i'}) = \exp\left(\frac{\|x_i - x_{i'}\|^2}{2\sigma^2}\right) \qquad 5.12$$

Where $\sigma > 0$

These MCSVM kernel functions can be broadly categorized as follows: local kernel functions and global kernel functions. Samples far apart have a great impact on the global kernel values while samples close to each other greatly influence the local kernel values. The linear and polynomial kernels are good examples of global kernels while the Gaussian radial basis function and the Gaussian are local kernels [233], [235]–[237], [242].

Relatively speaking, the linear kernel function has a better extraction of global features from samples, the polynomial kernel has good generalization ability and the gaussian kernel (the most widely used kernel) has a good learning ability among all the single kernel functions. Thus, it is evident that utilizing a single kernel function based MCSVM classifier in a given application such as gene expression data may neither attain good learning ability, proper global feature extraction ability and a better generalization capability. In trying to overcome this hiccup, two or more kernel functions can be combined [233]–[237].

5.2.4 Linear-Gaussian-Polynomial MCSVM (LGP- MCSVM)

In trying to build a kernel model that has a better global feature extraction capability, good learning and prediction abilities, the work presented in this chapter combines the merits of two global kernels (Linear and Polynomial) and one local kernel (Gaussian). This chapter therefore proposes a novel kernel "Linear-Gaussian -Polynomial (LGP)" kernel, which is formulated as follows:

$$\mathrm{G_{LGP}}(\mathrm{x_i}, \mathrm{x_{i'}}) = \beta_1.(\mathrm{x_i}.\mathrm{x_{i'}}) + \beta_2.\exp\left(-\beta_3.\left(\frac{(\eta\times(\mathrm{x_i}.\mathrm{x_{i'}}) + \delta)^{\mathrm{d}}}{2\times\sigma^2}\right)\right) \qquad 5.13$$

Where $\beta_1 + \beta_2 + \beta_3 = 1, \beta \in \mathrm{R}$ and $\delta, \mathrm{d} > 0$

In this chapter we utilize different values of β to mix the three standard kernels (different regions of the input space). In this case β is a vector i.e. $\beta = [\beta_1, \beta_2, \beta_3]$. Through this approach, the relative contribution of each kernel to the hybrid kernel i.e. $\mathrm{G_{lgpk}}(\mathrm{x_i}, \mathrm{x_{i'}})$ can be easily varied over the input space.

The LGP kernel function takes better global feature extraction ability from the linear kernel, good prediction ability from the polynomial kernel and better learning ability from the gaussian kernel. The Mercer's theorem provides the necessary and sufficient qualifiers of a valid kernel function. It states that a kernel function is a permissible kernel if the corresponding kernel matrix is symmetric and positive semi-definite (PSD) [212], [243].

A kernel matrix can be validated that it is PSD by determining its spectrum of eigenvalues. It is important to note that a symmetric is positive definite if and only if all its eigenvalues are non-negative. Considering this, for the proposed kernel to be permissible, it must satisfy the Mercer's theorem. This validity can be proved by using the Taylor expansion for exponential function of Equation 5.13:

$$\mathrm{G_{LGP}}(\mathrm{x_i}, \mathrm{x_{i'}}) = \beta_1.(\mathrm{x_i}.\mathrm{x_{i'}}) + \beta_2\left(-\sum_{i=0}^{\infty} \beta^{i}{}_{3}.\frac{(\eta\times(\mathrm{x_i}.\mathrm{x_{i'}}) + \delta)^{\mathrm{d.i}}}{2\times\sigma^{2\mathrm{i}}.i!}\right) \qquad 5.14$$

$$\mathrm{G_{LGP}}(\mathrm{x_i},\mathrm{x_{i'}}) = \beta_1(\mathrm{x_i}.\mathrm{x_{i'}}) + \beta_2\left(-1 + \sum_{i=1}^{\infty}\frac{-\beta^{i}{}_{3}}{2\times\sigma^{2\mathrm{i}}\,i!}\left((\eta(\mathrm{x_i}.\mathrm{x_{i'}}) + \delta)^{\mathrm{d.i}}\right)\right) \quad 5.15$$

$$\mathrm{G_{LGP}}(\mathrm{x_i},\mathrm{x_{i'}}) = \beta_1(\mathrm{x_i}.\mathrm{x_{i'}}) - \beta_2 + \beta_2\sum_{i=1}^{\infty}\frac{-\beta^{i}{}_{3}}{2\times\sigma^{2\mathrm{i}}\,i!}\left((\eta(\mathrm{x_i}.\mathrm{x_{i'}}) + \delta)^{\mathrm{d.i}}\right) \quad 5.16$$

$$\mathrm{G_{LGP}}(\mathrm{x_i},\mathrm{x_{i'}}) = \beta_1(\mathrm{x_i}.\mathrm{x_{i'}}) - \beta_2 + \beta_2\sum_{i=1}^{\infty}\frac{-\beta^{i}{}_{3}}{2\times\sigma^{2\mathrm{i}}\,i!}.K_{Poly(i)} \quad 5.17$$

$$\mathrm{G_{LGP}}(\mathrm{x_i},\mathrm{x_{i'}}) = \beta_1 K_{Linear} - \beta_2 + \beta_2\sum_{i=1}^{\infty}\frac{-\beta^{i}{}_{3}}{2\times\sigma^{2\mathrm{i}}\,i!}.K_{Poly(i)} \quad 5.18$$

$$\mathrm{G_{LGP}}(\mathrm{x_i},\mathrm{x_{i'}}) = \beta_1 K_{Linear} - \beta_2 + \beta_2\sum_{i=1}^{\infty}\frac{-\gamma^{i}\times\beta^{i}{}_{3}}{i!}.K_{Poly(i)} \quad 5.19$$

Where $K_{Poly(i)} = (\eta(\mathrm{x_i}.\mathrm{x_{i'}}) + \delta)^{\mathrm{d}}$ and $K_{Linear} = (\mathrm{x_i}.\mathrm{x_{i'}})$ and $\gamma^{i} = \frac{1}{2*\sigma^{2\mathrm{i}}}$

From Equation 5.19, it is evident that $\mathrm{G_{LGP}}(\mathrm{x_i},\mathrm{x_{i'}})$ is a mixed kernel comprising of a weighted linear kernel, a constant β_2 and a weighted summation of polynomial kernels. Using propositions 20,21and 22 of theorem 2.20 and propositions 23 and 24 of corollary 2.21 [243], Mercer's conditions are proved to be true for the proposed kernel and hence it is a valid kernel.

Theorem 2.20. *Functions of Mercer's kernels K1 and K2 are also Mercer's kernels.*

$$\mathrm{G}(\mathrm{x_i},\mathrm{x_{i'}}) = \mathrm{K1}(\mathrm{x_i},\mathrm{x_{i'}}) + \mathrm{K2}(\mathrm{x_i},\mathrm{x_{i'}}) \quad 5.20$$

$$\mathrm{G}(\mathrm{x_i},\mathrm{x_{i'}}) = \mathrm{c}.\mathrm{K1}(\mathrm{x_i},\mathrm{x_{i'}}),\ for\ all\ \mathrm{c} \in \mathrm{R}^{+} \quad 5.21$$

$$\mathrm{G}(\mathrm{x_i},\mathrm{x_{i'}}) = \mathrm{K1}(\mathrm{x_i},\mathrm{x_{i'}}) + c\ , for\ all\ \mathrm{c} \in \mathrm{R}^{+} \quad 5.22$$

Corollary 2.21. *Functions of a Mercer kernel K1 are also Mercer's kernels.*

$$\mathrm{G}(\mathrm{x_i}, \mathrm{x_{i'}}) = (\mathrm{K1}(\mathrm{x_i}, \mathrm{x_{i'}}) + \mathrm{c})^{d} \text{ , } \mathit{for\ all}\ \mathrm{c} \in \mathrm{R}^{+}\ \mathit{and}\ \ \mathrm{d} \in \mathrm{N} \qquad 5.23$$

$$\mathrm{G}(\mathrm{x_i}, \mathrm{x_{i'}}) = \exp\left(\frac{K1(x_i, x_{i'})}{\sigma^2}\right) \text{, , } \mathit{for\ all}\ \sigma \in \mathrm{R}^{+} \qquad 5.25$$

Since the proposed hybrid LGP kernel combines three valid Mercer's kernels i.e. linear, gaussian and polynomial kernels, it is also a valid Mercer's kernel that can be used for training and classification of the multi-class support vector machine (MCSVM).

By using the proposed LGP-MCSVM, the non-linear transformation of the microarray gene sample points to the corresponding kernel matrix so as to obtain the classification results during the training phase of the MCSVM classifier.

5.2.5 Particle swarm optimization (PSO)

Currently, there is no widely accepted method for optimizing these parameters. The "Grid-Search (GS)" with exponentially growing sequences of combination $\{\mathrm{C}, \eta\}$ for the commonly utilized Gaussian kernel is often applied in microarray analysis [208], [224]. Though easy to implement, it has a low computing efficiency. In addition, optimal result of the GS can only be generated from the pre-set grid-combinations while unknown possible optimal parameters cannot be explored and discovered.

In this chapter, particle swarm optimization (PSO) optimization technique is adopted to optimally search for the best parameter combinations for the considered models [224], [238]. The PSO technique is derived from the migration patterns of birds during foraging, which has a faster convergence, efficient parallel computing and a strong universality that is able to efficiently avoid local optimum [60]. In addition, the iteration velocity for its particles is largely influenced by the sum of current velocity; previous particle value, the current global optimal value and random interferences, which greatly helps, avoid the local optimal and improves the search coverage and effectiveness. In order to effectively evaluate the performance of the considered models, different values were considered for all kernel parameters within the ranges presented in Table 5.1.

Table 5.2 presents the initial PSO parameters of each considered algorithm. In this paper, as a rule of thumb with heuristic optimization algorithms, the swarm size for each model was set to

$10\times variable\ size$ [244].More information on the PSO algorithm is presented in [60], [224], [225], [238], [244]–[248].

Table 5.1: Parameters and their respective ranges

Parameter	Range
$\beta = [\beta_1, \beta_2, \beta_3]$	$0 < \beta_1, \beta_2, \beta_3 < 1$ and $\beta_1+\beta_2+\beta_3=1$
log_2C	$-5 \leq log_2\mathrm{C} \leq 15$
δ	$0 \leq \delta \leq 5$
d	$2 \leq \mathrm{d} \leq 5$
$log_2\gamma, log_2\eta$	$-15\leq log_2\gamma, log_2\eta \leq 3$

Table 5.2: Initial PSO parameters setting

Parameter	Range
Maximum number of iterations	50
Inertial weight,w	1
Number of particles/Swarm size	1) PSO+L-MCSVM=10
	2) PSO+G-MCSVM=20
	3) PSO+P -MCSVM=40
	4) PSO+LGP-MCSVM=80
Cognition learning factor, c_1	2.0
Social learning factor, c_2	2.0

β_1, β_2 and β_3 are scalers that indicating the relative contribution of the linear, gaussian and polynomial kernels to the proposed kernel. C is the penalize parameter that controls the trade-off between achieving a lower error on the traning data subset and minimizing the norm of weights. δ and η are the polynomial kernel constants while d is the power of the polynomial kernel. D determines the degree one wants to map the data. γ is the inverse of standard deviation of the radial basis kernel i.e $\gamma = \frac{1}{2*\sigma^2}$ which is utilized as a similarity measure between two points. For more information on setting of PSO paramters refer to detailed account given for Table 3.2.

5.2.6 PCA-PSO-LGP-MCSVM model

The main process of the proposed algorithm is outlined as follows:

1) Transforming the cancer microarray data into the right format for the SVM package.
2) Loading a cancer microarray dataset.
3) Randomly dividing the loaded microarray data into two sets: training set and testing set.
4) Initialize the PSO parameters like the population size, the maximum number of iterations, and the considered multi-class SVM parameters.
5) Adopt PSO to search for the optimal solution of particles in the global space by using 5-fold cross-validation that incorporates per fold PCA feature extraction. This process is presented below.
6) To achieve 5-fold cross-validation incorporating PCA, the following steps were followed:
    i) For j=1 to 5 repeat steps (ii) to (vi)
    ii) Carry out PCA on data present in the remaining 4 folds to generate a loadings matrix.
    iii) Transform this data (data in the remaining 4 folds i.e. calibration set) into a set of principal components (PC) scores using the first *P* components (that account for at least 95% cumulative variance) of the loadings matrix generated in step (ii).
    iv) Build a considered SVM classification model using a set of parameter values using the generated PC scores data in step (iii).
    v) Transform the held-out test fold data (i.e. data in fold j) into a set of principal component (PC) scores using the *P* components loadings matrix retained in step (iii).
    vi) Compute the classification accuracy of the built SVM classification model in step (iv) using the transformed test fold j data in step (v).
    vii) For the considered parameters set, store their optimal parameter values set (i.e. a set of parameters that yields the highest classification accuracy).
7) Report optimal parameters for the considered model.

8) Carry out PCA on the whole training set data (i.e. the training set obtained in step 3) to generate a loading matrix.
9) Transform this whole training set data into a set of PC scores using the first P components (that account for at least 95% cumulative variance).
10) Build an optimal model for the considered SVM classification model using the optimal parameter values set obtained in step vii) using the generated PC scores data in step 9.
11) Transform the whole testing set data (i.e. the testing set obtained in step 3) into a set of principal components (PC) scores using the P components loadings matrix retained in step 9.
12) Compute the classification accuracy of the built optimal SVM classification model in step 8 using the transformed whole testing set data in step 9
13) Report this test classification accuracy.

The schematic diagram in Figure 5.1 shows the process of the PSO-PCA-LGP-MCSVM algorithm.

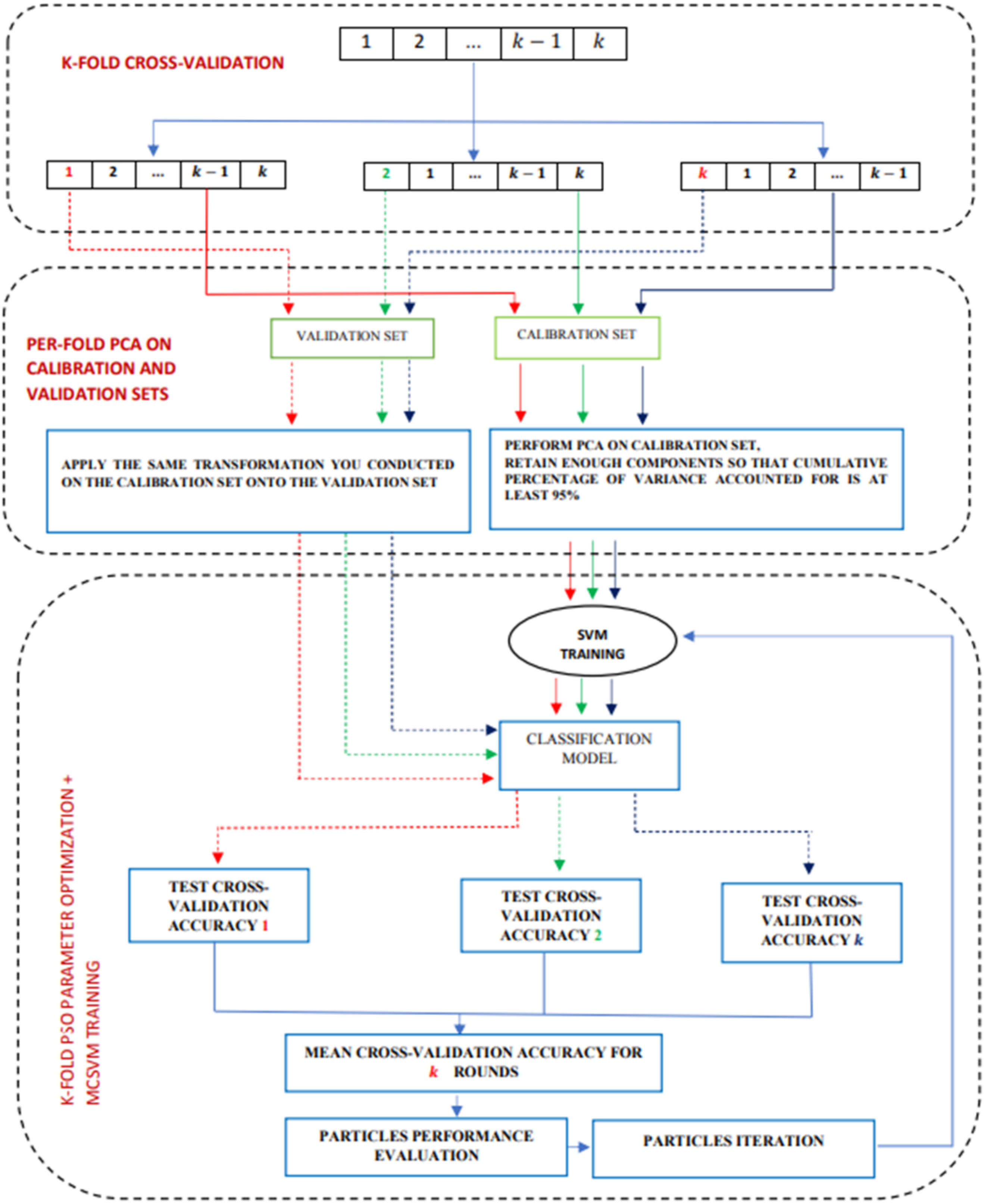

K-FOLD CROSS-VALIDATION
1 2 ... $k-1$ k
1 2 ... $k-1$ k
2 1 ... $k-1$ k
k 1 2 ... $k-1$
PER-FOLD PCA ON CALIBRATION AND VALIDATION SETS
VALIDATION SET
CALIBRATION SET
APPLY THE SAME TRANSFORMATION YOU CONDUCTED ON THE CALIBRATION SET ONTO THE VALIDATION SET
PERFORM PCA ON CALIBRATION SET, RETAIN ENOUGH COMPONENTS SO THAT CUMULATIVE PERCENTAGE OF VARIANCE ACCOUNTED FOR IS AT LEAST 95%
K-FOLD PSO PARAMETER OPTIMIZATION + MCSVM TRAINING
SVM TRAINING
CLASSIFICATION MODEL
TEST CROSS-VALIDATION ACCURACY 1
TEST CROSS-VALIDATION ACCURACY 2
TEST CROSS-VALIDATION ACCURACY k
MEAN CROSS-VALIDATION ACCURACY FOR k ROUNDS
PARTICLES PERFORMANCE EVALUATION
PARTICLES ITERATION

Figure 5.1: Scheme of the proposed PSO-PCA-LGP-MCSVM algorithm

It is important to mention that the analysis process is conducted using the LIBSVM framework in MATLAB [249]–[252] on Intel(R)Core (TM) i3-3240M CPU@ 3.4GHz with 12GB of RAM machine.

5.3 Performance evaluation

5.3.1 Microarray Datasets

To assess the performance of the proposed PSO-PCA-LGP-SVM algorithm, several experiments were conducted on four publicly available datasets. Summary of all the datasets utilized in this research can be found in Table 5.3 and following is a brief description of each dataset.

Colon dataset [214]: contains gene expression levels obtained from DNA based microarrays. It has 62 samples; 20 normal and 40 cancerous tissue samples, each described by 2000 features.

Leukemia (AMLALL) dataset [57]: contains gene expression levels from 72 leukemia patients; 47 with Acute Lymphoblastic Leukemia (ALL) and 25 with Acute Myeloid Leukemia (AML). Each patient data is described by expression levels of 7129 probes obtained from 6817 human genes.

Stjude Leukemia dataset [213]: This data was obtained from St. Jude children's research hospital. It is divided into 6 diagnostic groups: BCR-ABL(9 patients), E2A-PBX1(18 patients), Hyper- diploid>50 (42 patients), Mixed Lineage Leukemia(MLL)(14 patients), T-cell Acute Lymphoblastic Leukemia(T-ALL)(28 patients) and TEL-Leukemia(TEL-AML1)(52 patients) and other 52 patients that could not fit into any of the outlined diagnostic groups. This dataset contains 12558 genes.

Lung Cancer dataset [219]: Contains 3312 gene data obtained from 17 people with normal lungs and 186 lung cancer patients that is classified into 5 classes: Adenocarcinomas (139 patients), Squamous Cell Lung Carcinomas(21 patients), Pulmonary Carcinoids(20 patients), Small Cell Lung Carcinomas (6 patients) and Normal Lung (17 people).

Table 5.3: The cancer microarray datasets utilized in this chapter

Category	Dataset	Sample Size	Number of genes	Number of classes
Two-Class	AMLALL	72	7129	2
	COLON	62	2000	2
Multi-Class	STJUDE	215	12558	7

LUNG	203	3312	5

Due to the small number of instances in the considered datasets, all the datasets were initially split into two disjoint sets: the training set and the test set. Utilizing 5-fold cross-validation, the training set was randomly divided further into 5 subsets (approximately) equal in size. Each time 4 subsets were selected as the calibration set and the remaining subset was used as the validation set. This process was repeated 5 times. Finally, the average of classification accuracy on the validation set was used as one of the evaluation metrics. It is important to point out that by using 5-fold cross-validation to dynamically divide the microarray training samples, the considered models turn out to be more stable and objective.

The percentage proportion for the calibration, validation and test sets for all the considered microarray datasets are presented in Table 5.4.

Table 5.4: Percentage proportion for calibration, validation and test sets

Dataset	% Proportion for Calibration set	% Proportion for Validation set	%Proportion for Test set
AMLALL	61.1	15.3	23.6
COLON	58.1	14.5	27.4
STJUDE	57.7	14.4	27.9
LUNG	57.1	14.3	28.6

5.3.2 Performance measures for imbalanced microarray datasets

When the samples in a dataset are unevenly distributed among the classes (for instance in the case of microarray datasets), the task of classification in imbalanced domains must be defined. The majority class(es), as a result influences the data mining algorithms skewing their performances towards it [221]. Most algorithms simply compute the accuracy on the basis of the percentage of correct samples.

However, in the case of microarrays, these results are highly deceiving since the minority classes hold minimal effects on the overall classification accuracy. Thus, a consideration of a complete confusion matrix (Table 5.5) must be made to obtain the classification of both positive and negative classes independently [221].

Table 5.5: Confusion matrix for a two-class problem

	Positive prediction	**Negative prediction**
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

The description in Table 5.5 gives four baseline statistical components, where TP and FN denote the number of positive samples, which are accurately and falsely predicted, respectively, and TN and FP depict the number of negative samples that are predicted accurately and wrongly, respectively.

Two most frequently used metrics for class imbalance problem, namely F-measure and G-mean, can be regarded as functions of these four statistical components and are calculated as follows:

$$\mathrm{F-measure} = \frac{2 * Recall * Precision}{(Recall + recision)} \qquad 5.25$$

$$\mathrm{G-mean} = \sqrt{(TPR \times TNR)} \qquad 5.26$$

Where Precision, Recall, TPR and TNR are further defined as follows:

$$\mathrm{Precision} = \frac{TP}{(TP + FP)} \qquad 5.27$$

$$\mathrm{Recall\ (TPR)} = \frac{TP}{(TP + FN)} \qquad 5.28$$

$$\mathrm{TNR} = \frac{TN}{(TN + FP)} \qquad 5.29$$

The overall classification accuracy Acc can be calculated using equation 5.30.

$$\mathrm{Acc} = \frac{TP + TN}{(TP + TN + FP + FN)} \qquad 5.30$$

However, all these evaluation metrics are appropriate for estimating binary-class imbalance tasks. To extend them for multi-class, the following transformations should be considered [38]. $\mathrm{G-mean}$ computes the geometric mean of all the classes' accuracies and is defined by Equation 5.31.

$$\mathrm{G-mean} = (\prod_{i=1}^{C} Acc_i)^{1/c} \qquad 5.31$$

Where Acc_i denotes the accuracy of the i^{th} class. $\mathrm{F-measure}$ can be transformed as $\mathrm{F-Score}$ and is computed using equation 5.32.

$$\mathrm{F-Score} = \frac{\sum_{i=1}^{C} \mathrm{F-measure}_i}{C} \qquad 5.32$$

Where $\mathrm{F-measure}_i$ is calculated further using the equation 5.33.

$$\mathrm{F-measure}_i = \frac{2\times\mathrm{Precision}_i\times\mathrm{Recall}_i}{\mathrm{Precision}_i + \mathrm{Recall}_i} \qquad 5.33$$

Acc can be transformed as depicted by equation 5.34.

$$Acc = \sum_{i=1}^{C} (\mathrm{Acc}_i\times\mathrm{P}_i) \qquad 5.34$$

Where P_i is the percentage of samples in the i^{th} class. To impartially and comprehensively assess the classification performance of the proposed model in comparison with PSO-PCA-L-MCSVM, PSO-PCA-G-MCSVM and PSO-PCA-P-MCSVM models that utilize the standard linear, gaussian and polynomial kernels respectively, the three extended measures namely $\mathrm{F-Score}$, $\mathrm{G-mean}$ and Acc which are described in 5.32, 5.31 and 5.34 respectively.

5.4 Results and Discussions

The experimental results for the 4 classification models on the 4 microarray datasets are reported in Tables 5.6, 5.7 and 5.8, where the best result in each dataset is highlighted in bold and the worst is italicized.

From Tables 5.6 -5.8 the following observations can be made:

i) Lung and STJUDE datasets are slightly sensitive to the class imbalance while Colon and AMLALL are not, as shown by the difference between Accuracy and G-mean values. An accuracy slightly lower than the G-mean values imply that the MCSVM is affected by the imbalanced class distribution. This is largely attributed by a large number of True negatives (TNs) recorded achieved by all the models when analyzing both the Lung and STJUDE datasets.

ii) The hybrid kernel boosted the classification performance of the multi-class on three datasets i.e. Colon, Lung and STJUDE. These promotions are better portrayed by the F-

Score and G-Mean metrics, which are used to evaluate a balance level of classification results. However, a tie is reported for the AMLALL dataset. This implies that though the complementary characteristics of the three standard kernels i.e. linear, Gaussian and polynomial in the proposed hybrid linear-gaussian-polynomial (LGP) kernel may improve the multi-class support vector machine classifier's classification ability on most microarray datasets, datasets a single suitable kernel is sufficient.

iii) Of all the considered models, the PSO-PCA-P-MCSVM reported the least performance in all the considered metrics for all the four datasets. However, it is important to note that a promising kernel can be obtained if we embed into the exponential kernel.

Table 5.6: Accuracy of all considered models on the four microarray datasets, where bold represents the best result and the italics denotes the worst in each column respectively

Models	Colon	Lung	AMLALL	STJUDE
PSO+L-MCSVM	*0.7647*	0.9596	**0.9412**	0.9422
PSO+P -MCSVM	0.8235	*0.9592*	*0.8235*	*0.9395*
PSO+G-MCSVM	0.8235	0.9608	**0.9412**	0.9572
PSO+LGP-MCSVM **(New)**	**0.8824**	**0.9729**	**0.9412**	**0.9603**

Table 5.7: F-Score of all considered models on the four microarray datasets, where bold represents the best result and the italics denotes the worst in each column respectively

Models	Colon	Lung	AMLALL	STJUDE
PSO+L-MCSVM	*0.7572*	0.9246	0.9328	0.7870
PSO+P -MCSVM	0.8211	*0.7524*	*0.7733*	*0.6831*
PSO+G-MCSVM	0.8211	0.9306	**0.9377**	0.8477
PSO+LGP-MCSVM **(New)**	**0.8712**	**0.9586**	**0.9377**	**0.8989**

Table 5.8: G-mean of all considered models on the four microarray datasets, where bold represents the best result and the italics denotes the worst in each column respectively

Models	Colon	Lung	AMLALL	STJUDE
PSO+L-MCSVM	*0.7676*	0.9791	**0.9412**	0.9557
PSO+P -MCSVM	0.8235	*0.7524*	*0.8235*	*0.9512*
PSO+G-MCSVM	0.8235	0.9792	**0.9412**	0.9661
PSO+LGP-MCSVM **(New)**	**0.8824**	**0.9861**	**0.9412**	**0.9709**

In summary, compared with single-kernel-based models (i.e. PSO-PCA-L-MCSVM, PSO-PCA-G-MCSVM and PSO-PCA-P-MCSVM), the proposed PSO-PCA-LGP-MCSVM model that is based on a hybrid linear-gaussian-polynomial (LGP) kernel with a better global feature extraction ability, good prediction ability and better learning ability, has an attractive classification ability in cancer diagnosis using both imbalanced dual and multiclass microarray datasets. Moreover, due to excellent global searching ability of the particle swarm optimization, it can effectively optimize the hybrid kernel based MCSVM when solving a wider range of classification problems.

5.5 Chapter Conclusion

Techniques to choose or construct suitable kernel functions, and optimally tune its parameters for MCSVM has received a considerable and critical attention in imbalanced microarray-based cancer diagnosis. A novel classification model, PSO-PCA-LGP-MCSVM, that is based on MCSVM with a hybrid kernel i.e. linear-gaussian-polynomial (LGP), is proposed in this chapter. The LGP kernel combines the advantages of three standard kernels i.e. linear, gaussian and polynomial kernels in a novel manner where the linear kernel is linearly combined with a polynomial kernel that is embedded into a gaussian kernel. Using PSO to optimally tune the LGP kernel based MCSVM resulted into better generalization, learning and predicting ability as evidenced by the promising results in terms three extended measures F-Score, G-mean and Accuracy irrespective of imbalanced binary or multi-class microarray datasets. The performance of the proposed model was compared with those of 3 models i.e. PSO-PCA-L-MCSVM, PSO-PCA-G-MCSVM and PSO-PCA-P-MCSVM that are based on single linear, gaussian and polynomial kernels respectively and the experimental results show that the proposed algorithm is superior to the three single-kernel based models. This reflects the good practical value of the proposed model in the field of microarray based cancer diagnosis, which can also be extended to more applications of medical diagnostic classification to explore its potential.

CHAPTER SIX: CONCLUSION AND RECOMMENDATIONS

6.1 General Conclusions

To date, optimal gene selection and accurate classification of a given patient sample are the most sought topics in a DNA microarray based cancer disease diagnosis.This is because an effective gene selection phase derives a reduced informative gene subset from the gene-rich DNA microarray datasets which subsequently minimizes noise, computational overheads as well as model overfitting. On the other hand, an improved learning and classification stage builds an effective classifier that achieves a reliable and accurate classification of a DNA patient sample.

Optimal gene selection requires a stable, diverse and robust gene selector. This can only be achieved by a wrapper that maturely converges during the search process and thus ensuring an exhaustive search of the whole population of DNA microarray genes. On the other hand, mature convergence demands striking of a proper and optimal balance between exploitation and exploration in the design of a metaheuristic. Exploitation and exploration are two antagonistic principles which pose a big challenge in striking a proper balance between them in the design of a metaheuristic. A reason why majority of the existing wrappers have proved inadequate in solving the feature selection problem in DNA microarray based cancer disease diagnosis.

Designing an efficient gene selector without enhancing both the learning and classification phase will still render the DNA microarray based cancer classification pipeline incomplete.Though currently the SVM is a promising classifier in DNA microarray data classification, its performance largely depends on the kernel adopted for this classifier as well as tuning of the kernel parameters. The linear, polynomial and Gaussian kernels are the three standard kernels commonly adopted a large number of researchers for this classifier. The linear kernel function has a better extraction of global features from samples, the polynomial kernel has good generalization ability and the gaussian kernel (the most widely used kernel) has a good learning ability among all the single kernel functions. Thus, it is evident that utilizing a single kernel function based MCSVM classifier in a given application such as gene expression data may neither attain good learning ability, proper global feature extraction ability and a better generalization capability.To date, this has necessitated a combinination of two or more of these standard kernel functions.

This research has successfully tackled the aforementioned challenges by development of the following techniques:

i) To select a subset of informative genes from the highly dimensional DNA microarray chips, a novel excited binary grey wolf optimization (EBGWO) based wrapper utilizing the K-NN classifier is presented in Chapter 3. To overcome the local minima trapping of the existing BGWO that normally results into semi-optimal solutions, in the proposed EBGWO, a new position-updating criterion is formulated. The new position updating criterion utilizes the fitness values of vectors $X_1^{\rightarrow}$, $X_2^{\rightarrow}$and $X_3^{\rightarrow}$ to determine the new candidate individuals. These vectors are derived from the union of scalars X_1, X_2 and X_3 respectively of the existing BGWO. Moreover, to make full use of and strike a better balance between exploration and exploitation, which is also a challenge in the BGWO, a novel nonlinear control strategy is formulated. This non-linear strategy innovatively decreases parameter $a^{\rightarrow}$ via the concept of the complete current response of a direct current (DC) excited resistor-capacitor (RC) circuit. One induction algorithm i.e. the K-Nearest Neighbor (K-NN) is utilized in the proposed wrapper approach to evaluate the classification performance of subset of genes selected by the EBGWO, using 5-fold cross-validation technique.

The performance of EBGWO as a gene selector is evaluated on 7 standard DNA microarray chips derived from Irvine (UCI) repository namely Brain Tumour1 (5920 genes), Brain Tumour2 (30367 genes), Central Nervous System Cancer (7129 genes), Diffuse Large B-Cell Lymphoma (DLBL) (5469 genes), Leukemia (7129 genes), Colon Cancer (2000 genes) and Lung Cancer(12600). The EBGWO achieved the most compact informative gene subsets along with the highest classification accuracies as follows: Brain Tumour1 (501 genes, 92%), Brain Tumour2 (1151 genes, 88%), Central Nervous System Cancer (710 genes, 83%), DLBL (426 genes, 100%), Leukemia (649 genes, 90%), Colon Cancer (143 genes, 92%) and Lung Cancer(1005 genes, 98%). Binary Grey Wolf Optimization 2 (BGWO2), the second best state-of-the-art published algorithm, attained the following: Brain Tumour1 (1343 genes, 89%), Brain Tumour2 (3083 genes, 85%), Central Nervous System Cancer (2175 genes, 78%), DLBL (1408 genes, 98%), Leukemia (1805 genes, 87%), Colon Cancer (455 genes, 90%) and Lung Cancer(2413 genes, 97%).On average, the proposed EBGWO algorithm attained a reduced informative gene subset with 655 genes along with a classification accuracy of 92%. On the other hand, on average the BGWO2 (second best algorithm) attained a reduced informative gene subset with 1812 genes along with a classification accuracy of 89%. **Thus in comparison with BGWO2 (the current best gene selector that is based on the GWO algorithm), on average the proposed EBGWO algorithm reduced the number of selected genes from 1812 to 655 (i.e. a further reduction by 1157 genes) while improving the classification accuracy from 89% to 92% (i.e. an improvement by 3%)**.

ii) Though the proposed EBGWO wrapper has proved attractive in selecting informative genes from the highly dimensioned DNA microarray datasets due to its enhanced stability and diversity capabilities, it does not strike an optimal balance between exploitation and exploration during the search process. This is because exploitation and exploration are two contradicting principles, which must be balanced efficiently in order to achieve an improved performance of a metaheuristic. Moreover, attaining an optimal balance between these antagonist principles is difficult with a single metaheurist. In tying to attain the required optimal balance between exploitation and exploration, another innovative excited-ACS-IDGWO complementary hybrid model comprising of two improved wrappers i.e. adaptive cuckoo search algorithm (ACS) and intensification dedicated grey wolf optimizer (IDGWO) (a variant of the EBGWO wrapper presented in Chapter 3) and using the SVM classifier is presented in Chapter 4. The proposed model innovatively adopts the concept of the complete voltage and current responses of a direct current (DC) excited resistor-capacitor (RC) circuit to nonlinearly control parameter $a^{\rightarrow}$ of IDGWO and the step size of ACS. To handle the higher diversity of the search space during the early stages, both the ACS and IDGWO are jointly involved in the local exploitation. Conversely, to promote mature convergence during later stages of the search space, the role of ACS is shifted to global exploration while the IDGWO is left carrying out local exploitation. The performance of the proposed model is compared with those of four state-of-art wrappers. The proposed technique emerged to be superior in attaining a good learning from a few samples and optimally deriving a reduced feature subset from the information-rich datasets. The superiority of the proposed E-ACS-

IDGWO is further proved via a number of statistical approaches like ranking techniques and statistical analysis.

The performance of EACSIDGWO as a gene selector is evaluated on six standard DNA microarray chips derived from Irvine (UCI) repository namely Ovarian Cancer (4000 genes), Central Nervous System Cancer (7129 genes), Colon Cancer (2000 genes), Breast Cancer Wisconsin (prognosis) (33 genes), Breast Cancer Wisconsin (diagnostic) (30 genes) and SPECTF Heart Cancer (44 genes). The EACSIDGWO achieved the most compact informative gene subsets along with the highest classification accuracies as follows: Ovarian Cancer (274 genes, 100%), Central Nervous System Cancer (1208 genes, 72%), Colon Cancer (538 genes, 91%), Breast Cancer Wisconsin (prognosis) (5 genes, 87%), Breast Cancer Wisconsin (diagnostic) (3 genes, 98%) and SPECTF Heart Cancer (4 genes, 88%). Extended Binary Cuckoo Search (EBCS), the second best state-of-the-art published algorithm, attained the following: Ovarian Cancer (1811 genes, 99%), Central Nervous System Cancer (3446 genes, 67%), Colon Cancer (988 genes, 89%), Breast Cancer Wisconsin (prognosis) (6 genes, 86%), Breast Cancer Wisconsin (diagnostic) (3 genes, 97%) and SPECTF Heart Cancer (6 genes, 86%). On average, the proposed EACSIDGWO algorithm attained a reduced informative gene subset with 339 genes along with a classification accuracy of 89%. On the other hand, on average the EBCS (second best algorithm) attained a reduced informative gene subset with 1043 genes along with a classification accuracy of 87% **Thus in comparison with** EBCS **(the current best improved version of the Binary Cuckoo Search algorithm), on average the proposed EBGWO algorithm reduced the number of selected genes from 1043 to 339 (i.e. a further reduction by 704 genes) while improving the classification accuracy from 87% to 89% (i.e. an improvement by 2%)**.

iii) From the results presented in (ii) above, the proposed hybrid EACSIDGWO algorithm achieved an optimal balance between exploitation and exploration during the search thus overcoming EBGWO's shortcoming. However, this wrapper adopted the SVM classifier (a commonly utilized classifier in DNA microarray based cancer classification) whose performance is largely dependent on the kernel adopted for it as well as tuning of the kernel parameters. Moreover, utilizing a single kernel function based MCSVM classifier in a given application such as gene expression data does not attain both a good learning ability, proper global feature extraction ability and a better generalization capability. Thus, to enhance both the learning and classification ability of the SVM classifier a particle swarm optimized hybrid kernel-based multi-class support vector machine i.e. PSO-PCA-LPG-MCSVM is presented in Chapter 5. In this model, particle swarm optimization (PSO) algorithm, principal component algorithm (a gene extractor) and multiclass support vector machine (MCSVM) that is based on a hybrid kernel i.e. linear-gaussian-polynomial (LGP) are combined. The major contribution of this work is the novel hybrid kernel i.e. LGP that combines the advantages of three standard kernels (linear, Gaussian and polynomial) in a novel manner; where the linear kernel is linearly combined with a Gaussian kernel that is embedding a polynomial kernel. Further, the validity of the proposed kernel is proved. The effectiveness of the proposed model is revealed by carrying out a number of experiments and obtained results compared with those of three single kernel-based models i.e. PSO-PCA-L-MCSVM, PSO-PCA-G-MCSVM and PSO-PCA-P-MCSVM that utilize the standard alone linear, polynomial and Gaussian kernels respectively. Two dual and two multiclass imbalanced DNA microarray datasets that are publicly available were utilized. The obtained experimental results in terms of three extended evaluation metrics i.e. G-mean, F-score and accuracy reveal how superior the proposed model is in terms of global feature extraction, learning and prediction , compared to the other standalone kernel-based models.

To reveal the superior global gene extraction, prediction and learning ability of this model against three single kernel-based models: PSO-PCA-L-MCSVM (using a single Linear kernel), PSO-G-MCSVM (using a single Gaussian kernel) and PSO-P-MCSVM (using a single Polynomial kernel), four datasets: Colon cancer (2000 genes), Acute Lymphoblastic Leukemia-Acute myeloid Leukemia (ALL-AML) (7129 genes), St. Jude Leukemia dataset (12558 genes) and Lung cancer(3312 genes) were used. Adopting three extended evaluation metrics (G-mean, Accuracy (Acc) and F-score) the proposed model achieved the following: Colon Cancer (G-mean: 0.88, Acc: 0.88, F-score: 0.87), ALL-AML (G-mean: 0.94, Acc: 0.94, F-score: 0.94), Lung Cancer (G-mean: 0.99, Acc: 0.97, F-score: 0.96) and St. Jude Leukemia dataset (G-mean: 0.97, Acc: 0.96, F-score: 0.90). The PSO-G-MCSVM, the second best published model, attained the following: Colon Cancer (G-mean: 0.82, Acc:

0.82, F-score: 0.82), ALL-AML (G-mean: 0.94, Acc: 0.94, F-score: 0.94), Lung Cancer (G-mean: 0.98, Acc: 0.96, F-score: 0.93) and St. Jude Leukemia dataset (G-mean: 0.97, Acc: 0.95, F-score: 0.85). On average, the proposed PSO-PCA-LPG-MCSVM algorithm attained the following for the four datasets: G-mean: 0.95, Acc: 0.94 and F-score: 0.92. On the other hand, on average the PSO-G-MCSVM (second best published model) attained the following for the four datasets: G-mean: 0.93, Acc: 0.92 and F-score: 0.89**. Thus in comparison with PSO-G-MCSVM (the second best published model), on average the proposed PSO-PCA-LPG-MCSVM model improved both the G-mean and Acc by 0.02 (2%) and F-score by 0.03 (3%).**

6.2 Key Findings

Table 6.1: Summary of research findings

TECHNIQUE	SALIENT FEATURES	ISSUES ADDRESSED	WEAKNESSES
EBGWO	• Wrapper based approach for gene selection. • Adopts a new position-updating criterion that utilizes fitness values of vectors $\vec{X}_1, \vec{X}_2$ and $\vec{X}_3$ to determine new search agents. This criterion maintains and strengthens the social hierarchy of the pack. • A novel nonlinear control strategy inspired by the complete current response of DC excited RC-circuit is developed to make full use of and balance exploration and exploitation of the existing BGWO. • Because this novel control strategy allocates a large number of generations to diversification in comparison to intensification, the convergence speed of the EBGWO algorithm is heightened while reducing the local optimal trapping effects. • To enhance diversity and improve the quality of the reported solutions a weighting scheme utilizing the fitness values of the three leaders of the pack (alpha(α), beta(β) and delta(δ)), that of the currently considered wolf and that worst wolf is adopted. • Achieved a more compact set of informative genes along with highest classification accuracy in comparison with all the other considered wrappers	• A challenge of highly dimensional DNA microarray datasets • Need for better classification accuracies using a small subset of informative genes • Need of robust and stable soft computing techniques for selection of informative genes and cancer disease classification.	• Though, better, stable and reliable results were attained, the great search complexity due to the large number of genes within the DNA microarray datasets increased the computation time. • Doesn't attain optimal balance between exploitation and exploration during the search process.

E-ACS-IDGWO	• Hybrid (Ensemble) wrapper based approached for gene selection • The EACSIDGWO algorithm hybridizes IDGWO (a variant of the EBGWO) and another new improved cuckoo search algorithm i.e. ACS. • The step size of ACS is innovatively made adaptive via the concept of complete voltage response of the direct current (DC) excited resistor-capacitor (RC) circuit • To handle the higher diversity of the search space during early stages, both ACS and IDGWO jointly carry out local exploitation. • To enhance mature convergence during later stages of the proposed algorithm, the role of ACS is switched to global exploration while the IDGWO is still left carrying out local exploitation • Good learning ability from using a few samples	• The large and complex search space of the highly dimensional DNA microarray datasets. • Combined joint and standalone gene selection approach to derive a subset of informative genes related to a specific cancer disease. • Optimal balance between exploitation and exploration of soft computing techniques for gene selection	• Based on supervised machine learning approach, which requires all the DNA microarray datasets to be labelled.However, availability of large microarray sample sizes is still a challenge.
PSO-PCA-LPG-MCSVM	• Hybrid(Ensemble) technique for DNA microarray data analysis • Utilizes a novel hybrid linear-gaussian-polynomial (LGP) kernel-based multiclass support vector machine that enhances the performance of classification stage (last stage of the DNA microarray data analysis) • The hybrid LGP kernel innovatively combines the advantages of three standard kernels (linear, gaussian and polynomial); where the linear kernel is linearly combined with a gaussian kernel embedding the polynomial kernel.	• Class Imbalance problem in DNA microarray data • Kernel selection for the multi-class support vector machine. • Reduction of the computation complexity of DNA microarray data using PCA (feature extractor)	• High cost of classification in terms of computation time due to the large number of genes within the DNA microarray datasets.

	• A proof to ensure that the proposed kernel conforms to the features of a valid kernel is carried out. • To tackle the class imbalance problem in microarray data analysis, three extended evaluation metrics i.e. G-mean, Accuracy and F-score are utilized. • Better generalization, learning and prediction ability achieved by the system		

6.3 Recommendations Further Work

This research focused on the development of fast, stable and reliable diagnostic techniques for DNA microarray gene expression data for cancer disease diagnosis and classification.

The proposed techniques can be utilized in other microarray based clinical research areas such as toxicological studies, drug response analysis and patient's survival.

Moreover, the results obtained can be further analyzed to determine the biological relevance so that this information can aid biologists in providing accurate and timely interpretations of attained outcome.

In this research work, all the formulated models only adopted supervised machine learning methods for cancer disease diagnosis and classification, which require all the tissue samples to be labeled. However, there exists a number of DNA microarray datasets with unlabeled data. Thus, it will be important for semi-supervised learning techniques to be considered as well.

For the proposed EBGWO and E-ACS-IDGWO techniques, the sigmoid transfer function was used to convert the continuous solutions of the search agents to binary for feature selection. To further improve the performance of these approaches, it could be important to investigate on the fully-binary versions of these techniques.

Finally, optimized versions of the proposed algorithms can be adapted in collecting, analyzing cancerG data using mobile devices with limited computing resources.

REFERENCES

[1] N. Almugren and H. Alshamlan, “A Survey on Hybrid Feature Selection Methods in Microarray Gene Expression Data for Cancer Classification,” *IEEE Access*, vol. 7, pp. 78533–78548, 2019, doi: 10.1109/ACCESS.2019.2922987.

[2] M. Qaraad, S. Amjad, H. Fathi, and I. I. M. Manhrawy, “Feature Selection Techniques for Cancer Classification applied to Microarray Data: A survey,” in *2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS)*, Dec. 2019, pp. 1–8, doi: 10.1109/ISACS48493.2019.9068865.

[3] Z. Ahmed, S. Zeeshan, D. Mendhe, and X. Dong, “Human gene and disease associations for clinical- genomics and precision medicine research,” *Clin. Transl. Med.*, vol. 10, no. 1, pp. 297–318, May 2020, doi: 10.1002/ctm2.28.

[4] W. Guo, T. Zeng, T. Huang, and Y.-D. Cai, “Disease Cluster Detection and Functional Characterization,” *IEEE Access*, vol. 8, pp. 141958–141966, 2020, doi: 10.1109/ACCESS.2020.3013666.

[5] G. D. Syu, J. Dunn, and H. Zhu, “Developments and Applications of Functional Protein Microarrays,” *Mol. Cell. Proteomics MCP*, vol. 19, no. 6, pp. 916–927, Jun. 2020, doi: 10.1074/mcp.R120.001936.

[6] M. W. Farouq, W. Boulila, M. Abdel-aal, A. Hussain, and A.-B. Salem, “A Novel Multi-Stage Fusion based Approach for Gene Expression Profiling in Non-Small Cell Lung Cancer,” *IEEE Access*, vol. 7, pp. 37141–37150, 2019, doi: 10.1109/ACCESS.2019.2898897.

[7] T. Khorshed, M. N. Moustafa, and A. Rafea, “Deep Learning for Multi-Tissue Cancer Classification of Gene Expressions (GeneXNet),” *IEEE Access*, vol. 8, pp. 90615–90629, 2020, doi: 10.1109/ACCESS.2020.2992907.

[8] R. K. Singh and M. Sivabalakrishnan, “Feature Selection of Gene Expression Data for Cancer Classification: A Review,” *Procedia Comput. Sci.*, vol. 50, pp. 52–57, Jan. 2015, doi: 10.1016/j.procs.2015.04.060.

[9] “Kenya Cancer Statistics & National Strategies,” *Kenyan Network of Cancer Organizations*, Feb. 18, 2013. https://kenyacancernetwork.wordpress.com/kenya-cancer-facts/ (accessed Oct. 19, 2020).

[10] F. W. Wambalaba, B. Son, A. E. Wambalaba, D. Nyong’o, and A. Nyong’o, “Prevalence and Capacity of Cancer Diagnostics and Treatment: A Demand and Supply Survey of Health-Care Facilities in Kenya,” *Cancer Control J. Moffitt Cancer Cent.*, vol. 26, no. 1, Dec. 2019, doi: 10.1177/1073274819886930.

[11] N. Hawkes, “Cancer survival data emphasise importance of early diagnosis,” *BMJ*, vol. 364, Jan. 2019, doi: 10.1136/bmj.l408.

[12] R. Govindarajan, J. Duraiyan, K. Kaliyappan, and M. Palanisamy, “Microarray and its applications,” *J. Pharm. Bioallied Sci.*, vol. 4, no. Suppl 2, pp. S310–S312, Aug. 2012, doi: 10.4103/0975-7406.100283.

[13] M. Kenn, D. Cacsire Castillo-Tong, C. F. Singer, M. Cibena, H. Kölbl, and W. Schreiner, “Microarray Normalization Revisited for Reproducible Breast Cancer Biomarkers,” *BioMed Research International*, Aug. 06, 2020. https://www.hindawi.com/journals/bmri/2020/1363827/ (accessed Oct. 19, 2020).

[14] L. Zhang, J. Chen, T. Cheng, H. Yang, C. Pan, and H. Li, “Identification of Differentially Expressed Genes and miRNAs Associated with Esophageal Squamous Cell Carcinoma by Integrated Analysis of Microarray Data,” *BioMed Research International*, Jul. 02, 2020. https://www.hindawi.com/journals/bmri/2020/1980921/ (accessed Oct. 19, 2020).

[15] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, “Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray,” *Science*, vol. 270, no. 5235, pp. 467–470, Oct. 1995, doi: 10.1126/science.270.5235.467.

[16] D. J. Lockhart *et al.*, “Expression monitoring by hybridization to high-density oligonucleotide arrays,” *Nat. Biotechnol.*, vol. 14, no. 13, pp. 1675–1680, Dec. 1996, doi: 10.1038/nbt1296-1675.

[17] H. Zhang and S. Li, “DNA Microarray Assay Helps to Identify Functional Genes Specific for Leukemia Stem Cells,” *Dataset Papers in Science*, Sep. 04, 2013. https://www.hindawi.com/journals/dpis/2013/520285/ (accessed Oct. 19, 2020).

[18] J. Zhang and Y. Zhou, “Identification of Key Genes and Pathways Associated with Age-Related Macular Degeneration,” *Journal of Ophthalmology*, Aug. 21, 2020. https://www.hindawi.com/journals/joph/2020/2714746/ (accessed Oct. 19, 2020).

[19] “Microarray Applications,” *News-Medical.net*, Jun. 13, 2017. https://www.news-medical.net/life-sciences/Microarray-Aplications.aspx (accessed Oct. 19, 2020).

[20] T. Worku and D. Negassu, “Review on DNA Micro Array Technologyand Its Application,” *Am. J. Zool.*, vol. 2, no. 4, Art. no. 4, Jan. 2020, doi: 10.11648/j.ajz.20190204.11.

[21] “15.6: Genomic Approaches- The DNA Microarray,” *Biology LibreTexts*, Dec. 20, 2018. https://bio.libretexts.org/Bookshelves/Cell_and_Molecular_Biology/Book%3A_Basic_Cell_and_Molecular_Biology_(Bergtrom)/15%3A_DNA_Technologies/15.06%3A_Genomic_Approaches-_The_DNA_Microarray (accessed Oct. 19, 2020).

[22] J. Pati, “Gene Expression Analysis for Early Lung Cancer Prediction Using Machine Learning Techniques: An Eco-Genomics Approach,” *IEEE Access*, vol. 7, pp. 4232–4238, 2019, doi: 10.1109/ACCESS.2018.2886604.

[23] “Oligonucleotide Microarrays - an overview | ScienceDirect Topics.” https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/oligonucleotide-microarrays (accessed Oct. 19, 2020).

[24] J. DeRisi *et al.*, “Use of a cDNA microarray to analyse gene expression patterns in human cancer,” *Nat. Genet.*, vol. 14, no. 4, pp. 457–460, Dec. 1996, doi: 10.1038/ng1296-457.

[25] N. Behera, S. Sinha, R. Gupta, A. Geoncy, N. Dimitrova, and J. Mazher, “Analysis of Gene Expression Data by Evolutionary Clustering Algorithm,” in *2017 International Conference on Information Technology (ICIT)*, Dec. 2017, pp. 165–169, doi: 10.1109/ICIT.2017.41.

[26] “(496) Pinterest,” *Pinterest*. https://www.pinterest.com/pin/537124693039577832/ (accessed Oct. 19, 2020).

[27] K. Raza, “Analysis of Microarray Data using Artificial Intelligence Based Techniques,” *Handbook of Research on Computational Intelligence Applications in Bioinformatics*, 2016. www.igi-global.com/chapter/analysis-of-microarray-data-using-artificial-intelligence-based-techniques/157490 (accessed Oct. 19, 2020).

[28] F. Rafii, B. D. R. Hassani, and M. A. Kbir, “New Approach for Microarray Data Decision Making with Respect to Multiple Sources,” in *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*, New York, NY, USA, Mar. 2017, pp. 1–5, doi: 10.1145/3090354.3090463.

[29] G. Paumier, *English: A DNA microarray.* 2008.

[30] F. Maleki, K. Ovens, D. J. Hogan, and A. J. Kusalik, “Gene Set Analysis: Challenges, Opportunities, and Future Research,” *Front. Genet.*, vol. 11, 2020, doi: 10.3389/fgene.2020.00654.

[31] F. Tang *et al.*, “Identification of differentially expressed genes and biological pathways in bladder cancer,” *Mol. Med. Rep.*, vol. 17, no. 5, pp. 6425–6434, May 2018, doi: 10.3892/mmr.2018.8711.

[32] J. M. Garcia-Manteiga *et al.*, “Epigenomics of Neural Cells: REST-Induced Down- and Upregulation of Gene Expression in a Two-Clone PC12 Cell Model,” *BioMed Research International*, Aug. 27, 2015. https://www.hindawi.com/journals/bmri/2015/202914/ (accessed Oct. 19, 2020).

[33] G. A. Churchill, “Using ANOVA to Analyze Microarray Data,” *BioTechniques*, vol. 37, no. 2, pp. 173–177, Aug. 2004, doi: 10.2144/04372TE01.

[34] M. Kumar, N. K. Rath, A. Swain, and S. K. Rath, “Feature Selection and Classification of Microarray Data using MapReduce based ANOVA and K-Nearest Neighbor,” *Procedia Comput. Sci.*, vol. 54, pp. 301–310, Jan. 2015, doi: 10.1016/j.procs.2015.06.035.

[35] A. L. Cope, B. C. O’Meara, and M. A. Gilchrist, “Gene expression of functionally-related genes coevolves across fungal species: detecting coevolution of gene expression using phylogenetic comparative methods,” *BMC Genomics*, vol. 21, no. 1, p. 370, May 2020, doi: 10.1186/s12864-020-6761-3.

[36] S. van Dam, U. Võsa, A. van der Graaf, L. Franke, and J. P. de Magalhães, “Gene co-expression analysis for functional classification and gene–disease predictions,” *Brief. Bioinform.*, vol. 19, no. 4, pp. 575–592, Jan. 2017, doi: 10.1093/bib/bbw139.

[37] “Microarray data clustering and visualization tool using self-organizing maps - IEEE Conference Publication.” https://ieeexplore.ieee.org/document/7387955 (accessed Oct. 19, 2020).

[38] A. Rouhi and H. Nezamabadi-Pour, “Feature Selection in High-Dimensional Data,” in *Optimization, Learning, and Control for Interdependent Complex Networks*, M. H. Amini, Ed. Cham: Springer International Publishing, 2020, pp. 85–128.

[39] U. Shruthi, V. Nagaveni, and B. K. Raghavendra, “A Review on Machine Learning Classification Techniques for Plant Disease Detection,” in *2019 5th International Conference on Advanced Computing Communication Systems (ICACCS)*, Mar. 2019, pp. 281–284, doi: 10.1109/ICACCS.2019.8728415.

[40] C. E. Crangle, R. Wang, M. Perreau-Guimaraes, M. U. Nguyen, D. T. Nguyen, and P. Suppes, “Machine learning for the recognition of emotion in the speech of couples in psychotherapy using the Stanford Suppes Brain Lab Psychotherapy Dataset,” *ArXiv190104110 Cs Eess*, Jan. 2019, Accessed: Oct. 26, 2020. [Online]. Available: http://arxiv.org/abs/1901.04110.

[41] A. Rouhi, M. Spitale, F. Catania, G. Cosentino, M. Gelsomini, and F. Garzotto, “Emotify: emotional game for children with autism spectrum disorder based-on machine learning,” in *Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion*, New York, NY, USA, Mar. 2019, pp. 31–32, doi: 10.1145/3308557.3308688.

[42] R. O. Duda, P. E. Hart, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley, 2001.

[43] M. Fernandes, A. Canito, V. Bolón-Canedo, L. Conceição, I. Praça, and G. Marreiros, “Data analysis and feature selection for predictive maintenance: A case-study in the metallurgic industry,” *Int. J. Inf. Manag.*, vol. 46, pp. 252–262, Jun. 2019, doi: 10.1016/j.ijinfomgt.2018.10.006.

[44] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Springer US, 1998.

[45] H. Handels, Th. Roß, J. Kreusch, H. H. Wolff, and S. J. Pöppl, “Feature selection for optimized skin tumor recognition using genetic algorithms,” *Artif. Intell. Med.*, vol. 16, no. 3, pp. 283–297, Jul. 1999, doi: 10.1016/S0933-3657(99)00005-6.

[46] B. Nikpour and H. Nezamabadi-pour, “HTSS: a hyper-heuristic training set selection method for imbalanced data sets,” *Iran J. Comput. Sci.*, vol. 1, no. 2, 2018, doi: 10.1007/s42044-018-0009-2.

[47] K. Borowska and J. Stepaniuk, “A rough-granular approach to the imbalanced data classification problem,” *Appl. Soft Comput.*, vol. 83, p. 105607, Oct. 2019, doi: 10.1016/j.asoc.2019.105607.

[48] A. Reyes-Nava, H. Cruz-Reyes, R. Alejo, E. Rendón-Lara, A. A. Flores-Fuentes, and E. E. Granda-Gutiérrez, “Using Deep Learning to Classify Class Imbalanced Gene-Expression Microarrays Datasets,” in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Cham, 2019, pp. 46–54, doi: 10.1007/978-3-030-13469-3_6.

[49] P. Branco, L. Torgo, and R. Ribeiro, “A Survey of Predictive Modelling under Imbalanced Distributions,” *ArXiv150501658 Cs*, May 2015, Accessed: Oct. 26, 2020. [Online]. Available: http://arxiv.org/abs/1505.01658.

[50] H. He and E. A. Garcia, “Learning from Imbalanced Data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009, doi: 10.1109/TKDE.2008.239.

[51] R. J. Hickey, “Noise modelling and evaluating learning from examples,” *Artif. Intell.*, vol. 82, no. 1, pp. 157–179, Apr. 1996, doi: 10.1016/0004-3702(94)00094-8.

[52] Z. Sun, Q. Song, X. Zhu, H. Sun, B. Xu, and Y. Zhou, “A novel ensemble method for classifying imbalanced data,” *Pattern Recognit.*, vol. 48, no. 5, pp. 1623–1637, May 2015, doi: 10.1016/j.patcog.2014.11.014.

[53] C. E. Brodley and M. A. Friedl, “Identifying mislabeled training data,” *J. Artif. Intell. Res.*, vol. 11, no. 1, pp. 131–167, Jul. 1999.

[54] B. Frénay and A. Kaban, “A Comprehensive Introduction to Label Noise,” *Proc. 2014 Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn. ESANN 2014*, 2014, Accessed: Oct. 26, 2020. [Online]. Available: https://researchportal.unamur.be/en/publications/a-comprehensive-introduction-to-label-noise-proceedings-of-the-20.

[55] F. Barani, M. Mirhosseini, and H. Nezamabadi-pour, “Application of binary quantum-inspired gravitational search algorithm in feature subset selection,” *Appl. Intell.*, vol. 47, no. 2, pp. 304–318, Sep. 2017, doi: 10.1007/s10489-017-0894-3.

[56] A. P. Dawid and A. M. Skene, “Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm,” *J. R. Stat. Soc. Ser. C Appl. Stat.*, vol. 28, no. 1, pp. 20–28, 1979, doi: 10.2307/2346806.

[57] T. R. Golub *et al.*, “Molecular classification of cancer: class discovery and class prediction by gene expression monitoring,” *Science*, vol. 286, no. 5439, pp. 531–537, Oct. 1999, doi: 10.1126/science.286.5439.531.

[58] I. Kamkar, S. K. Gupta, D. Phung, and S. Venkatesh, “Stable feature selection for clinical prediction: Exploiting ICD tree structure using Tree-Lasso,” *J. Biomed. Inform.*, vol. 53, pp. 277–290, Feb. 2015, doi: 10.1016/j.jbi.2014.11.013.

[59] A. Rouhi and H. Nezamabadi-pour, “A hybrid feature selection approach based on ensemble method for high-dimensional data,” in *2017 2nd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, Mar. 2017, pp. 16–20, doi: 10.1109/CSIEC.2017.7940163.

[60] N. Almugren and H. Alshamlan, “A Survey on Hybrid Feature Selection Methods in Microarray Gene Expression Data for Cancer Classification,” *IEEE Access*, vol. 7, pp. 78533–78548, 2019, doi: 10.1109/ACCESS.2019.2922987.

[61] S. Tabakhi, A. Najafi, R. Ranjbar, and P. Moradi, “Gene selection for microarray data classification using a novel ant colony optimization,” *Neurocomputing*, vol. 168, pp. 1024–1036, Nov. 2015, doi: 10.1016/j.neucom.2015.05.022.

[62] M. K. Ebrahimpour, H. Nezamabadi-pour, and M. Eftekhari, “CCFS: A cooperating coevolution technique for large scale feature selection on microarray datasets,” *Comput. Biol. Chem.*, vol. 73, pp. 171–178, Apr. 2018, doi: 10.1016/j.compbiolchem.2018.02.006.

[63] A. Rouhi and H. Nezamabadi-pour, “Filter-based feature selection for microarray data using improved binary gravitational search algorithm,” in *2018 3rd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, Mar. 2018, pp. 1–6, doi: 10.1109/CSIEC.2018.8405411.

[64] Y.-W. Chen and C.-J. Lin, “Combining SVMs with Various Feature Selection Strategies,” in *Feature Extraction: Foundations and Applications*, I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, Eds. Berlin, Heidelberg: Springer, 2006, pp. 315–324.

[65] J. R. Quinlan, “Induction of decision trees,” *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986, doi: 10.1007/BF00116251.

[66] I. Kononenko, “Estimating attributes: Analysis and extensions of RELIEF,” in *Machine Learning: ECML-94*, Berlin, Heidelberg, 1994, pp. 171–182, doi: 10.1007/3-540-57868-4_57.

[67] Hanchuan Peng, Fuhui Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005, doi: 10.1109/TPAMI.2005.159.

[68] L. Yu and H. Liu, “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution,” in *Proceedings, Twentieth International Conference on Machine Learning*, Dec. 2003, pp. 856–863, Accessed: Oct. 29, 2020. [Online]. Available: https://asu.pure.elsevier.com/en/publications/feature-selection-for-high-dimensional-data-a-fast-correlation-ba.

[69] J. Li *et al.*, “Feature Selection: A Data Perspective,” *ACM Comput. Surv.*, vol. 50, no. 6, p. 94:1–94:45, Dec. 2017, doi: 10.1145/3136625.

[70] A. Rouhi and H. Nezamabadi-pour, “A hybrid method for dimensionality reduction in microarray data based on advanced binary ant colony algorithm,” *2016 1st Conf. Swarm Intell. Evol. Comput. CSIEC*, 2016, doi: 10.1109/CSIEC.2016.7482124.

[71] N. Taheri and H. Nezamabadi-pour, “A hybrid feature selection method for high-dimensional data,” in *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, Oct. 2014, pp. 141–145, doi: 10.1109/ICCKE.2014.6993381.

[72] Q. Gu, Z. Li, and J. Han, “Generalized Fisher score for feature selection,” in *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, Arlington, Virginia, USA, Jul. 2011, pp. 266–273, Accessed: Oct. 28, 2020. [Online].

[73] X. He, D. Cai, and P. Niyogi, “Laplacian score for feature selection,” in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, Cambridge, MA, USA, Dec. 2005, pp. 507–514, Accessed: Oct. 28, 2020. [Online].

[74] M. A. Hall and L. A. Smith, “Feature Selection for Machine Learning: Comparing a Correlation-based Filter Approach to the Wrapper,” p. 5.

[75] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd Edition. Cambridge, UK ; New York: Cambridge University Press, 2007.

[76] J. C. Davis and R. J. Sampson, *Statistics and Data Analysis in Geology*. New York: John Wiley and Sons, 1986.

[77] H. Lee *et al.*, “Feature selection practice for unsupervised learning of credit card fraud detection,” *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 2, pp. 408–417, Jan. 2018.

[78] Y. Saeys, I. Inza, and P. Larrañaga, “A review of feature selection techniques in bioinformatics,” *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Oct. 2007, doi: 10.1093/bioinformatics/btm344.

[79] A. Rouhi and H. N. pour, “A hybrid-ensemble based framework for microarray data gene selection,” *Int. J. Data Min. Bioinforma.*, vol. 19, no. 3, p. 221, 2017, doi: 10.1504/IJDMB.2017.090987.

[80] S. Kashef, H. Nezamabadi- pour, and B. Nikpour, “Multilabel feature selection: A comprehensive review and guiding experiments,” *WIREs Data Min. Knowl. Discov.*, vol. 8, no. 2, p. e1240, 2018, doi: 10.1002/widm.1240.

[81] M. B. Dowlatshahi, V. Derhami, and H. Nezamabadi-pour, “Ensemble of Filter-Based Rankers to Guide an Epsilon-Greedy Swarm Optimizer for High-Dimensional Feature Subset Selection,” *Information*, vol. 8, no. 4, Art. no. 4, Dec. 2017, doi: 10.3390/info8040152.

[82] M. Dorigo and G. D. Caro, “Ant colony optimization: a new meta-heuristic,” in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Jul. 1999, vol. 2, pp. 1470-1477 Vol. 2, doi: 10.1109/CEC.1999.782657.

[83] S. Kashef and H. Nezamabadi-pour, “An advanced ACO algorithm for feature subset selection,” *Neurocomputing*, vol. 147, pp. 271–279, Jan. 2015, doi: 10.1016/j.neucom.2014.06.067.

[84] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, Nov. 1995, vol. 4, pp. 1942–1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[85] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, “GSA: A Gravitational Search Algorithm,” *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009, doi: 10.1016/j.ins.2009.03.004.

[86] A. Mahanipour and H. Nezamabadi-pour, “A multiple feature construction method based on gravitational search algorithm,” *Expert Syst. Appl.*, vol. 127, pp. 199–209, Aug. 2019, doi: 10.1016/j.eswa.2019.03.015.

[87] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, “BGSA: binary gravitational search algorithm,” *Nat. Comput.*, vol. 9, no. 3, pp. 727–745, Sep. 2010, doi: 10.1007/s11047-009-9175-3.

[88] E. Rashedi and H. Nezamabadi-pour, “Feature subset selection using improved binary gravitational search algorithm,” *J. Intell. Fuzzy Syst.*, vol. 26, no. 3, pp. 1211–1221, Jan. 2014, doi: 10.3233/IFS-130807.

[89] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[90] E. Emary, H. M. Zawbaa, and A. E. Hassanien, “Binary grey wolf optimization approaches for feature selection,” *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016, doi: 10.1016/j.neucom.2015.06.083.

[91] J. Too, A. R. Abdullah, N. Mohd Saad, N. Mohd Ali, and W. Tee, "A New Competitive Binary Grey Wolf Optimizer to Solve the Feature Selection Problem in EMG Signals Classification," *Computers*, vol. 7, no. 4, Art. no. 4, Dec. 2018, doi: 10.3390/computers7040058.

[92] R. A and N. P. H, "A Hybrid-Based Feature Selection Method For High-Dimensional Data Using Ensemble Methods," vol. 15, no. 4, pp. 283–294, Jan. 2018.

[93] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Inf. Sci.*, vol. 282, pp. 111–135, Oct. 2014, doi: 10.1016/j.ins.2014.05.042.

[94] P. A. Mundra and J. C. Rajapakse, "SVM-RFE With MRMR Filter for Gene Selection," *IEEE Trans. NanoBioscience*, vol. 9, no. 1, pp. 31–37, Mar. 2010, doi: 10.1109/TNB.2009.2035284.

[95] H. Uğuz, "A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm," *Knowl.-Based Syst.*, vol. 24, no. 7, pp. 1024–1032, Oct. 2011, doi: 10.1016/j.knosys.2011.04.014.

[96] L.-Y. Chuang, C.-H. Yang, K.-C. Wu, and C.-H. Yang, "A hybrid feature selection method for DNA microarray data," *Comput. Biol. Med.*, vol. 41, no. 4, pp. 228–237, Apr. 2011, doi: 10.1016/j.compbiomed.2011.02.004.

[97] C.-P. Lee and Y. Leu, "A novel hybrid feature selection method for microarray data analysis," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 208–213, Jan. 2011, doi: 10.1016/j.asoc.2009.11.010.

[98] S. S. Shreem, S. Abdullah, M. Z. A. Nazri, and M. Alzaqebah, "Hybridizing Relieff, Mrmr Filters And Ga Wrapper Approaches For Gene Selection," . *Vol.*, vol. 46, p. 6, 2005.

[99] J. Apolloni, G. Leguizamón, and E. Alba, "Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments," *Appl. Soft Comput.*, vol. 38, pp. 922–932, Jan. 2016, doi: 10.1016/j.asoc.2015.10.037.

[100] B. Venkatesh and J. Anuradha, "A Hybrid Feature Selection Approach for Handling a High-Dimensional Data," in *Innovations in Computer Science and Engineering*, Singapore, 2019, pp. 365–373, doi: 10.1007/978-981-13-7082-3_42.

[101] Z. Manbari, F. AkhlaghianTab, and C. Salavati, "Hybrid fast unsupervised feature selection for high-dimensional data," *Expert Syst. Appl.*, vol. 124, pp. 97–118, Jun. 2019, doi: 10.1016/j.eswa.2019.01.016.

[102] C. Yan, J. Liang, M. Zhao, X. Zhang, T. Zhang, and H. Li, "A novel hybrid feature selection strategy in quantitative analysis of laser-induced breakdown spectroscopy," *Anal. Chim. Acta*, vol. 1080, pp. 35–42, Nov. 2019, doi: 10.1016/j.aca.2019.07.012.

[103] T. Gangavarapu and N. Patil, “A novel filter–wrapper hybrid greedy ensemble approach optimized using the genetic algorithm to reduce the dimensionality of high-dimensional biomedical datasets,” *Appl. Soft Comput.*, vol. 81, p. 105538, Aug. 2019, doi: 10.1016/j.asoc.2019.105538.

[104] L. Sun, X. Kong, J. Xu, Z. Xue, R. Zhai, and S. Zhang, “A Hybrid Gene Selection Method Based on ReliefF and Ant Colony Optimization Algorithm for Tumor Classification,” *Sci. Rep.*, vol. 9, no. 1, Art. no. 1, Jun. 2019, doi: 10.1038/s41598-019-45223-x.

[105] W. You, Z. Yang, and G. Ji, “PLS-based recursive feature elimination for high-dimensional small sample,” *Knowl.-Based Syst.*, vol. 55, pp. 15–28, Jan. 2014, doi: 10.1016/j.knosys.2013.10.004.

[106] T. Prasartvit, A. Banharnsakun, B. Kaewkamnerdpong, and T. Achalakul, “Reducing bioinformatics data dimension with ABC-kNN,” *Neurocomputing*, vol. 116, pp. 367–381, Sep. 2013, doi: 10.1016/j.neucom.2012.01.045.

[107] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene Selection for Cancer Classification using Support Vector Machines,” *Mach. Learn.*, vol. 46, no. 1, pp. 389–422, Jan. 2002, doi: 10.1023/A:1012487302797.

[108] S. Maldonado, R. Weber, and J. Basak, “Simultaneous feature selection and classification using kernel-penalized support vector machines,” *Inf. Sci.*, vol. 181, no. 1, pp. 115–128, Jan. 2011, doi: 10.1016/j.ins.2010.08.047.

[109] J. Canul-Reich, L. O. Hall, D. B. Goldgof, J. N. Korecki, and S. Eschrich, “Iterative feature perturbation as a gene selector for microarray data,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 26, no. 05, p. 1260003, Aug. 2012, doi: 10.1142/S0218001412600038.

[110] S. Maldonado and J. López, “Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification,” *Appl. Soft Comput.*, vol. 67, pp. 94–105, Jun. 2018, doi: 10.1016/j.asoc.2018.02.051.

[111] H. Liu, M. Zhou, and Q. Liu, “An embedded feature selection method for imbalanced data classification,” *IEEECAA J. Autom. Sin.*, vol. 6, no. 3, pp. 703–715, May 2019, doi: 10.1109/JAS.2019.1911447.

[112] C. Peng, X. Wu, W. Yuan, X. Zhang, and Y. Li, “MGRFE: multilayer recursive feature elimination based on an embedded genetic algorithm for cancer classification,” *IEEE/ACM Trans. Comput. Biol. Bioinform.*, pp. 1–1, 2019, doi: 10.1109/TCBB.2019.2921961.

[113] A. B. Brahim and M. Limam, “Robust ensemble feature selection for high dimensional data sets,” in *2013 International Conference on High Performance Computing Simulation (HPCS)*, Jul. 2013, pp. 151–157, doi: 10.1109/HPCSim.2013.6641406.

[114] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, “Data classification using an ensemble of filters,” *Neurocomputing*, vol. 135, pp. 13–20, Jul. 2014, doi: 10.1016/j.neucom.2013.03.067.

[115] F. Yang and K. Z. Mao, “Robust Feature Selection for Microarray Data Based on Multicriterion Fusion,” *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 8, no. 4, pp. 1080–1092, Jul. 2011, doi: 10.1109/TCBB.2010.103.

[116] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, “An ensemble of filters and classifiers for microarray data classification,” *Pattern Recognit.*, vol. 45, no. 1, pp. 531–539, Jan. 2012, doi: 10.1016/j.patcog.2011.06.006.

[117] S. Sayed, M. Nassef, A. Badr, and I. Farag, “A Nested Genetic Algorithm for feature selection in high-dimensional cancer Microarray datasets,” *Expert Syst. Appl.*, vol. 121, pp. 233–243, May 2019, doi: 10.1016/j.eswa.2018.12.022.

[118] B. Pes, “Ensemble feature selection for high-dimensional data: a stability analysis across multiple domains,” *Neural Comput. Appl.*, vol. 32, no. 10, pp. 5951–5973, May 2020, doi: 10.1007/s00521-019-04082-3.

[119] B. Singh, K. Kumar, S. Mohan, and R. Ahmad, “Ensemble of Clustering Approaches for Feature Selection of High Dimensional Data,” Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 3349018, Feb. 2019. doi: 10.2139/ssrn.3349018.

[120] J. Wang, J. Xu, C. Zhao, Y. Peng, and H. Wang, “An ensemble feature selection method for high-dimensional data based on sort aggregation,” *Syst. Sci. Control Eng.*, vol. 7, no. 2, pp. 32–39, Nov. 2019, doi: 10.1080/21642583.2019.1620658.

[121] X. Song, L. R. Waitman, Y. Hu, A. S. L. Yu, D. Robins, and M. Liu, “Robust clinical marker identification for diabetic kidney disease with ensemble feature selection,” *J. Am. Med. Inform. Assoc.*, vol. 26, no. 3, pp. 242–253, Mar. 2019, doi: 10.1093/jamia/ocy165.

[122] V. P. Singh, D. J. Kalita, and D. S. Tripathi, “Classifying Gene Expression Data of Cancer Using Multistage Ensemble of Neural Networks,” Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 3349578, Mar. 2019. doi: 10.2139/ssrn.3349578.

[123] H. Lu, J. Chen, K. Yan, Q. Jin, Y. Xue, and Z. Gao, “A hybrid feature selection algorithm for gene expression data classification,” *Neurocomputing*, vol. 256, pp. 56–62, Sep. 2017, doi: 10.1016/j.neucom.2016.07.080.

[124] M. Dashtban and M. Balafar, “Gene selection for microarray cancer classification using a new evolutionary method employing artificial intelligence concepts,” *Genomics*, vol. 109, no. 2, pp. 91–107, Mar. 2017, doi: 10.1016/j.ygeno.2017.01.004.

[125] F. Vafaee Sharbaf, S. Mosafer, and M. H. Moattar, "A hybrid gene selection approach for microarray data classification using cellular learning automata and ant colony optimization," *Genomics*, vol. 107, no. 6, pp. 231–238, Jun. 2016, doi: 10.1016/j.ygeno.2016.05.001.

[126] S. A. A. Vijay and P. GaneshKumar, "Fuzzy Expert System based on a Novel Hybrid Stem Cell (HSC) Algorithm for Classification of Micro Array Data," *J. Med. Syst.*, vol. 42, no. 4, p. 61, Feb. 2018, doi: 10.1007/s10916-018-0910-0.

[127] M. Dashtban, M. Balafar, and P. Suravajhala, "Gene selection for tumor classification using a novel bio-inspired multi-objective approach," *Genomics*, vol. 110, no. 1, pp. 10–17, Jan. 2018, doi: 10.1016/j.ygeno.2017.07.010.

[128] R. Aziz, C. K. Verma, and N. Srivastava, "A novel approach for dimension reduction of microarray," *Comput. Biol. Chem.*, vol. 71, pp. 161–169, Dec. 2017, doi: 10.1016/j.compbiolchem.2017.10.009.

[129] H. Alshamlan, G. Badr, and Y. Alohali, "mRMR-ABC: A Hybrid Gene Selection Algorithm for Cancer Classification Using Microarray Gene Expression Profiling," *BioMed Research International*, Apr. 15, 2015. https://www.hindawi.com/journals/bmri/2015/604910/ (accessed Oct. 30, 2020).

[130] I. Jain, V. K. Jain, and R. Jain, "Correlation feature selection based improved-Binary Particle Swarm Optimization for gene selection and cancer classification," *Appl. Soft Comput.*, vol. 62, pp. 203–215, Jan. 2018, doi: 10.1016/j.asoc.2017.09.038.

[131] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Appl. Soft Comput.*, vol. 43, pp. 117–130, Jun. 2016, doi: 10.1016/j.asoc.2016.01.044.

[132] E. Pashaei, M. Ozen, and N. Aydin, "Gene selection and classification approach for microarray data based on Random Forest Ranking and BBHA," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, Feb. 2016, pp. 308–311, doi: 10.1109/BHI.2016.7455896.

[133] X. Li and M. Yin, "Multiobjective Binary Biogeography Based Optimization for Feature Selection Using Gene Expression Data," *IEEE Trans. NanoBioscience*, vol. 12, no. 4, pp. 343–353, Dec. 2013, doi: 10.1109/TNB.2013.2294716.

[134] S. S. Shreem, S. Abdullah, and M. Z. A. Nazri, "Hybrid feature selection algorithm using symmetrical uncertainty and a harmony search algorithm," *Int. J. Syst. Sci.*, vol. 47, no. 6, pp. 1312–1329, Apr. 2016, doi: 10.1080/00207721.2014.924600.

[135] P. Tumuluru, “GOA-based DBN: Grasshopper Optimization Algorithm-based Deep Belief Neural Networks for Cancer Classification,” vol. 12, no. 24, p. 14, 2017.

[136] H. Djellali, S. Guessoum, N. Ghoualmi-Zine, and S. Layachi, “Fast correlation based filter combined with genetic algorithm and particle swarm on feature selection,” in *2017 5th International Conference on Electrical Engineering - Boumerdes (ICEE-B)*, Oct. 2017, pp. 1–6, doi: 10.1109/ICEE-B.2017.8192090.

[137] H. M. Alshamlan, G. H. Badr, and Y. A. Alohali, “Genetic Bee Colony (GBC) algorithm: A new gene selection method for microarray cancer classification,” *Comput. Biol. Chem.*, vol. 56, pp. 49–60, Jun. 2015, doi: 10.1016/j.compbiolchem.2015.03.001.

[138] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, “Recent advances and emerging challenges of feature selection in the context of big data,” *Knowl.-Based Syst.*, vol. 86, pp. 33–45, Sep. 2015, doi: 10.1016/j.knosys.2015.05.014.

[139] S. Nogueira and G. Brown, “Measuring the Stability of Feature Selection,” in *Machine Learning and Knowledge Discovery in Databases*, Cham, 2016, pp. 442–457, doi: 10.1007/978-3-319-46227-1_28.

[140] J. Pirgazi, M. Alimoradi, T. Esmaeili Abharian, and M. H. Olyaee, “An Efficient hybrid filter-wrapper metaheuristic-based gene selection method for high dimensional datasets,” *Sci. Rep.*, vol. 9, no. 1, p. 18580, Dec. 2019, doi: 10.1038/s41598-019-54987-1.

[141] P. Shunmugapriya and S. Kanmani, “A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid),” *Swarm Evol Comput*, vol. 36, pp. 27–36, 2017, doi: 10.1016/j.swevo.2017.04.002.

[142] Y. Sun, C. Lu, and X. Li, “The Cross-Entropy Based Multi-Filter Ensemble Method for Gene Selection,” *Genes*, vol. 9, no. 5, Art. no. 5, May 2018, doi: 10.3390/genes9050258.

[143] E. Zorarpacı and S. A. Özel, “A hybrid approach of differential evolution and artificial bee colony for feature selection,” *Expert Syst. Appl. Int. J.*, vol. 62, no. C, pp. 91–103, Nov. 2016, doi: 10.1016/j.eswa.2016.06.004.

[144] C. De Stefano, F. Fontanella, C. Marrocco, and A. Scotto Di Freca, “A GA-based feature selection approach with an application to handwritten character recognition,” *Pattern Recognit. Lett.*, vol. 35, pp. 130–141, Jan. 2014, doi: 10.1016/j.patrec.2013.01.026.

[145] M. H. Aghdam, N. Ghasem-Aghaee, and M. E. Basiri, “Text feature selection using ant colony optimization,” *Expert Syst. Appl.*, vol. 36, no. 3, Part 2, pp. 6843–6853, Apr. 2009, doi: 10.1016/j.eswa.2008.08.022.

[146] J. Too, A. R. Abdullah, N. Mohd Saad, and W. Tee, “EMG Feature Selection and Classification Using a Pbest-Guide Binary Particle Swarm Optimization,” *Computation*, vol. 7, no. 1, Art. no. 1, Mar. 2019, doi: 10.3390/computation7010012.

[147] A. A. M. El-Gaafary, Y. S. Mohamed, A. M. Hemeida, and A.-A. A. Mohamed, “Grey Wolf Optimization for Multi Input Multi Output System,” *Univers. J. Commun. Netw.*, vol. 3, no. 1, pp. 1–6, Feb. 2015, doi: 10.13189/ujcn.2015.030101.

[148] W. Long, J. Jiao, X. Liang, and M. Tang, “An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization,” *Eng. Appl. Artif. Intell.*, vol. 68, pp. 63–80, Feb. 2018, doi: 10.1016/j.engappai.2017.10.024.

[149] J. Luo, Q. Wang, and X. Xiao, “A modified artificial bee colony algorithm based on converge-onlookers approach for global optimization,” *Appl. Math. Comput.*, vol. 219, no. 20, pp. 10253–10262, Jun. 2013, doi: 10.1016/j.amc.2013.04.001.

[150] N. Mittal, U. Singh, and B. S. Sohi, “Modified Grey Wolf Optimizer for Global Engineering Optimization,” *Applied Computational Intelligence and Soft Computing*, 2016. https://www.hindawi.com/journals/acisc/2016/7950348/ (accessed Feb. 28, 2020).

[151] C. Alexander and M. Sadiku, *Fundamentals of Electric Circuits*. 2016.

[152] Q. Tu, X. Chen, and X. Liu, “Hierarchy Strengthened Grey Wolf Optimizer for Numerical Optimization and Feature Selection,” *IEEE Access*, vol. 7, pp. 78012–78028, 2019, doi: 10.1109/ACCESS.2019.2921793.

[153] S. Salesi and G. Cosma, “A novel extended binary cuckoo search algorithm for feature selection,” in *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*, Oct. 2017, pp. 6–12, doi: 10.1109/ICKEA.2017.8169893.

[154] S. Arlot and A. Celisse, “A survey of cross-validation procedures for model selection,” *Stat. Surv.*, vol. 4, pp. 40–79, 2010, doi: 10.1214/09-SS054.

[155] J. Han, M. Kamber, and J. Pei, *Data Mining concepts and techniques*. 2012.

[156] M. Bennasar, Y. Hicks, and R. Setchi, “Feature selection using Joint Mutual Information Maximisation,” *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8520–8532, Dec. 2015, doi: 10.1016/j.eswa.2015.07.007.

[157] M. Dash and H. Liu, “Feature selection for classification,” *Intell. Data Anal.*, vol. 1, no. 1, pp. 131–156, Jan. 1997, doi: 10.1016/S1088-467X(97)00008-5.

[158] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, no. null, pp. 1157–1182, Mar. 2003.

[159] H. Liu and Z. Zhao, “Manipulating Data and Dimension Reduction Methods: Feature Selection,” in *Computational Complexity: Theory, Techniques, and Applications*, R. A. Meyers, Ed. New York, NY: Springer, 2012, pp. 1790–1800.

[160] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, “Feature Selection: An Ever Evolving Frontier in Data Mining,” in *Feature Selection in Data Mining*, May 2010, pp. 4–13, Accessed: Nov. 03, 2020. [Online]. Available: http://proceedings.mlr.press/v10/liu10b.html.

[161] A. Zarshenas and K. Suzuki, “Binary coordinate ascent: An efficient optimization technique for feature subset selection for machine learning,” *Knowl.-Based Syst.*, vol. 110, pp. 191–201, Oct. 2016, doi: 10.1016/j.knosys.2016.07.026.

[162] E.-G. Talbi, *Metaheuristics: from design to implementation*, New. Wiley, 2009.

[163] H. Liu and H. Motoda, Eds., *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Springer US, 1998.

[164] “Two-Step Particle Swarm Optimization to Solve the Feature Selection Problem - IEEE Conference Publication.” https://ieeexplore.ieee.org/document/4389688 (accessed Nov. 03, 2020).

[165] Q. Al-Tashi, H. Rais, and S. Jadid, “Feature Selection Method Based on Grey Wolf Optimization for Coronary Artery Disease Classification,” in *Recent Trends in Data Science and Soft Computing*, Cham, 2019, pp. 257–266, doi: 10.1007/978-3-319-99007-1_25.

[166] Md. M. Kabir, Md. Shahjahan, and K. Murase, “A new local search based hybrid genetic algorithm for feature selection,” *Neurocomputing*, vol. 74, no. 17, pp. 2914–2928, Oct. 2011, doi: 10.1016/j.neucom.2011.03.034.

[167] “Binary Dragonfly Algorithm for Feature Selection - IEEE Conference Publication.” https://ieeexplore.ieee.org/document/8250257 (accessed Nov. 03, 2020).

[168] A. H. Gandomi and X.-S. Yang, “Chaotic bat algorithm,” *J. Comput. Sci.*, vol. 5, no. 2, pp. 224–232, Mar. 2014, doi: 10.1016/j.jocs.2013.10.002.

[169] N. Abd-Alsabour, “A Review on Evolutionary Feature Selection,” in *2014 European Modelling Symposium*, Oct. 2014, pp. 20–26, doi: 10.1109/EMS.2014.28.

[170] S. Mirjalili, “SCA: A Sine Cosine Algorithm for solving optimization problems,” *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016, doi: 10.1016/j.knosys.2015.12.022.

[171] H. Ma and D. Simon, “Blended biogeography-based optimization for constrained optimization,” *Eng. Appl. Artif. Intell.*, vol. 24, no. 3, pp. 517–525, Apr. 2011, doi: 10.1016/j.engappai.2010.08.005.

[172] R. Sindhu, R. Ngadiran, Y. M. Yacob, N. A. Hanin Zahri, M. Hariharan, and K. Polat, "A Hybrid SCA Inspired BBO for Feature Selection Problems," *Mathematical Problems in Engineering*, Apr. 02, 2019. https://www.hindawi.com/journals/mpe/2019/9517568/ (accessed Nov. 03, 2020).

[173] P. J. Gaidhane and M. J. Nigam, "A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems," *J. Comput. Sci.*, vol. 27, pp. 284–302, Jul. 2018, doi: 10.1016/j.jocs.2018.06.008.

[174] H. M. Zawbaa, E. Emary, C. Grosan, and V. Snasel, "Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach," *Swarm Evol. Comput.*, vol. 42, pp. 29–42, Oct. 2018, doi: 10.1016/j.swevo.2018.02.021.

[175] A. A. Alomoush, A. A. Alsewari, H. S. Alamri, K. Aloufi, and K. Z. Zamli, "Hybrid Harmony Search Algorithm With Grey Wolf Optimizer and Modified Opposition-Based Learning," *IEEE Access*, vol. 7, pp. 68764–68785, 2019, doi: 10.1109/ACCESS.2019.2917803.

[176] S. Arora, H. Singh, M. Sharma, S. Sharma, and P. Anand, "A New Hybrid Algorithm Based on Grey Wolf Optimization and Crow Search Algorithm for Unconstrained Function Optimization and Feature Selection," *IEEE Access*, vol. 7, pp. 26343–26361, 2019, doi: 10.1109/ACCESS.2019.2897325.

[177] B. N. Gohil and D. R. Patel, "A hybrid GWO-PSO Algorithm for Load Balancing in Cloud Computing Environment," in *2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)*, Aug. 2018, pp. 185–191, doi: 10.1109/ICGCIoT.2018.8753111.

[178] A. Zhu, C. Xu, Z. Li, J. Wu, and Z. Liu, "Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC," *J. Syst. Eng. Electron.*, vol. 26, no. 2, pp. 317–328, Apr. 2015, doi: 10.1109/JSEE.2015.00037.

[179] Z. Yue, S. Zhang, and W. Xiao, "A Novel Hybrid Algorithm Based on Grey Wolf Optimizer and Fireworks Algorithm," *Sensors*, vol. 20, no. 7, Apr. 2020, doi: 10.3390/s20072147.

[180] J. Cheng, L. Wang, and Y. Xiong, "Ensemble of cuckoo search variants," *Comput. Ind. Eng.*, vol. 135, pp. 299–313, Sep. 2019, doi: 10.1016/j.cie.2019.06.015.

[181] W. Long, S. Cai, J. Jiao, M. Xu, and T. Wu, "A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models," *Energy Convers. Manag.*, vol. 203, p. 112243, Jan. 2020, doi: 10.1016/j.enconman.2019.112243.

[182] Z. Zhang, S. Ding, and W. Jia, "A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 254–268, Oct. 2019, doi: 10.1016/j.engappai.2019.06.017.

[183] X. Zhang, X. tao Li, and M. hao Yin, “Hybrid cuckoo search algorithm with covariance matrix adaption evolution strategy for global optimisation problem,” *Int. J. Bio-Inspired Comput.*, vol. 13, no. 2, p. 102, 2019, doi: 10.1504/IJBIC.2019.098403.

[184] D. Gareth Huw, “The Life of Birds | Parenthood.” http://www.pbs.org/lifeofbirds/home/index.html (accessed Nov. 04, 2020).

[185] K. Khan and A. Sahai, “Neural-Based Cuckoo Search of Employee Health and Safety (HS),” *Int. J. Intell. Syst. Appl.*, vol. 5, no. 2, pp. 76–83, Jan. 2013, doi: 10.5815/ijisa.2013.02.09.

[186] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms: Second Edition*. Luniver Press, 2010.

[187] C. T. Brown, L. S. Liebovitch, and R. Glendon, “Lévy Flights in Dobe Ju/’hoansi Foraging Patterns,” *Hum. Ecol.*, vol. 35, no. 1, pp. 129–138, Feb. 2007, doi: 10.1007/s10745-006-9083-4.

[188] I. Pavlyukevich, “Lévy flights, non-local search and simulated annealing,” *J. Comput. Phys.*, vol. 226, no. 2, pp. 1830–1844, Oct. 2007, doi: 10.1016/j.jcp.2007.06.008.

[189] I. Pavlyukevich, “Cooling down Lévy flights,” *J. Phys. Math. Theor.*, vol. 40, no. 41, pp. 12299–12313, Sep. 2007, doi: 10.1088/1751-8113/40/41/003.

[190] A. M. Reynolds and M. A. Frye, “Free-Flight Odor Tracking in Drosophila Is Consistent with an Optimal Intermittent Scale-Free Search,” *PLoS ONE*, vol. 2, no. 4, Apr. 2007, doi: 10.1371/journal.pone.0000354.

[191] M. F. Shlesinger, G. M. Zaslavsky, and U. Frisch, Eds., *Lévy Flights and Related Topics in Physics: Proceedings of the International Workshop Held at Nice, France, 27–30 June 1994*. Berlin Heidelberg: Springer-Verlag, 1995.

[192] M. F. Shlesinger, “Search research,” *Nature*, vol. 443, no. 7109, Art. no. 7109, Sep. 2006, doi: 10.1038/443281a.

[193] X. Yang and Suash Deb, “Cuckoo Search via Lévy flights,” in *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, Dec. 2009, pp. 210–214, doi: 10.1109/NABIC.2009.5393690.

[194] X. S. Yang and S. Deb, “Engineering optimisation by cuckoo search,” *Int. J. Math. Model. Numer. Optim.*, vol. 1, no. 4, p. 330, 2010, doi: 10.1504/IJMMNO.2010.035430.

[195] L. Wang, Y. Yin, and Y. Zhong, “Cuckoo search with varied scaling factor,” *Front. Comput. Sci.*, vol. 9, no. 4, pp. 623–635, Aug. 2015, doi: 10.1007/s11704-015-4178-y.

[196] M. Reda, A. Y. Haikal, M. A. Elhosseini, and M. Badawy, “An Innovative Damped Cuckoo Search Algorithm With a Comparative Study Against Other Adaptive Variants,” *IEEE Access*, vol. 7, pp. 119272–119293, 2019, doi: 10.1109/ACCESS.2019.2936360.

[197] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[198] Q. Al-Tashi, S. J. A. Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, “Binary Optimization Using Hybrid Grey Wolf Optimization for Feature Selection,” *IEEE Access*, vol. 7, pp. 39496–39508, 2019, doi: 10.1109/ACCESS.2019.2906757.

[199] M. M. Mafarja and S. Mirjalili, “Hybrid Whale Optimization Algorithm with simulated annealing for feature selection,” *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017, doi: 10.1016/j.neucom.2017.04.053.

[200] M. Črepinšek, S.-H. Liu, and M. Mernik, “Exploration and exploitation in evolutionary algorithms: A survey,” *ACM Comput. Surv.*, vol. 45, no. 3, p. 35:1–35:33, Jul. 2013, doi: 10.1145/2480741.2480752.

[201] J. Wei and Y. Yu, “An Effective Hybrid Cuckoo Search Algorithm for Unknown Parameters and Time Delays Estimation of Chaotic Systems,” *IEEE Access*, vol. 6, pp. 6560–6571, 2018, doi: 10.1109/ACCESS.2017.2738006.

[202] J. Huang, X. Li, and L. Gao, “A new hybrid algorithm for unconstrained optimisation problems,” *Int. J. Comput. Appl. Technol.*, vol. 46, no. 3, p. 187, 2013, doi: 10.1504/IJCAT.2013.052808.

[203] R. Chi, Y. Su, Z. Qu, and X. Chi, “A Hybridization of Cuckoo Search and Differential Evolution for the Logistics Distribution Center Location Problem,” *Mathematical Problems in Engineering*, Feb. 06, 2019. https://www.hindawi.com/journals/mpe/2019/7051248/ (accessed Nov. 04, 2020).

[204] M. K. Naik and R. Panda, “A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition,” *Appl. Soft Comput.*, vol. 38, pp. 661–675, Jan. 2016, doi: 10.1016/j.asoc.2015.10.039.

[205] S. Mirjalili and S. Z. M. Hashim, “BMOA: Binary Magnetic Optimization Algorithm,” *Int. J. Mach. Learn. Comput.*, pp. 204–208, 2012, doi: 10.7763/IJMLC.2012.V2.114.

[206] “UCI Machine Learning Repository.” http://archive.ics.uci.edu/ml/index.php (accessed Nov. 04, 2020).

[207] D. Segera, M. Mbuthia, and A. Nyete, “Particle Swarm Optimized Hybrid Kernel-Based Multiclass Support Vector Machine for Microarray Cancer Data Analysis,” *BioMed Research International*, Dec. 16, 2019. https://www.hindawi.com/journals/bmri/2019/4085725/ (accessed Nov. 04, 2020).

[208] Adiwijaya, U. N. Wisesty, E. Lisnawati, A. Aditsania, and D. S. Kusumo, “Dimensionality Reduction using Principal Component Analysis for Cancer Detection based on Microarray Data

Classification," *J. Comput. Sci.*, vol. 14, no. 11, Art. no. 11, Nov. 2018, doi: 10.3844/jcssp.2018.1521.1530.

[209] S. Sun, Q. Peng, and A. Shakoor, "A Kernel-Based Multivariate Feature Selection Method for Microarray Data Classification," *PLOS ONE*, vol. 9, no. 7, p. e102541, Jul. 2014, doi: 10.1371/journal.pone.0102541.

[210] A. Osareh and B. Shadgar, "Microarray data analysis for cancer classification," in *2010 5th International Symposium on Health Informatics and Bioinformatics*, Apr. 2010, pp. 125–132, doi: 10.1109/HIBIT.2010.5478893.

[211] G. C. Cawley and N. L. C. Talbot, "Gene selection in cancer classification using sparse logistic regression with Bayesian regularization," *Bioinformatics*, vol. 22, no. 19, pp. 2348–2355, Oct. 2006, doi: 10.1093/bioinformatics/btl386.

[212] M. Mollaee and M. H. Moattar, "A novel feature extraction approach based on ensemble feature selection and modified discriminant independent component analysis for microarray data classification," *Biocybern. Biomed. Eng.*, vol. 36, no. 3, pp. 521–529, Jan. 2016, doi: 10.1016/j.bbe.2016.05.001.

[213] E.-J. Yeoh *et al.*, "Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling," *Cancer Cell*, vol. 1, no. 2, pp. 133–143, Mar. 2002, doi: 10.1016/s1535-6108(02)00032-6.

[214] U. Alon *et al.*, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Natl. Acad. Sci.*, vol. 96, no. 12, pp. 6745–6750, Jun. 1999, doi: 10.1073/pnas.96.12.6745.

[215] J. Khan *et al.*, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nat. Med.*, vol. 7, no. 6, pp. 673–679, Jun. 2001, doi: 10.1038/89044.

[216] S. A. Armstrong *et al.*, "MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia," *Nat. Genet.*, vol. 30, no. 1, Art. no. 1, Jan. 2002, doi: 10.1038/ng765.

[217] C. D. A. Vanitha, D. Devaraj, and M. Venkatesulu, "Gene Expression Data Classification Using Support Vector Machine and Mutual Information-based Gene Selection," *Procedia Comput. Sci.*, vol. 47, pp. 13–21, Jan. 2015, doi: 10.1016/j.procs.2015.03.178.

[218] A. Nurfalah, Adiwijaya, and A. A. Suryani, "Cancer Detection Based On Microarray Data Classification Using Pca And Modified Back Propagation," *Far East J. Electron. Commun.*, vol. 16, no. 2, pp. 269–281, May 2016, doi: 10.17654/EC016020269.

[219] A. Bhattacharjee *et al.*, “Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses,” *Proc. Natl. Acad. Sci.*, vol. 98, no. 24, pp. 13790–13795, Nov. 2001, doi: 10.1073/pnas.191502998.

[220] M. Xi, J. Sun, L. Liu, F. Fan, and X. Wu, “Cancer Feature Selection and Classification Using a Binary Quantum-Behaved Particle Swarm Optimization and Support Vector Machine,” *Computational and Mathematical Methods in Medicine*, Aug. 24, 2016. https://www.hindawi.com/journals/cmmm/2016/3572705/ (accessed Nov. 04, 2020).

[221] H. Yu, S. Hong, X. Yang, J. Ni, Y. Dan, and B. Qin, “Recognition of Multiple Imbalanced Cancer Types Based on DNA Microarray Data Using Ensemble Classifiers,” *BioMed Research International*, Aug. 26, 2013. https://www.hindawi.com/journals/bmri/2013/239628/ (accessed Nov. 04, 2020).

[222] R. F. Wahyu Pratama, S. W. Purnami, and S. P. Rahayu, “Boosting Support Vector Machines for Imbalanced Microarray Data,” *Procedia Comput. Sci.*, vol. 144, pp. 174–183, Jan. 2018, doi: 10.1016/j.procs.2018.10.517.

[223] A. Bir-Jmel, S. M. Douiri, and S. Elbernoussi, “Gene Selection via a New Hybrid Ant Colony Optimization Algorithm for Cancer Classification in High-Dimensional Data,” *Computational and Mathematical Methods in Medicine*, Oct. 13, 2019. https://www.hindawi.com/journals/cmmm/2019/7828590/ (accessed Nov. 04, 2020).

[224] M. N. A. Wahab, S. Nefti-Meziani, and A. Atyabi, “A Comprehensive Review of Swarm Optimization Algorithms,” *PLOS ONE*, vol. 10, no. 5, p. e0122827, May 2015, doi: 10.1371/journal.pone.0122827.

[225] A. Tharwat, “Classification assessment methods,” *Appl. Comput. Inform.*, vol. ahead-of-print, no. ahead-of-print, Jan. 2020, doi: 10.1016/j.aci.2018.08.003.

[226] S. Mocellin, Ed., *Microarray Technology and Cancer Gene Profiling*. New York: Springer-Verlag, 2007.

[227] L. Zou and Z. Wang, “Microarray Gene Expression Cancer Diagnosis Using Multiclass Support Vector Machines,” in *2007 1st International Conference on Bioinformatics and Biomedical Engineering*, Jul. 2007, pp. 260–263, doi: 10.1109/ICBBE.2007.70.

[228] V. Bhuvaneswari, “Classification of Microarray Gene Expression Data by Gene Combinations using Fuzzy Logic MGC-FL,” *Int. J. Comput. Sci. Eng. Appl.*, vol. 2, no. 4, pp. 79–98, Aug. 2012, doi: 10.5121/ijcsea.2012.2409.

[229] G. B. Singh, *Fundamentals of Bioinformatics and Computational Biology: Methods and Exercises in MATLAB*. Springer International Publishing, 2015.

[230] V. Bevilacqua, G. Mastronardi, F. Menolascina, A. Paradiso, and S. Tommasi, "Genetic Algorithms and Artificial Neural Networks in Microarray Data Analysis: a Distributed Approach," *Eng. Lett.*, vol. 13, Nov. 2006.

[231] W. Chen, H. Lu, M. Wang, and C. Fang, "Gene Expression Data Classification Using Artificial Neural Network Ensembles Based on Samples Filtering," in *2009 International Conference on Artificial Intelligence and Computational Intelligence*, Nov. 2009, vol. 1, pp. 626–628, doi: 10.1109/AICI.2009.441.

[232] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, Oct. 2000, doi: 10.1093/bioinformatics/16.10.906.

[233] S. Niazmardi, A. Safari, and S. Homayouni, "A Novel Multiple Kernel Learning Framework for Multiple Feature Classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 8, pp. 3734–3743, Aug. 2017, doi: 10.1109/JSTARS.2017.2697417.

[234] H. Bhavsar and A. Ganatra, "Radial Basis Polynomial Kernel (RBPK): A Generalized Kernel for Support Vector Machine," *Int. J. Comput. Sci. Inf. Secur. IJCSIS*, Apr. 2016.

[235] H. Song, Z. Ding, C. Guo, Z. Li, and H. Xia, "Research on Combination Kernel Function of Support Vector Machine," in *2008 International Conference on Computer Science and Software Engineering*, Dec. 2008, vol. 1, pp. 838–841, doi: 10.1109/CSSE.2008.1231.

[236] W. An-na, Z. Yue, H. Yun-tao, and L. I. Yun-lu, "A novel construction of SVM compound kernel function," in *2010 International Conference on Logistics Systems and Intelligent Management (ICLSIM)*, Jan. 2010, vol. 3, pp. 1462–1465, doi: 10.1109/ICLSIM.2010.5461210.

[237] A. Ben-Hur, C. S. Ong, S. Sonnenburg, B. Schölkopf, and G. Rätsch, "Support Vector Machines and Kernels for Computational Biology," *PLOS Comput. Biol.*, vol. 4, no. 10, p. e1000173, Oct. 2008, doi: 10.1371/journal.pcbi.1000173.

[238] D. Kaya, "Optimization of SVM Parameters with Hybrid CS-PSO Algorithms for Parkinson's Disease in LabVIEW Environment," *Parkinson&#x2019;s Disease*, May 02, 2019. https://www.hindawi.com/journals/pd/2019/2513053/ (accessed Nov. 04, 2020).

[239] R. Aziz, C. K. Verma, and N. Srivastava, "Dimension reduction methods for microarray data: a review," *AIMS Bioeng.*, vol. 4, no. 1, p. 179, 2017, doi: 10.3934/bioeng.2017.1.179.

[240] M. Lenz, F.-J. Müller, M. Zenke, and A. Schuppert, "Principal components analysis and the reported low intrinsic dimensionality of gene expression microarray data," *Sci. Rep.*, vol. 6, no. 1, Art. no. 1, Jun. 2016, doi: 10.1038/srep25696.

[241] E. Ahmed, N. El-Gayar, and I. A. El-Azab, "Support Vector Machine ensembles using features distribution among subsets for enhancing microarray data classification," in *2010 10th International Conference on Intelligent Systems Design and Applications*, Nov. 2010, pp. 1242–1246, doi: 10.1109/ISDA.2010.5687078.

[242] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, Illustrated edition. Cambridge, MA: The MIT Press, 2012.

[243] R. Herbrich, *Learning kernel classifiers: theory and algorithms*. MIT Press, 2002.

[244] "(16) On what basis do we select the swarm size for any application in Particle Swarm Optimization? Does it vary with the type of PSO used?," *ResearchGate*. https://www.researchgate.net/post/On_what_basis_do_we_select_the_swarm_size_for_any_application_in_Particle_Swarm_Optimization_Does_it_vary_with_the_type_of_PSO_used (accessed Nov. 04, 2020).

[245] M. Alam, *Particle Swarm Optimization: Algorithm and its Codes in MATLAB*. 2016.

[246] M. A. Meziane, Y. Mouloudi, B. Bouchiba, and A. Laoufi, "Impact of inertia weight strategies in particle swarm optimization for solving economic dispatch problem," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 13, no. 1, Art. no. 1, Jan. 2019, doi: 10.11591/ijeecs.v13.i1.pp377-383.

[247] E. Mezura-Montes and J. I. Flores-Mendoza, "Improved Particle Swarm Optimization in Constrained Numerical Search Spaces," in *Nature-Inspired Algorithms for Optimisation*, R. Chiong, Ed. Berlin, Heidelberg: Springer, 2009, pp. 299–332.

[248] J. Zheng, Q. Wu, and W. Song, "An Improved Particle Swarm Algorithm for Solving Nonlinear Constrained Optimization Problems," in *Third International Conference on Natural Computation (ICNC 2007)*, Aug. 2007, vol. 4, pp. 112–117, doi: 10.1109/ICNC.2007.221.

[249] H. Dong and G. Jian, "Parameter Selection of a Support Vector Machine, Based on a Chaotic Particle Swarm Optimization Algorithm," *Cybern. Inf. Technol.*, vol. 15, no. 3, pp. 140–149, Sep. 2015, doi: 10.1515/cait-2015-0047.

[250] S. M. Elsayed, R. A. Sarker, and E. Mezura-Montes, "Particle Swarm Optimizer for constrained optimization," in *2013 IEEE Congress on Evolutionary Computation*, Jun. 2013, pp. 2703–2711, doi: 10.1109/CEC.2013.6557896.

[251] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27:1–27:27, May 2011, doi: 10.1145/1961189.1961199.

[252] "A Practical Guide to Support Vector Classification | BibSonomy." https://www.bibsonomy.org/bibtex/2c04ef97dc3c3de168e684c3e4abe061b/zeno (accessed Nov. 04, 2020).

LIST OF PUBLICATIONS

1) Davies Segera[1] , Mwangi Mbuthia[2] , and Abraham Nyete[3] , “Particle Swarm Optimized Hybrid Kernel-Based Multiclass Support Vector Machine for Microarray Cancer Data Analysis,” Biomed Research International Volume 2019, Article ID 4085725, 11 pages.

2) Davies Segera[1] , Mwangi Mbuthia[2] , and Abraham Nyete[3] , “Particle Swarm Optimized Hybrid Kernel-Based Multiclass Support Vector Machine for Microarray Cancer Data Analysis,” In: Prime Archives in Medicine. Hyderabad,India: Vide Leaf 2020.

3) Davies Segera[1] , Mwangi Mbuthia[2] , and Abraham Nyete[3] , “An Innovative Excited-ACS-IDGWO Algorithm for Optimal Biomedical Data Feature Selection,” Biomed Research International 2020, Article ID 8506365, 17 pages.

4) Davies Segera[1] , Mwangi Mbuthia[2] , and Abraham Nyete[3] , “An Excited Binary Grey Wolf Optimizer for Feature Selection in Highly Dimensional Datasets,” Proc. Of the 17th ,Int. Conf. On Informatics in Control, Automation and Robotics, ICINCO 2020, Online Streaming 7-9Th July, 2020, Oleg Gusikhin, Kurosh Madani, Janan Zaytoon, Eds. Paris:SCITEPRESS,2020, 125-133.

Hindawi

BioMed Research International

Research Article

An Innovative Excited-ACS-IDGWO Algorithm for Optimal Biomedical Data Feature Selection

Davies Segera,[1] Mwangi Mbuthia,[1] Abraham Nyete,[1]

[1] *Department of Electrical and Information Engineering, University of Nairobi, Nairobi 30197, Kenya.*

Correspondence should be addressed to Davies Segera; davies.segera@uonbi.ac.ke

Abstract. Finding an optimal set of discriminative features is still a crucial but challenging task in biomedical science. The complexity of the task is intensified when any of the two scenarios arise: a highly dimensioned dataset; a small sample-sized dataset. The first scenario poses a big challenge to existing machine learning approaches since the search space for identifying the most relevant feature subset is so diverse to be explored quickly while utilizing minimal computational resources. On the other hand, the second aspect pose a challenge of too few samples to learn from. Though many hybrid metaheuristic approaches (i.e. combining multiple search algorithms) have been proposed in the literature to address these challenges with very attractive performance compared to their counterpart standard standalone metaheuristics, more superior hybrid approaches can be achieved if the individual metaheuristics within the proposed hybrid algorithms are improved prior to the hybridization. Motivated by this, we propose a new hybrid Excited (E)-Adaptive Cuckoo Search (ACS)-Intensification Dedicated Grey Wolf Optimizer (IDGWO) i.e. EACSIDGWO. EACSIDGWO is an algorithm where the step size of ACS and the non-linear control strategy of parameter $\vec{a}$ of the IDGWO are innovatively made adaptive via the concept of the complete voltage and current responses of a direct current (DC) excited resistor-capacitor (RC) circuit. Since the population has a higher diversity at early stages of the proposed EACSIDGWO algorithm, both the ACS and IDGWO are jointly involved in local exploitation. On the other hand, to enhance mature convergence at latter stages of the proposed algorithm, the role of ACS is switched to global exploration while the IDGWO is still left conducting the local exploitation. To prove that the proposed algorithm is superior in providing a good learning from fewer instances and an optimal feature selection from information-rich biomedical data, all these while maintaining a high classification accuracy of the data, the EACSIDGWO is employed to solve the feature selection problem. The

EACSIDGWO as a feature selector is tested on six standard biomedical datasets from the university of California at Irvine (UCI) repository. The experimental results are compared with the state-of-the-art feature selection techniques, including binary anti-colony optimization (BACO), binary genetic algorithm (BGA), binary particle swarm optimization (BPSO) and extended binary cuckoo search algorithm (EBCSA).These results reveal that the EACSIDGWO has comprehensive superiority in tackling the feature selection problem, which proves the capability of the proposed algorithm in solving real-world complex problems. Furthermore, the superiority of the proposed algorithm is proved via various numerical techniques like ranking methods and statistical analysis.

Introduction

Currently, there is a growing research interest in developing and deploying population-based metaheuristics to tackle combinatorial optimization challenges. This is because they are simple, flexible with an inexpensive computational cost and are gradient-free [1].

Many researchers have applied these optimization algorithms in various research domains because of their ability to achieve best solutions.

The optimization challenge grows bigger when tackling highly dimensioned datasets. This is because these datasets have a vast feature space with many classes. Due to the presence of redundant and non-informative attributes within these datasets, the process of effective machine learning greatly hindered. Thus, the construction of efficient classifiers with high predictive power largely depends on selection of informative features [2].

Feature selection (FS) is one of main steps in data preprocessing that aims at selecting a subset of attributes out of the whole dataset resulting into removal of noisy non-informative and redundant features. This in turn increases the accuracy of a considered classifier or clustering model [3].

FS algorithms can be broadly categorized into two classes: filter and wrapper techniques [4-5]. Filters include techniques independent of classifiers and work directly on presented data. Moreover, these methods in many situations determine the correlations between features. On the contrary, wrapper approaches engage classifiers and mainly determine interactions between dataset features. From literature, wrapper approaches have proved to be superior compared to filters for classification algorithms [6-7].

To utilize wrapper-based techniques, three key factors need to be outlined; considered classifiers (i.e. k-nearest neighbor (KNN), support vector machine (SVM)), evaluation criteria for the identified feature subset and a search technique utilized in determining a subset of optimal features [8].

Many researchers have pointed out that determining an optimal subset of attributes is not only challenging but computationally expensive as well. Though, in the recent past metaheuristics have proved to be reliable and efficient tools in tackling many optimization tasks (e.g., engineering designs problems, machine learning, feature selection and data mining), they are not efficient in solving problems with high computational complexity [9 - 12].

In the recent past, a number of metaheuristic search algorithms have been utilized for FS using highly dimensioned datasets. Some of these metaheuristics are grey wolf optimization (GWO) [13-14], Genetic Algorithm (GA) [15], particle swarm optimization

(PSO) [11], anti-colony optimization (ACO) [16], differential evolution algorithm (DEA) [17], cuckoo search algorithm (CSA) [18], dragonfly algorithm (DA) [19]. Though, many of these algorithms have already made an important contribution in the field of feature selection, in many cases they offer acceptable solutions without a guarantee of determining optimal solutions since they do not explore the entire search space [11].

Some of the new modifications that have been proposed to improve the performance of these metaheuristics include chaotic maps [20], evolutionary methods [21], sine cosine algorithms [22], biogeography based optimization and local searches [23].

While designing or utilizing a metaheuristic, it should be noted that diversification (exploring the search space) and intensification (exploiting optimal solutions obtained so far) are two contradicting principles, that must be balanced efficiently in order to achieve an improved performance of the metaheuristic [9].

In this regard, one promising alternative is developing a memetic algorithm whereby an integration of (at least) two algorithms is done with the aim of enhancing the overall performance.

Motivated by this, a good number of hybrid algorithms have proposed in the recent past to solve a variety of optimizations and feature selection problems [24]. However, to enhance diversification and intensification of these hybrid algorithms, exploration and fine-tuning within their basic constituent algorithms is needed prior to hybridization [25].

This emphasizes too, that there are a number of techniques lying within these memetic algorithms that are yet to be investigated.

Foremost, the technique of combining one or more nature inspired algorithms (NIAs) need to be determined. Secondly, the criterion of determining how many NIAs need to be combined within the search space has to be accomplished. Thirdly, the method of determining the application area upon which the proposed memetic algorithm has to be done. Finally, the criterion of applying the memetic algorithm in a specific domain has to be accomplished [25].

Inspired by the aforementioned, this paper propose a new hybrid algorithm called Excited (E) - Adaptive Cuckoo Search (ACS)-Intensification Dedicated Grey Wolf Optimizer (IDGWO) i.e. EACSIDGWO algorithm to solve the feature selection problem in biomedical science. In the proposed algorithm, the concept of the complete voltage and current responses of a direct current (DC) excited resistor capacitor (RC) circuit are innovatively utilized to make the step size of ACS and the non-linear control strategy of parameter $\vec{a}$ of the IDGWO adaptive. Since the population has a higher diversity during early stages of the proposed algorithm, both the ACS and IDGWO are jointly utilized to attain accelerated convergence. However, to enhance mature convergence while striking an effective balance between exploitation and exploration in latter stages, the role of ACS is switched to global exploration while the IDGWO is still left conducting the local exploitation.

The remainder of this paper is organized as follows: Sections 2 discussed the existing literature within the same research domain. Section 3 presents the background information of the CS and the GWO respectively where their inspirations and mathematical models are given emphasis. The continuous version of the proposed EACSIDGWO algorithm is presented in section 4 while the details of its binary version are given section 5. The experimental methodology considered in this paper is presented in section 6 while the results on feature selection are discussed in section 7. Finally, conclusions and the suggested future works are given in section 8.

Literature Reviews

2.1. Review of Hybridization of GWO with other Search Algorithms. Combining two or more metaheuristics to attain better solutions is currently a new insight in the area of optimization. In the literature, many researchers have utilized GWO in the field of hybrid metaheuristics. For instance, in [26] a hybrid of GWO and Artificial Bee Colony (ABC) is proposed to improve performance of a complex system. In [27], GWO is hybridized with Ant Lion Optimizer (ALO) for wrapper feature selection. Alomoush [28] proposed a hybrid of GWO and Harmony Search (HS). In this memetic, GWO updates the bandwidth and pitch adjustment rate in HS, which in return improves the global optimization abilities of the hybrid algorithm. In [29], Sankalap Arora combined GWO with Crow Search Algorithm (CSA). The performance of the derived memetic as a feature selector is evaluated using 21 datasets. The obtained results reveal that the combined algorithm is superior in solving complex optimization algorithms. In [30], a novel combination between GWO and PSO is utilized as load balancing technique in the cloud-computing arena. The conclusions point out that the hybrid algorithm improved both the convergence speed and the simplicity in comparison with other algorithms. Zhu [31], hybridized GWO with Differential Evolution (DE). The hybrid algorithm was tested on 23 different functions and a non-deterministic polynomial hard problem. The obtained results indicate that this combination achieved superior exploration. In [32], a new memetic combining the exploration ability of Fireworks algorithm (FWA) with the exploitation ability of GWO is proposed. Utilizing 16 benchmark functions with varied dimensions and complexities, the experimental results indicate that the hybrid algorithm attained attractive global search abilities and convergence speeds.

2.2. Review of Hybridization of CS with other Search Algorithms. Utilizing the concept of rand and best agents within a population, Cheng [33] developed an ensemble cuckoo search variant combining three different CS approaches that coexist within the entire search domain. These CS variants actively compete to derive superior generations for numerical optimization. To maintain population diversity, he introduced an external archive. The statistical results obtained reveal that the ensemble CS attained attractive converge speeds as well as robustness. In [34], GWO is hybridized with CS i.e. GWOCS for the extraction of parameters for different PV cell models situated in different conditions. Zhang [35] developed an ensemble CS algorithm that foremost divides a population into two smaller groups and then utilizes CS and differential evolution (DE) on the derived subgroups independently. The subgroups are free to share useful information by division. Further, the CS and DE algorithms can freely utilize each other's merits to complement their weaknesses. This approach proved to balance the quality of solutions and the computation consumption. In [35], CS is hybridized with a covariance matrix adaptation evolution approach i.e. CMA-CS to improve the performance of CS in different optimization problems.

Despite the advantages portrayed by the aforementioned hybrid GWO and CS metaheuristics for optimization and feature selection, superior hybrid approaches can be achieved if the single GWO and CS algorithms are improved prior to hybridization. Furthermore, the No-Free-Lunch (NFL) theorem has logically proved that there has been, is and will be no single metaheuristic capable of solving all optimization and feature selection problems [34]. While a given metaheuristic can, show an attractive performance on specific datasets, its performance might degrade when applied to similar or different types of datasets [35]. Thus, there is still a dire need to improve existing algorithms or develop new ones to solve function optimization problems as well as feature selection problems efficiently.

Standard Cuckoo Search (CS)

2.1. Inspiration of CS

The behavior of cuckoo birds. To date more than a thousand different species of birds are in existence in nature [36]. For most of these species, the female birds lay eggs in nests they have built themselves [37]. However, there exists some types of birds that do not build nests of their own, but instead lay their eggs in other different species' nests, leaving the responsibility of taking care of their eggs to the host birds. The cuckoos are the most famous of these brood parasites [38].

There are three types of brood parasites: intra-specific brood parasites, cooperative breeding and nest take-over [39].The cuckoo strategy is full of amazing traits, foremost it replaces one host egg with its own to increase the chances of its egg being hatched by the host bird. Next, it tries to mimic the pattern and color (s) of this host eggs with the aim of reducing the chances of its egg being noticed and discarded by the host bird. It is also important to point out that, the timing of laying its egg is amazing since it cleverly selects a nest where a host bird has just laid eggs, implying that the cuckoo's egg will hatch prior to the host eggs. The first action taken by the hatched cuckoo is evicting the host eggs that are yet to hatch out of the nest by blind propelling in order to increase its chances of being fed well by the host bird [38]. In addition, this young cuckoo mimics the call of host chicks thus enhancing more access to the food provided by the host bird [40].

However, if this host bird is able to identify the cuckoo's egg, it can either discard it from the nest or quit this nest to build a completely new nest in a different location.

Le'vy flights. From literature, many researchers have shown that the behavior of many flying animals, birds and insects can be demonstrated by a Le'vy flight [41, 42, 43, 44]. Le'vy flights are evident when some birds, insects and animals follow a long path with sudden turns in combination with random-short moves [44].These Le'vy flights have been successfully applied in optimization [42,44,45,46].A Le'vy flight is a random walk characterized with step-lengths whose distribution is according to a heavy-tailed probability distribution.

2.2. Cuckoo search (CS) algorithm

CS is a metaheuristic swarm-based global optimization based on cuckoos that was proposed by Yang and Deb in 2009.The CS combines the obligate brood parasitic nature of cuckoos with the le'vy flight existing in fruit flies and some birds [39].There are three basic idealized rules for the CS namely:

i) A female cuckoo lays one egg at a time, and puts it in a randomly chosen nest.

ii) The best nests with high quality eggs (highest fitness/solutions) will carry over to the next generations

iii) The number of available host nests is kept fixed, and the host bird can discover the egg laid by the female cuckoo (alien egg) with a probability $P_a \in [0,1]$. Depending on the value of P_a, the host bird can either throw away the alien egg or abandon the nest. An assumption that only a fraction of P_a nests is replaced by new ones.

Based on the above rules, an illustration of the CS is shown in Algorithm 1

Algorithm 1: Pseudo-code for the standard CS

```
Begin:
 Initialize $P_a = 0.25$
 Define objective function $f(x), x = (x_1, x_2, \dots, x_d)$, where $d$ is the number of dimensions
Generate initial population of $n$ host bird nests, $X_i (i = 1,2, \dots, n)$
while $g \leq g_{max}$ or any other stopping criteria
Generate a new cuckoo (solution) randomly via Le'vy flight according to Equation (1)
Evaluate the fitness of the new cuckoo, $F_i$
Randomly choose a nest from among the host nests $n$ (For example $j$)
if $F_i > F_j$ then
    Replace nest j by the new cuckoo $i$
end
Abandon a fraction of $P_a$ worst nests and generate new ones according to Equation (6)
Keep best solutions(or those nests with quality solutions)
Rank these solutions, then keep the current best
end while
Report the final best
end
```

2.3. Mathematical modelling of the standard CS

Considering Algorithm 1, the standard CS has three major steps [47, 48, 49]:

1) Exploitation (Intensification) by the use of Le'vy flight random walk (LFRW)
2) Exploration(Diversification) using biased-selective random walk(BSRW)
3) Elitist scheme via greedy selection

Intensification using Le'vy flight random walk (LFRW). In this phase, new solutions are generated around the current best solution, which in return enhances the speed of the local search. This phase is achieved via the LFRW that is generally presented in (1) where the step size is derived from the Le'vy distribution.

$$X_{i,gen+1} = X_{i,gen} + \alpha \oplus \text{Le'vy}(\lambda) \quad (1)$$

Where $X_{i,gen}$ is the i^{th} nest in the gen^{th} generation and $X_{i,gen+1}$ is a new nest generated by the Le'vy flight. $\oplus$ implies entry-wise multiplications and α is the step size where $\alpha > 0$ and is formulated in (2).The formula in equation (1) ensures that a new solution will be close to the current best-solution.

$$\alpha = \alpha_0 \times (X_{i,gen} - X_{best}) \qquad (2)$$

Where X_{best} is the current solution and α_0 is a scaler that is set to 0.01 in the standard CSA [39, 50]. Le'vy(λ) is a random number derived from the Le'vy distribution and is formulated in (3)

$$\text{Le'vy}(\lambda) \sim \frac{\partial \times \varepsilon}{|\varphi|^{\frac{1}{\lambda}}} \qquad (3)$$

Where λ is a constant whose value is 1.5 as suggested by Yang is in the standard CS [39]. ε and φ are random numbers derived from a normal distribution whose mean and standard deviation is 1. ∂ is a parameter computed in (4)

$$\partial = \left(\frac{\lceil(1+\lambda) \times \sin\left(\frac{\pi \times \lambda}{2}\right)}{\lceil\left(\frac{1+\lambda}{2} \times \lambda \times 2^{\frac{\lambda-1}{2}}\right)} \right)^{\frac{1}{\lambda}} \qquad (4)$$

Where ⌈ is a gamma function. The final form of Le'vy flight random walk (LFRW) is a combination of equations (1) to (4) as presented in equation (5).

$$X_{i,gen+1} = X_{i,gen} + \alpha_0 \frac{\partial \times \varepsilon}{|\varphi|^{\frac{1}{\lambda}}} (X_{i,gen} - X_{best}) \qquad (5)$$

Diversification by the use of biased-selective random walk (BSRW). In this phase, new solutions are randomly generated in locations far from the current best solution. An approach that ensures that the CSA is not trapped in the local optimum thus enhancing suitable diversity and exploration of the entire search space [49]. This phase of the CSA is achieved by utilizing the BSRW which is efficient in exploring the entire search space especially when it is large since the step-size in the Le'vy flight is much longer in the long run [47,49].

To find new solutions that are far from the current best solution, foremost, a trial solution is obtained by using a mutation of the current best solution and a differential step size from two solutions selected randomly. Then a new solution is derived from a crossover operator between the current best solution and the two trial solutions [49].The formulation of the BSRW is given in (6) [48].

$$X_{i,gen+1} = \begin{cases} X_{i,gen} + s\times\left(x_{a,j,gen} - x_{b,j,gen}\right) \mathit{with}\ l \\ X_{i,gen}\ \mathit{with\ the\ remaining}\ P_a \end{cases} \quad (6)$$

Where a and b are two random indexes, s is a random number in the range [0, 1] and P_a is the probability discovery whose best value is 0.25 [39, 49].

Elitist scheme via greedy selection.

After each random walk process, the cuckoo search algorithm utilizes the greedy strategy to select solutions with better fitness values that will be passed to the next generation. This facilitates maintenance of good solutions [49].

Grey Wolf Optimization (GWO) Algorithm

GWO is recent nature-inspired metaheuristic algorithm that was proposed by Mirjalili et al in 2014 [51-53].The GWO imitates both the hunting and leadership traits of the grey wolves. The grey wolves belong to the Canidae family and follow a social hierarchy that is very strict. In most cases, a pack of between 5 and 12 wolves is involved in hunting. To efficiently simulate the leadership hierarchy of the conventional GWO algorithm, four levels are considered: alpha (α), beta (β), delta (δ) and omega (ω). Alpha, which is either a male or female is at the topmost of the hierarchy and is regarded as the leader of the pack. This leader makes all suitable decisions for the pack which are not limited to discipline and order, hunting, sleeping location and waking-up time for the entire pack. Beta are known to assist the Alpha in decision-making and their main task is the feedback suggestions. Delta behave like scouts, caretakers, sentinels, hunters and elders. They control and guide the omega wolves by obeying both the beta and alpha wolves. The omega wolves are least in the hierarchy and must obey all the other wolves [51-53].

The GWO algorithm is modelled mathematically in four stages that are described as follows:

3.1. Leadership hierarchy

The mathematical model of the GWO is anchored on the social hierarchy of the grey wolves. The alpha (α) is considered the best solution in the population while beta (β) and delta (δ) are termed as the second, third best solutions respectively. Lastly, the omega (ω) are assumed the rest of the solutions in the population [51-53].

3.2. Encircling the prey

Equation (6) and Equation (7) represent the mathematical model for the wolves' encircling trait [51].

$$\vec{D} = \left|\vec{C}\,.\vec{X}_p(t) - \vec{X}(t)\right| \tag{6}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}.\vec{D} \tag{7}$$

Where $\vec{D}$ is the distance between the prey and a given wolf. $\vec{X}$ is the wolf's position vector and $\vec{X}_p$ depicts the prey's position vector at iteration t. $\vec{A}$ and $\vec{C}$ are random vectors computed as shown in Equation (8) and Equation (9) [51].

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \tag{8}$$

$$\vec{C} = 2.\vec{r}_2 \tag{9}$$

Where $\vec{r}_1$ and $\vec{r}_1$ are randomly generated vectors in the range [0, 1] and $\vec{a}$ is a set vector that is linearly decreases from 2 to 0 over the iterations.

3.3. Hunting the prey

In the hunting stage, the alpha is considered the best applicant for the solution while its two assistants (beta and delta) are expected to know the possible location of the prey. Thus, the best three solutions that have been achieved until a given iteration are preserved and are used to compel the remaining wolves in the pack (i.e. omega) to update their positions in the search space consistent with the optimal location.

The mechanism utilized in updating the wolves' positions is given in Equation 10.

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{10}$$

Where$\vec{X}_1, \vec{X}_2$ and $\vec{X}_3$ are defined and computed using Equation 11, Equation 12 and Equation 13 respectively.

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1.\vec{D}_\alpha \tag{11}$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2.\vec{D}_\beta \tag{12}$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3.\vec{D}_\delta \tag{13}$$

Where$\vec{X}_\alpha$, $\vec{X}_\beta$ and $\vec{X}_\delta$ are the three best wolves (solutions) in the pack at a given iteration t. $\vec{A}_1$, $\vec{A}_2$ and $\vec{A}_3$ are calculated using Equation 8. While $\vec{D}_\alpha$, $\vec{D}_\beta$ and $\vec{D}_\delta$ are calculated using Equation (14), Equation (15) and Equation (16) respectively.

$$\vec{D}_{\alpha} = \left|\vec{C}_1 . \vec{X}_{\alpha} - \vec{X}\right| \tag{14}$$

$$\vec{D}_{\beta} = \left|\vec{C}_2 . \vec{X}_{\beta} - \vec{X}\right| \tag{15}$$

$$\vec{D}_{\delta} = \left|\vec{C}_3 . \vec{X}_{\delta} - \vec{X}\right| \tag{16}$$

Where $\vec{C}_1, \vec{C}_2$ and $\vec{C}_3$ are calculated based on Equation (9).

3.4. Searching and attacking the prey

The grey wolves can only attack the prey when it stops moving. This is modelled mathematically based on vector $\vec{A}$ that is utilized in Equation (8). Vector $\vec{A}$ comprises of values that span within the range $[-2a, 2a]$ and the value of $\vec{a}$ is decreased from 2 to 0 over the course of iterations using Equation (17).

$$\vec{a} = 2 - \left(\frac{2\times iter}{Max_{iter}}\right) \tag{17}$$

Where $iter$ is the iteration number and Max_{iter} is the optimal total number of iterations.

When $\left|\vec{A}\right| < 1$ the wolves are forced to attack the prey and when$\left|\vec{A}\right| > 1$, the wolf diverges out from the current prey .Searching for the prey is the exploration phase while attacking it is the exploitation phase.

Algorithm 2: Pseudo-code for the GWO

```
Begin:
 Initialize population size $n$ , parameter $a$, coefficient vectors $A, C$ and maximum number of iterations $Max_{iter}$
 Set $t \coloneqq 0$ {Counter initialization}
 for ($i$ =1: $i \leq n$) do
    Randomly generate an initial population $X_i(t)$
   Evaluate the fitness function of each agent(solution) i.e. $f(X_i)$
 end for
 Assign the values of the 1st ,2nd and 3rd best solutions i.e. $X_{\alpha}, X_{\beta}$ and $X_{\delta}$ respectively
 repeat
 for ($i$ =1: $i \leq n$) do
   Update each search agent in the population using Equation (10)
 Decrease the value of $a$ using Equation (17)
 Update the coefficients $A$ and $C$ as shown in Equation (8) and Equation (9) respectively
```

```
    Evaluate the fitness function of each search agent (vector) $f(X_i)$
  end for
  Update the vectors $X_\alpha$, $X_\beta$ and $X_\delta$
  Set $t = t + 1$ {Iteration counter increasing}
Until ($t < Max_{iter}$) {termination criteria satisfied}
Report the best solution $X_\alpha$
```

Excited -Adaptive Cuckoo Search-Intensification dedicated Grey Wolf Optimization (EACSIDGWO)

In general, effective balancing between diversification (global search) and intensification (local search) in a metaheuristic plays a beneficial and crucial role in achieving excellent performance of an algorithm [54, 55, 56]. However, it is difficult to achieve this balance with a single metaheuristic (for example either using CSA or GWO) [54, 55]. For instance, CSA is efficient at exploring the promising area of the whole search space (diversification) but ineffective at fine-tuning the end of the search space (exploitation/intensification) [57, 58]. On the other hand, GWO is good at intensification (exploitation) but inefficient at diversification (exploration) [33, 59].

For this reason, in trying to enhance mature convergence while ensuring the required effective balance between diversification and intensification is met, a hybrid algorithm called Excited- Adaptive Cuckoo Search-Intensification Dedicated Grey Wolf Optimization (EACSIDGWO) utilizing the strengths of each algorithm (i.e. CSA's diversification and GWO's intensification abilities) is proposed in this paper. Moreover, the adaptability of the proposed EACSIDGWO is guided innovatively by the complete voltage and current responses of a dc excited RC circuit (whose analysis results in first order differential equations) that find continual applications in electronics, communications and control systems [60].

4.1. Adaptive cuckoo search (ACS)

4.1.1. Adaptive step size via the complete voltage response of the dc excited RC circuit

From the details of the standard CS algorithm presented in section 2, it is evident that the algorithm lacks a criterion to control its step size through the iteration process. Control of the step size is key in guiding the CS algorithm to reach either its global maxima or minima [49, 61].

Inspired by the complete voltage response of a direct current (dc) excited RC circuit which increases with time, a novel mechanism to control the step size is proposed. Contrary to prior research [49, 61] where the step size decays with generations, in this research the step size grows with generations with the aim of strengthening the diversification (exploration) ability of the CS, which is a component of the proposed EACSIDGWO algorithm.

The solution to first order differential equation of the direct current excited RC circuit motivated the formulation of a new variant of ACS in this paper.

The complete voltage response of the RC circuit to a sudden application of a dc voltage source, with the assumption that the capacitor is initially not charged is given in equation 18.

$$v(t) = \begin{cases} 0, & t < 0 \\ V_s\left(1 - e^{-t/\tau}\right), & t > 0 \end{cases} \tag{18}$$

Where $\tau = R * C$ is the time constant, which expresses the rapidity with which this the voltage $v(t)$ rises to the value of V_s which is a constant dc voltage source. R and C are the equivalent resistance and capacitance in the circuit.

Considering the situation when $t > 0$, equation (18) can be rewritten as presented in equation (19)

$$v(t) = V_s(1 - (e^{-t})^{\tau}) \tag{19}$$

$$v(t) = V_s(1 - (\frac{1}{e^t})^{\tau}) \tag{20}$$

As $t \to \infty$, the component $\frac{1}{e^t} \to 0$ forcing $v(t \to \infty) \to V_s$. We adopt this concept i.e. the exponential growth of $v(t)$ to control the step size of the cuckoo search algorithm by introducing the proposed equation (21).

$$step_{gen+1} = step_{Max} \times (1 - (\frac{gen_{Max} - gen}{gen_{Max}})^{\tau}) \tag{21}$$

Where $gen$the current generation (iteration) is, $step_{Max}$ is the upper bound of the step size $step$ and gen_{Max} is the maximum number of generations (iterations).

To ensure that the $step_{gen+1}$ is proportional to the fitness of a given individual nest within the search space in the current generation, the non-linear modulation index τ is formulated in equation 22.

$$\tau_{i\,,gen} = \left| \frac{\left(\frac{\alpha_{nestf_{gen}} + \beta_{nestf_{gen}} + \delta_{nestf_{gen}}}{3}\right) - \mathrm{i}_{nestf_{gen}}}{\left(\frac{\alpha_{nestf_{gen}} + \beta_{nestf_{gen}} + \delta_{nestf_{gen}}}{3}\right) - \mathrm{worst}_{nestf_{gen}}} \right| \quad (22)$$

Where $\tau_{i\,,gen}$ is the the non-linear modulation index for i^{th} nest in generation gen, $\alpha_{nestf_{gen}}$is the fitness value of the alpha(α) nest (overall best nest) in generation gen, $\beta_{nestf_{gen}}$ is the fitness value of the beta (β) nest (2nd best nest) in generation gen, $\delta_{nestf_{gen}}$ is the fitness value of the delta (δ) nest (3rd best nest) in generation gen, $\mathrm{i}_{nestf_{gen}}$ is the fitness value of the i^{th} nest in generation gen and $\mathrm{worst}_{nestf_{gen}}$ is the fitness value of the worst nest among the remaining omega(ω) nests (i.e. nests whose fitness values do not feature among the top three fitness values).

Thus, equation (21) is further modified as equation (23).

$$step_{i\,,gen+1} = step_{Max} \times (1 - (\frac{gen_{Max} - gen}{gen_{Max}})^{\tau_{i\,,gen}}) \quad (23)$$

Where $step_{i\,,gen+1}$ is the step size for the for i^{th} nest in generation $gen + 1$.

From equation (23), the step size $step_{i\,,gen+1}$ is non-linearly increasing from relatively small values to values close to $step_{Max}$. The reason for proposing a non-linearly increasing strategy are as follows. Foremost, at the early stages of the proposed EACSIDGWO algorithm, whereby *ACS* is a component, the population has a higher diversity. A higher diversity imply a stronger ability to explore the global space. Our aim at this point is to accelerate convergence. Therefore, the value of the step size $step_{i\,,gen+1}$ is set to a smaller value.

It is important to point out that the anticipated accelerated convergence is a joint effort attained by foremost setting the $step_{i\,,gen+1}$ of the *ACS* to a small value at early stages, and utilizing the IDGWO *(*whose details are presented in section 4.2*)* whose core task is exploitation.

On the other hand, since the proposed EACSIDGWO algorithm is a hybrid algorithm where the ACS cooperatively works with the *IDGWO,* all the nests will be attracted to the global optima i.e. the alpha (α) nest at the later stage. This will compel them to converge prematurely without being given enough room to explore the search space. Such a situation will lead the nests away from a local optimum, and encourage diversification. For this reason, the value of the step size $step_{i\,,gen+1}$ is set to a larger value i.e. $step_{Max}$. In this paper the $step_{Max}$ is set to 1.

In other words, our main reason for proposing a non-linearly increasing step size $step_{i\,,gen+1}$ is that its small values at the initial stages of the proposed EACSIDGWO algorithm facilitates "local exploitation" while its larger values in the later stages will facilitate "global exploration".

The *ACS* can then be modeled as presented in equation 24.

$$X_{i,gen+1} = X_{i,gen} + randn\times\, step_{i,gen+1} \qquad (24)$$

Equation (24) is a formulation of the new search space for the *ACS* from the current solution.

Moreover, if this step size is considered proportional to the global best solution, then equation (24) can be formulated as given in equation (25).

$$X_{i,gen+1} = X_{i,gen} + randn\times\, step_{i,gen+1}{*}(X_{i,gen} - X_{gbest,gen}) \qquad (25)$$

Where $X_{gbest,gen}$ is the global best solution among all X_i for $i = 1,2, \ldots, n$ at generation gen ,and n is the number of host bird nests.

Thus, from equations (21) – (25) it is evident that the diversification ability of the ACS is heightened as the number of generations (gen) approach the maximum number of generations (gen_{Max}). This because the value of the step size rapidly increases towards the set maximum value of step ($step_{Max}$).

4.2. Intensification dedicated grey wolf optimizer (IDGWO)

4.2.1. Nonlinearly controlling parameter $\vec{a}$ via the complete current response of the dc excited RC circuit

It is evident from sub-section 3.4 that parameter $\vec{a}$ plays a critical role in balancing the diversification (exploration) and the intensification (exploitation) of a search agent.

A large value of control parameter $\vec{a}$ facilitates diversification while a smaller value of this parameter facilitates intensification. Thus, a suitable selection of the control parameter $\vec{a}$ can enhance a good balance between global diversification (exploration) and local intensification (exploitation).

In the original GWO (described in section 3), the value of $\vec{a}$ linearly decreases from 2 to 0.(refer to Equation 17). However, the search process of the GWO algorithm is both non-linear and complicated, which cannot be truly reflected by the linear control strategy of $\vec{a}$ presented in equation 17.

In addition, Mittal [62] proposed that an attractive performance can be attained if parameter $\vec{a}$ is non-linearly decreased rather than decreased linearly.

Inspired by the complete current response of a direct current (dc) excited RC circuit which increases with time, a novel nonlinear adjustment mechanism of control parameter $\vec{a}$ is formulated in this paper.

The complete current response of the RC circuit to a sudden application of a dc voltage source, with the assumption that the capacitor is initially not charged is given in equation 26.

$$i(t) = \frac{V_s}{R}((\frac{1}{e^t})^\tau) \qquad (26)$$

As $t \to \infty$, the component $\frac{1}{e^t} \to 0$ forcing $i(t \to \infty) \to 0$. We adopt this concept i.e. the exponential decay of $i(t)$ to formulate a novel improved strategy i.e. equation 27 to generate the values for control parameter $\vec{a}$.

$$\vec{a}_{i,gen}=a_o\times(\frac{gen_{Max}-gen}{gen_{Max}})^{\tau_{i,gen}} \quad (27)$$

Where gen is the current generation (iteration), a_o is the initial higher value of parameter a and gen_{Max} is the maximum number of generations (iterations). $\tau_{i,gen}$ is the non-linear modulation index described earlier by equation 22.

Consequently, vector $\vec{A}$ is computed as given in equation 28.

$$\vec{A}=2\vec{a}_{i,gen}.\vec{r}_1-\vec{a}_{i,gen} \quad (28)$$

Equation 27 is a non-linear decreasing control parameter for $\vec{a}_{i,gen}$ whose initial upper limit is equal to the value a_o while its final lower limit is zero.

From the original literature of GWO, the value $|\vec{A}|<1$ compels the grey wolves to move towards the prey (exploitation) while $|\vec{A}|>1$ compels them to move away from the prey in search of a fitter prey (exploration).Thus, setting a_o to 1 will always force the wolves to move to the prey which will enable us dedicate modified GWO algorithm, a component of proposed EACSIDGWO, for intensification.

4.2.2. Enhanced mature convergence via a fitness value based position-updating criterion

Both diversification and intensification are crucial for population-based optimization algorithms [62]. However, from the detailed account of the conventional GWO (refer to section 3), it is evident that all the other wolves are attracted towards the three leaders α, β and δ , a scenario that will force the algorithm to converge prematurely without attaining sufficient diversification of the search space. In other words, the conventional GWO is prone to pre-mature convergence.

In reference to the position-updated criterion of GWO described by equation 10, a new candidate individual is obtained by moving the old individual towards the best leader ($\alpha\ wolf$), the second best leader (β $wolf$) and the third best leader (δ $wolf$). This approach will force all the other grey wolves to crowd a in a reduced section of the search space that might be different from the optimal region, and without giving them a leeway to escape from such a region. In an effort to overcome this major drawback, in this paper a scheme that promotes mature converge is devised.

Instead of averaging values of vectors $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ (a form of recombining them) as a mechanism of updating the wolves' positions (refer to equation 10), in this paper we make full use of information of their fitness values as a criteria of arriving at new positions for the wolves.

Foremost the search agents of the populations $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ are computed as given in equations 29-31.

$$\vec{X}_1(i,j) = \vec{X}_\alpha(j) - \vec{A}_1.\vec{D}_\alpha \tag{29}$$

$$\vec{X}_2(i,j) = \vec{X}_\beta(j) - \vec{A}_2.\vec{D}_\beta \tag{30}$$

$$\vec{X}_3(i,j) = \vec{X}_\delta(j) - \vec{A}_3.\vec{D}_\delta \tag{31}$$

Where $i = 1,2,\dots,n$ and $j = 1,2,\dots,d$. n is the population size while d is the dimension of the search space.

Next, the fitness value for each search agent in each of the derived populations i.e. $\vec{X}_1, \vec{X}_2$ and $\vec{X}_3$ is evaluated. Further a new population with the fittest values is derived from these three populations i.e. $\vec{X}_1, \vec{X}_2$ and $\vec{X}_3$.

Equations 32-33 represents the process undertaken to derive this new population.

$$[Fit_{max}, Index] = \max\ (\bigcup_{j=1}^{3} X_j f_{i,gen}) \tag{32}$$

$$X_{i,gen+1} = \bigcup_{j=1}^{3} {\vec{X}_{j\ i,gen}}_{Index} \tag{33}$$

Where $\vec{X}_{j\ i,gen}$ is vector j computed using search agent i during iteration gen, $X_j f_{i,gen}$ is the fitness value of vector $\vec{X}_{j\ i,gen}$.

4.3. Proposed EACSIDGWO (Continuous version)

We cooperatively combined the proposed adaptive cuckoo search (ACS) and the intensification-dedicated grey wolf optimization (IDGWO), and developed the EACSIDGWO. In the EACSIDGWO algorithm, the ACS is actively involved in intensification (exploitation) during the early stage when the population has higher diversity and diversification at later stages. On the other hand, the IDGWO is only actively involved in intensification in all the stages of the proposed algorithm. By doing so, an effective balance between diversification and intensification is achieved. In addition, mature convergence is enhanced which in the end leads to high quality solutions.

Proposed EACSIDGWO (Binary version)

Selection of features is binary by nature [63]. Therefore, the proposed EACSIDGWO algorithm cannot be utilized in selection of features without further modifications.

In the proposed EACSIDGWO algorithm, the new positions of the search agents will have continuous solutions, which must be converted into corresponding binary values.

In this paper, this conversion is achieved by foremost applying squashing of the continuous solutions in each dimension using a sigmoid (S-shaped) transfer function [63].This will compel the search agents to move into a binary search space as depicted by equation 35.

$$S = \frac{1}{1 + e^{-10({X^d}_{i,gen} - 0.5)}} \tag{34}$$

Where ${X^d}_{i,gen}$ is a continuous-valued position of the i^{th} search agent in the d^{th} dimension during generation gen.

The output S of the sigmoid transfer function is still a continuous value and thus it has to be the threshold to reach the binary-value one. Normally, the sigmoid function maps smoothly the infinite input to a finite output [63].To arrive at the binary solution when a sigmoid function is used, the commonly stochastic threshold is applied as presented in equation 35.

$$y^{d}{}_{i,gen} = \begin{cases} 0 & if\ rand < S \\ 1 & if\ rand \geq S \end{cases} \tag{35}$$

$$Y_{i,gen} = \bigcup_{i=1}^{n} y^{d}{}_{i,gen} \tag{36}$$

Where $y^{d}{}_{i,gen}$ is the binary updated position at generation gen in the d^{th} dimension and $rand$ is a random number drawn from a uniform distribution $\in [0,1]$. $Y_{i,gen}$ is the equivalent binary vector of the i^{th} search agent at generation gen.

Using this approach, the original solutions remain in the continuous domain of the proposed EACSIDGWO algorithm and can be converted to binary when need arises.

The pseudocode of the binary version of the proposed EACSIDGWO algorithm is presented in Algorithm 3.

Algorithm 1: Pseudo-code for the EACSIDGWO (Binary Version)

Input: labelled biomedical dataset *D*, *MaxIter*, ACS and IDGWO parameters value, number of host bird nests (n), number of dimensions (features) d, Lower bound (L_b) and Upper bound (U_b)

Output: Best Fitness , Best Search Agent

```
for each nest i (i =1, 2...n) do
  for each dimension j(j=1,2,...,d) do
    $X^{j}{}_{i,0}$=random number drawn from $[L_b, U_b]$
  end
  Convert continuous values of $X_{i,0}$ to binary using Eq. 35, 36 and 37
  Train a classifier to evaluate the accuracy of the equivalent binary vector of $X_{i,0}$ and store the value in $Xf_{i,0}$
end
[~, Index]=Sort ( $Xf_0$,'descend')
$\alpha_{nestf_0} = Xf_0(Index(1))$
$\beta_{nestf_0} = Xf_0(Index(2))$
$\delta_{nestf_0} = Xf_0(Index(3))$
worst$_{nestf_0} = Xf_0(Index(n))$
$\alpha_{nest_0} = X_0(Index(1))$
$\beta_{nest_0} = X_0(Index(2))$
$\delta_{nest_0} = X_0(Index(3))$
While ($gen \leq MaxIter$)
```

```
17      for each nest i (i =1, 2...n) do
18          Calculate $\tau_{i\ ,gen}$ and $step_{i\ ,gen+1}$ using Eq. 22 and 23 respectively
19          Generate a new cuckoo nest $X_{i,gen+1}$ using Eq. 25
20  Convert continuous values of $X_{i,gen+1}$ to binary using Eq. 35, 36 and 37
21  Train a classifier to evaluate the accuracy of the equivalent binary vector of $X_{i,gen+1}$ and store the value in $Xf_{i,gen+1}$
22  if( $Xf_{i,gen+1} > Xf_{i,0}$) then
23      $Xf_{i,0}$= $Xf_{i,gen+1}$
24      $X_{i,0}$=$X_{i,gen+1}$
25  end
26    end
27  Repeat step 8 to 15
28  for each nest i (i =1, 2...n) do
29  Calculate $\tau_{i\ ,gen}$ and $a_{i\ ,gen}$ using Eq. 22 and 27 respectively
30    for each dimension j(j=1,2,...,d) do
31  Calculate coefficients $A$ and $C$ as shown in Equation (28) and Equation (9) respectively
32  Compute vectors$\vec{X}_{1_{i,gen}}(j)$, $\vec{X}_{2_{i,gen}}(j)$ and $\vec{X}_{3_{i,gen}}(j)$ using Equations 29, 30 and 31 respectively.
33  end
34  Convert continuous values of $\vec{X}_{1_{i,gen}}$,$\vec{X}_{2_{i,gen}}$ and $\vec{X}_{3_{i,gen}}$ to binary using Eq. 35, 36 and 37
35  Consecutively, train a classifier to evaluate the accuracies of the equivalent binary vectors of $\vec{X}_{1_{i,gen}}$,$\vec{X}_{2_{i,gen}}$ and $\vec{X}_{3_{i,gen}}$ and store the value in $X_1f_{i,gen}$,$X_2f_{i,gen}$ and $X_3f_{i,gen}$ respectively.
36  Determine $\vec{X}_{i,gen+1}$ using equations 32 and 33 respectively
37  end
38  Repeat step 8 to 15
39  Abandon a fraction of $P_a$ worst nests and generate new ones according to Equation (6)
40  Keep best solutions(or those nests with quality solutions)
41  Repeat step 8 to 15
    end
42  Best Search Agent=$\alpha_{nest_0}$
```

```
43  Best Fitness=$\alpha_{nestf_0}$
```

Experimental methodology

In this section, detailed accounts of the biomedical datasets, evaluation metrics, proposed fitness function and the parameter setting for the considered metaheuristic algorithms are outlined.

6.1. Considered Biomedical Datasets

To validate the performance of the considered metaheuristic algorithms, six benchmark biomedical datasets extracted from the UCI Irvine Machine [64] were utilized. Each dataset has two classes and the performance of each of these algorithms is evaluated based on its ability to classify these classes correctly. Details of these datasets are given in Table 1.

Table 1: Considered Biomedical Datasets

Dataset	**Number of Features**	**Number of Cases**
Breast Cancer Wisconsin (Prognosis)	33	198
Breast Cancer Wisconsin (Diagnostic)	30	569
SPECTF Heart	44	267
Ovarian Cancer	4000	216
CNS	7129	60
Colon	2000	62

6.2. Evaluation Metrics

For the considered feature selection problem, the following evaluation metrics were utilized to compare the performance of each considered feature selection technique.

Average Accuracy (Avg_Acc)- It is one of the commonly used classification metric that represents the number of correctly classified instances by using a particular feature set. The mathematical formulation of this metric is given in Equation 37.

$$Avg\_Acc = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{k}\sum_{j=1}^{k}Acc_j \quad (34)$$

Where N is the number of times (runs) a given metaheuristic algorithm is run, k represents the number folds utilized and Acc_j is the accuracy reported during fold j. Acc_j is defined in Equation 35.

$$Acc_j = \frac{TP_j + TN_j}{TP_j + TN_j + FP_j + FN_j} \quad (35)$$

Where TP and FN denote the number of positive samples in fold j , that are accurately and falsely predicted, respectively, and TN and FP represent the number of negative samples in the same fold that are predicted accurately and wrongly, respectively [65].

Average Feature Length (Avg_NFeat)-This metric characterizes the average length of selected features to the total number of features in the dataset. Equation 36 gives its mathematical formulation.

$$Avg\_NFeat = \frac{1}{N}\sum_{i=1}^{N}Sel\_Feat_i \quad (36)$$

Where Sel_Feat_i is the number of selected features in the testing dataset during run i.

Minimum Accuracy (Min_Acc) - Is the least value of accuracy reported during N runs. Equation 37 depicts its formulation.

$$Min\_Acc = \min\left(\bigcup_{j=1}^{N}Avg\_crossAcc_j\right) \quad (37)$$

Where $Avg\_crossAcc_i$ is given by Equation 38

$$Avg\_crossAcc_i = \frac{1}{k}\sum_{j=1}^{k}Acc_j \quad (38)$$

Maximum Accuracy (Max_Acc) - Is the largest value of accuracy reported during N runs. Its mathematical formulation is given by Equation 39.

$$Max\_Acc = \max\left(\bigcup_{j=1}^{N}Avg\_crossAcc_j\right) \quad (39)$$

Maximum Features Selected (Max_NFeat) - Is the largest number of selected features during N runs. Equation 40 gives its mathematical formulation.

$$Max\_NFeat = \max\left(\bigcup_{i=1}^{N} Sel\_Feat_i\right) \tag{40}$$

Minimum Features Selected (Min_NFeat) - Is the least number of selected features during N runs. Equation 41 gives its mathematical formulation.

$$Min\_NFeat = \min\left(\bigcup_{i=1}^{N} Sel\_Feat_i\right) \tag{41}$$

6.3. Evaluation of the classifier performance

Since the Support Vector machine classifier has already made immense contributions in the field of microarray-based cancer classification [65], it was adopted in this paper to evaluate the classification accuracy using the selected subset of features returned by the various considered metaheuristic feature selection approaches. The Matlab fitcsvm function that trains and cross-validates an SVM model was adopted in this paper. We specified the kernel scale parameter to "auto" to allow the function select the appropriate scale factor using a heuristic search.

With the SVM classifier, the data items are mapped points in an n −dimensional feature space (i.e. n=number of features) and the each feature's value is a value of a given coordinate. The final output of this classifier is an optimal hyperplane which can be used to classify new cases[18, 65].

However, the performance of the SVM classifier is highly dependent on the selection of its kernel function [18,65].A reason why experiments were conducted using various kernels in this paper.

Selecting a suitable kernel is both dataset and problem specific and selected experimentally [18, 65]. Based on the conducted experiments, suitable kernel functions were selected for the considered datasets . The considered datasets and their suitable kernel functions are presented in Table 2.

More information of selecting suitable SVM kernel functions is presented in [65].

Table 2: Selection of suitable kernel functions

Dataset	**Kernel function**
Breast Cancer Wisconsin (Prognosis)	Radial Basis Function (RBF)

Breast Cancer Wisconsin (Diagnostic)	Radial Basis Function (RBF)
SPECTF Heart	Radial Basis Function (RBF)
Ovarian Cancer	Linear Function
CNS	Linear Function
Colon	Linear Function

6.4. Fitness function

The main aim of a feature selection exercise is to discover a subset of features from the whole set of existing features in a given dataset such that, the considered optimization algorithm is able to achieve the highest possible accuracy using that subset. For instance in datasets with many features (attributes), the objective is to minimize the number of selected features while improving the classification accuracy of the feature selection approach.

In classifications tasks, there exists higher chances that two feature subsets containing different number of features will have the same accuracy [18].However, if a subset with a large number of features is discovered earlier by a given optimization algorithm, it is likely that the one with least features will be ignored[18].

In trying to overcome this challenge, a fitness function proposed in [18] to evaluate the classification performance of optimization algorithms for feature selection tasks is adopted. This fitness function is given in Equation 42.

$$Fit = \alpha * \frac{|R|}{|N|} - \beta * Avg\_crossAcc_i \tag{42}$$

Where $|N|$ represents the total number of features within a given dataset, $|R|$ represents the number of selected features during run i and $Avg\_crossAcc_i$ is the average crossvalidation accuracy reported during run i (refer to Equation 38). β and α are two weights corresponding to the significance of the classification quality and the subset length respectively. In this paper, β is set to 0.8 and $\alpha = 0.2$ as adopted from [18].

It is important to point out that both terms are normalized by dividing by their largest possible values i.e. the number of selected features $|R|$ is divided by the total number of features $|N|$, and average accuracy $Avg\_crossAcc_i$ is divided by the value 1.

6.5. Parameter setting for the considered feature selection techniques

The performance of the proposed EACSIDGWO algorithm was compared to those of Extended Binary Cuckoo Search (EBCS), Binary Anti-Colony Optimization (BACO), Binary Genetic Algorithm (BGA) and Binary Particle Swarm Optimization (BPSO) that were reported earlier in [18].

Table 3 indicates the selected parameter values for both the proposed BEACSIDGWO algorithm and each of other algorithms as reported in [18].

Table 3: Selection of parameter values for the considered approaches

Algorithm	Parameter values
EACSIDGWO	$step_{Max} = 1, a_o = 1,$ P_a=0.25
EBCS	$N_{mut} = 10, \lambda = 1, \alpha = 1,$ P_a=0.4
BACO	$\Gamma_{initial} = 0.1, \alpha = 1, p = 0.1$
BGA	$M_r = 0.1, C_r = 0.1$
BPSO	$C_1 = 1, C_2 = 2,$ $\omega_{initial} = 0.9,$ $\omega_{vary-for} = 0.9$

To be consistent with the setup proposed in [18], the population size for the proposed EACSIDGWO was set to 30. Then the algorithm was run 10 times to perform the feature selection task for each considered dataset. In addition, each run terminated when 10000 fitness function evaluations was attained. This approach, allowed the proposed algorithm to utilize the fitness function at an equal number of times.

In this paper, all the experiments were conducted using Matlab 2017 running on Windows 10 operating system on a HP desktop with Intel(R) Core (TM) i7-3770CPU @ 3.4GHZ with 12.0GB of RAM.

7. Results and Discussion

To examine the diversification and intensification of the proposed EACSIDGWOA, detailed comparative study is presented in this section.

The efficiency and the optimization performance of the proposed algorithm has been verified by comparing and analyzing its results with those of four other state-of-the-art optimization algorithms.

The experimental classification results have been probed through statistical tests, comparative analysis and ranking methods.

Tables 4-9 provides the performance of all the considered optimizations approaches for feature selection using the datasets described in subsection

6.1. It is important to point out that the best result achieved in each column for all the considered biomedical datasets is highlighted in bold while the worst is italicized.

To prove that the proposed EACSIDGWO is superior over the other four-optimization algorithms, Wilcoxon rank-sum test i.e. a non-parametric statistical test is also performed. The statistical results for the p, h and z values obtained from the pairwise comparisons of the four groups are tabulated in Table 10. Tables 11-12 present a comparison of the overall ranking of the results obtained by the considered algorithms.

7.1 Discussion

7.1.1 Investigation of the obtained classification results. From Tables 4-9, the following observations can be made.

(i) The proposed EACSIDGWO algorithm outperformed all the other considered algorithms in terms of classification accuracy for all the utilized datasets. It recorded the highest classification accuracy on the three highly dimensioned datasets (i.e. Ovarian, CNS and Colon) as well as the remaining three small sample sized datasets. This promising performance is largely attributed to the cooperative exploitation conducted by ACS and IDGWO components of the proposed algorithm during the early generations, as well as the single-handedly exploitation and exploration by IDGWO and ACS respectively at later generations.

(ii) For four datasets i.e. Ovarian, Heart, CNS and Colon, the proposed algorithm attained a value for Avg_Acc that is larger than the value for Max_Acc attained by the EBCS. EBCS is a variant of Cuckoo Search, which is a component of the proposed EACSIDGWO algorithm. This superior performance proves the competency of the proposed approach to efficiently determine the optima within the search space.

(iii) With regard to the average feature length (Avg_NFeat), the proposed B-EACSIDGWO algorithm demonstrated a superior performance by selecting the least number of features compared to the other algorithms. According to the results reported in Tables 4-9, the proposed algorithm performed better on all the considered datasets.

(iv) In comparison with the original number of features in the considered datasets, there is a notable reduction in the number selected features by the proposed approach. For instance, the actual number of features in ovarian cancer, CNS and Colon cancer datasets is 4000, 7129 and 2000 respectively, whereas the number of selected features by the proposed EACSIDGWO is 274.8, 1208.1 and 538.5 respectively. This clearly indicates the proposed algorithm is able to reduce the number of features as well as locate the most significant optimal feature subsets. The strength of the proposed EACSIDGWO lies in its well-formulated algorithm (refer to section 5) that enhances both its diversification and intensification capabilities which enables it to eliminate redundant (non-informative) attributes and then actively search within the high-performance regions of the feature space.

Table 4: Experimental results for the ovarian cancer dataset

Algorithm	**Accuracy**			**Number Of Features**		
	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat
EACSIDGWO	**1.000**	**1.000**	**1.000**	**292**	**264**	**274.8**
EBCS	*0.991*	0.991	0.991	1855	1747	1811.6
BACO	*0.991*	*0.986*	*0.990*	*1971*	*1912*	*1945.7*
BGA	*0.991*	0.991	0.991	1830	1755	1887.3
BPSO	*0.991*	*0.986*	*0.990*	1913	1777	1857

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 5: Experimental results for the breast cancer Wisconsin (Diagnostic) dataset

Algorithm	**Accuracy**			**Number Of Features**		

	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat
EACSIDGWO	0.977	**0.974**	**0.975**	**3**	**3**	**3**
EBCS	**0.981**	**0.974**	0.973	4	**3**	3.1
BACO	*0.972*	*0.960*	*0.969*	*8*	*6*	*7*
BGA	0.975	0.965	0.972	6	**3**	3.6
BPSO	**0.981**	0.963	0.974	*8*	**3**	5.4

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 6: Experimental results for the breast cancer Wisconsin (Prognosis) dataset

Algorithm	**Accuracy**			**Number Of Features**		
	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat
EACSIDGWO	**0.879**	**0.864**	**0.873**	**7**	**3**	**5.6**
EBCS	0.874	0.828	0.856	8	4	6.2
BACO	*0.818*	*0.768*	*0.794*	*12*	*5*	*8.4*
BGA	0.874	0.793	0.843	10	4	6.5
BPSO	*0.848*	*0.798*	0.821	11	4	8.3

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 7: Experimental results for the SPECTF Heart dataset

Algorithm	**Accuracy**			**Number Of Features**		
	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat
EACSIDGWO	**0.884**	**0.861**	**0.875**	**6**	**3**	**4.5**

EBCS	0.873	0.846	0.861	8	5	6.2
BACO	*0.846*	*0.813*	*0.831*	*15*	*10*	*12.1*
BGA	**0.884**	0.846	0.866	11	4	8.4
BPSO	0.865	0.846	0.854	*15*	9	10.9

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 8: Experimental results for the CNS dataset

Algorithm	**Accuracy**			**Number Of Features**		
	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat
EACSIDGWO	**0.767**	**0.700**	**0.718**	**1623**	**807**	**1208.1**
EBCS	*0.667*	0.667	0.667	3490	3391	3446.7
BACO	*0.667*	*0.650*	*0.660*	*3589*	3432	*3522.9*
BGA	0.683	0.667	0.668	3566	*3438*	3489.7
BPSO	*0.667*	0.667	0.667	3547	3359	3474.3

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

Table 9: Experimental results for the colon dataset

Algorithm	**Accuracy**			**Number Of Features**		
	Max_Acc	Min_Acc	Avg_Acc	Max_NFeat	Min_NFeat	Avg_NFeat
EACSIDGWO	**0.919**	**0.887**	**0.905**	**637**	**397**	**538.5**
EBCS	0.903	0.871	0.887	*1016*	*961*	*988.7*
BACO	0.903	0.871	*0.881*	1002	932	976

BGA	*0.887*	0.871	0.882	1003	944	962.8
BPSO	*0.887*	*0.855*	0.879	1003	933	971.2

Values in bold represent the best result and values in italic denote the worst in each column, respectively.

7.1.2 Statistical analysis. The superiority of the proposed EACSIDGWO algorithm has been verified via Wilcoxon rank-sum test i.e. a non-parametric test with a significance level of 5%. The results obtained for the pairwise comparison of the four groups are presented in Table 10. Observations from Table 10 reveal the statistical significance of the obtained experimental results for all the considered datasets. This clearly indicates that the proposed approach has an attractive performance in relation to the other four approaches. Thus, the overall statistical results by our algorithm are highly significant from the results of the four algorithms for all the considered datasets.

7.1.3 Ranking methods. Tables 11-12 outline detailed ranking of all the considered algorithms with their respective comparative analysis. The ranking is based on maximum accuracy (Max_Acc), minimum accuracy (Min_Acc), average accuracy (Avg_Acc), maximum number of selected features (Max_NFeat), minimum number of selected features (Min_NFeat) and average number of selected features (Avg_NFeat). From the ranking, it is evident that that the proposed EACSIDGWO algorithm obtained the best values in all these measures for all the datasets. Considering the final ranks, the proposed algorithm attained an attractive performance whose overall rank value is 37.This clearly reveals the superiority of EACSIDGWO algorithm in relation to the four state-of-the-art algorithms.

Table 10: Using Wilcoxon's rank sum test at $p = 0.05$ to compare EACSIDGWO with other algorithms

Dataset	Wilcoxon's rank sum test	EBCS Vs EACSIDGWO	BACO Vs EACSIDGWO	BGA Vs EACSIDGWO	BPSO Vs EACSIDGWO
Ovarian Cancer	p value	0.000181651	0.000181651	0.000182672	0.000181651

	h value	1.000000000	1.000000000	1.000000000	1.000000000
	z value	3.743255786	3.743255786	3.741848283	3.743255786
Breast Cancer Wisconsin (Diagnostic)	p value	0.022591996	0.000146767	0.017044126	0.000582314
	h value	1.000000000	1.000000000	1.000000000	1.000000000
	z value	2.28026466	3.796476695	2.38575448	3.439721266
Breast Cancer Wisconsin (Prognosis)	p value	0.000730466	0.0001707	0.00073729	0.000174624
	h value	1.000000000	1.0000000	1.00000000	1.000000000
	z value	3.377881495	3.758843896	3.375323463	3.753152986
SPECTF Heart	p value	0.000321376	0.000176611	0.000176611	0.000177611
	h value	1.000000000	1.000000000	1.000000000	1.000000000
	z value	3.597430949	3.750317207	3.750317207	3.748901726
CNS	p value	0.000182672	0.000182672	0.000182672	0.000182672
	h value	1.000000000	1.000000000	1.000000000	1.000000000
	z value	3.741848283	3.741848283	3.741848283	3.741848283
COLON	p value	0.000182672	0.000182672	0.000182672	0.000181651
	h value	1.000000000	1.000000000	1.000000000	1.000000000
	z value	3.741848283	3.741848283	3.741848283	3.743255786

Table 11: Overall ranking of considered algorithms

Algorithm	Measures	Datasets									
		Ovarian Cancer	Breast Cancer Wisconsin (Diagnostic)	Breast Cancer Wisconsin (Prognosis)	SPECTF Heart	CNS	Colon	Sum of ranks	Overall rank	Total sum	Final ranks

EACSIDGWO	Max_Acc	1	2	1	1	1	1	7	1	37	1
	Min_Acc	1	1	1	1	1	1	6	1		
	Avg_Acc	1	1	1	1	1	1	6	1		
	Max_NFea	1	1	1	1	1	1	6	1		
	Min_NFea	1	1	1	1	1	1	6	1		
	Avg_NFea	1	1	1	1	1	1	6	1		
EBCS	Max_Acc	2	1	2	3	3	2	13	2	84	2
	Min_Acc	2	1	2	2	2	2	11	2		
	Avg_Acc	2	2	2	3	3	2	14	2		
	Max_NFea	3	2	2	2	2	4	15	2		
	Min_NFea	2	1	2	3	3	5	16	2		
	Avg_NFea	2	2	2	2	2	5	15	2		
BACO	Max_Acc	2	4	4	4	2	2	18	4	138	5
	Min_Acc	3	4	5	3	3	2	20	4		
	Avg_Acc	3	5	5	5	4	4	26	5		
	Max_NFea	5	4	5	4	5	2	25	5		
	Min_NFea	5	2	3	5	3	2	20	3		
	Avg_NFea	5	5	5	5	5	4	29	5		

Table 12: Overall ranking of considered algorithms

Algorithm	Measures	Datasets									
		Ovarian Cancer	Breast Cancer Wisconsin (Diagnostic)	Breast Cancer Wisconsin (Prognosis)	SPECTF Heart	CNS	Colon	Sum of ranks	Overall rank	Total sum	Final ranks

BGA	Max_Acc	2	3	2	1	2	3	13	2	95	3
	Min_Acc	2	2	5	2	2	2	15	3		
	Avg_Acc	2	4	3	2	2	3	16	3		
	Max_NFeat	2	3	3	3	4	3	18	3		
	Min_NFeat	3	1	2	2	4	4	16	2		
	Avg_NFeat	3	3	3	3	3	2	17	3		
BPSO	Max_Acc	2	1	3	3	3	3	15	3	110	4
	Min_Acc	3	3	2	2	2	3	15	3		
	Avg_Acc	3	2	4	4	3	5	21	4		
	Max_NFeat	4	4	4	4	3	3	22	4		
	Min_NFeat	4	1	2	4	2	3	16	2		
	Avg_NFeat	3	4	4	4	3	3	21	4		

8. Conclusion

This paper proposed a new hybrid Excited (E) - Adaptive Cuckoo Search (ACS)-Intensification Dedicated Grey Wolf Optimizer (IDGWO) i.e. EACSIDGWO algorithm to solve the feature selection problem in biomedical science. In the proposed algorithm, the the concept of the complete voltage and current responses of a direct current (DC) excited resistor capacitor (RC) circuit are innovatively utilized to make the step size of ACS and the non-linear control strategy of parameter $\vec{a}$ of the IDGWO adaptive. Since the population has a higher diversity during early stages of the proposed algorithm, both the ACS and IDGWO are jointly utilized to attain accelerated convergence. However, to enhance mature convergence while striking an effective balance between exploitation and exploration in latter stages, the role of ACS is switched to global exploration while the IDGWO is still left conducting the local exploitation. In order to test the efficiency of the proposed EACSIDGWO as a feature selector, six standard biomedical datasets from the University of California at Irvine

(UCI) repository were utilized. The experimental results obtained prove that the proposed algorithm is superior to the state-of-the-art feature selection techniques i.e. BACO, BGA, BPSO and EBCSA in attaining a good learning from fewer instances, and optimal feature selection from information-rich biomedical data, all these while maintaining a high . classification accuracy of the utilized data. In future, utilizing this hybrid algorithm as a filter-feature selection approach seeking to evaluate the generality of the selected features will be a valuable contribution.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Funding Statement

This work is fully supported by the African Development Bank (AfDB), through the Ministry of Education, Kenya Support for Capacity Building.

References

[1] J. Han, J. Pei, and M. Kamber, Data Mining: Concepts and Techniques. Amsterdam, The Netherlands: Elsevier, 2011.

[2] H. Liu and H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, vol. 454. Springer, 2012.

[3] M.Bennasar,Y.Hicks,andR.Setchi,''Features electionusingjointmutual information maximisation,'' Expert Syst. Appl., vol. 42, pp. 8520–8532, Sep. 2015.

[4] M. Dash and H. Liu, ''Feature selection for classification,'' Intell. Data Anal., vol. 1, nos. 1–4, pp. 131–156, 1997.

[5] I. Guyon and A. Elisseeff, ''An introduction to variable and feature selection,'' J. Mach. Learn. Res., vol. 3, pp. 1157–1182, Jan. 2003

[6] H. Liu and Z. Zhao, ''Manipulating data and dimension reduction methods: Feature selection,'' in Encyclopedia Complexity Systems Science. New York, NY, USA: Springer, 2009, pp. 5348–5359.

[7] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, ‘‘Feature selection: An ever evolving frontier in data mining,’’ in Proc.4thWorkshop Feature Selection Data Mining, May 2010, pp. 4–13.

[8] A. Zarshenas and K. Suzuki, ‘‘Binary coordinate ascent: An efficient optimization technique for feature subset selection for machine learning,’’ Knowl.-Based Syst., vol. 110, pp. 191–201, Oct. 2016.

[9] E.-G. Talbi, Metaheuristics: From Design to Implementation, vol. 74, Hoboken, NJ, USA: Wiley, 2009.

[10] H. Liu and H. Motoda, Feature Extraction, Construction and Selection: A Data Mining Perspective, vol. 453. Springer, 1998

[11] R.Bello,Y.Gomez,A.Nowe,andM.M.Garcia,‘ ‘Two-step particle swarm optimization to solve the feature selection problem,’’in Proc.7th Int. Conf. Intell. Syst. Des. Appl. (ISDA), Oct. 2007, pp. 691–696.

[12] I. Guyon and A. Elisseeff, ‘‘An introduction to variable and feature selection,’’ J. Mach. Learn. Res., vol. 3, pp. 1157–1182, Jan. 2003.

[13] E. Emary, H. M. Zawbaa, and A. E. Hassanien, ‘‘Binary grey wolf optimization approaches for feature selection,’’ Neurocomputing, vol. 172, pp. 371–381, Jan. 2016.

[14] Q. Al-Tashi, H. Rais, and S. Jadid, ‘‘Feature selection method based on grey wolf optimization for coronary artery disease classification,’’ in Proc. Int. Conf. Reliable Inf. Commun. Technol., 2018, pp. 257–266

[15] M. M. Kabir, M. Shahjahan, and K. Murase, ‘‘A new local search based hybrid genetic algorithm for feature selection,’’ Neurocomputing, vol. 74, no. 17, pp. 2914–2928, Oct. 2011.

[16] S. Kashef and H. Nezamabadi-pour, ‘‘An advanced ACO algorithm for feature subset selection,’’ Neurocomputing, vol. 147, pp. 271–279, Jan. 2015.

[17] E. ZorarpacI and S. A. Özel, ‘‘A hybrid approach of differential evolution and artificial bee colony for feature selection,’’ Expert Syst. Appl., vol. 62, pp. 91–103, Nov. 2016.

[18] S. Salesi and G. Cosma, “A novel extended binary cuckoo search algorithm for feature selection,” in 2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA). IEEE, 2017, pp. 6–12.

[19] M. M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, and S. Mirjalili, ‘‘Binary

dragonfly algorithm for feature selection,'' in Proc. Int. Conf. New Trends Comput. Sci. (ICTCS), Oct. 2017, pp. 12–17.

[20] A. H. Gondomi and X. S. Yang, ''Chaotic bat algorithm,'' Journal of Computational Science, vol. 5, no. 2, pp. 224-232,2014.

[21] N. Abd-Alsabour, ''A review on evolutionary feature selection,'' in Proceedings of the European Modelling Symposium (EMS), IEEE, 2014.

[22] S. Mirjalili, ''SCA: a sine cosine algorithm for solving optimization problems,'' Knowledge- Based Systems, vol. 96, pp. 120-133, 2016.

[23] H. Ma and D. Simon, ''Blended biogeography-based optimization for constrained optimization,'' Engineering Applications of Artificial Intelligence, vol. 24, no. 3, pp. 517-525, 2011.

[24] N. Almugren and H. Alshamlan, ''A survey on hybrid feature selection methods in microarray gene expression data for cancer classification,'' IEEE Access, vol. 7, pp. 78533–78548, 2019.

[25] R. Sindhu, R. Ngadiran, Y. M. Yacob, N. A. H. Zahri, M. Hariharan and K. Polat, ''A hybrid SCA Inspired BBO for feature selection problems,'' Mathematical Problems in Engineering Volume 2019, Article ID 9517568, 18 pages, 2019.

[26] P. J. Gaidhane and M. J. Nigam, ''A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems,'' J. Comput. Sci., vol. 27, pp. 284–302, Jul. 2018.

[27] H. M. Zawbaa, E. Emary, C. Grosan and V. Snasel, ''Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach,'' Swarm and Evolutionary Computation, vol. 42, pp. 29–42, 2018.

[28] A. A. Alomoush, A. A. Alsewari, H. S. Alamri, K. Aloufi and K. Z. Zamli, ''Hybrid harmony search algorithm with grey wolf optimizer and modified opposition-based learning,'' in IEEE Acess, vol. 7, pp. 68764–68785, 2019.

[29] S. Arora, H. Singh, M. Sharma and P. Anand, ''A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection,'' in IEEE Acess, vol. 7, pp. 26343–26361, 2019.

[30] B. N. Gohil and D. R. Patel, ''A hybrid GWO-PSO algorithm for load balancing in cloud computing environment,'' 2018 Second

International Conference on Green Computing and Internet of Things (ICGCIoT), Bangalore, India, 2018, pp. 185–191.

[31] A. Zhu, C. Xu, Z. Li, J. Wu and Z. Liu, ‘‘Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC,’’ Journal of Systems Engineering and Electronics, vol. 26, no. 2, April 2015, pp. 317-328

[32] Z. Yue, S. Zhang and W. Xiao, ‘‘A novel hybrid algorithm based on grey wolf optimizer and fireworks algorithm,’’ Sensors (Basel), vol. 20, no. 7, April 2020, pp. 1-7.

[33] J. Cheng, L. Wang and Y. Xiong, ‘‘Ensemble of cuckoo search variants,’’ Computer and Industrial Engineering,September 2019, vol. 135, pp. 299-313.

[34] W. Long, S. Cai, J. Jiao, M. Xu and T. Wu, ‘‘A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models,’’ Energy Conversion and Management, January 2020, vol. 203, 112243.

[35] Z. Zhang, S. Ding and W. Jia, ‘‘A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems,’’ Engineering Applications of Artificial Intelligence, October 2019, vol. 85, pp. 254-268.

[36] X. Zhang, Li, X. T., M. H. and Yin, ‘‘Hybrid cuckoo search algorithm with covariance matrix adaption evolution strategy for global optimisation problem,’’ International Journal of Bio-Inspired Computation, vol. 13, no. 2, pp. 102-110.

[37] G. H. Davies, “The life of birds, parenthood,” 1970. [Online]

[38] K. Khan and A. Sahai, “Neural-based cuckoo search of employee health and safety (hs),” International Journal of Intelligent Systems and Applications, vol. 5, no. 2, p. 76, 2013.

[39] X.-S. Yang and S. Deb, “Cuckoo search via lévy flights,” in 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 210–214, IEEE, 2009.

[40] X.-S. Yang and L. Press, “Nature-inspired metaheuristic algorithms second edition,” 2010.

[41] C. T. Brown, L. S. Liebovitch, and R. Glendon, “Lévy flights in dobe ju/’hoansi foraging patterns,” Human Ecology, vol. 35, no. 1, pp. 129–138, 2007.

[42] I. Pavlyukevich, “Lévy flights, non-local search and simulated annealing,” Journal of Computational Physics, vol. 226, no. 2, pp. 1830–1844, 2007.

[43] I. Pavlyukevich, “Cooling down lévy flights,” Journal of Physics A: Mathematical and Theoretical, vol. 40, no. 41, p. 12299, 2007.

[44] A. M. Reynolds and M. A. Frye, “Free-flight odor tracking in drosophila is consistent with an optimal intermittent scale-free search,” PloS one, vol. 2, no. 4, p. e354, 2007.

[45] M. F. Shlesinger, G. M. Zaslavsky, and U. Frisch, “Lévy flights and related topics in physics,” 1995.

[46] M. F. Shlesinger, “Mathematical physics: Search research,” Nature, vol. 443, no. 7109, p. 281, 2006.

[47] X.-S. Yang and S. Deb, “Engineering optimisation by cuckoo search,” arXiv preprint arXiv:1005.2908, 2010.

[48] L.Wang,Y.Yin,andY.Zhong,“Cuckoo search with varied scaling factor,” Frontiers of Computer Science, vol. 9, no. 4, pp. 623–635, 2015.

[49] Mohamed Reda1, Amira Y. Haikal1, Mostafa El-hosseini1,2, and Mahmoud Badawy1, “An Innovative Damped Cuckoo Search Algorithm with a Comparative Study against Other Adaptive Variants,” IEEE Access, vol. 7, pp. 78533–78548, 2019.

[50] X.-S. Yang and S. Deb, “Cuckoo search: recent advances and applications,” Neural Computing and Applications, vol. 24, no. 1, pp. 169–174, 2014.

[51] S. Mirjalili, S. M. Mirjalili, and A. Lewis, ‘‘Grey wolf optimizer,’’ Adv. Eng. Softw., vol. 69, pp. 46–61, Mar. 2014.

[52] S. Arora, H. Singh, M. Sharma, S. Sharma, and P. Anand, ‘‘A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection,’’ IEEE Access, vol. 7, pp. 26343–26361, 2019.

[53] Qasem Al-Tashi, Said Jadid Abdul Kadir, Helmi Md Rais, Seyedali Mirjalili, and Hitham Alhussian. “Binary optimization using hybrid grey wolf optimization for feature selection” . IEEE Access, 7:39496–39508, 2019.

[54] M. M. Mafarja and S. Mirjalili, ‘‘Hybrid whale optimization algorithm with simulated annealing for feature selection,’’

Neurocomputing, vol. 260, pp. 302–312, Oct. 2017.

[55] M. Črepinšek, S.-H. Liu, and M. Mernik, ‘‘Exploration and exploitation in evolutionary algorithms: A survey,’’ ACM Comput. Surveys (CSUR), vol. 45, no. 3, p. 35, 2013.

[56] J. Wei and Y. Yu, ‘‘An effective hybrid cuckoo search algorithm for unknown parameters and time delays estimation of chaotic systems,’’IEEE Access, vol. 6, pp. 6560–6571, 2017.

[57] J. Huang, X. Li, and L. Gao, "A New Hybrid Algorithm for Unconstrained Optimisation Problems," International Journal of Computer Applications in Technology, vol. 46, no. 3, pp. 187– 194, 2013.

[58] Rui Chi ,Yixin Su, Zhijian Qu, and Xuexin Chi, "A Hybridization of Cuckoo Search and Differential Evolution for the Logistics Distribution Center Location Problem," Hindawi Mathematical Problems in Engineering Volume 2019, pp. 1– 16, 2013.

[59] Long W,Jiao J,Liang X,Tang M ,Yixin Su, Zhijian Qu, and Xuexin Chi, "An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization," Engineering applications of artificial intelligence, vol. 68, pp. 63-80, 2018.

[60] Charles K. Alexander, Matthew N. O. Sadiku, Fundamentals of Electric Circuits, McGraw-Hill Education, New York, England, 6th edition, 2017.

[61] M.K. Naik, R. Panda, "A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition," Applied Soft Computing, vol. 38, pp. 661-675, 2016.

[62] Nitin Mittal, Urvinder Singh and Balwinder SinghSohi, "Modified Grey Wolf Optimizer for Global Engineering Optimization," Applied Computational Intelligence and Soft Computing, vol. 2016, pp. 1-16, 2016.

[63] S. Mirjalili and S. Z. M. Hashim, "BMOA: Binary magnetic optimization algorithm," International Journal of Machine learning and Computing, vol. 2, no. 3, pp. 204-208, 2012.

[64] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[65] Davies Segera, Mwangi Mbuthia, and Abraham Nyete, "Particle Swarm Optimized Hybrid Kernel-Based Multiclass Support Vector Machine for Microarray Cancer Data Analysis," BioMed Research International Volume 2019, Article ID 4085725, 11 pages, 2019

Hindawi
BioMed Research International

Research Article

Particle Swarm Optimized Hybrid kernel based Multi-Class Support Vector Machine for Microarray Cancer Data Analysis

Davies Segera,[1] Mwangi Mbuthia,[2] Abraham Nyete,[3]

[1,2,3] *Department of Electrical and Information Engineering, University of Nairobi, Nairobi 30197, Kenya.*

Correspondence should be addressed to Davies Segera; davies.segera@uonbi.ac.ke

Abstract. Determining an optimal decision model is an important but difficult combinatorial task in imbalanced microarray-based cancer classification. Though the multi-class support vector machine (MCSVM) has already made an important contribution in this field, its performance solely depends on three aspects; the penalty factor C, the type of kernel and its parameters. To improve the performance of this classifier in microarray- based cancer analysis, this paper proposes PSO-PCA-LGP-MCSVM model that is based on particle swarm optimization (PSO), principal component analysis (PCA) and multi-class support vector machine (MCSVM). The MCSVM is based on a hybrid kernel i.e. linear-gaussian-polynomial (LGP) that combines the advantages of three standard kernels (linear, gaussian and polynomial) in a novel manner; where the linear kernel is linearly combined with the gaussian kernel embedding the polynomial kernel. Further, this paper proves and makes sure that the LGP kernel confirms the features of a valid kernel. In order to reveal the effectiveness of our model, several experiments were conducted and the obtained results compared between our model and other three single kernel-based models, namely, PSO-PCA-L-MCSVM (utilizing a linear kernel), PSO-PCA-G-MCSVM (utilizing a gaussian kernel) and PSO-PCA-P-MCSVM (utilizing a polynomial kernel). In comparison, two dual and two multi-class imbalanced standard microarray datasets were used. Experimental results in terms of three extended assessment metrics (F-Score, G-mean and Accuracy)

reveal the superior global feature extraction, prediction and learning abilities of this model against three single kernel-based models.

Introduction

Cancer is a disorder caused by excessive and uncontrolled cell division in a body. A total of 9.6 million people died of cancer in 2018[1]. As a matter of fact, death due to cancer can be reduced to nearly half if the cancer types are detected early and the right treatment administered in time. However, it is still a challenge for researchers to effectively diagnose cancer on the basis of morphological structure since different cancer types exhibit thin differences [2].

This challenge encourages application of data mining techniques, especially the use of gene - expression data in determining the types of cancer cells. The level of gene expression can duly indicate the activity of a gene in a body cell based on the number of messenger ribonucleic acids (mRNAs). It is well known to contain information about the disease that may be in the gene sample, which may help experts in treating or preventing the disease [3].

Though next generation sequencing (NGS) especially RNA-sequencing (RNA-Seq) are slowly replacing microarrays when analyzing and identifying complex mechanism in gene expression e.g. in the gene-expression based cancer classification problem, they are relatively expensive compared to microarrays. Since microarrays have been used for a long time, there exists robust statistical and operational methods for their processing [4-13].In addition, many significant microarray experiments have been conducted and are publicly available to the research community [37-43]. For microarrays, there exists large and well-maintained repositories that have collected these type of data for long. While the pre-processing and analysis steps of microarray data are mostly standardized, the establishment of RNA-Seq data analysis techniques are still ongoing in the field of transcriptomics. Because of these reasons, to date microarrays are still utilized in many gene-expressions based cancer classification studies as presented in the most recent survey of hybrid feature selection methods in microarray gene expression for data for cancer classification [43].

The DNA microarray technology has the capability of determining the level of thousands of genes concurrently in a given experiment, which so far has facilitated the development of cancer classification by the use of gene expression data [4-13].

Clinical decision support is the most recent application of DNA microarrays in the medical domain. This support can take the form of disease diagnosis or predicting clinical outcomes in response to a treatment. Currently, the two major areas in medicine that are drawing much attention in this regard are management of cancer and other contagious diseases [14].

With the rapid development of artificial intelligence (AI), machine-learning algorithms such as artificial neural network (ANN), support vector machine (SVM), K-nearest neighbor (KNN) , many researchers have immensely applied them in the gene-expression base cancer diagnosis. For instance, the artificial neural networks (ANN) have been proposed for the microarray gene classification due to their superior ability to map input-output structured data. Khan and Meltzer utilized the ANN in analyzing microarray gene data from patients with small round blue-cell tumours [9]. Bevilacqua and Tommasi developed an accurate classifier model based on the feed-forward ANN for estrogen receptor (ER) +/- metastasis recurrence of breast cancer tumours [20]. Chen and Cheng [19] also modeled a classifier for microarray gene data using ANN ensembles that were based on filtering of samples .In all these studies attractive classification accuracies were obtained.

Furey proposed an SVM based on a simple kernel to carry out gene expression data analysis, which turned out to perform remarkably [21]. Vanitha utilized SVM alongside mutual information gained (MI-SVM) for feature selection [11]. In his research, he used various SVM models; linear SVM, radial basis function (RBF) SVM, Quadratic SVM and Polynomial SVM. He further compared the results obtained from the proposed scheme with the k-nearest neighbor (K-NN) and ANN classifier results. Based on the obtained result, utilization of the MI-SVM obtained better results compared to K-NN and ANN, and even in some datasets, 100% accuracy was achieved.

Based on these previous researches, it is evident that SVM has already made an important contribution in the field of microarray-based cancer classification. However, many researchers have pointed out that though the SVM is a promising classifier in microarray-based cancer classification, its performance solely depends on three aspects; the penalty parameter C of this classifier, the type of kernel utilized and its parameters [22, 24, 30, 31, 32].

To improve the classification accuracy of the SVM classifier, some techniques have been presented to search for the optimal model parameters, such as the grid-search and the gradient descent [1]. Although, these approaches have proven their effectiveness in the corresponding experiments, in most cases they fall into the local optimum point easily and have a defect of low efficiency [1, 41].

Recently, some meta-heuristic techniques, such as particle swarm optimization (PSO), genetic algorithm (GA), bat algorithm (BA) and dragonfly algorithm (DA) have attained promising results when utilized in tuning SVM classifier's parameters [41]. However, most of these research has not been applied to gene-expression based cancer analysis. In addition, they only focus on SVM with a single kernel function. Though some research [22] point out that combining multiple kernel functions can achieve better performance compared to a single kernel function, little research has provided an in-depth formulation and analysis of the performance of a multi-class support vector machine (MCSVM) with a combined kernel function. Thus, there is a definite need to systematically study the complex optimization problem in the MCSVM classifier with a combined kernel applicable to gene-expression based cancer classification.

Considering PSO is easy to implement, has a few parameters to adjust, is computationally efficient compared to other optimization techniques [44] ,and existence of few studies on MCSVM classifier with combined kernels in microarray-based cancer classification, this paper proposes a novel gene-expression based cancer classification model i.e. PSO-PCA-LGP-MCSVM. This model is based on particle swarm optimization (PSO), principal component analysis (PCA) and multi-class support vector machine (MCSVM) with a novel hybrid kernel function i.e. linear-gaussian-polynomial (LGP) kernel

The objective of this research is to construct a MCSVM classifier with three different standard kernel functions (linear, gaussian and polynomial). Use PCA to reduce the dimensional complexity of the considered microarray datasets and optimize all the parameters of this model using PSO.

The overall structure of this paper takes the form of five chapters, including this introductory chapter. The remaining part of this paper proceeds as follows: A detailed presentation of the proposed model is presented in section 2. Section 3 deals with the considered cancer microarray datasets. Section 4 focusses on the experimental results and discussions. Finally, conclusions and recommendations are given in section 5.

PSO-PCA-LGP-MCSVM PRINCIPLES

2.1. Normalization

Microarray gene expressions can differ by an order of magnitude. Thus, it is necessary to normalize these data to improve the performance of subsequent microarray data analysis stages like gene selection/ feature extraction, clustering and classification [1].

In this paper, the microarray gene expressions are linearly transformed from the interval $[X_{min}, X_{max}] \rightarrow [0,1]$ uniformly utilizing equation 1 [1];

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2}$$

Where, X' is the new normalized value of the gene expression level, X is the value of the gene expression level before normalization, while X_{max} and X_{min} respectively declare the largest and least values of all the data in an attribute (gene) to be normalized.

Since the min-max normalization has the advantage of preserving exactly all the relationships among the original gene data values and does not introduce any bias [1], it is considered in this paper.

2.2. Principal component analysis (PCA)

One of the major challenges encountered in working with DNA microarray data is their high dimensionality that is coupled with a relatively small sample size. While there is a plethora of crucial information that can be derived from these large datasets, their high dimensional nature can often hide the critical information. Thus, a process that can reduce the dimensionality complexity of this type of data is required. In addition, a dimensionality reduction step will minimize errors obtained in the subsequent classification stage [1, 12, 44].

In this paper, principal component analysis (PCA) that includes the calculation of variance of proportion for eigenvector is used. The steps of this algorithm are as follows:

k) Let X'(the normalized microarray gene expression data) be the input matrix for PCA. Each row vectors of X' represent the normalized expression gene values for each of the genes.

l) Compute the mean (centroid) $\overline{X}$ of each gene j using equation 2 where the sum goes through all M samples (tissues):

$$\overline{X} = \frac{1}{M}\sum_{i=1}^{M} X'_{ij} \tag{3}$$

Where M is the number of tissues and $X'_{ij}$is gene j data.

m) Compute the covariances (degree to which the genes are linearly correlated) as per equation 3:

$$C_{kj} = \frac{1}{M-1}\sum_{i=1}^{M}(X'_{ki} - \overline{X}_k)(X'_{ji} - \overline{X}_j) \tag{4}$$

Where, C_{kj} is the covariance of gene k and gene j, M is the number of samples(tissues), X'_{ki} is the expression level of gene k in sample i, X'_{ji} is the expression level of gene j in sample i, $\overline{X}_k$ is the mean of expression levels of gene k and $\overline{X}_j$ is the mean of expression levels of gene j

n) Form a covariance matrix C using the computed covariances and transform it into a diagonal matrix as depicted in equation 4:

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{1M} \\ \vdots & \vdots & \vdots \\ C_{M1} & C_{M2} & C_{MM} \end{bmatrix} \to \begin{bmatrix} \partial_1 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & \partial_M \end{bmatrix} \tag{5}$$

The diagonal elements of the transformed matrix are the eigenvalues $\partial_1, \partial_2, \ldots., \partial_M$ which denotes the amount of variability captured along a particular new dimension.

o) Calculate corresponding eigenvectors as $\rho_1, \rho_2, \ldots\ldots \rho_M$ using equation 5:

$$\partial_k \rho_k = C\partial_k \qquad (6)$$

p) Sort the eigenvalues in descending order i.e. $\partial_1 \geq \partial_2 \geq \partial_2, \ldots \partial_{M-1} \geq \partial_M$

q) The eigenvectors corresponding to the k largest eigenvalues (where $k < M$) are the first k *principal components*

r) Select the first k *eigenvectors* via the cumulative proportion of variance (eigenvalues). The proportion of variance (PPV) for each principal component is determined as follows:

$$PPV = \frac{\partial_i}{\sum_{i=1}^{M} \partial_i} \times 100\% \qquad (7)$$

s) Form the principal component matrix P ,a matrix consisting of selected k eigenvectors that correspond to the largest k eigenvalues. Where the k eigenvectors are derived from eigenvalues that meet the criterion in equation 7;

$$\frac{\sum_{i=1}^{k} \partial_i}{\sum_{i=1}^{M} \partial_i} \times 100\% \geq 95\% \qquad (8)$$

t) Compute dimensionally reduced microarray gene expression data X'_{DimRed} using equation 8;

$$X'_{DimRed} = X' \times P \qquad (9)$$

Hence, the analysis reduces the highly dimensioned original microarray datasets to P for each sample, which are the inputs for the multi-class support vector machine (MCSVM).

To be able to measure the generalization error for each considered model, per-fold PCA was adopted. This is achieved by first conducting a separate PCA on each calibration set and then applying this transformation on the validation set. This same transformation is achieved by first subtracting the means of the calibration set from the validation set and then projecting these, data onto the principal components of the training set achieved this. The underlying assumption is that the testing and training set should be derived from the same distribution, which justifies this process.

2.3. Multi-class support vector machine (MCSVM)

The MCSVM classifier is based on Vapnik Chervonenkis (VC) dimension of the statistical learning theory and the structural risk minimization [1, 5, 7, 11, 23].

The main objective of MCSVM is to map the preprocessed, on-linear inseparable microarray gene expression data into a linear highly dimensioned manifold θ by the use of a transformation $\emptyset: \mathrm{R}^{\mathrm{N}} \rightarrow \theta$, then obtaining the optimal hyper-plane $\Psi: \psi(\mathrm{x}) = (\omega. \phi(\mathrm{x}) + \mathrm{b})$ by solving the following optimization convex problem(the soft margin problem) [23]:

$$\mathrm{MIN}(\Omega, \Xi) = \frac{1}{2}\|\Omega\|^2 + \mathrm{B}\sum_{\mathrm{I}=1}^{\mathrm{N}} \Xi_{\mathrm{I}} \qquad (10)$$

Subject to $y_i(\omega.\phi(x) + b) \geq 1 - \xi_i$ for all $1 \leq i \leq n$

Where ω is a coefficient vector of the hyper-plane in the manifold (feature space), b is the threshold value of the hyper-plane, ξ_i is a slack factor introduced for classification errors and β is a penalty factor for errors.

The parameter β controls the penalty of misclassification and its value is normally determined via cross-validation. Larger values of β normally leads to a small margin which minimizes classification errors while smaller values of β may produce a wider margin resulting to many misclassifications.

The feature space θ is highly dimensioned, so its direct computation can lead to "dimension disaster". However, since $\omega = \sum_{i=1}^{n} \delta_i\, y_i \emptyset(x_i)$, then all the operations of the support vector machine (MCSVM) in the feature space θ are only dot products. And since kernel functions i.e$G(x_i, x_{i'}) = \emptyset(x_i).\emptyset(x_{i'})$, are efficient at handling dot products, they were introduced into the SVM. This implies there is no need to know how to map the microarray gene expression data from its original space to the feature space θ. Thus, selection of a kernel and its coefficients are vital in the computational efficiency and accuracy of an MCSVM classifier model [22, 24, 30, 31, 32].

The common kernel functions that are utilized as continuous predictors include [1, 5, 22]:

4) Linear Kernel:

$$\mathrm{G}(\mathrm{X}_{\mathrm{I}}, \mathrm{X}_{\mathrm{I}'}) = \mathrm{X}_{\mathrm{I}}.\mathrm{X}_{\mathrm{I}'} \qquad (11)$$

5) Polynomial Kernel:

$$\mathrm{G}(\mathrm{X}_{\mathrm{I}}, \mathrm{X}_{\mathrm{I}'}) = (\mathrm{H} * (\mathrm{X}_{\mathrm{I}}.\mathrm{X}_{\mathrm{I}'}) + \Delta)^{\mathrm{D}} \qquad (12)$$

Where $\eta > 0$, $\delta \in \mathrm{R}$ and $\mathrm{d} \in \mathrm{Z}^{+}$

6) Gaussian kernel:

$$\mathrm{G}(\mathrm{X}_{\mathrm{I}}, \mathrm{X}_{\mathrm{I}'}) = \mathrm{EXP}\left(\frac{\|\mathrm{X}_{\mathrm{I}} - \mathrm{X}_{\mathrm{I}'}\|^2}{2\Sigma^2}\right) \qquad (13)$$

Where $\sigma > 0$

These MCSVM kernel functions can be broadly categorized as follows: local kernel functions and global kernel functions. Samples far apart have a great impact on the global kernel values while samples close to each other greatly influence the local kernel values. The linear and polynomial kernels are good examples of global kernels while the Gaussian radial basis function and the Gaussian are local kernels [22, 30, 31, 32].

Relatively speaking, the linear kernel function has a better extraction of global features from samples, the polynomial kernel has good generalization ability and the gaussian kernel (the most widely used kernel) has a good learning ability among all the single kernel functions. Thus, it is evident that utilizing a single kernel function based MCSVM classifier in a given application such as gene expression data may neither attain good learning ability, proper global feature extraction ability and a better generalization capability. In trying to overcome this hiccup, two or more kernel functions can be combined [22, 24, 30, 31, 32].

2.4. Linear-Gaussian-Polynomial MCSVM (LGP-MCSVM)

In trying to build a kernel model that has a better global feature extraction capability, good learning and prediction abilities, the work presented in this paper combines the merits of two global kernels (Linear and Polynomial) and one local kernel (Gaussian). This paper therefore proposes a novel kernel "Linear-Gaussian -Polynomial (LGP)" kernel, which is formulated as follows:

$$\mathrm{G}_{\mathrm{LGP}}(\mathrm{X}_{\mathrm{I}}, \mathrm{X}_{\mathrm{I}'}) = \mathrm{B}_1.(\mathrm{X}_{\mathrm{I}}.\mathrm{X}_{\mathrm{I}'}) + \mathrm{B}_2.\mathrm{EXP}\left(-\mathrm{B}_3.\left(\frac{(\mathrm{H}\times(\mathrm{X}_{\mathrm{I}}.\mathrm{X}_{\mathrm{I}'}) + \Delta)^{\mathrm{D}}}{2\times\Sigma^2}\right)\right) \tag{13}$$

Where $\beta_1 + \beta_2 + \beta_3 = 1, \beta \in \mathrm{R}$ and $\delta, \mathrm{d} > 0$

In this paper we utilize different values of β to mix the three standard kernels (different regions of the input space). In this case β is a vector i.e. $\beta = [\beta_1, \beta_2, \beta_3]$. Through this approach, the relative contribution of each kernel to the hybrid kernel i.e. $\mathrm{G}_{\mathrm{lgpk}}(\mathrm{x}_{\mathrm{i}}, \mathrm{x}_{\mathrm{i}'})$ can be easily varied over the input space.

The LGP kernel function takes better global feature extraction ability from the linear kernel, good prediction ability from the polynomial kernel and better learning ability from the gaussian kernel. The Mercer's theorem provides the necessary and sufficient qualifiers of a valid kernel function. It states that a kernel function is a permissible kernel if the corresponding kernel matrix is symmetric and positive semi-definite (PSD) [5, 29].

A kernel matrix can be validated that it is PSD by determining its spectrum of eigenvalues. It is important to note that a symmetric is positive definite if and only if all its eigenvalues are non-negative. Considering this, for the proposed kernel to be permissible, it must satisfy the Mercer's theorem. This validity can be proved by using the Taylor expansion for exponential function of equation 13:

$$\mathrm{G}_{\mathrm{LGP}}(\mathrm{X}_{\mathrm{I}}, \mathrm{X}_{\mathrm{I}'}) = \mathrm{B}_1.(\mathrm{X}_{\mathrm{I}}.\mathrm{X}_{\mathrm{I}'}) + \mathrm{B}_2\left(-\sum_{i=0}^{\infty} \mathrm{B}^{i}{}_{3}.\frac{(\mathrm{H}\times(\mathrm{X}_{\mathrm{I}}.\mathrm{X}_{\mathrm{I}'}) + \Delta)^{\mathrm{D.I}}}{2\times\Sigma^{2\mathrm{I}}.i!}\right) \tag{14}$$

$$\mathrm{G}_{\mathrm{LGP}}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}) = \mathrm{B}_1(\mathrm{X}_{\mathrm{I}}.\mathrm{X}_{\mathrm{I}'}) + \mathrm{B}_2\left(-1 + \sum_{i=1}^{\infty}\frac{-\mathrm{B}^{i}{}_{3}}{2\times\Sigma^{2\mathrm{I}}\,i!}((\mathrm{H}(\mathrm{X}_{\mathrm{I}}.\mathrm{X}_{\mathrm{I}'}) + \Delta)^{\mathrm{D.I}})\right) \tag{15}$$

$$\mathrm{G}_{\mathrm{LGP}}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}) = \mathrm{B}_1(\mathrm{X}_{\mathrm{I}}.\mathrm{X}_{\mathrm{I}'}) - \mathrm{B}_2 + \mathrm{B}_2\sum_{i=1}^{\infty}\frac{-\mathrm{B}^{i}{}_{3}}{2\times\Sigma^{2\mathrm{I}}\,i!}((\mathrm{H}(\mathrm{X}_{\mathrm{I}}.\mathrm{X}_{\mathrm{I}'}) + \Delta)^{\mathrm{D.I}}) \tag{16}$$

$$\mathrm{G}_{\mathrm{LGP}}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}) = \mathrm{B}_1(\mathrm{X}_{\mathrm{I}}.\mathrm{X}_{\mathrm{I}'}) - \mathrm{B}_2 + \mathrm{B}_2\sum_{i=1}^{\infty}\frac{-\mathrm{B}^{i}{}_{3}}{2\times\Sigma^{2\mathrm{I}}\,i!}.K_{Poly(i)} \tag{17}$$

$$\mathrm{G}_{\mathrm{LGP}}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}) = \mathrm{B}_1 K_{Linear} - \mathrm{B}_2 + \mathrm{B}_2\sum_{i=1}^{\infty}\frac{-\mathrm{B}^{i}{}_{3}}{2\times\Sigma^{2\mathrm{I}}\,i!}.K_{Poly(i)} \tag{18}$$

$$\mathrm{G}_{\mathrm{LGP}}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}) = \mathrm{B}_1 K_{Linear} - \mathrm{B}_2 + \mathrm{B}_2\sum_{i=1}^{\infty}\frac{-\gamma^{i}\times\mathrm{B}^{i}{}_{3}}{i!}.K_{Poly(i)} \tag{19}$$

Where $K_{Poly(i)} = (\eta(\mathrm{x_i}.\mathrm{x_{i'}}) + \delta)^{\mathrm{d}}$ and $K_{Linear} = (\mathrm{x_i}.\mathrm{x_{i'}})$ and $\gamma^{i} = \frac{1}{2*\sigma^{2\mathrm{i}}}$

From equation 19, it is evident that $\mathrm{G}_{\mathrm{LGP}}(\mathrm{x_i},\mathrm{x_{i'}})$ is a mixed kernel comprising of a weighted linear kernel, a constant β_2 and a weighted summation of polynomial kernels. Using propositions 20,21and 22 of theorem 2.20 and propositions 23 and 24 of corollary 2.21 [29], Mercer's conditions are proved to be true for the proposed kernel and hence it is a valid kernel.

Theorem 2.20. *Functions of Mercer's kernels K1 and K2 are also Mercer's kernels.*

$$\mathrm{G}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}) = \mathrm{K1}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}) + \mathrm{K2}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}) \tag{20}$$

$$\mathrm{G}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}) = \mathrm{C}.\mathrm{K1}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}),\quad for\ all\ \mathrm{C} \in \mathrm{R}^{+} \tag{21}$$

$$\mathrm{G}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}) = \mathrm{K1}(\mathrm{X}_{\mathrm{I}},\mathrm{X}_{\mathrm{I}'}) + c\quad ,for\ all\ \mathrm{C} \in \mathrm{R}^{+} \tag{22}$$

Corollary 2.21. *Functions of a Mercer kernel K1 are also Mercer's kernels.*

$$G(X_I, X_{I'}) = (K1(X_I, X_{I'}) + C)^d \qquad (23)$$
$$, for\ all\ C \in R^+\ and\ D \in N$$

$$G(X_I, X_{I'}) = EXP\left(\frac{K1(x_i, x_{i'})}{\sigma^2}\right) \qquad (24)$$
$$, , for\ all\ \sigma \in R^+$$

Since the proposed hybrid LGP kernel combines three valid Mercer's kernels i.e. linear, gaussian and polynomial kernels, it also a valid Mercer's kernel that can be used for training and classification of the multi-class support vector machine (MCSVM).

By using the proposed LGP-MCSVM, the non-linear transformation of the microarray gene sample points to get the corresponding kernel matrix so as to obtain the classification results during the training phase of the MCSVM classifier.

2.5. Particle swarm optimization (PSO)

Currently, there is no widely accepted method for optimizing these parameters. The "Grid-Search (GS)" with exponentially growing sequences of combination $\{C, \eta\}$ for the commonly utilized Gaussian kernel is often applied in microarray analysis [1, 41]. Though easy to implement, it has a low computing efficiency. In addition, optimal result of the GS can only be generated from the pre-set grid-combinations while unknown possible optimal parameters cannot be explored and discovered.

In this paper, particle swarm optimization (PSO) optimization technique is adopted to optimally search for the best parameter combinations for the considered models [41, 44]. The PSO technique is derived from the migration patterns of birds during foraging, which has a faster convergence, efficient parallel computing and a strong universality that is able to efficiently avoid local optimum [43]. In addition, the iteration velocity for its particles is largely influenced by the sum of current velocity; previous particle value, the current global optimal value and random interferences, which greatly helps, avoid the local optimal and improves the search coverage and effectiveness. In order to effectively evaluate the performance of the considered models, different values were considered for all kernel parameters within the following ranges presented in Table 1.

Table 2 presents the initial PSO parameters of each considered algorithm. In this paper, as a rule of thumb with heuristic optimization algorithms, the swarm size for each model was set to $10\times$ $variable\ size$ .More information on the PSO algorithm is presented in [41-44].

Table1: Parameters and their respective ranges

PARAMETER	RANGE

$B = [B_1, B_2, B_3]$	$0 < B_1, B_2, B_3 < 1$ and $B_1 + B_2 + B_3 = 1$
$log_2 C$	$-5 \leq log_2 C \leq 15$
Δ	$0 \leq \Delta \leq 5$
D	$2 \leq D \leq 5$
$log_2 \gamma, log_2 H$	$-15 \leq log_2 \gamma, log_2 H \leq 3$

Table 2: Initial PSO parameters setting

Parameter	Range
Maximum number of iterations	50
Inertial weight, w	1
Number of particles/Swarm size	5) PSO+L-MCSVM=10 6) PSO+G-MCSVM=20 7) PSO+P -MCSVM=40 8) PSO+LGP-MCSVM=80
Cognition learning factor, c_1	2.0
Social learning factor, c_2	2.0

2.6. PCA-PSO-LGP-MCSVM model

The main process of the proposed algorithm is outlined as follows:

14) Transforming the cancer microarray data into the right format for the SVM package.
15) Loading a cancer microarray dataset.
16) Randomly dividing the loaded microarray data into two sets: training set and testing set.
17) Initialize the PSO parameters like the population size, the maximum number of iterations, and the considered multi-class SVM parameters.
18) Adopt PSO to search for the optimal solution of particles in the global space by using 5-fold cross-validation that incorporates per fold PCA feature extraction. This process is presented below.

19) To achieve 5-fold cross-validation incorporating PCA, the following steps were followed:

    viii) For j=1 to 5 repeat steps (ii) to (vi)

    ix) Carry out PCA on data present in the remaining 4 folds to generate a loadings matrix.

    x) Transform this data (data in the remaining 4 folds i.e. calibration set) into a set of principal components (PC) scores using the first *P* components (that account for at least 95% cumulative variance) of the loadings matrix generated in step (ii).

    xi) Build a considered SVM classification model using a set of parameter values using the generated PC scores data in step (iii).

    xii) Transform the held-out test fold data (i.e. data in fold j) into a set of principal component (PC) scores using the *P* components loadings matrix retained in step (iii).

    xiii) Compute the classification accuracy of the built SVM classification model in step (iv) using the transformed test fold j data in step (v).

    xiv) For the considered parameters set, store their optimal parameter values set (i.e. a set of parameters that yields the highest classification accuracy).

20) Report optimal parameters for the considered model.

21) Carry out PCA on the whole training set data (i.e. the training set obtained in step 3) to generate a loading matrix.

22) Transform this whole training set data into a set of PC scores using the first *P* components (that account for at least 95% cumulative variance).

23) Build an optimal model for the considered SVM classification model using the optimal parameter values set obtained in step vii) using the generated PC scores data in step 9.

24) Transform the whole testing set data (i.e. the testing set obtained in step 3) into a set of principal components (PC) scores using the *P* components loadings matrix retained in step 9.

25) Compute the classification accuracy of the built optimal SVM classification model in step 8 using the transformed whole testing set data in step 9

26) Report this test classification accuracy.

The schematic diagram in Figure 1 shows the all process of the PSO-PCA-LGP-MCSVM algorithm.

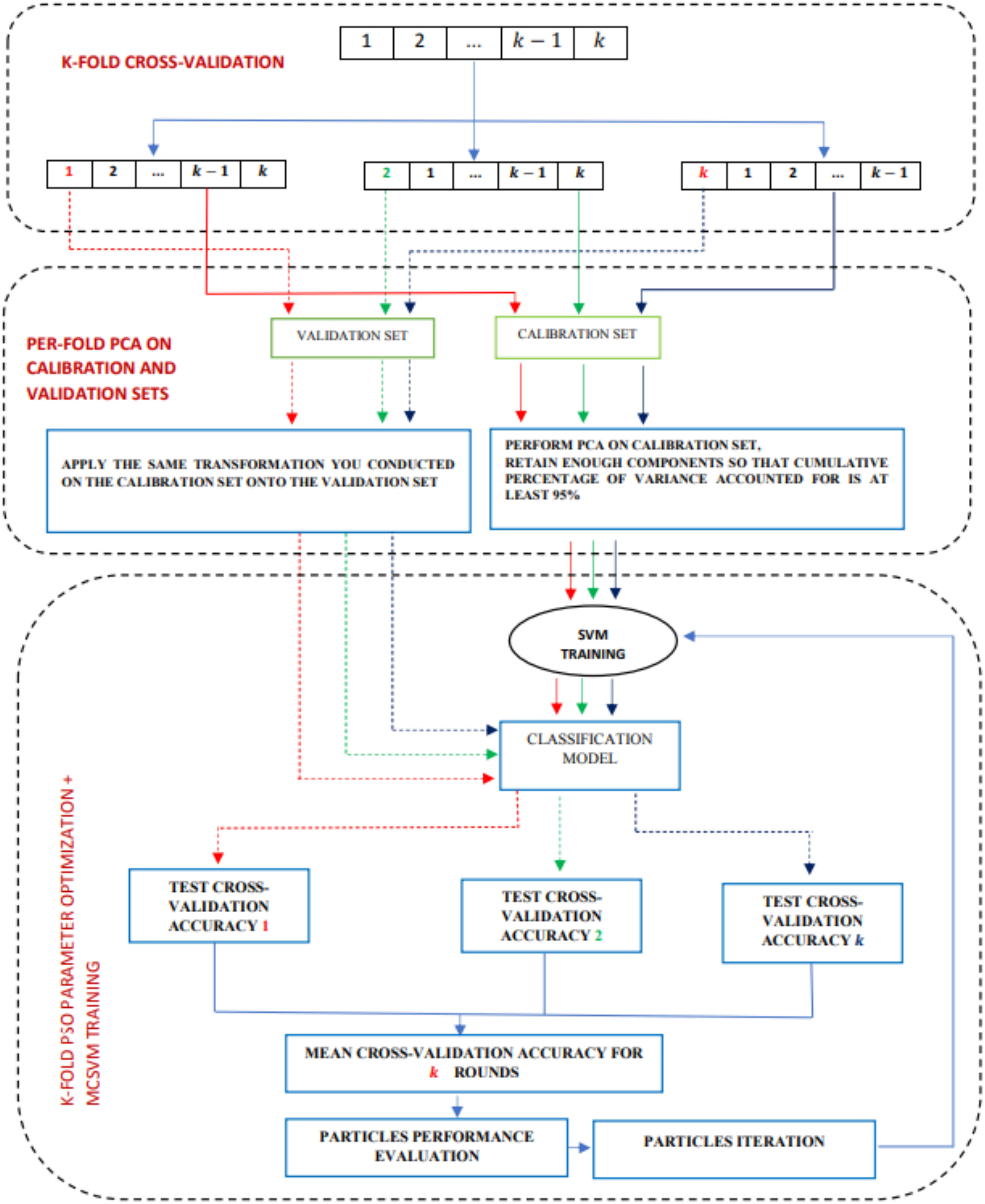

K-FOLD CROSS-VALIDATION
1
2
...
$k-1$
k
1
2
...
$k-1$
k
2
1
...
$k-1$
k
k
1
2
...
$k-1$
PER-FOLD PCA ON CALIBRATION AND VALIDATION SETS
VALIDATION SET
CALIBRATION SET
APPLY THE SAME TRANSFORMATION YOU CONDUCTED ON THE CALIBRATION SET ONTO THE VALIDATION SET
PERFORM PCA ON CALIBRATION SET, RETAIN ENOUGH COMPONENTS SO THAT CUMULATIVE PERCENTAGE OF VARIANCE ACCOUNTED FOR IS AT LEAST 95%
K-FOLD PSO PARAMETER OPTIMIZATION + MCSVM TRAINING
SVM TRAINING
CLASSIFICATION MODEL
TEST CROSS-VALIDATION ACCURACY 1
TEST CROSS-VALIDATION ACCURACY 2
TEST CROSS-VALIDATION ACCURACY k
MEAN CROSS-VALIDATION ACCURACY FOR k ROUNDS
PARTICLES PERFORMANCE EVALUATION
PARTICLES ITERATION

Figure 1: Scheme of the proposed PSO-PCA-LGP-MCSVM algorithm

It is important to mention that the whole analysis process is conducted using the LIBSVM framework in MATLAB [33,34] on Intel(R)Core (TM) i3-3240M CPU@ 3.4GHz with 12GB of RAM machine.

Performance evaluation

3.1. Microarray Datasets

To assess the performance of the proposed PSO-PCA-LGP-SVM algorithm, several experiments were conducted four publicly available datasets. Summary of all the datasets utilized in this research can be found in Table 3 and following is a brief description of each dataset.

Colon dataset [8]: contains gene expression levels obtained from DNA based microarrays. It has 62 samples; 20 normal and 40 cancerous tissue samples, each described by 2000 features.

Leukemia (AMLALL) dataset [6]: contains gene expression levels from 72 leukemia patients; 47 with Acute Lymphoblastic Leukemia (ALL) and 25 with Acute Myeloid Leukemia (AML). Each patient data is described by expression levels of 7129 probes obtained from 6817 human genes.

Stjude Leukemia dataset [7]: This data was obtained from St. Jude children's research hospital. It is divided into 6 diagnostic groups: BCR-ABL(9 patients), E2A-PBX1(18 patients), Hyper- diploid>50 (42 patients), Mixed Lineage Leukemia(MLL)(14 patients), T-cell Acute Lymphoblastic Leukemia(T-ALL)(28 patients) and TEL-Leukemia(TEL-AML1)(52 patients) and other 52 patients that could not fit into any of the outlined diagnostic groups. This dataset contains 12558 genes.

Lung Cancer dataset [13]: Contains 3312 gene data obtained from 17 people with normal lungs and 186 lung cancer patients that is classified into 5 classes: Adenocarcinomas (139 patients), Squamous Cell Lung Carcinomas(21 patients), Pulmonary Carcinoids(20 patients), Small Cell Lung Carcinomas (6 patients) and Normal Lung (17 people).

Table 3: The cancer microarray datasets utilized in this paper

Category	Dataset	Sample Size	Number of genes	Number of classes
Two-Class	AMLALL	72	7129	2
	COLON	62	2000	2
Multi-Class	STJUDE	215	12558	7
	LUNG	203	3312	5

Due to the small number of instances in the considered datasets, all the datasets were initially split into two disjoint sets: the training set and the test set. Utilizing 5-fold cross-validation, the training set was randomly divided further into 5 subsets (approximately) equal in size. Each time 4 subsets were selected as the calibration set and the remaining subset was used as the validation set. This process was repeated 5 times. Finally, the average of classification accuracy on the validation set was used as one of the evaluation metrics. It is important to point out that by using 5-fold cross-validation to dynamically divide the microarray training samples, the considered models turn out to be more stable and objective.

The percentage proportion for the calibration, validation and test sets for all the considered microarray datasets are presented in table 4.

Table 4: Percentage proportion for calibration, validation and test sets

DATASET	% PROPORTION FOR CALIBRATION SET	% PROPORTION FOR VALIDATION SET	%PROPORTION FOR TEST SET
AMLALL	61.1	15.3	23.6
COLON	58.1	14.5	27.4
STJUDE	57.7	14.4	27.9
LUNG	57.1	14.3	28.6

3.2. Performance measures for imbalanced microarray datasets

When the samples in a dataset are unevenly distributed among the classes (for instance in the case of microarray datasets), the task of classification in imbalanced domains must be defined. The majority class(es), as a result influences the data mining algorithms skewing their performances towards it [38].

Most algorithms simply compute the accuracy on the basis of the percentage of correctly samples.

However, in the case of microarrays, these results are highly deceiving since the minority classes hold minimal effects on the overall classification accuracy. Thus, a consideration of a complete confusion matrix (Table 5) must be made to obtain the classification of both positive and negative classes independently [38].

Table 5: Confusion matrix for a two-class problem

	Positive prediction	Negative prediction
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

The description in table 5 gives four baseline statistical components, where TP and FN denote the number of positive samples, which are accurately and falsely predicted, respectively, and TN and FP depict the number of negative samples that are predicted accurately and wrongly, respectively.

Two most frequently used metrics for class imbalance problem, namely F-measure and G-mean, can be regarded as functions of these four statistical components and are calculated as follows:

$$\text{F} - \text{MEASURE} = \frac{2*Recall*Precision}{(Recall+recision)} \tag{25}$$

$$\text{G} - \text{MEAN} = \sqrt{(TPR \times TNR)} \tag{26}$$

Where Precision, Recall, TPR and TNR are further defined as follows:

$$\text{Precision} = \frac{TP}{(TP+FP)} \tag{27}$$

$$\text{Recall (TPR)} = \frac{TP}{(TP+FN)} \tag{28}$$

$$\text{TNR} = \frac{TN}{(TN+FP)} \tag{29}$$

The overall classification accuracy Acc can be calculated using equation 26.

$$\text{Acc} = \frac{TP+TN}{(TP+TN+FP+FN)} \tag{30}$$

However, all these evaluation metrics are appropriate for estimating binary-class imbalance tasks. To extend them for multi-class, the following transformations should be considered [38].

$\text{G} - \text{mean}$ computes the geometric mean of all the classes' accuracies and is defined by

$$\mathrm{G-MEAN} = \left(\prod_{i=1}^{C} Acc_i\right)^{1/c} \quad (31)$$

Where Acc_i denotes the accuracy of the i^{th} class. $\mathrm{F-measure}$ can be transformed as $\mathrm{F-Score}$ and is computed using equation 32.

$$\mathrm{F-SCORE} = \frac{\sum_{i=1}^{C} \mathrm{F-MEASURE}_i}{C} \quad (32)$$

Where $\mathrm{F-measure}_i$ is calculated further using the equation 33.

$$\mathrm{F-MEASURE}_i = \frac{2\times \mathrm{PRECISION}_i \times \mathrm{RECALL}_i}{\mathrm{PRECISION}_i + \mathrm{RECALL}_i} \quad (33)$$

Acc can be transformed as depicted by equation 34.

$$Acc = \sum_{i=1}^{C} (\mathrm{Acc}_i \times \mathrm{P}_i) \quad (34)$$

Where P_i is the percentage of samples in the i^{th} class. To impartially and comprehensively assess the classification performance of the proposed model in comparison with PSO-PCA-L-MCSVM, PSO-PCA-G-MCSVM and PSO-PCA-P-MCSVM models that utilize the standard linear, gaussian and polynomial kernels respectively, the three extended measures namely $\mathrm{F-Score}$, $\mathrm{G-mean}$ and Acc which are described in (32),(31) and (34) respectively.

Results and Discussions

The experimental results for the 4 classification models on the 4 microarray datasets are reported in Tables 6, 7 and 8, where the best result in each dataset is highlighted in bold and the worst is italicized.

From Tables 6 to 8 the following observations can be made

(i) Lung and STJUDE datasets are slightly sensitive to the class imbalance while Colon and AMLALL are not, as shown by the difference between Accuracy and G-mean values. An accuracy slightly lower than the G-mean values imply that the MCSVM is affected by the imbalanced class distribution. This is largely attributed by a large number of True negatives (TNs) recorded achieved by all the models when analyzing both the Lung and STJUDE datasets.

Table 6: Accuracy of all considered models on the four microarray datasets, where bold represents the best result and the italics denotes the worst in each column respectively

MODELS	COLON	LUNG	AMLALL	STJUDE
PSO+L-MCSVM	*0.7647*	0.9596	**0.9412**	0.9422

PSO+P -MCSVM	0.8235	*0.9592*	*0.8235*	*0.9395*
PSO+G-MCSVM	0.8235	0.9608	**0.9412**	0.9572
PSO+LGP-MCSVM	**0.8824**	**0.9729**	**0.9412**	**0.9603**

Table 7: F-Score of all considered models on the four microarray datasets, where bold represents the best result and the italics denotes the worst in each column respectively

MODELS	COLON	LUNG	AMLALL	STJUDE
PSO+L-MCSVM	*0.7572*	0.9246	0.9328	0.7870
PSO+P -MCSVM	0.8211	*0.7524*	*0.7733*	*0.6831*
PSO+G-MCSVM	0.8211	0.9306	**0.9377**	0.8477
PSO+LGP-MCSVM	**0.8712**	**0.9586**	**0.9377**	**0.8989**

Table 8: G-mean of all considered models on the four microarray datasets, where bold represents the best result and the italics denotes the worst in each column respectively

MODELS	COLON	LUNG	AMLALL	STJUDE
PSO+L-MCSVM	*0.7676*	0.9791	**0.9412**	0.9557
PSO+P -MCSVM	0.8235	*0.7524*	*0.8235*	*0.9512*
PSO+G-MCSVM	0.8235	0.9792	**0.9412**	0.9661
PSO+LGP-MCSVM	**0.8824**	**0.9861**	**0.9412**	**0.9709**

(ii) The hybrid kernel boosted the classification performance of the multi-class on three datasets i.e. Colon, Lung and STJUDE. These promotions are better portrayed by the F-Score and G-Mean metrics, which are used to evaluate a balance level of classification results. However, a tie is reported for the AMLALL dataset. This implies that though the complementary characteristics of the three standard kernels i.e. linear, Gaussian and polynomial in the proposed hybrid linear-gaussian-polynomial (LGP) kernel may improve the multi-class support vector machine classifier's classification ability on most microarray datasets, datasets a single suitable kernel is sufficient.

(iii) Of all the considered models, the PSO-PCA-P-MCSVM reported the least performance in all the considered metrics for all the four datasets. However, it is important to note that a promising kernel can be obtained if we embed into the exponential kernel.

In summary, compared with single-kernel-based models (i.e. PSO-PCA-L-MCSVM, PSO-PCA-G-MCSVM and PSO-PCA-P-MCSVM), the proposed PSO-PCA-LGP-MCSVM model that is based on a hybrid linear-gaussian-polynomial (LGP) kernel with a better global feature extraction ability, good prediction ability and better learning ability, has an attractive classification ability in cancer diagnosis using both imbalanced dual and multiclass microarray datasets. Moreover, due to excellent global searching ability of the particle swarm optimization, it can effectively optimize the hybrid kernel based MCSVM when solving a wider range of classification problems.

Conclusion

Techniques to choose or construct suitable kernel functions, and optimally tune its parameters for MCSVM has received a considerable and critical attention in imbalanced microarray-based cancer diagnosis. A novel classification model, PSO-PCA-LGP-MCSVM, that is based on MCSVM with a hybrid kernel i.e. linear-gaussian-polynomial (LGP), is proposed in this paper. The LGP kernel combines the advantages of three standard kernels i.e. linear, gaussian and polynomial kernels in a novel manner where the linear kernel is linearly combined with a polynomial kernel that is embedded into a gaussian kernel. Using PSO to optimally tune the LGP kernel based MCSVM resulted into better generalization, learning and predicting ability as evidenced by the promising results in terms three extended measures F-Score, G-mean and

Accuracy irrespective of imbalanced binary or multi-class microarray datasets. The performance of the proposed model was compared with those of 3 models i.e. PSO-PCA-L-MCSVM, PSO-PCA-G-MCSVM and PSO-PCA-P-MCSVM that are based on single linear, gaussian and polynomial kernels respectively and the experimental results show that the proposed model is superior to the three single-kernel based models. This reflects the good practical value of the proposed model in the field of microarray based cancer diagnosis, which can also be extended more applications of medical diagnostic classification to explore its potential.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Funding Statement

This work is fully supported by the African Development Bank (AfDB), through the Ministry of Education, Kenya Support for Capacity Building.

Supplementary Materials

The results presented in Tables 6 to 8 are based on confusion matrices attached as supplementary materials where Figure 1, Figure 2, Figure 3 and Figure 4 represents the confusion matrices obtained when the trained PSO-PCA-L-MCSVM, PSO-PCA-G-MCSVM, PSO-PCA-P-MCSVM and PSO-PCA-LGP-MCSVM models were evaluated using the Colon ,LUNG, AMLALL and STJUDE test set samples respectively .

References

[1] Adiwijaya, Untari N. Wisesty, E. Lisnawati, A. Aditsania and Dana S. Kusumo, "Dimensionality reduction using Principal Component Analysis for Cancer Detection based on Microarray Data Classification,"

Journal of Computer Science, vol. 14, pp. 1521-1530, 2018.

[2] Peng Q, Shakoor A Sun S, "A Kernel-Based Multivariate Feature Selection Method for Microarray Data Classification," PLOS ONE, vol. 9, no. 7, pp. 1-12, July 2014.

[3] A. Osareh and B. Shadgar, "Microarray Data Analysis for Cancer Classification," 5th International Symposium on Health Informatics and Bioinformatics (HIBIT), vol. 9, pp. 459-468, 2010.

[4] Nicola L, Talbot C. Cawley GC, "Gene selection in cancer classification using sparse logistic regression with Bayesian regularization," Bioinformatics , vol. 22, no. 19, 2006.

[5] M., & Moattar, M. H. Mollaee, "A novel feature extraction approach based on ensemble feature selection and modified discriminant independent component analysis for microarray data classification," Biocybernetics and Biomedical Engineering, vol. 36, no. 3, pp. 521-529, 2016.

[6] Slonim DK, Tamayo P, Huard C, Gaasenbeek M, et al Golub TR, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," Science 531–537, vol. 286, pp. 531-537, 1999.

[7] Ross ME, Shurtleff SA, Williams WK, Patel D, et al Yeoh EJ, "Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling," Cancer Cell, vol. 1, pp. 133-143, 2002

[8] Alon,U., Barkai, N., Notterman,D., Gish, K., Ybarra,S., Mark, D.,and Levine, A, "Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays," Proceedings of National Academy of Sciencesof the Unitewd States of America, vol. 96, pp. 531–537, 1999.

[9] Wei JS, Ringner M, Saal LH, Ladanyi M, et al. Khan J, "Classification and diagnostic preFchengdiction of cancers using gene expression profiling and artificial neural networks," Nature Medicine, vol. 7, pp. 673–679, 2001.

[10] Staunton JE, Silverman LB, Pieters R, de Boer ML, et al. Armstrong SA,

"Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia," Nature Genetics 30:., vol. 30, pp. 41–47, 2002.

[11] C.D.A., Devaraj, D. and Venkatesulu, M Vanitha, "Gene Expression Data Classification Using Support Vector Machine and Mutual Information-based Gene Selection," Procedia Computer Science, vol. 47, pp. 13-21, 2015.

[12] Adiwijaya and Arie Ardiyanti Suryani Adiyasa Nurfalah,"Cancer Detection Based On Microarray Data Classification Using PCA and ModifiedBack Propagation," Far East Journal of Electronics and Communications, vol. 16, no. 2, pp. 269 - 281, June 2016.

[13] Richards WG, Staunton J, Li C, Monti S, et al. Bhattacharjee A, "Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses," Proceedings Of the National Academy Of Sciences Of the United States Of America, vol. 98, pp. 13790–13795, 2001.

[14] Gautam B. Simgh, Fundamentals of Bioinformatics and Computational Biology: Methods and Exercises in Matlab, 1st ed. New York, United States of America: Springer, 2015.

[15] Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, 1st ed. London, England: The MIT Press, 2012.

[16] M.D., Ph.D Simone Mocellin, Microarray Technology and Cancer Gene Profiling. New York, United States of America: Springer science, 2007.

[17] Zhengzhi Wang Lingyun Zou, "Microarray Gene Expression Cancer Diagnosis Using Multiclass Support Vector Machines," 1st International Conference on Bioinformatics and Biomedical Engineering, pp. 260-263, 2007.

[18] Vanitha V.Bhuvaneswari, "Classification of Microarray Gene Expression Data by Gene Combinations using Fuzzy Logic MGC-FL," International Journal of Computer Science, Engineering and Applications (IJCSEA) , vol. 2, no. 4, pp. 79-98, August 2012.

[19] Huijuan Lu, Mingyi Wang and Cheng Fang Wutao Chen, "Gene Expression Data Classification Using Artificial Neural Network Ensembles Based on Samples Filtering," International Conference on Artificial Intelligence and Computational Intelligence, vol. 1, pp. 626 – 628, 2009.

[20] G. Mastronardi, F. Menolascina, A. Paradiso ,S. Tommasi V. Bevilacqua, "Genetic Algorithms and Artificial Neural Networks in Microarray Data Analysis: a Distributed Approach ," Engineering Letters Special Issue Bioinformatics, vol. 13, no. 3, pp. 335–343, 2006..

[21] Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D. Furey T.S., "Support vector machine classification and validation of cancer tissue samples using microarray expression data," Bioinformatics, vol. 16, pp. p.906–914, October 2000.

[22] Abdolreza Safari ,Saeid Homayouni Saeid Niazmardi, "A Novel Multiple Kernel Learning Framework for Multiple Feature Classification," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 8, pp. 3734 - 3743, August 2017.

[23] Neamat El-Gayar ,Iman A. El-Azab Eman Ahmed, "Support Vector Machine ensembles using features distribution among subsets for enhancing microarray data classification," in 2010 10th International Conference on Intelligent Systems Design and Applications, Cairo, 2010, pp. 1242-1246.

[24] Amit Ganatra Hetal Bhavsar, "Radial Basis Polynomial(RBPK): A Generalized Kernel for Support Vector Machine," International Journal of Computer Science and Information Security(IJCSIS), vol. XIV, no. 4, pp. 296-315, aPRIL 2016.

[25] C.J. Lin C. C. Chang, "LIBSVM: A Library for Support Vector Machine," ACM Transactions on Intelligent Systems and Technology, vol. II, no. 3, pp. 1-27, 2011..

[26] Chih-Chung Chang, Chih-Jen Lin Chih-Wei Hsu, A practical Guide to Support Vector Classification, May 19,2016.

[27] C.K. Verma,D. Namita Srivastava Rabia Aziz, "Dimension reduction methods for microarray data: a review

," AIMS Bioengineering, vol. IV, no. 2, pp. 179-197, March 2017

[28] Franz-Josef Müller, Martin Zenke , Andreas Schuppert Michael Lenz, "Principal components analysis and the reported low intrinsic dimensionality of gene expression microarray data," Scientific Reports, vol. 6, no. 25696, pp. 1-11, June 2016.

[29] Ralf Herbrich, Learning Kernel Classifiers: Theory and Algorithms (Adaptive Computation and Machine Learning) , 1st ed. Cambridge, England: The MIT Press, 2002.

[30] Zichun Ding, Cuicui Guo, Zhe Li, Hongxia Xia Huazhu Song, "Research on Combination Kernel Function of Support Vector Machine," in International Conference on Computer Science and Software Engineering, 2008, pp. 838-841.

[31] Zhao Yue,Hou Yun-tao,LI Yun-lu Wang Anna, "A Novel Construction of SVM Compound Kernel Function," in International Conference on Logistics Systems and Intelligent Management (ICLSIM), Harbin, China, 2010, pp. 1462-1465.

[32] Asa Ben, Cheng Soon, Soren Sonnenburg, Gunna Ratsch, "Support Vector Machines and Kernels for Computational Biology," PLoS Computational Biology, vol. 4, no. 10, pp. 1-10, October 2008.

[33] Huang Dong, Gao Jian, "Parameter Selection of a Support Vector Machine, Based on a Chaotic Particle Swarm Optimization Algorithm," Cybernetics and Information Technologies, vol. 15, no. 3, pp. 140-148, 2015..

[34] Saber M. Elsayed, Ruhul A. Sarker and Efrén Mezura-Montes, "Particle Swarm Optimizer for Constrained Optimization," in IEEE Congress on Evolutionary Computation, Cancún, México, 2013, pp. 2703-2711.

[35] Zhicheng Qu, Qingyan Li, Lei Yue, "Improved Particle Swarm Optimization for Constrained Optimization," in International Conference on Information Technology and Applications, Chengdu, China, November 2013, pp. 244-247.

[36] Kang Hu, Guo-Li Zhang, Bo Xiong, "An Improved Particle Swarm Algorithm For Constrained Optimization Problem," in International Conference on Machine

Learning and Cybernetics (ICMLC), Chengdu, China, November 2018, pp. 393-398.

[37] Maolong Xi, Jun Sun, Li Liu, Fangyun Fan, and Xiaojun Wu, "Cancer Feature Selection and Classification Using a Binary Quantum-Behaved Particle Swarm Optimization and Support Vector Machine" in Hindawi Computational and Mathematical Methods in Medicine, 2016, pp. 1-9.

[38] Hualong Yu, Shufang Hong, Xibei Yang, Jun Ni, Yuanyuan Dan, and Bin Qin, " Recognition of Multiple Imbalanced Cancer Types Based on DNA Microarray Data Using Ensemble Classifiers" in Hindawi BioMed Research International, 2013, pp. 1-13.

[39] Risky Frasetio Wahyu Pratamaa, Santi Wulan Purnamia,Santi Puteri Rahayua, " Boosting Support Vector Machines for Imbalanced Microarray Data" in Elsevier Procedia Computer Science, 2018, pp. 174-183.

[40] Ahmed Bir-Jmel , Sidi Mohamed Douiri, and Souad Elbernoussi, " Gene Selection via a New Hybrid Ant Colony Optimization Algorithm for Cancer Classification in High-Dimensional Data" in Hindawi Computational and Mathematical Methods in Medicine, 2019, pp. 1-20.

[41] Mohd Nadhir Ab Wahab, Samia Nefti-Meziani, Adham Atyabi , " A Comprehensive Review of Swarm Optimization Algorithms," .PLoSONE10(5), pp. 1-36, 2015.

[42] Alaa Tharwat, “Classification assessment methods," in Applied Computing and Informatics, 2018, pp. 1-11.

[43] Nada Almugren, Hala Alshamlan, "A Survey on Hybrid Feature Selection methods in Microarray Gene Expression Data for Cancer Classification," in IEEE Access, Chengdu, China, November Vol 7, 2019, pp. 78533-78548.

[44] Duygu Kaya, "Optimization of SVM Parameters with Hybrid CS-PSO Algorithms for Parkinson’s Disease in LabVIEW Environment," in Hindawi Parkinson’s Disease, 2019, pp. 1-10.

SOURCE CODES

```
%-------------------------------------------------------------------------%
%  Binary Excited Grey Wolf Optimization (BGWO) source codes         %
%                                                                         %
%  Programmer: Davies Rene Segera
%
%                                                                         %
%  E-Mail: davies.segera@uonbi.ac.ke %
%-------------------------------------------------------------------------%

function [Alpha_score,Alpha_acc,nfeat]=jBEGWO2_ver2(feat,label,N,T)
%---Inputs-----------------------------------------------------------------
% feat:  features
% label: labelling
% N:     Number of wolves
% T:     Maximum number of iterations
%---Outputs----------------------------------------------------------------
% sFeat: Selected features
% Sf:    Selected feature index
% Nf:    Number of selected features
% curve: Convergence curve
%--------------------------------------------------------------------------

% Objective function
fun=@jFitnessFunction2;
% Number of dimensions
D=size(feat,2);
% Initial Population
X=zeros(N,D);
fit=zeros(1,N);
for i=1:N
  for d=1:D
    if rand() > 0.5
      X(i,d)=1;
    end
  end
end
% Fitness
for i=1:N
%   [fit(i),acc(i)]=fun(feat,label,X(i,:));
  [fit(i),acc(i)]=jwrapperSVM3(feat,label,X(i,:));
end
% Sort fitness
[~,idx]=sort(fit);
% Update alpha, beta & delta wolves
Xalpha=X(idx(1),:);
Xbeta=X(idx(2),:);
Xdelta=X(idx(3),:);
Falpha=fit(idx(1));
Accalpha=acc(idx(1));
Fbeta=fit(idx(2));
Accbeta=acc(idx(2));
Fdelta=fit(idx(3));
```

```
Accdelta=acc(idx(3));
Fworst=max(fit);

a_max=2;

% Pre
curve=inf;
t=1;
% figure(1);
% clf; axis([1 100 0 0.5]);
% xlabel('Number of Iterations');
% ylabel('Fitness Value'); title('Convergence Curve'); grid on;
%---Iterations start-------------------------------------------------------
while t <= T
  % Coefficient decreases linearly from 2 to 0 Eq(17)
%   a=2-2*(t/T);
  MyBase=(T-t)/T;
  for i=1:N
      tau=abs((fit(i)-
(1/3)*(Falpha+Fbeta+Fdelta))/((1/3)*(Falpha+Fbeta+Fdelta)-Fworst));
      a=a_max*(MyBase)^tau;
    for d=1:D
      % Parameter C Eq(16)
      C1=2*rand();
      C2=2*rand();
      C3=2*rand();
      % Compute Dalpha, Dbeta & Ddelta Eq(22-24)
      Dalpha=abs(C1*Xalpha(d)-X(i,d));
      Dbeta=abs(C2*Xbeta(d)-X(i,d));
      Ddelta=abs(C3*Xdelta(d)-X(i,d));
      % Parameter A Eq(15)
      A1=2*a*rand()-a;
      A2=2*a*rand()-a;
      A3=2*a*rand()-a;
      % Compute X1, X2 & X3 Eq(19-21)
      X1(i,d)=Xalpha(d)-A1*Dalpha;
      X2(i,d)=Xbeta(d)-A2*Dbeta;
      X3(i,d)=Xdelta(d)-A3*Ddelta;
%       % Update wolf Eq(18)
%       Xn=(X1+X2+X3)/3;
%       % Sigmoid function Eq(37)
%       TF=1/(1+exp(-10*(Xn-0.5)));
%       % Position update Eq(36)
%       if TF >= rand()
%         X(i,d)=1;
%       else
%         X(i,d)=0;
%       end
    end
    X1B(i,:)=Vec2Bin(X1(i,:));
    X2B(i,:)=Vec2Bin(X2(i,:));
    X3B(i,:)=Vec2Bin(X3(i,:));

    if nnz(X1B(i,:))>0
%         FX1B(i)=fun(feat,label,X1B(i,:));
        [FX1B(i),~]=jwrapperSVM3(feat,label,X1B(i,:));
    else
```

```
        FX1B(i)=Inf;
    end

    if nnz(X2B(i,:))>0
%         FX2B(i)=fun(feat,label,X2B(i,:));
        [FX2B(i),~]=jwrapperSVM3(feat,label,X2B(i,:));
    else
        FX2B(i)=Inf;
    end

    if nnz(X3B(i,:))>0
%         FX3B(i)=fun(feat,label,X3B(i,:));
        [FX3B(i),~]=jwrapperSVM3(feat,label,X3B(i,:));
    else
        FX3B(i)=Inf;
    end

    FVec=[FX1B(i),FX2B(i),FX3B(i)];
    BMatrix=[X1B(i,:);X2B(i,:);X3B(i,:)];
    % Sort fitness
    [~,idx]=sort(FVec);
    X(i,:)=BMatrix(idx(1),:);

  end


  for i=1:N
   if nnz(X(i,:))>0
    % Fitness
%     fit(i)=fun(feat,label,X(i,:));
%     [fit(i),acc(i)]=fun(feat,label,X(i,:));
    [fit(i),acc(i)]=jwrapperSVM3(feat,label,X(i,:));
    % Update alpha, beta & delta
    if fit(i) < Falpha
      Falpha=fit(i);
      Xalpha=X(i,:);
      Accalpha=acc(i);
    end
    if fit(i) < Fbeta && fit(i) > Falpha
      Fbeta=fit(i);
      Xbeta=X(i,:);
      Accbeta=acc(i);
    end
    if fit(i) < Fdelta && fit(i) > Falpha && fit(i) > Fbeta
      Fdelta=fit(i);
      Xdelta=X(i,:);
      Accdelta=acc(i);
    end
  else
  fit(i)=Inf;
  end
  end
  Fworst=max(fit);

  curve(t)=Falpha;
  % Plot convergence curve
%   pause(0.000000001); hold on;
%   CG=plot(t,Falpha,'Color','r','Marker','.'); set(CG,'MarkerSize',5);
  t=t+1;
end
% Select features based on selected index
Pos=1:D;
Sf=Pos(Xalpha==1);
Alpha_acc=Accalpha;
nfeat=length(Sf);
sFeat=feat(:,Sf);
Alpha_score=Falpha;
end




%---------------------------------------------------------------------------%
%  Binary Adaptive Cuckoo Search -Intensification Dedicated Grey Wolf
Optimization (BACSIDGWO) source codes         %
%                                                                           %
%  Programmer: Davies Rene Segera
%
%                                                                           %
%  E-Mail: davies.segera@uonbi.ac.ke %
%---------------------------------------------------------------------------%

function
[Alpha_score,Alpha_acc,nfeat]=jBACSIDGWO(feat,label,n,N_IterTotal,pa)

%number of variables
nd=size(feat,2);
% initialize alpha, beta, and delta_pos
Alpha_nest=zeros(1,nd);
Alpha_score=inf; %change this to -inf for maximization problems
Alpha_acc=0;

Beta_nest=zeros(1,nd);
Beta_score=inf; %change this to -inf for maximization problems
Beta_acc=0;

Delta_nest=zeros(1,nd);
Delta_score=inf; %change this to -inf for maximization problems
Delta_acc=0;

% nest=zeros(n,nd);

% Random initial solutions
% Lower bounds
Lb=0*ones(1,nd);
% Upper bounds
Ub=1*ones(1,nd);


% Random initial solutions

nest=zeros(n,nd);
```

```
X1=zeros(n,nd);
X2=zeros(n,nd);
X3=zeros(n,nd);


for i=1:n
  for d=1:nd
    if rand() > 0.5
      nest(i,d)=1;
    end
  end
end


% Get the current best
fitness=10^10*ones(n,1);
accuracy=[];
%
[Alpha_score,Alpha_nest,Beta_score,Beta_nest,Delta_score,Delta_nest,nest,fi
tness,accuracy]=get_best_nest(Alpha_score,Alpha_nest,Beta_score,Beta_nest,D
elta_score,Delta_nest,nest,nest,fitness,feat,label,accuracy);
[Alpha_score,Alpha_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score
,Delta_nest,Delta_acc,nest,fitness,accuracy]=get_best_nest(Alpha_score,Alph
a_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score,Delta_nest,Delta
_acc,nest,nest,fitness,feat,label,accuracy);
iter=0;

%% Starting iterations
 for iter=1:N_IterTotal


[new_nest1,new_nest2,new_nest3]=GWO(fitness,nest,Alpha_nest,Beta_nest,Delta
_nest,Lb,Ub,iter,N_IterTotal);


[Alpha_score,Alpha_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score
,Delta_nest,Delta_acc,nest,fitness,accuracy]=get_best_nest(Alpha_score,Alph
a_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score,Delta_nest,Delta
_acc,nest,new_nest1,fitness,feat,label,accuracy);

[Alpha_score,Alpha_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score
,Delta_nest,Delta_acc,nest,fitness,accuracy]=get_best_nest(Alpha_score,Alph
a_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score,Delta_nest,Delta
_acc,nest,new_nest2,fitness,feat,label,accuracy);

[Alpha_score,Alpha_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score
,Delta_nest,Delta_acc,nest,fitness,accuracy]=get_best_nest(Alpha_score,Alph
a_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score,Delta_nest,Delta
_acc,nest,new_nest3,fitness,feat,label,accuracy);



new_nest=get_cuckoos_acs(nest,Alpha_nest,fitness,Lb,Ub,iter,N_IterTotal);

[Alpha_score,Alpha_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score
,Delta_nest,Delta_acc,nest,fitness,accuracy]=get_best_nest(Alpha_score,Alph
a_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score,Delta_nest,Delta
_acc,nest,new_nest,fitness,feat,label,accuracy);


     new_nest=empty_nests(nest,Lb,Ub,pa);

[Alpha_score,Alpha_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score
,Delta_nest,Delta_acc,nest,fitness,accuracy]=get_best_nest(Alpha_score,Alph
a_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score,Delta_nest,Delta
_acc,nest,new_nest,fitness,feat,label,accuracy);

     % Select features based on selected index
     nfeat=size(find(round(Alpha_nest)==1),2);

end
end




%% Find the current best nest
function
[Alpha_score,Alpha_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score
,Delta_nest,Delta_acc,nest,fitness,accuracy]=get_best_nest(Alpha_score,Alph
a_nest,Alpha_acc,Beta_score,Beta_nest,Beta_acc,Delta_score,Delta_nest,Delta
_acc,nest,newnest,fitness,feat,label,accuracy)
% Evaluating all new solutions
for j=1:size(nest,1)
            newnest(j,:)=Vec2Bin(newnest(j,:));

          if nnz(newnest(j,:))>0

             %fnew=jFitnessFunction(feat,label,newnest(j,:));
%               [fnew,acc]=svm(feat,label,newnest(j,:));
              [fnew,acc]=jwrapperSVM3(feat,label,newnest(j,:));
%            [fnew,acc]=svm2(X1,Y1,round(newnest(j,:)));
             if fnew<=fitness(j)
                fitness(j)=fnew;
                accuracy(j,:)=acc;
                nest(j,:)=newnest(j,:);
             end
            if fitness(j) < Alpha_score
               Alpha_score=fitness(j);
               Alpha_nest=nest(j,:);
               Alpha_acc=accuracy(j,:);
            end

            if fitness(j) < Beta_score && fitness(j) > Alpha_score
               Beta_score=fitness(j);
               Beta_nest=nest(j,:);
               Beta_acc=accuracy(j,:);
            end
            if fitness(j) < Delta_score && fitness(j) > Alpha_score &&
fitness(j) > Beta_score
               Delta_score=fitness(j);
               Delta_nest=nest(j,:);
               Delta_acc=accuracy(j,:);
           end


          else
           fitness(j)=Inf;
         end





end

end





function [fit,Acc]=jwrapperSVM3(feat,label,X)
kfold=12;
    Model =
fitcsvm(feat(:,[find(X==1)]),label,'Standardize',true,'KernelFunction','rbf
','KernelScale','auto');
%     CVSVMModel =crossval(SVMModel,'KFold',kfold);

    % Perform cross-validation
    C=crossval(Model,'KFold',kfold);
    % Accuracy for each fold
    Afold=1*(1-kfoldLoss(C,'mode','individual'));
    Acc=mean(Afold);
    ErrorRate=1-Acc;
    D=size(X,2);
    Pos=1:D;
    Nfalpha=length(Pos(X==1));
%     fit=-Acc;
%     fit=(0.2*(Nfalpha/D))-0.8*Acc; %proposed fitness function
    Alpha=0.9;
    fit=(Alpha*ErrorRate)+((1-Alpha)*(Nfalpha/D)); %proposed fitness
function
%     fit=-Acc;
%      fit=(0.15*(Nfalpha/D))-0.85*Acc; %proposed fitness function
end


function BinVec=Vec2Bin(Vec)
    Cstep=1./(1+exp(-10*(Vec-0.5)));
    Val=rand(1,size(Vec,2));
%     Val2=rand(1,size(Vec,2));
%     Val3=rand(1,size(Vec,2));
%     for i=1:1:size(Vec,2)
%         Val(i)=jCrossover3(Val1(i),Val2(i),Val3(i));
%     end
%     Sampled=randsample(Val,size(Vec,2));
```

```
    Bstep=(Cstep>=Val);
    BinVec=Bstep;
end


% Adaptive Cuckoo Search without levy flight
function nest=get_cuckoos_acs(nest,best,fit,Lb,Ub,iter,max_iter)
% Levy flights
n=size(nest,1);

% Sort fitness
[~,idx]=sort(fit);
FAlpha_nest=fit(idx(1));
FBeta_nest=fit(idx(2));
FDelta_nest=fit(idx(3));

worst_fit=max(fit);
stepsize_max=1;
MyBase=(max_iter-iter)/max_iter;
for j=1:n
    s=nest(j,:);
    tau=abs((fit(j)-
(1/3)*(FAlpha_nest+FBeta_nest+FDelta_nest))/((1/3)*(FAlpha_nest+FBeta_nest+
FDelta_nest)-worst_fit));

    %     stepsize=stepsize_max*(iter/max_iter)^tau;
%     stepsize=stepsize_max*(1-exp(-iter*6/max_iter))^tau;
    stepsize=stepsize_max*(1-(MyBase)^tau);
%     stepsize=stepsize_max*(1-tau*exp(-(iter/max_iter)));
    s=s+stepsize.*randn(size(s));
    nest(j,:)=simplebounds(s,Lb,Ub);
end


function
[X1,X2,X3]=GWO(fit,nest,Alpha_nest,Beta_nest,Delta_nest,Lb,Ub,iter,max_iter
)

   X1=zeros(size(nest,1),size(nest,2));
   X2=zeros(size(nest,1),size(nest,2));
   X3=zeros(size(nest,1),size(nest,2));

   % Sort fitness
   [~,idx]=sort(fit);
   FAlpha_nest=fit(idx(1));
   FBeta_nest=fit(idx(2));
   FDelta_nest=fit(idx(3));
   a_max=1;
   worst_fit=max(fit);
   MyBase=(max_iter-iter)/max_iter;
   for i=1:size(nest,1)
       tau=abs((fit(i)-
(1/3)*(FAlpha_nest+FBeta_nest+FDelta_nest))/((1/3)*(FAlpha_nest+FBeta_nest+
FDelta_nest)-worst_fit));
%        a=a_max*(exp(-iter*10/max_iter))^tau;
```

```
       a=a_max*(MyBase)^tau;
%        a=a_max*(1-(iter/max_iter))^tau;
%        a=(1-(1-(iter/max_iter)*exp(-(iter*tau/max_iter))));
       for j=1:size(nest,2)
             % Parameter C Eq(16)
             C1=2*rand();
             C2=2*rand();
             C3=2*rand();
             % Compute Dalpha, Dbeta & Ddelta Eq(22-24)
             Dalpha=abs(C1*Alpha_nest(j)-nest(i,j));
             Dbeta=abs(C2*Beta_nest(j)-nest(i,j));
             Ddelta=abs(C3*Delta_nest(j)-nest(i,j));

             %compute a

             % Parameter A Eq(15)
             A1=2*a*rand()-a;
             A2=2*a*rand()-a;
             A3=2*a*rand()-a;
             % Compute X1, X2 & X3 Eq(19-21)
             X1(i,j)=Alpha_nest(j)-A1*Dalpha;
             X2(i,j)=Beta_nest(j)-A2*Dbeta;
             X3(i,j)=Delta_nest(j)-A3*Ddelta;
%              new_nest(i,j)=(X1(i,j)+X2(i,j)+X3(i,j))/3;
       end

%        new_nest(i,:)=simplebounds( new_nest(i,:),Lb,Ub);
       X1(i,:)=simplebounds( X1(i,:),Lb,Ub);
       X2(i,:)=simplebounds( X2(i,:),Lb,Ub);
       X3(i,:)=simplebounds( X3(i,:),Lb,Ub);

   end



end


%% Replace some nests by constructing new solutions/nests
function new_nest=empty_nests(nest,Lb,Ub,pa)
% A fraction of worse nests are discovered with a probability pa
n=size(nest,1);
% Discovered or not -- a status vector
K=rand(size(nest))>pa;

% In the real world, if a cuckoo's egg is very similar to a host's eggs,
then
% this cuckoo's egg is less likely to be discovered, thus the fitness
should
% be related to the difference in solutions.  Therefore, it is a good idea
% to do a random walk in a biased way with some random step sizes.
%% New solution by biased/selective random walks
stepsize=rand*(nest(randperm(n),:)-nest(randperm(n),:));
new_nest=nest+stepsize.*K;
for j=1:size(new_nest,1)
```

```
    s=new_nest(j,:);
     new_nest(j,:)=simplebounds(s,Lb,Ub);

end
end




%-------------------------------------------------------------------------%
%  Particle Swarm Optimization-Pricipal Component Analysis-Linear Gaussian
Polynomial-MultiClass Support Vector Machine source codes         %
%                                                                         %
%  Programmer: Davies Rene Segera
%
%                                                                         %
%  E-Mail: davies.segera@uonbi.ac.ke %
%-------------------------------------------------------------------------%

% READING MICROARRAY DATASET
clear;clc;close all;
trainData=xlsread('AMLALL_Nature_Dataset.xls','trainData');
trainLabels=xlsread('AMLALL_Nature_Dataset.xls','trainLabels','A:A');
testData=xlsread('AMLALL_Nature_Dataset.xls','testData');
testLabels=xlsread('AMLALL_Nature_Dataset.xls','testLabels','A:A');
% END

nfold = 5;
sheetname = {'CrossValidationIndices',};
filename = 'AMLALL_Nature_Dataset.xls';          % can be named without
'.xls'
[num,txt,raw] = xlsread(filename,'CrossValidationIndices');
if isempty(num)
   cv = cvpartition(trainLabels, 'kfold',nfold);          %# Statistics
toolbox
   indices = zeros(size(trainLabels));
   for b=1:nfold
       indices(cv.test(b)) = b;
   end
   [raw{:, :}]=deal(NaN);
   xlswrite(filename, raw, 'CrossValidationIndices');
   xlswrite(filename,indices,'CrossValidationIndices');
else
    indices=xlsread(filename,'CrossValidationIndices');
end




%  SHUTTING DOWN OF ALL WARNINGS
warning('off','all');
warning;
% END
Variance=0.95;
LGPdims=0;
```

```
nVar=8;            % Number of Decision Variables
VarSize=[1 nVar];   % Size of Decision Variables Matrix
%       Cost   sigma   gamma   coef  degree  beta1   beta2   beta3
VarMin=[2^-20  2^-20   2^-20   0     2       0.0001  0.0001  0.0001]; %
Lower Bound of Variables
VarMax=[2^20   2^20    2^20    5     3       1       1       1];   % Upper
Bound of Variables
%% PSO Parameters
MaxIt=50;      % Maximum Number of Iterations
nPop=80;        % Population Size (Swarm Size)

% Constriction Coefficients
phi1=2.05;
phi2=2.05;
phi=phi1+phi2;
chi=2/(phi-2+sqrt(phi^2-4*phi));
w=chi;          % Inertia Weight
wdamp=1;        % Inertia Weight Damping Ratio
c1=chi*phi1;    % Personal Learning Coefficient
c2=chi*phi2;    % Global Learning Coefficient

% Velocity Limits
VelMax=0.1*(VarMax-VarMin);
VelMin=-VelMax;

%% Initialization
empty_particle.Position=[];
empty_particle.Cost=[];
empty_particle.Velocity=[];
empty_particle.Best.Position=[];
empty_particle.Best.Cost=[];
particle=repmat(empty_particle,nPop,1);
GlobalBest.Cost=-inf;
tic
for i=1:nPop

    % Initialize Position
    particle(i).Position=unifrnd(VarMin,VarMax,VarSize);

particle(i).Position(6:end)=particle(i).Position(6:end)/sum(particle(i).Pos
ition(6:end));
    particle(i).Position(4:5)=round(particle(i).Position(4:5));
    % Initialize Velocity
    particle(i).Velocity=zeros(VarSize);
    %% Conduct 5-fold cross-validation using particle(i).Position
    cv = cvpartition(trainLabels, 'kfold',nfold);          %# Statistics
toolbox
    indices = zeros(size(trainLabels));
    for b=1:nfold
        indices(cv.test(b)) = b;
    end
    for a=1:nfold
        %SEGMENT DATA INTO FOLDS
        %disp(['fold: ' num2str(a)]);
        testIdx = (indices == a);
        trainIdx = ~testIdx;
        %CLASSES
        labels = unique(trainLabels(trainIdx));
        numLabels = numel(labels);
        %CARRY OUT PCA
        mn = mean(trainData(trainIdx,:));
        train_out = bsxfun(@minus,trainData(trainIdx,:),mn); % substract
mean
        test_out = bsxfun(@minus,trainData(testIdx,:),mn);
        [coefs,scores,variances] = princomp(train_out,'econ'); % PCA
        pervar = cumsum(variances) / sum(variances);
        var_frac=Variance;
        LGPdims = max(find(pervar < var_frac));
        train_out = train_out*coefs(:,1:LGPdims); % dims - keep this many
dimensions
        test_out = test_out*coefs(:,1:LGPdims); % result is in train_out
and test_out
        %PRE-COMPUTED LINEAR KERNEL
        XTrainsize = size(train_out,1);
        K_TrainLinear = train_out*train_out';
        Gamma_K_TrainLinear = particle(i).Position(3).*K_TrainLinear;
        K_TrainPolynomial = Gamma_K_TrainLinear;
        for count = 2:particle(i).Position(5)
            K_TrainPolynomial = K_TrainPolynomial.*(particle(i).Position(4)
+ Gamma_K_TrainLinear);
        end

K_TrainLinearRBFPolynomial=((particle(i).Position(6).*K_TrainLinear)+(parti
cle(i).Position(7).*exp(-
(particle(i).Position(8)*particle(i).Position(2)).*K_TrainPolynomial)));
        K1_TrainLinearRBFPolynomial = [(1:XTrainsize)',
K_TrainLinearRBFPolynomial];
        cmd = ['-s 0 -q -b 1 -t 4  -c ',num2str(particle(i).Position(1))];
        % BUILD NEW MODEL
        model =svmtrain(trainLabels(trainIdx,:),
K1_TrainLinearRBFPolynomial, cmd);
        XTestsize = size(test_out,1);
        K_TestLinear = test_out*train_out';
        Gamma_K_TestLinear = particle(i).Position(3).*K_TestLinear;
        K_TestPolynomial = Gamma_K_TestLinear;
        for count = 2:particle(i).Position(5)
            K_TestPolynomial = K_TestPolynomial.*(particle(i).Position(4) +
K_TestPolynomial);
        end

K_TestLinearRBFPolynomial=((particle(i).Position(6).*K_TestLinear)+(particl
e(i).Position(7).*exp(-
(particle(i).Position(8)*particle(i).Position(2)).*K_TestPolynomial)));
        K1_TestLinearRBFPolynomial=[(1:XTestsize )',
K_TestLinearRBFPolynomial];
        % EVALUATE WITH TEST DATA
        [~, accuracy, ~] = svmpredict(trainLabels(testIdx,:),
K1_TestLinearRBFPolynomial, model,'-b 1 -q');
        % Store fold accuracy
        LinearAcc(a) = accuracy(1);
    end
    %Average Cross-validation accuracy
    AverageLinearAcc = mean(LinearAcc);
    % Evaluation
    particle(i).Cost=AverageLinearAcc;
    % Update Personal Best
    particle(i).Best.Position=particle(i).Position;
    particle(i).Best.Cost=particle(i).Cost;
    % Update Global Best
    if particle(i).Best.Cost>GlobalBest.Cost

        GlobalBest=particle(i).Best;

    end



end

BestCost=zeros(MaxIt,1);
%% PSO Main Loop
for it=1:MaxIt

    for i=1:nPop

        % Update Velocity
        particle(i).Velocity = w*particle(i).Velocity ...
            +c1*rand(VarSize).*(particle(i).Best.Position-
particle(i).Position) ...
            +c2*rand(VarSize).*(GlobalBest.Position-particle(i).Position);

        % Apply Velocity Limits
        particle(i).Velocity = max(particle(i).Velocity,VelMin);
        particle(i).Velocity = min(particle(i).Velocity,VelMax);

        % Update Position
        particle(i).Position = particle(i).Position + particle(i).Velocity;

particle(i).Position(6:end)=particle(i).Position(6:end)/sum(particle(i).Pos
ition(6:end));
        particle(i).Position(4:5)=round(particle(i).Position(4:5));
        % Velocity Mirror Effect
        IsOutside=(particle(i).Position<VarMin |
particle(i).Position>VarMax);
        particle(i).Velocity(IsOutside)=-particle(i).Velocity(IsOutside);

        % Apply Position Limits
        particle(i).Position = max(particle(i).Position,VarMin);
        particle(i).Position = min(particle(i).Position,VarMax);

        %% Conduct 5-fold cross-validation using particle(i).Position
        cv = cvpartition(trainLabels, 'kfold',nfold);          %#
Statistics toolbox
        indices = zeros(size(trainLabels));
        for b=1:nfold
            indices(cv.test(b)) = b;
        end
        for a=1:nfold
            %SEGMENT DATA INTO FOLDS
            %disp(['fold: ' num2str(a)]);
            testIdx = (indices == a);
            trainIdx = ~testIdx;
            %CLASSES
            labels = unique(trainLabels(trainIdx));
            numLabels = numel(labels);
            %CARRY OUT PCA
            mn = mean(trainData(trainIdx,:));
            train_out = bsxfun(@minus,trainData(trainIdx,:),mn); %
substract mean
            test_out = bsxfun(@minus,trainData(testIdx,:),mn);
            [coefs,scores,variances] = princomp(train_out,'econ'); % PCA
            pervar = cumsum(variances) / sum(variances);
            var_frac=Variance;
            LGPdims = max(find(pervar < var_frac));
            train_out = train_out*coefs(:,1:LGPdims); % dims - keep this
many dimensions
            test_out = test_out*coefs(:,1:LGPdims); % result is in
train_out and test_out
            %PRE-COMPUTED LINEAR KERNEL
            XTrainsize = size(train_out,1);
            K_TrainLinear = train_out*train_out';
            Gamma_K_TrainLinear = particle(i).Position(3).*K_TrainLinear;
            K_TrainPolynomial = Gamma_K_TrainLinear;
            for count = 2:particle(i).Position(5)
                K_TrainPolynomial =
K_TrainPolynomial.*(particle(i).Position(4) + Gamma_K_TrainLinear);
            end

K_TrainLinearRBFPolynomial=((particle(i).Position(6).*K_TrainLinear)+(parti
cle(i).Position(7).*exp(-
(particle(i).Position(8)*particle(i).Position(2)).*K_TrainPolynomial)));
            K1_TrainLinearRBFPolynomial = [(1:XTrainsize)',
K_TrainLinearRBFPolynomial];
            cmd = ['-s 0 -q -b 1 -t 4  -c
',num2str(particle(i).Position(1))];
            % BUILD NEW MODEL
            model =svmtrain(trainLabels(trainIdx,:),
K1_TrainLinearRBFPolynomial, cmd);
            XTestsize = size(test_out,1);
            K_TestLinear = test_out*train_out';
            Gamma_K_TestLinear = particle(i).Position(3).*K_TestLinear;
            K_TestPolynomial = Gamma_K_TestLinear;
            for count = 2:particle(i).Position(5)
                K_TestPolynomial =
K_TestPolynomial.*(particle(i).Position(4) + K_TestPolynomial);
            end

K_TestLinearRBFPolynomial=((particle(i).Position(6).*K_TestLinear)+(particl
e(i).Position(7).*exp(-
(particle(i).Position(8)*particle(i).Position(2)).*K_TestPolynomial)));
            K1_TestLinearRBFPolynomial=[(1:XTestsize )',
K_TestLinearRBFPolynomial];
            % EVALUATE WITH TEST DATA
            [~, accuracy, ~] = svmpredict(trainLabels(testIdx,:),
K1_TestLinearRBFPolynomial, model,'-b 1 -q');
            % Store fold accuracy
            LinearAcc(a) = accuracy(1);
         end
        %Average Cross-validation accuracy
        AverageLinearAcc = mean(LinearAcc);
        particle(i).Cost=AverageLinearAcc;
        % Update Personal Best
        if particle(i).Cost>particle(i).Best.Cost

                particle(i).Best.Position=particle(i).Position;
                particle(i).Best.Cost=particle(i).Cost;

                % Update Global Best
                if particle(i).Best.Cost>GlobalBest.Cost

                    GlobalBest=particle(i).Best;

                end

       end

    end

    BestCost(it)=GlobalBest.Cost;

    disp(['Iteration ' num2str(it) ': Best Cost = '
num2str(BestCost(it))]);

    w=w*wdamp;

end
BestSol = GlobalBest;
toc
LGP_MCSVM_TRAINING_TIME=toc;
tic
train_out=trainData; % save original data
test_out=testData;
mn = mean(train_out);
train_out = bsxfun(@minus,train_out,mn); % substract mean
test_out = bsxfun(@minus,test_out,mn);
[coefs,scores,variances] = princomp(train_out,'econ'); % PCA
pervar = cumsum(variances) / sum(variances);
var_frac=Variance;
LGPdims = max(find(pervar < var_frac));
train_out = train_out*coefs(:,1:LGPdims); % dims - keep this many
dimensions
test_out = test_out*coefs(:,1:LGPdims); % result is in train_out and
test_out
XTrainsize = size(train_out,1);
K_TrainLinear = train_out*train_out';
Gamma_K_TrainLinear = BestSol.Position(3).*K_TrainLinear;
K_TrainPolynomial = Gamma_K_TrainLinear;
for count = 2:BestSol.Position(5)
```

```
    K_TrainPolynomial = K_TrainPolynomial.*(BestSol.Position(4) +
Gamma_K_TrainLinear);
end
K_TrainLinearRBFPolynomial=((BestSol.Position(6).*K_TrainLinear)+(BestSol.P
osition(7).*exp(-
(BestSol.Position(8)*BestSol.Position(2)).*K_TrainPolynomial)));
K1_TrainLinearRBFPolynomial = [(1:XTrainsize)',
K_TrainLinearRBFPolynomial];
cmd = ['-s 0 -q -b 1 -t 4  -c ',num2str(BestSol.Position(1))];
% BUILD NEW MODEL
model =svmtrain(trainLabels, K1_TrainLinearRBFPolynomial, cmd);
testsetsize = size(test_out,1);
XTestsize = size(test_out,1);
K_TestLinear = test_out*train_out';
Gamma_K_TestLinear = BestSol.Position(3).*K_TestLinear;
K_TestPolynomial = Gamma_K_TestLinear;
for count = 2:BestSol.Position(5)
    K_TestPolynomial = K_TestPolynomial.*(BestSol.Position(4) +
K_TestPolynomial);
end
K_TestLinearRBFPolynomial=((BestSol.Position(6).*K_TestLinear)+(BestSol.Pos
ition(7).*exp(-
(BestSol.Position(8)*BestSol.Position(2)).*K_TestPolynomial)));
K1_TestLinearRBFPolynomial=[(1:XTestsize )', K_TestLinearRBFPolynomial];
[LGPSVMpredict_label, LGPSVMAccuracy, prob] = svmpredict(testLabels,
K1_TestLinearRBFPolynomial, model);
toc
LGP_MCSVM_TESTING_TIME=toc;
%% CONFUSION MATRIX FOR THE LINEAR-BASED SVM
ConfMat1 = confusionmat(testLabels,LGPSVMpredict_label);
isLabels = unique(testLabels);
nLabels = numel(isLabels);
[n,p] = size(testData);
% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(LGPSVMpredict_label,isLabels);
oofLabelMat1 = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
oofLabelMat1(idxLinear) = 1; % Flags the row corresponding to the class
[~,grpY] = ismember(testLabels,isLabels);
YMat1 = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMat1(idxLinearY) = 1;

%% END
 plotconfusion(YMat1,oofLabelMat1,'LGP Kernel-Multi-Class SVM Model');
```