# HYBRID DIMENSIONALITY REDUCTION MODEL FOR CLASSIFICATION OF RIBONUCLEIC ACID SEQUENCING MALARIA VECTOR DATASET

**BY**

**AROWOLO, MICHEAL OLAOLU**

**(18PGCD000016)**

**MAY, 2021**

# HYBRID DIMENSIONALITY REDUCTION MODEL FOR CLASSIFICATION OF RIBONUCLEIC ACID SEQUENCING MALARIA VECTOR DATASET

**Ph.D THESIS**

**BY**

**AROWOLO, MICHEAL OLAOLU**

**(18PGCD000016)**

A THESIS SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF DOCTOR OF PHILOSOPHY (PH.D.) IN COMPUTER SCIENCE IN THE DEPARTMENT OF COMPUTER SCIENCE, COLLEGE OF PURE AND APPLIED SCIENCE, LANDMARK UNIVERSITY. OMU-ARAN, NIGERIA.

**MAY, 2021**

# DECLARATION

I, Micheal, Olaolu AROWOLO, a Doctor of Philosophy (Ph.D.) student in the Department of Computer Science, College of Pure and Applied Sciences, Landmark University, Omu-Aran, hereby declare that this thesis entitled "Hybrid Dimensionality Reduction Model for Classification of Ribonucleic Acid Sequencing Malaria Vector Dataset", submitted by me is based on my original work. Any material(s) obtained from other sources or work done by any other persons or institutions have been duly acknowledged.

**Micheal Olaolu, AROWOLO (18PGCD000016)**

---

Signature & Date

# CERTIFICATION

This is to certify that this thesis has been read and approved as meeting the requirements of the Department of Computer Science, Landmark University, Omu-Aran, Nigeria, for the Award of Doctor of Philosophy (Ph.D.) Degree.

Prof. A.A. Adebiyi
**Supervisor** Signature and Date

Dr. M.O. Adebiyi
**(Co- Supervisor)** Signature and Date

Dr. M.O. Adebiyi
**(Head of Department)** Signature and Date

Prof. S.O. Olabiyisi
**(External Examiner)** Signature and Date

# DEDICATION

This thesis is dedicated to God Almighty, to my loving family and friends.

# ACKNOWLEDGEMENTS

Firstly, I wish to express my most profound gratitude and appreciation to my supervisors, Prof. Ayodele A. Adebiyi, and Dr Marion O. Adebiyi. They are great mentors that have inspired me with valuable advice, boundless generosity, outstanding knowledge, lessons, guidance, ongoing support, constructive criticism, and persistent encouragement, invaluable comments and advice, through the accomplishment of my thesis development. It has been an honor and a pleasure to have experienced the privilege to be tutored by leading intellectuals, who have exposed me to the basics of computer science, critical analysis of scientific issues, and the art of neat technical writing. I sincerely thank them for being the kind of supervisors every student need, for being astute, helpful, keen, and inspirational. The perfect role models for the pursuit and the unsurpassed imaginable top academics to supervise a determined development study. What more can I say better than to ask for God's blessings and favour to be on them in all their endeavors.

I Thank Landmark University, Omu-aran for their financial sustenance and for providing access to research resources. I want to express gratitude all my colleagues for all their sustenance and acquaintance through the journey. My sincere thanks to my past supervisors and staff of the Department of Computer Science, Kwara State University, Malete and Al-Hikmah University, Ilorin, for their technical support and help to harmonize numerous milestones by presenting helpful feedback, Thank you Dr. Yakub Saheed.

I am blessed with significant numbers of wonderful family and friends. Incredibly, my adored parents, Mr and Mrs M. O. Arowolo, who have unreservedly supported me at every stage of my life, have always encouraged me to overcome various challenges in

life, and cherish every moment towards cultivating a meaningful life. A special thanks to my siblings Olatoye and Ayoola for being there all the time.

Thank God, I have been blessed with so many wonderful people who have played impactful roles in making my journey a reality, they have enriched my life. I am extremely lucky to have great people surrounding me, with an endless list. Thank you all for your prayers, well wishes, and thoughtfulness. I appreciate an celebrate you all for been a catalyst I needed to the changes in my life.

My sincere and elite thanks to: My spiritual mentor Pastor Habuh-Rajan Kenneth, you have been profoundly needed for my growth. Prof. Gbolagade Kazeem, Dr. Isiaka Rafiu and Dr. Abdulsalam Sulaiman O., you all built my research skills, thank you for believing and mentoring me. Special appreciations to; Prof. Okeyinka A.E for his fatherly guide, Dr. Gbadamosi B., for his stewardship. My sincere gratitude to Prof. Aremu Charity for been a wonderful mother and the Dean of the School of Postgraduate Studies Landmark University, Thanks to Prof. Osemwegie O., for all the supports through the journey. The staff of The Department of Computer Science, Landmark University, Omu-aran, Kwara State University, Malete, and Al-Hikmah University, Ilorin.

My sincere appreciations to; the Ojuolape family, Olarewaju Adeola, Omole Tomileye, Ilesanmi Olawale and The Monday Prayer Group COZA, Ojo Ayorinde, Ahmed Bolaji, Oguayansi Innocent, Ogidan Akinyemi, Salawu Damola, Dr. Saheed Yakub, Oyewale Oloyede, Salaudeen Dhikrillah, Jaji Adeola, Ojulari Sodiq, Olayiwola Idris, Akanni Sabur, Akeem Kadri, Mrs. Bukola Balogun, Dr. Ajao Falilat, Mrs. Shakirat Yusuf, Martin Mavis, Mrs Ogundokun Oluwaseun, Mr. Olawepo Samuel, Ms. Peace Ayegba, Ms. Ayoola Joyce, Mr. Igbekele Emmanuel, Mr. Asani O., Mr. Akeem Femi Kadri, all

my friends and well-wishers as the list is endless. I thank them for their prayers, affections, friendship and all the help provided.

I profoundly appreciate all researchers, authors, originators, theorists and all great individuals who have remarkably paid, made profound and valuable input to delighted features of life researches. They have greatly contributed to the success of this thesis; may the good lord richly bless you all.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ALGORITHMS

# LIST OF ABBREVIATIONS AND SYMBOLS

| | |
|---|---|
| **ACO** | Ant Colony Optimization |
| **AdaBoost** | Adaptive Boosting |
| **ANOVA** | Analysis of Variance |
| **Bagging** | Bootstrap Aggregating |
| **CBFS** | Correlated Based Feature Selection |
| **CCA** | Canonical Component Analysis |
| **CDC** | Centers for Disease Control |
| **CSUMI** | Component Selection Using Mutual Information |
| **DEG** | Differential Expressed Genes |
| **DM** | Data Mining |
| **DNA** | Deoxyribonucleic acid |
| **DT** | Decision Tree |
| **EBI** | European Bioinformatics |
| **FN** | False Negative |
| **FP** | False Positive |
| **GA** | Genetic Algorithm |
| **GA-O** | Genetic Algorithm Optimization |
| **GO** | Gene Ontology |
| **GUI** | Graphical User Interface |
| **ID3** | Iterative Dichotomized 3 |

| | |
|---|---|
| ***K-NN*** | K-nearest Neighbors |
| **LLE** | Locally Linear Embedding |
| **ML** | Machine Learning |
| **NCBI** | National Center for Biotechnology Information |
| **NGS** | Next Generation Sequencing |
| **ICA** | Independent Component Analysis |
| **K-PCA** | Kernel-Principal Component Analysis |
| **LDA** | Linear Discriminant Analysis |
| **L-SVM** | Linear Support Vector Machine |
| **MATLAB** | Matrix Laboratory |
| **PCA** | Principal Component Analysis (PCA) |
| **PLS** | Partial Least Square |
| **PPI** | Protein-Protein Interaction |
| **RNA** | Ribonucleic acid |
| **RNA-Seq** | Ribonucleic acid Sequencing |
| **SOM** | Self-Organizing Map |
| **SVM** | Support Vector Machine |
| **SVM-RFE** | Support Vector Machine Recursive Feature Elimination |
| **SVM-RBF** | Support Vector Machine Radial Basis Function |
| **TP** | True Positive |
| **TN** | True Negative |
| **ZIFA** | Zero Inflated Factor Analysis |
| **ZINB** | Zero Inflated Negative Binomial |

# ABSTRACT

Malaria is a life-threatening disease caused by plasmodium falciparum parasite and spread to people from infected mosquitoes. Gene expression data analysis is an essential procedure that reveals critical genes responsible for the biological processes involved in the infection and treatment of malaria in humans. Ribonucleic Acid Sequencing (RNA-Seq) is the technology that generates profiles of transcriptional data. This data is fundamental to a variety of scientific and clinical research and applications. The RNA-Seq data, in its raw form, is however blighted by noise, redundancy and other limitations associated with high dimensional data, thus making classification of genes challenging due to *"curse of dimensionality"* and becomes too computationally expensive for high dimensional data.

Numerous approaches have been proposed to address the problem of *"curse of dimensionality"*. For instance, several dimensionality reduction, clustering and classification techniques have been suggested for analyzing RNA-Seq data. While these techniques detect interesting features in high dimensional data effectively, it is difficult to identify the relevant features of genes as there are inherent orthogonal problems, causing reductions to maximize its variances and making hidden correlation difficult. Essential information hidden in higher dimensions have been ignored, with some data loss and making classification output insufficient. The aim of this study is to overcome the limitations related to high dimensional data by introducing an optimized hybrid dimensionality reduction approach to better uncover relevant features for enhancing classification accuracy.

This study involved, two hybrid dimensionality reduction techniques, experimented using the Anopheles gambiae dataset. They include an Optimized Genetic Algorithm

(GA-O) and Principal Component Analysis (PCA) - (GA-O+PCA), and GA-O with Independent Component Analysis (ICA) - (GA-O+ICA). The low-dimensional data generated were then classified using the Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), Decision Tree and Ensemble classifiers. Experimental results showed that (GA-O+ICA) using Ensemble classifier outperformed the other techniques with a 93% accuracy. To validate the performance of the proposed work, other approaches conducted yielded distinguishing performances of the classification accuracy with GA-O+ICA+SVM 91.7%, GA-O+ICA+KNN 90%, and GA-O+PCA+DT 80% accuracies. This technique outperformed many existing methods and is thus very useful in significantly improving the performances of classification techniques. This study develops an enhanced approach in terms of computation, the obtained results are easily interpreted and can be used for the classification of other procedures and ailments.

**Keywords: RNA-Seq; Genetic Algorithm Optimization; Principal Component Analysis; Independent Component Analysis; Mosquito Anopheles; Machine Learning, Prediction; Support Vector Machine; Decision Tree; K-Nearest Neighbor, Ensemble.**

# CHAPTER ONE

## 1.0 INTRODUCTION

### 1.1. Background of the Study

Significant aspect of studies in Bioinformatics are influenced by issues relating to Deoxyribonucleic Acids (DNA) and Ribonucleic acids (RNA) profiling, as well as genetics and genomic epidemiology. Works on genomics entail the generation, analysis, and interpretation of enormous amounts of data, structured as a 3-D representation of protein-protein association (Usman et al., 2017). Practitioners use the results of these analyses in predicting, detecting, and understanding ailments, as well as getting insight into probable designs of both prophylactic and therapeutic vaccines (Liu et al., 2020).

Microarray gene expression technology has been deployed widely in the generation of a sequence of samples of genes. Microarray technologies are microscopic slides containing thousands of prearranged sequences of samples of DNA, RNA, genes, protein, or tissues among others that signify the human genomes (Guia et al., 2018). A lot of works have been done in microarray to help in identifying human diseases, through the classification of microarray data into a standard form of samples (Sheela & Rangarajan, 2018). In recent times, however, Ribonucleic Acid Sequencing (RNA-Seq) has been used as an alternative to microarray technology in computing gene expression data on diseases, such as; cancer, malaria infections, typhoid, and so on (Rao et al., 2019). RNA-Seq is an advanced, effective system for gene expression description of organisms which deploys the potentials of Next Generation Sequencing (NGS) knowledge. The RNA-Seq is generally more

reliable and preferable to the microarray technology because of the reduced amount of noise in the data and the vivid insight it provides on transcriptional features (Zararsız et al., 2017). RNA-Seq is capable of having a variety of applications, such as; determining narrative proofs, identifying, and calculating variations.

Perhaps owing to the devastating rate of fatalities caused by malaria, several studies have been suggested over the past decade on the generation and investigation of gene expression data on Anopheles of various species, using RNA-Seq technology (Lee et al., 2018). The objective of these efforts are to eliminate or drastically reduce the occurrences and transmission of malaria infection in endemic areas, especially the sub-Sahara Africa, where it is most prevalent (Bonizzoni et al., 2015). While there are different species of malaria vectors, the most pervasive carrier of malaria in West Africa is the *Anopheles gambiae* (*Ag*) (Hien et al., 2017). The analysis and gene expression of the *Anopheles gambiae* reveals the genetic and molecular properties that offer insights into the management, prevention, treatment and control of the malaria parasite (Jiang et al., 2014). Informative sequence variation, synthesis detection, and classifications, based on gene expression can help discover, differentiate genetic models, and forecast results from the tremendous amount of gene expression record that may form in a single path (Witten, 2011).

Gene expression data generated by RNA-Seq are often high dimensional, due to their volume and structure. For instance, RNA-Seq datasets needed for human interpretation and gene transcription of diseases are generated in trillions and stored using varying mediums (Prathusha & Jyothi, 2017). These data contain noises, redundancy, and other limitations associated with high dimensional data. The computation efforts needed for processing such data tend to *explode* exponentially with increasing dimensionality. This phenomenon refers

to the "*curse of dimensionality*" in handling high-dimensional data, a variety of issues arise when classifying, arranging, and evaluating high-dimensional data that does not exist in lower dimensions. When processing and arranging data in high-dimensions, phenomena that does not arise in low-dimensional settings emerges (Chattopadhyay et al., 2019). Lower dimensional data, however, increase the possibility of representing and retaining some meaningful properties of the original high dimensional data. It is, therefore, imperative to carry out the process of dimension reduction, as a pre-requisite to other forms of techniques such as classification (Hira & Gillies, 2015). Dimensionality reduction is a helpful and vital method. It endeavors to identify, lessen, distinguish, and show the set of recurrent data by altering a high dimensional data into lesser collection of data dimensions. Additionally, it also identifies the parsimonious, but important set of variables which help improve the classification process of ailments detection (Nguyen & Holmes, 2019). Dimensionality reduction may entail either Feature Selection, Feature Extraction (Hira & Gillies, 2015).

Different machine learning procedures are proposed in the literature, for analyzing RNA-Seq data enhancement. These techniques involve the dimensionality reduction phase, which includes feature selection and feature extraction. The classification phase uses algorithms for instance Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest, and others (Dagliyan et al., 2011; Luo et al., 2019; Susmi, 2016; Chen et al., 2016; Verma et al., 2018; Zahoor & Zafar, 2020). Some up-to-date feature selection and feature extraction include Analysis of Variance (ANOVA), Chi-Square, Genetic Algorithms, Partial Least Square (PLS), Principal Component Analysis (PCA), Canonical Component Analysis (CCA), Independent Component Analysis (ICA), and so on (Xia,

2020). Efforts have been made in the literature to improve on the performances of these techniques, to effectively enhance their applications for predictions and classification of gene expression data. Documented experimental results; however, show that there is still room for improvement (Duncan et al., 2020; Kumar, 2014; Zahoor & Zafar, 2020).

This study carried out, an optimized hybrid dimensionality reduction procedure, for classifying RNA-Seq dataset of malaria vector, Anopheles gambiae. The technique improved the genetic algorithm by introducing an optimization routine to enhance the prediction of mosquitoes' insecticidal compound classification. The method evaluates and compares its performance with state-of-the-art procedures using system of measurement such as accuracy, sensitivity, specificity, precision, recall, and f-score.

## 1.2. Statement of the Problem

Clinical data, especially on public health issues such as malaria are generally voluminous and can be said to be high dimensional. These data, like other high-dimensional data, suffer from problems such as complexity, recurrent inconsistency and *"curse of dimensionality"*. It is therefore hard to determine knowledge and deduce useful information from such non-structured and complicated sets of data (Parva et al., 2017). For instance, RNA-Seq gene expression data are noisy. They contain redundant features, thus making it challenging to understand correlations amongst large records of gene information existing in the data. (Kong et al., 2008; Luecken & Theis, 2019). High dimensionality is a challenge in analyzing RNA-Seq data, especially when correlations among variables are complex. It results in problems such as singularity, overfitting, increase in computational costs and so on (Hira & Gillies, 2015). RNA-Seq technology needs efficient approaches to for the

enormous amount of collected data as well as useful tools to extract meta-data and information (Han et al., 2015).

Development of several dimensionality reduction techniques for RNA-seq data exists. Researchers have suggested approaches such as Zero Inflated Factor Analysis (ZIFA), Clustering and Dimensionality reduction models, among other methods (Pierson & Yau, 2015; Zhu et al., 2015; Lachmann et al., 2018). However prevailing approaches are incapable of modelling raw count of RNA-seq data and time-consuming, conducting a huge number of cells ($for\ instance\ n > 500$) (Sun et al., 2019). Feature extraction and selection techniques have their limitations. Feature extraction suffers from loss of data interpretability and transformation, while feature selection algorithms suffer from overfitting, and training time (Hira & Gillies, 2015). Grouping of feature selection and feature extraction to form a general model will enhance finding an optimal subset of the features (Li et al., 2008; Nisar & Tariq, 2016; Hira & Gillies, 2015). It is essential to discover an optimum subset of features among the genes, utilized clinically to reduce the data complexity (Dagliyan et al., 2011).

Hence, there is a need to develop a hybrid dimensionality reduction model to progress the performance of RNA-Seq data classification for prediction of insecticidal compounds against malaria transmission.

## 1.3. Justification for the Study

Malaria infection in Africa is a scourge. It is a public health that requires major intervention. Several investigators have carried out abundant investigations, existing machine learning approaches are not satisfactory enough to tackle malaria transmission in human, especially in Africa (Gachelin et al., 2018). One reason adduced to this poor performance of machine learning task on gene expression is the high dimensional structure of the data; thus, dimensionality reduction techniques have been proposed (Alanni et al., 2019; Zahoor & Zafar, 2020). While existing approaches are proficient in making high accuracies, they still require optimization to achieve better performance (Sun et al., 2020). Hence, there is a need for an improved approach, such as an optimized hybrid dimensionality reduction approach to improve classification prediction.

## 1.4. Aim and Objectives of the Study

The aim of this thesis is to develop a hybrid dimensionality reduction model for Anopheles gambiae RNA-Seq data classification, to enhance insecticidal target discoveries against malaria infection transmissions and control.

The specific objectives required to accomplish the aim of this study includes:

i. Identifying relevant dimensionality reduction techniques features for improving classification performance;

ii. Developing a hybrid dimensionality reduction technique, using feature selection (Genetic Algorithm Optimization) with feature extraction techniques (PCA and ICA) algorithms.

iii. Simulate the developed model using SVM, KNN, Ensemble and Decision Tree classification techniques.

iv. Evaluating the performance of the model in terms of; accuracy, specificity, precision, sensitivity, f-score and computational time.

## 1.5. Research Questions

Premised on the statement of the problem, this study will make investigations on the following research questions:

i. How can an efficient model be evolved to generate a lower-dimensional transcription dataset of the Anopheles, proteins, compounds, pathways, and reactions among others?

ii. How can the optimal features generated lead to significant enhancement in RNA-Seq data analysis?

## 1.6. Scope of the Study

In this study, machine learning approaches which includes dimensionality reduction techniques (GA-O, PCA and ICA) and classification algorithms (SVM, KNN, Ensemble and Decision tree) on a publicly available RNA-Seq datasets of the Anopheles mosquito were used. The technique used is the MATLAB data mining tool package on a Windows Operating system.

## 1.7. Significance of the Study

Malaria is an epidemic in Africa. Suggestions of several clinical kinds of research in preventing, detecting and predicting the ailment in human are eminent. There is a need for proper and better prediction application procedure for human, to help in reducing the death rate. This work will contribute immensely to clinical researches on insecticidal targets and resistance classification, as well as in the diagnosis and prediction of infectious diseases and other RNA-Seq experiments. The developed model will be useful to Computational Biologist and will enhance the reliability of insecticide designs.

## 1.8. Organization of the Thesis

The structure of this thesis is ordered as follows:

Chapter One contains the background of the study, statement of the problem, justification of the study, aim and objectives, research questions, the scope of the study and the significance of the study. The remaining sections of this study are prepared as follows:

Chapter Two: The literature review, assesses studies that have been carried out in the aspect of bioinformatics, sequence alignment algorithms, machine learning, dimensionality reduction, classification, and related works.

Chapter Three: The methodology gives a detailed description of the existing approach, the proposed model, dataset, research design, the layout, and the performance evaluation metrics.

Chapter Four. The results and findings of the study are discussed. First, the proposed hybrid technique is evaluated, and the result compared with existing methods. The findings are then discussed.

Chapter Five: The Conclusion and Recommendation, captures the completion of the study by giving a transitory summary of the work done in this thesis with its discoveries and supplementary research tips.

# CHAPTER TWO

## 2.0 REVIEW OF LITERATURE

This chapter presents reviews of the literature: Bioinformatics with essential reference to RNA-Seq technology, and machine learning techniques. This study reviews the literature and related works on dimensionality reduction and classification techniques.

### 2.1. RNA-Seq

RNA sequencing is one of the critical options in evaluating expression levels (Costa-Silva et al., 2017). RNA-Seq is capable of executing previous understanding of the essential sequences and permits diversity of applications such as; assessment of nucleotide variations, reforming of the transcriptome, methylation prototype evaluation, among others. RNA-seq technology contains several advantages that surpass microarray technology (Moreno et al., 2009; Conesa et al., 2016). For example; the high intensity of data reproducibility throughout the flow-cells, reducing the number of procedural copies for research. RNA-seq identifies and enumerates the expression of useful related proteins with comparable but distinguishable amino acid sequence, predetermined by miscellaneous genes (isoforms) (Agarwal et al., 2010; Chowdhary et al., 2016). The rate of next-generation sequencing research has fallen significantly in the facet of high-throughput sequencing methodologies. A stimulating considerate quantitative and qualitative study of RNA-Seq is yet to be realized, specifically comparison with early procedures like the microarray technology (White, 1996; Kratz & Carninci, 2014).

RNA-Seq is an exceptional expression analysis technology helpful for several particular states (Raut et al., 2010; Zhang et al., 2014). With the advancing fame of RNA-Seq technology, numerous software and channels have been put in place for gene expression analysis from these data (Oshlack et al., 2010; Lee et al., 2018). The invasion of high-dimensional and noisy information into genetic knowledge has to taunt out the capability of lower-dimensional data structure to be critical. Numerous examples have been provided recently of how lower-dimensional structure is capable of delivering better insight in the biology world, helping as an understanding and visualization tools. Biological data has experienced an inundated high-dimensional and noisy data; impressive structures can be uncovered using dimensionality reduction techniques in high-dimensional RNA-Seq data. Little insight about biological and procedural simulations that can clarify uncovered arrangements of individual genes is significant (Simmons et al., 2015; Poostchi et al., 2018). There is no compromise regarding which methodology is most suitable to certify the strength of outcomes in terms of robustness and accuracy. Figure 2.1 shows the RNA-Seq generation of data. A sampling technique that utilizes next-generation sequencing to indicate the existence and magnitude of RNA in a single experiment at a specified instant, evaluating the constantly evolving cellular transcriptome.

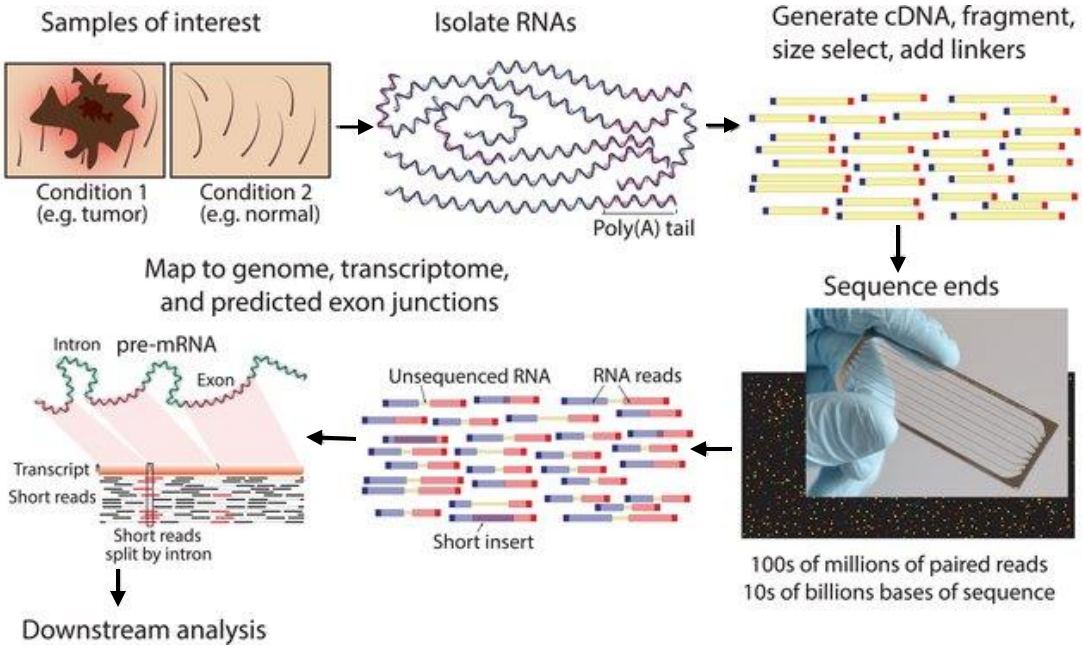



**Figure 2. 1 RNA-Seq Data Generation**

**Source: Griffith et al., (2015)**

## 2.2. Machine Learning

Human attention has always considered the knowledge of intelligent machines. Machine learning study and its development adopted comprehensive applications of neurophysiology, automation, psychology, mathematics, biology, computer science, among others to form theoretical sources, by exploring numerous learning methods, to develop an integrated learning structure. Several elementary difficulties of artificial intelligence and machine learning are designed, and utilization areas of multiple knowledge approaches have continually expanded (Kodratoff & Michalski, 2014). Over the years, Facebook, Amazon, Google, Microsoft, Twitter, Netflix, among other international Information Technology hulks have exposed the essence of machine learning and enhanced its correlated studies (Bell, 2014).

Machine learning is a computational technique that uses prior knowledge. It uses past information available in the form of digitized labelled training sets, to progress performances by the learner and makes specific interpretations. Its value and dimensions are significant to the success of predictions by the learner (Mohri *et al.*, 2019). Predicting models involve searching through data for expressions. This is well known in several aspects, like online shopping advertisements. It is recommended by the engines using machine learning to initialize online advertisements distribution in virtually real-time. Other than the modified advertising, further available machine learning uses comprises of detection of fraud, filtering spam and phishing, security detection of threats in networks, building news feeds and predictive maintenance.

Machine learning procedures are characterized as supervised or unsupervised learning. The supervised algorithms entail the delivering response about the prediction accuracies during the training of the algorithms. As soon as the training is complete, the algorithm applies what was learned to the new data. While the unsupervised algorithms do not require training by chosen result information, subsequently the achievement of a learning algorithm is dependent on data usage, machine learning is associated with data investigation and analysis. Machine learning methods remain data-driven and combine essential ideas in computer science with statistical concepts, optimization and probability (Karthik & Sudha, 2018).

Machine learning methods are efficient and applicable to numerous applications such as the network security, bioinformatics, banking, healthcare, economics, transports, among others. Bioinformatics and associated medical information are designed and gathered unceasingly, notable to the size of data. Innovative methods of big data, genomics, 3D

imaging, a biometric sensor, among others. Presenting information, the quick discovery of diseases and application of proper cares might diminish patient illness and death. Capability to achieve simultaneous investigations in contradiction of extensive torrent information transversely in all spheres transform healthcare. Within are information with dimensions, speed, diversity, among others (Kashyap et al., 2016).

Machine learning responsibilities necessitating extensive studies include the following (Jagga & Gupta, 2014), Figure 2.2 depicts an overview representation of a machine learning approach. The steps involved includes the following:

i. **Classification:** classifiers are used for restricted outputs, to limited conventional values.
ii. **Regression:** regressors predict constant outputs within a range of real values.
iii. **Ranking:** ranks are used to yield variation of items in hidden lists, it is a fundamental part of various information recovery difficulties, such as document retrieval, sentiment analysis, among others.
iv. **Clustering:** cluster helps in grouping sets of objects in similar forms.
v. **Dimensionality reduction:** helps in transforming the incomplete representation of objects into lower-dimensional form while conserving around properties of the unique model.

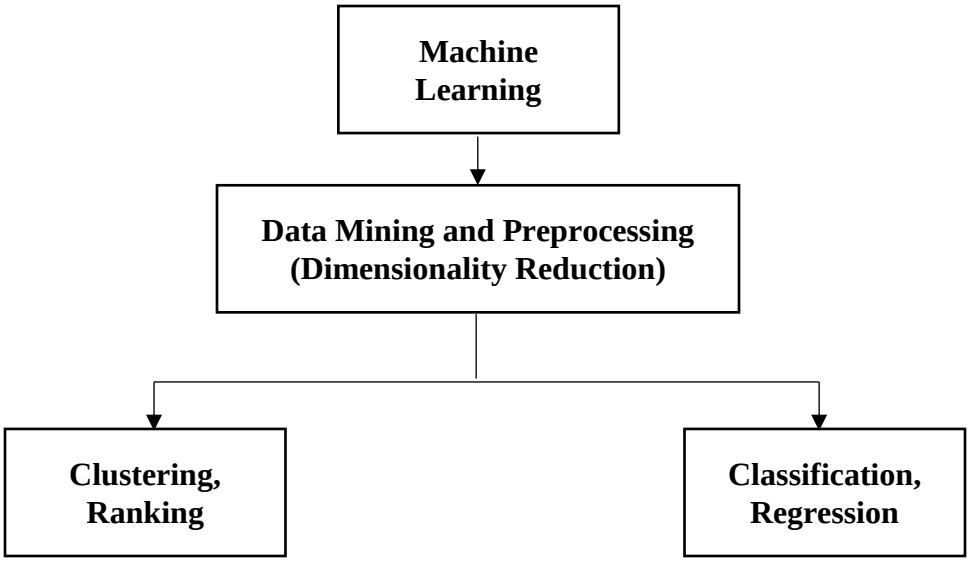



**Figure 2. 2 Overview of Machine Learning**

**Source: Frank et al., (2020)**

### 2.2.1. Machine Learning in Bioinformatics

There is a rapid rise in biomedical data dimension in the advent of data mining, and the acquisition rate has become stimulating with its predictable investigational approaches. Current machine learning methods, for instance, the deep learning, promise to control massive datasets for discovering hidden structures and predicting accurately (Angermueller et al., 2016).

Machine learning remains a computing science aspect that trains computational approaches acquired from the information. The potentials of machine learning in evaluating biomedical datasets helps in improving and exploiting the accessibility of progressively massive and high-dimensional datasets by training complex models that capture their structure. The learned models discover high-level features, intensify understanding and deliver further consideration about the building of the biomedical data (Angermueller et al., 2016).

Generally five forms of data are massive in dimensions and utilized severely in bioinformatics study (Kashyap et al., 2016):

i. protein-protein interaction (PPI) information,
ii. RNA, DNA, and protein sequence data
iii. Gene ontology (GO).
iv. Pathway data
v. Gene expression data

There are numerous kinds of information such as human ailment system and ailment genetic factor connotation system which are very significant for several studies like diagnosing ailments.

In the expression of gene analysis, levels of expressing thousands of protein sequence are investigated using diverse settings, for instance, discrete evolving phases of treatments or ailments. Expression of gene analysis can distinguish affected genes from infectious bodies or diseases, by relating the expression parameters from fit and unfit cells. Analysis outcomes are utilized in suggesting biomarkers for diagnosing and predicting diseases, among others. Several free database sources are well known, such as the Kaggle, Github, Gene Expression Omnibus, NCBI, EBI, Stanford database, among others (Zhou et al., 2014).

RNA/DNA sequences are prepared to utilize several investigative approaches to realize their attributes, features, structures, evolution and function. Sequence analysis methodologies comprise sequential alignments and biological exploration record, among others (Zielezinski et al., 2017). RNA sequencing remains mostly utilized as a microarray

substitute. Identification of post-transcriptional execution, mutation determination, the finding of diseases also exogenic RNAs, and finding Polyadenylation are some of the resolutions. Sequence analysis remains efficient than microarray analysis. Subsequently, sequence data embeds better-off evidence. It necessitates additional intelligent investigative tools with computing systems, to handle a massive volume of sequence information, significant sequence records comprise of DNA Information Bank, miRbase, among others (Nekrutenko & Taylor, 2012). There is several information the bioinformatics domain needs discovery; such as identifying current big data thriving in bioinformatics, and an urgent necessity to discourse these diverse kinds of issues.

#### 2.2.1.1. Microarray Data Analysis

The number and size of microarray sets of data remain speedily increasing, owing to diminishing cost and prevalent usage of microarray tests. Besides, microarray tests designed for gene-sample time-space are conducted towards recording variations in the values of expression over a period or multi-phase of diseases. Machine learning knowledge is essential for speedy creation of demonstration and voluminous microarray control network. One's gene expression results are collected at various periods of development of the disease; genes affected are identified to recognize biomarkers for diseases. In Computational terms, the calculation of third-dimension and period renders analytics more difficult according to terms of complexity than the initial gene investigation developments (Kashyap et al., 2016).

#### 2.2.1.2. Sequence Analysis

The upsurge in the amount (petabytes) of DNA information explosion has been initiated from thousands of generators. DNA sequencing instruments presently are insufficient, advancing high output and straightforward design for analysis of DNA sequence with transformed motivation for managing big data remains a bioinformatics concern with many requests in recent times. The emergence of RNA-seq technology as a durable substitute for microarray knowledge, owing to its added precise and predictable expression of gene dimensions. RNA-seq information comprises of extra data that are ignored frequently, and necessitate composite removal of machine learning procedures. Big data applications are used for detecting mutations, exogenous RNA contents and allele-specific expressions, for instance, diseases, from RNA-seq data with intelligent machine learning approaches. Next-generation gene sequencing makes available data on the broad human genome, in magnitude orders of more extensive than microarray-based genetic calculation methods. Significant procedures remain required to learn precise deviations in genome sequences owing to a specific virus and comparing with overall outcomes of similar or dissimilar associated diseases (Kashyap et al., 2016).

Analyzing machine learning data techniques requires the following properties:

i. **Accessible to large size:** The method is intelligent to hold a considerable amount of information with small dimension complexity and a reduced amount of overhead disk.

ii. **Full high speed:** The technique has less complexity of time, intelligent to abstract and develops stream information in real period lacking degeneration in achievement.

iii. **Translucent to assortment:** Big data are semi-organized or unorganized. Most conventional machine learning approaches, processing datasets utilizing standing illustration, generally generated from one source. Using representation, state well-organized feature sets and relations among them. Machine learning technique must handle information from numerous sources with diverse representation.

iv. **Additive:** Machine learning approaches works on the whole sets of data lacking system for the state at once, where the set of data is vigorously produced. Machine learning technique for big data processing can put into justification the unpredictable influx of data and manage these data at the lowest cost, without losing consistency.

v. **Dispersed:** Machine learning technique requires distributed partial data processing and integration of limited results. Big databases distributed worldwide; big data analytics are not accessible from only one source.

In machine learning, two major types of learning approaches exist, known as supervised and unsupervised learning approaches (Bhattacharyya & Kalita, 2013). A system learns in supervised learning from the collection of class-labelled sets, also known as the training set. The attained information is utilized to allocate labels to unidentified items known as test objects. In comparison, unsupervised learning approaches are independent on the

accessibility of preceding information or class labels training examples. All machine learning approaches necessitate dataset preprocessing for effective outcomes.

### 2.2.2. Supervised Machine Learning

Individual dataset samples in supervised learning are a pair of input values with external output value (vector), for prediction. Contingent functions are produced by evaluating supervised learning procedure training sets. The contingent functions, such as the training model, predicts new samples or mapped (Ariga, 2014).

Classification and regression are supervised learning methods with input vector X, output Y, and mission T for learning experiencing E from input X to output Y. Some supervised learning procedure categories are listed as follows: (Kuhn & Johnson, 2013):

i. Ordinary Linear Regression
ii. Linear Regression
iii. Penalized Regression
iv. Multivariate Adaptive Regression Splines
v. Partial Least Squares Regression
vi. Nonlinear Regression
vii. Artificial Neural Networks
viii. Bagging Tree
ix. Boosted Tree
x. K-Nearest Neighbors
xi. Support Vector Machines
xii. Regression Trees

xiii. Random Forest

### 2.2.3. Unsupervised Machine Learning

Unsupervised learning has no external output but owns the input vector during the learning procedure. It finds comparisons between unlabeled dataset samples. Two approaches for realizing unsupervised learning exists; indicating achievement over reward systems, and the result is completed through exploiting approvals, and not offering clear classifications. It also rewards the vectors by performing and reproving the other vectors (Oladipupo, 2010). Unsupervised knowledge is a data mining algorithm with no exact or improper response, making learning maintain results and patterns after running its algorithms. Methods of unsupervised learning comprise several learning methods (Wuest et al., 2016), figure 2.3 shows the learning approaches:

i. Clustering
ii. Expectation-Maximization algorithm
iii. Latent Variable Models
iv. Blind Signal Separation methods (for instance, PCA, ICA, Singular Value Decomposition, Non-negative Matrix Factorization).
v. Methods of Moments

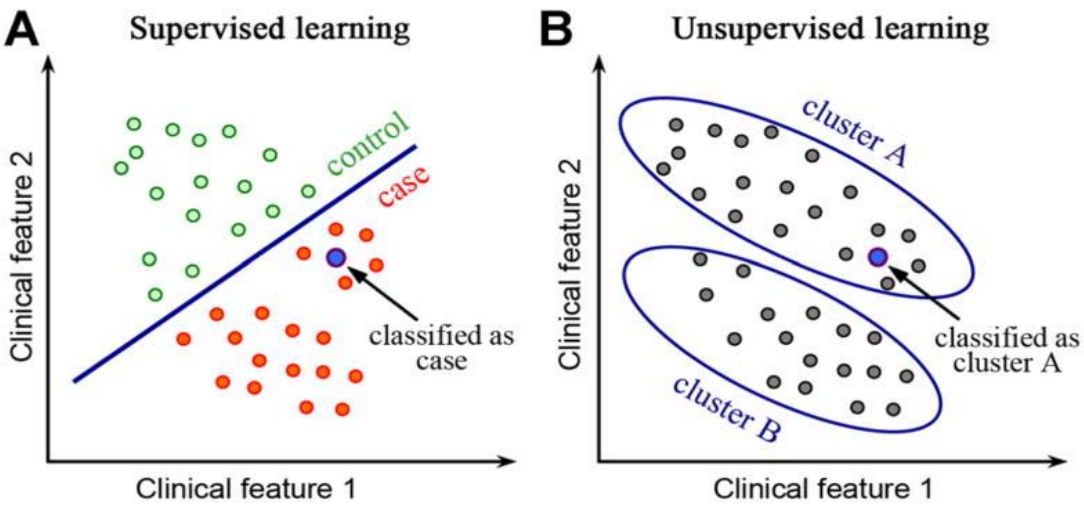



**Figure 2. 3 Supervised and Unsupervised Machine Learning Model**

**Source: Greene et al., (2014)**

## 2.3. Dimensionality Reduction

Dimensionality reduction has grown to be expected in pre-processing high-dimensional Gene expression data knowledge, for example, RNA-Seq, microarray, among others. RNA-Seq Gene expression data exhibits a significant sum of features of genes concurrently with small samples. Much information on genes is frequently made available to a learning algorithm for constructing and fetching an absolute description of the classification task. Most often, genetic factors are unrelated or unnecessary to learning studies. It deteriorates the precision and train speed, which results in the issue of overfitting (Ding et al., 2018).

Consequently, RNA-Seq data dimension reduction is a critical preprocessing stage for prediction and ailment classifications. A variation of dimensionality reduction methods has been suggested to differentiate genes having influence directly on various machine learning procedures for the classification, getting rid of residual ones. This study defines the dimension reduction method organization with its features, evaluation principles, pros and

cons and suggests an analysis of several dimension reduction methods for RNA-Seq data expression (Qi et al., 2020).

RNA-Seq data classification, has a significant complexity with most of the machine learning methods, which it is acquiring training through a massive amount of genetic factor. A lot of features (genes) are frequently made available to a learning procedure, for building a comprehensive classification task description. The applications of machine learning, with the aspect of features, has expanded in variables or features used in those applications. Several machine learning methods are developed to address the problem of reducing unrelated and redundant features which are burdens for different motivating tasks (Aziz et al., 2017a). High dimensional data are challenging. They contain substantial computational rate and usage of memory (Jindal & Kumar, 2017).

In machine learning, dimensionality reduction comprises of feature selection and feature extraction procedures. Feature selection technique discovers related features from a unique set of information using objective measures, to lessen the amount of features and to eliminate the unrelated, redundant and noisy features from high-dimensional data (Cheng et al., 2015). Feature extraction method obtains the most relevant data from unique data and denotes data in lesser dimensional space; it picks new feature sets and transforms its features into a direct or indirect grouping of unique features (Arul & Elavarasan, 2014). Dimensionality reduction techniques are vital, and they are used to lessen the features of original information, it may be used remotely or in combination to improve performances such as accuracy among other parameters.

### 2.3.1. Feature Selection

Feature selection is labelled a variable selection, an attribute selection, or a variable subset selection. It is the process of picking subsets of pertinent features from a huge set of information to advance classification performance (Jovic et al., 2015).

Feature selection have been extensively utilized in preprocessing data for machine learning technology, and fundamentally utilized for reducing information by getting rid of unrelated and unnecessary features in a data (Jain & Singh, 2018). Feature selection is a dimensionality reduction technique that improves the clarity of information benefiting precise data, trims downtime of training the learning algorithms, improves prediction performance and enhances visualization of data. Feature selection consists of three relevant variable selection types; Filter approaches, wrapper approaches and embedded approaches (Jindal & Kumar, 2017). Diverse learning algorithms perform proficiently and provide better precise results when data holds non-redundant and significant attributes. Datasets have a considerable amount of unrelated and unnecessary features. There is a necessity for a proficient feature selection technique in extracting relevant features. Feature selection methods are essential for selecting revealing genes proceeding to the classification of RNA-Seq data for prediction and diagnosis of diseases, to advance the classification accuracy (Aziz et al., 2017a).

There are numerous feature selection procedures applied to malaria vectors among other ailments, such as typhoid, tuberculosis, diarrhea, measles, among others. Filter, wrapper, ensemble and embedded systems remain its forms of techniques (Hira & Gillies, 2015).

Before models are employed on data, eliminate noisy and unreliable information to fetch precise outcomes in a lesser amount of period. Reducing the dimensionality of a set of data is of principal significance in applications. Furthermore, if the most significant features are picked, the difficulty reduces exponentially. Numerous feature selections methods have been applied on multiple ailment datasets. Exploring relevant proofs, application of feature selection approaches has been carried out on medical records to predict several degenerative diseases such as heart disease, diabetes, hypertension, strokes, among others. Numerous algorithms for learning perform proficiently and gives more precise outcomes if the information comprises of more essential and necessary attributes (Chaturvedi et al., 2018). As healthcare sets of the data content of a huge amount of unnecessary and unsuitable features, an effective feature selection method is required to abstract appropriate and exciting features. Feature selection system for classifying gene expression data model, utilized information depth to limit gene markers suitable for tumor (Pavithra & Lakshmanan, 2017; Wenyan et al., 2017). They explored an interval-based feature selection technique for two-biological-group classification delinquent. Bhattacharyya interval was employed for picking gene markers (Bhattacharyya & Kalita, 2013). To measure dissimilarities in the gene expression levels amongst collections. They utilized SVM classification on the fetched genes marker, genes marker selection and classification performances are demonstrated using the simulation training and real data investigation.

#### 2.3.1.1. Filter-Based Feature Selection Method.

Filter-based feature selection is based on a particular evaluation principle (Kumar, 2014); it is a non-dependent approach, giving various performance on prediction. Filter based approach provides rapid and proficient results on execution. As a result, they are ideal for big databases. They perform efficiently with huge databases, computationally less expensive and efficient. Filter approach provides performance of results very fast with a high-quality overview, and it is low computational complexity (Hasan & Adnan, 2012). They pick subsets of features by using relevant model learning algorithm and ranks features on assessment condition basis. They depend on the fundamental uniqueness of data, and their variable selection procedure requires execution formerly utilizing this approach, they use statistical techniques for conveying scores to features due to their robustness against over-fitting in comparison to other methods and procedures (Hazrati et al., 2013).

The drawback of these approaches is that no attention is paid to the classification interaction, feature dependencies and failure in picking the most "useful" features (Chen et al., 2010; Lavanya et al., 2014). Filter method also has several drawbacks because it pays no consideration on the classifier's interaction, features are not considered, and neglects several features not functional themselves but valuable when shared. Filter algorithms are evaluated on different criteria; distance, information, dependency and consistency (Kumar, 2014). Filter based feature selection comprises of the following relatively RNA-Seq beneficial better performance algorithms (Jain & Singh, 2018). ANOVA, T-test feature selection, Information gain, Genetic Algorithm, Fisher score, Chi-squared test, Correlation-based Feature Selection, among others. Algorithm depict the filter feature selection algorithm (Ghareb et al., 2016).

**Algorithm 2.1: Filter Algorithm Source: Ghareb et al., (2016)**

**Input:-**

$X(F_0, F_1, \ldots\ldots\ldots\ldots, F_{n-1})$ //a training data with n features

$Subset_N$ //a subset from where to start the search

$Stop_{cr}$ //a stopping Criterian

**Output:-**

$Y_{opt}$ // an Optimal subset

**Algorithm:-**

```
begin
Initialize
Yopt = Subset N;
Newopt=Evaluate (Subset N, X ,M);  //Evaluate subset by a Independent Measure M
do begin
S=Selectsubset(X) ;   //generate a subset for evaluation
New=Evaluate(S, X, M); //Evaluate the current subset S by M
If(New is better than Newopt)
Newopt=New;
Yopt=S;
End until(Stopcr is met);
Return Yopt;
End
```

### 2.3.1.2.Wrapper-Based Feature Selection Method

Wrapper-based feature selection picks features by giving proper consideration to the usage of knowledge algorithm. The significant benefit over filter methods is locating the major constructive features, and best-selected features are carried out for the learning system (Kumar, 2014). Wrapper-based feature selection method explores the best subset in a feature by taking into account the learning system to be used. It utilizes accurate classifier to estimate the selected features class, the classifier runs severally to evaluate the value of the features, in terms of the accuracy of a model, scoring is assigned. A wrapper-based method performs an optimal selection of features. It calculates the estimated accuracy for the particular feature using the partiality of the training system to pick features of the learning algorithm (Alelyani et al., 2018; Tang et al., 2014). Wrapper method considers

reliance along with features. It has an enhanced presentation in terms of predictive metrics, improved classifier relations, and optimizes the classifier performance and provides a better precise outcome in contrast to filter approaches (Maldonado & Weber, 2009). However, there is difficulty in utilizing additional learning algorithms, resulting from the needs of executing the algorithms over again. More computational complexity is encountered with a larger time of implementation. Over-fitting of the dataset, huge computational resources, expensive than filter methods computationally, lacks generality, and large datasets are less scalable. Wrapper based methods are complex and result in over-fitting on small training datasets. Procedures that may be useful to RNA-Seq are; Simulated annealing, Sequential forward selection, Genetic Algorithms, Recursive feature elimination method, backward elimination Method, among others (Jain & Singh, 2018). Algorithm 2.2 depicts the wrapper-based feature selection algorithm (Sahu et al., 2018).

**Algorithm 2.2: Wrapper Algorithm Source: Sahu et al., (2018)**

**Input:-**
X ($F_0$,$F_1$,……,$F_{n-1}$) //a training data with n features
Subset N //a subset from where to start the search
$Stop_{cr}$ //a stopping Criterian
**Output:-**
$Y_{opt}$ // an Optimal subset
**Algorithm:-**
begin
Initialize
$Y_{opt}$=Subset N;
$New_{opt}$=Evaluate (Subset N, X ,MinAlg);//Evaluate subset by a mining //Algorithm MinAlg
do begin
S=Selectsubset(X) ; //generate a subset for evaluation
New=Evaluate(S, X, MinAlg); //Evaluate the current subset S by A
If(New is better than $New_{opt}$)
End until($Stop_{cr}$ is met);
$New_{opt}$=New;
$Y_{opt}$=S;
Return $Y_{opt}$;
end

#### 2.3.1.3. Embedded Feature Selection Method

The embedded feature selection procedure is generally steered by the learning technique search recognized as the nested-subset technique (Kumar et al., 2015). It evaluates the "worth" of feature subsets, then the feature selection procedure is carried out as a training progression (Hira & Gillies, 2015). They work specifically for enhancing the execution of the learning procedure using data obtainable then generates solutions quicker. Search is steered by learning procedure in this method by carrying out the training process, the services of filter and wrapper methods are aggregated and precise to the learning machines. They stand computationally inexpensive and prone less to over-fitting. They stand better classifiers with its dependencies between features capturing effectively, available data usage and providing faster solutions.

Moreover, the computational complexity is better (Kumar et al., 2015). Its major limitation is taking dependent classification decisions, hence affecting the selected features by the hypothesis that the varying classifiers (Peng et al., 2010). They are specific, with poor simplification, considerate selection of appropriate features for classifier usage and computationally costly. Embedded feature selection methods include; Decision Trees "ID3, C4.5/5.0 algorithms", CART-Random forest algorithm, LASSO technique, SVM-Recursive Feature Elimination approach, Elastic Net, Artificial neural networks, Ridge Regression, Weighted Naïve Bayes, Feature selection with SVM weighted vector, Sequential Forward, Selection (SFS), among others. Algorithm 2.3 and Figure 2.4 depicts the embedded feature selection algorithm and feature selection mechanisms respectively (Aziz et al., 2017).

**Algorithm 2.3: Embedded Method Source: Aziz et al., 2017**

**Input:-**
$X(F_0, F_1, \ldots\ldots\ldots\ldots, F_{n-1})$ //a training data with n features
Subset N //a subset from where to start the search
$Stop_{cr}$ //a stopping Criterian
**Output:-**
$Y_{opt}$ // an Optimal subset
**Algorithm:-**
Begin
Initialize
$Y_{opt}$=Subset N;
$C_0$ = Cardinality(Subset N) ; //Calculate the cardinality of subset N
$\theta_{best}$ = Evaluate (Subset n, X, M);
$\sigma_{best}$ = Evaluate(Subset N, X , MinAlg); //Evaluate Subset N by a mining algorithm MinAlg
for $c=c_0+1$ to N begin
for i=0 to N-c begin
$S=Y_{opt} U \{ F_j \}$; //generate a subset with cardinality c for evaluation
$\sigma$ = evaluate(S, X, M); //Evaluate the current subset S by M
If($\sigma$ is better than $\sigma_{best}$)
$\sigma_{best}=\sigma$;
$Y^i_{opt}$ = S;
End
$\sigma$ = Evaluate ($Y^i_{opt}$, X , MinAlg); // Evaluate $Y^i_{opt}$ by MinAlg
If($\sigma$ is better than $\sigma_{best}$)
$Y_{opt}= Y^i_{opt}$;
$\sigma_{best}=\sigma$;
else
break and return $Y_{opt}$;
end;
return $Y_{opt}$; end;

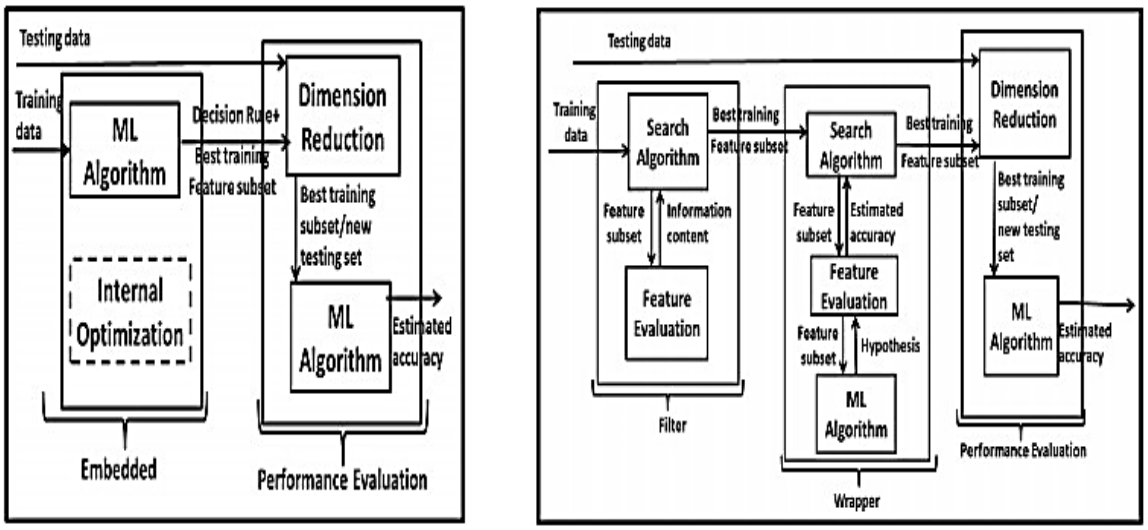



**Figure 2.4 Feature Selection Mechanisms and Approaches**

**Source: Aziz *et al.*, 2017**

### 2.3.2. Feature Extraction

Feature extraction is an intelligent substitute to feature selection, for diminishing sizes of huge- dimensional information. In literature, it is known as “Feature projection or construction” on a lower-dimensional-subspace. Feature extraction method changes the original feature in lesser dimensional space; in this way, problems are represented in a more discriminating (informative) space that makes the further analysis more efficient. Two main categories of feature extraction algorithms known as the linear and non-linear approaches exist. Linear approaches are usually faster, more robust and more interpretable than non-linear methods. The non-linear methods discover complicated structures of data (embedment’s) where linear methods fail to distinguish (Bartenhagen et al., 2010).

Feature extraction is used in obtaining new latent optimal component features from a given dataset by transforming the data into a reduced complexity form of features. It gives a frank data illustration of the respective variable in a feature subspace as a grouping of linear input variables. Furthermore, feature extraction is a general method, with various procedures existing, for example; PCA, Non-Linear PCA, Self-Organizing Map (SOM), Ant Colony Optimization, ICA, Locally Linear Embedding (LLE), Kernel-PCA, Local Directional Pattern (LDP), Linear Discriminant Analysis (LDA), among others (Jindal & Kumar, 2017). PCA is the most common and extensively utilized feature extraction approach, presented by Karl (Nandhini & Porkodi., 2019). It consists of orthogonal conversion for models fitting to connected variables with linearly uncorrelated variables models.

### 2.3.3. Hybrid Methods for Dimension Reduction

Hybrid examination method has been utilized for dimension reduction, due to its rewards of both filter/extraction and wrapper technique (Huang & Lowe, 2007). A hybrid dimension reduction system involves two phases, with the initial phase, a filter/extraction technique are utilized to recognize best appropriate features of the sets of data. The next phase institutes another method to validate the earlier recognized germane feature subsets by confirming the method to give higher classification accuracy rates (Aziz *et al.,* 2017). It utilizes different assessment conditions in diverse search phases, to advance the classification efficiency and accuracy with improved computational performance. In hybrid exploration procedure, the initial feature subset is picked or removed, based on the filter or extraction technique and after other techniques are utilized to pick the concluding feature set. Hence, due to the computational rates, hybrid methods have become suitable due to the utilization of reduced feature sizes, authors have lately utilized the hybrid technique in solving the issues of dimensionality reduction in the RNA-Seq technology (Yong et al., 2016).

## 2.4. Classification

Classification methods have been established, approved, and useful for distinguishing and analysis of data. In recent time, innovative session of ranked probabilistic representations based on several classification techniques has to turn out to be one of the trending insights for analyzing data. These representations are initially established utilizing gene expression data, and additionally enhanced for classification difficulties under a combining outline. Unambiguously, an accommodative algorithm with an arranged structure is constructed to

fetch suitable highlighting kernels, to determine possibly important genes, and make best prediction classes for disease with related classification subsequent probabilities (Guia et al., 2018).

In recent years, numerous classification algorithms for data have evolved and improved from recent machine learning systems. A lot of researchers in the past have ardent their hard work to the learning of ensemble-decision tree classification approaches. This approach combines decision trees produced by many training sets through re-selecting the training sets. Boosting, Bagging, and Random forest is recognized ensemble approaches in the machine learning area (Almasri et al., 2019).

Classification methods can be utilized in gene expression analysis to foretell model phenotypes premised on patterns of gene expression. Although innovation and precise gene expression classification tools are developed continually, the dominant form of prediction systems offers operative tools (Dudoit et al., 2002), suggested an applied evaluation of approaches for classifying tumor gene expression data.

Dataset samples belong to classes like the malignant or non-malignant dataset. Its goal is classifying samples and yielding classified samples grounded on its measurements in RNA-Seq. Classifier training of high-dimensional datasets is a great challenge receiving varieties of consideration from researchers. A standard way of addressing these challenges are majorly done by using pre-processors and applications of classification algorithm that controls the complexity model through regularization (Mohamed et al., 2016). Machine learning is an approach that scientifically addresses some questions such as, how systems can be programmed to learn and improve using knowledge inevitably. Learning is not

measured as an actual learning process but identifying multifaceted designs and making intelligent conclusions built on data. Machine learning advances procedures that realize knowledge from precise data and knowledge, grounded on principles of computation. Classification aims to develop rule decisions that distinguish among models of distinct classes grounded on the profiles of the gene expression. Discovering important rules of classification to achieve the task of classification is fit for medical investigations. Some of the extensively utilized classifiers are Decision Tree, Neural Network, Bat Algorithm, Artificial bee colony, Particle swarm optimization, SVM, K-NN, among others (Sumathi et al., 2012), Table 2.1 and Table 2.2 show the characteristics of dimensionality reduction techniques.

## 2.5. Dimensionality Reduction Approaches

**Table 2. 1 Feature Selection Algorithms with Their Respective Characteristics.**

| Feature Selection Method | Algorithms | Characteristics | Limitations | Assessments |
|---|---|---|---|---|
| **Filter-Based approaches** | Correlation-based feature selection (CBFS) (Alzubi et al., 2018). | Assesses subsets considering the predictive skill of its features individually with their quality of correlation or redundancy | It is feature dependent but slower than univariate techniques. | Heuristic merit |
| | Mutual Information (Hira & Gillies, 2015) | Examined most probable cancer-associated genes, to enhance classification accuracy | Evaluate features and class dependencies. Features contribute to classification redundancy (Pavithra & Lakshmanan, 2017). | Symmetric relationship |
| | Analysis of Variance (ANOVA) (M. Kumar et al., 2015) | The dependent variable is continuous and categorized as nominal or ordinal. Its data are normally distributed | It gives an overall test of equality of group means. It tests against the specific hypothesis. | Hypothesis test |
| | Information Gain (Uysal & Gunal, 2012). | It measures known features of relevant and predicted Information; features often occurring in | It evaluates based on entropy, and it involves mathematical theorems, complex theories | Ranking |

| | | | | |
|---|---|---|---|---|
| | | positive samples can be obtained. | and entropy formulas. | |
| | Chi-Square (Arul & Elavarasan, 2014) | evaluates correlation among two variables and limits if independent or correlated | | |
| **Wrapper Based Approaches** | Genetic Algorithm (Pavithra & Lakshmanan, 2017) | By getting a set of sequences to describe possible answers, it imitates mutation and integrates it to create more fits. | Produce a random population search. But has a lower training time | Crossover and mutation |
| | Recursive feature elimination method (Nalband et al., 2016) | Backward selection of predictor fitting models and removing weakest features. | Comprises of essential partition predictors. Ranks features are based on the order of elimination and multicollinearity. | Greedy optimization |
| **Embedded approaches** | Info Gain-SVM (Sivapriya & Kamal, 2013) | Selects attributes and improves correlation | Diminishes the outcome of partiality resultant from information gain. Corrects attributes allows extensiveness and consistency of attribute values. | Wavelength |

| | | | | |
|---|---|---|---|---|
| | SVM-Recursive Feature Elimination (RFE) (Hira & Gillies, 2015) | makes implicit orthogonality assumptions; it considers a combination of univariate classifiers. The decision function is based solely on support vectors with "borderline" cases characterizing "typical" cases. | lower risk of overfitting | ranking criterion |

**Table 2. 2 Feature Extraction Algorithms and Their Respective Characteristics.**

| **Feature Extraction Algorithms** | **Algorithms** | **Characteristics** | **Limitations** |
|---|---|---|---|
| **Unsupervised or Nonlinear Learning Approach** | Principal Component Analysis (Pinto da Costa et al., 2011) | Selects the most important genes and identifies transcriptional programs by *extracting* groups of genes covering across sample sets. | Each variable number taken do not have the same status and where information may be containing noises and outliers. |
| **Supervised or Linear Learning Approach** | Independent Component Analysis (ICA) (Lucas, 2013) | New variables are limited in the *S* rows, variables detected are linearly poised independent components. | Blind separation of independent bases from their linear grouping |
| | Partial Least Square (PLS) (Tan et al., 2014) | An insignificant number of hidden features determined it. It goes for determining the unrelated linear conversion of the preliminary indicator features with high covariance and response features. | Inherent components, PLS requires y response characteristics, the validation task, it repeats conceptual matrix X, the knowledge modeling operation. Optimizing covariance between variable *y* and primary predictor variables |

### 2.5.1. Genetic Algorithm (GA)

GA is a capable process used in examining the appropriate high dimensional dataset features. Evolving GAs are wrapper-based methods of feature selection. There are many uses of parameters for genetic algorithms whereby mutation and crossover operatives typically remain related to the concepts of binary parameters. The use of a genetic algorithm recognizes suitable characteristics (Duval & Hao, 2010). The RNA receives the related number of components describing characteristics with values of 1 and 0 as selected and unselected. Presenting the value of features, GA is used to find an optimal set of attributes with voted specific usability function classification estimates. In Algorithm 2.4, the generalized GA framework is specified by adopting (Shukla et al., 2019):

**Algorithm 2.4. Genetic Algorithm**

Involve: Prime the parameters bPop = a, $t_{max}$, t = 0;

```
Certify Optimum feature subset with the highest fitness rate.
      1: while (t<=t_max) do
      2:     Create pop a, t_max;
      3:     For c = 1 to a do
      4:           Parents [a_1, a_2] = system selection (a, nPop)
      5:           Child = Xor[a_1, a_2]
      6:           M u = mutation {Child}
      7:     End for
      8:     Swap a through Child_1, Child_2, …, Child_a
      9:     t = t+ 1;
      10:    End while
      11:    Store Highest fitness value;
```

*a* is a population dimensions, *r* = random sum lying lined by 0 - 1, process chrome signifies, picked or unpicked feature through, aid of threshold δ sets the rate to be 0.5, and α = selected features of threshold value. The key difficulties of the specific technique are picking the determined suitable features from recognized datasets figure 2.8 shows the Genetic algorithm flowchart.

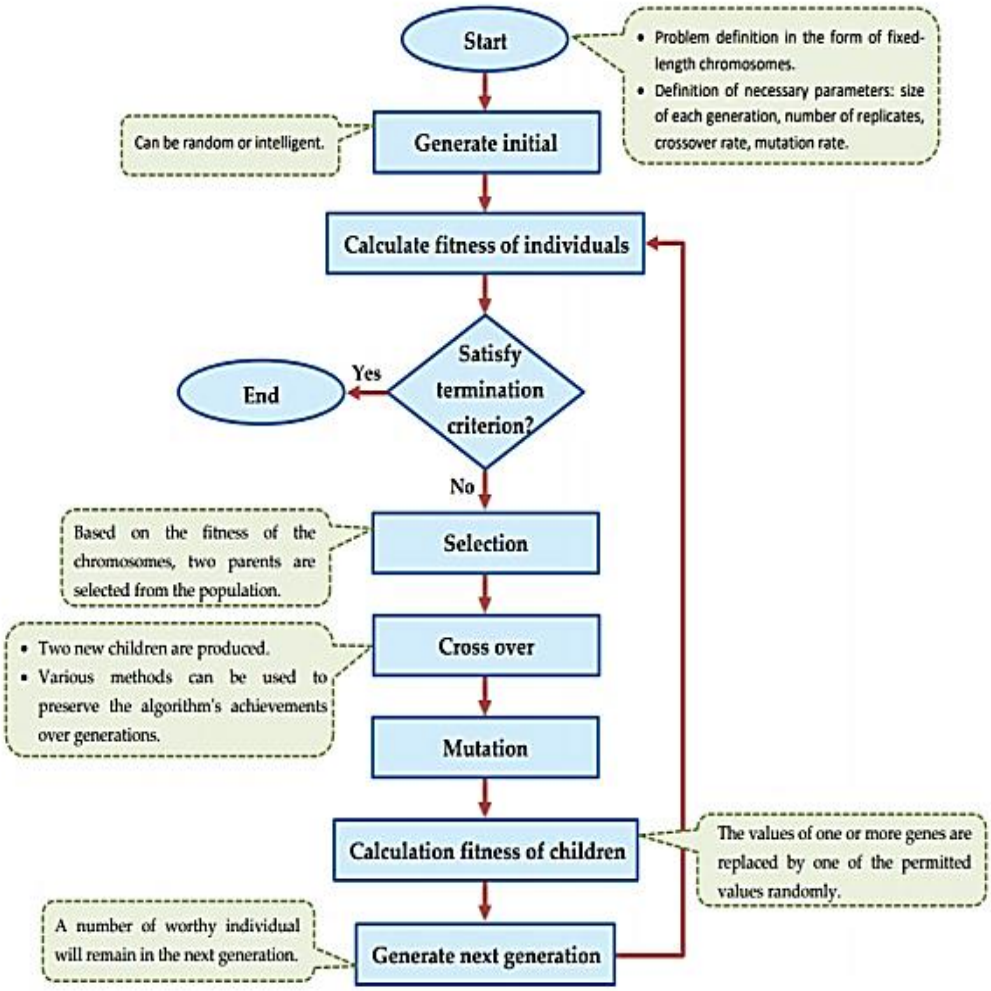



**Figure 2. 5 Flowchart of a General Genetic Algorithm**

**Source: Momeni & Abadeh (2019)**

### 2.5.2. Principal Component Analysis (PCA)

A broadly utilized unsupervised feature extraction dimensionality decrease procedure because of its simplicity (Sofie, 2017). It utilizes a straightforward procedure to implant information into a linear subspace of lower dimensionality. PCA plans each occurrence of the specified dataset existing in a dimensional space to a *j* dimensional subspace where *j* < *a*. The set of *j* new dimensions produced are recognized as the Principal Components (PC), and all principal component remains matched to determined variance without the variance previously represented in the primary components (Keerthi Vasan & Surendiran, 2016). PCA is widely the most prevalent (unsupervised) linear technique; it builds a low-dimensional illustration of the information depicting an abundance of the difference in the information as could be expected. It is carried out by discovering a linear premise of decreased dimensionality for information; the volume of difference in the information is greatest. PCA computation transformation matrix U adopted Rasan & Mani, (2015) and given as:

$$U = (\sum_{i=1}^{n}(B_i - l)(B_i - l)^S) \qquad 2.1$$

Where; n is the instances; *Bi* is the *i-th* instance; l is the mean vector of the input data.

The given high-dimensional input data are standardized as each attribute falls within the same range, to ensure that all attributes with larger domains in the data do not overwhelm attributes with little domain. PCA computes the symmetrical vector, which gives a premise to standardized data.

The input data are a linear combination of PC. It is arranged in diminishing order of their quality or criticalness, the size of data can be decreased by weaker component, implying that PCs with lower variance.

Adopting Shon et al., (2019), the model mean $\bar{x}$ and information covariance matrix S are as below:

$$\bar{X} = \frac{1}{N}\sum_{n=1}^{N} X_n \qquad 2.2$$

$$S = \frac{1}{N}\sum_{n=1}^{N} (X_n - \bar{X})(X_n - \bar{X})^T \qquad 2.3$$

Equation 2.2 and 2.3 assumes the component vector on the principal subspace exploiting the variance of an assumed set of data is shown in equation 2.4.

$$Su_i = \lambda_i u_i u_i{}^T \; Su_i = \lambda_i \qquad 2.4$$

The vector maximizing the variance of a predictable data develops an eigenvector, $u_i$, of matrix *S*, and maximal variance dimensions in the path of the eigenvector develop the eigenvalue $\lambda_i$. Principal subspace collected for the principal component resultant from PCA is *M* eigenvectors bits of maximal eigenvalues for matrix S. Figure 2.9 shows the flowchart for PCA algorithm.

**Algorithm 2.5. Principal Component Analysis (PCA) Source: Shamir, (2009)**

PCA procedure ($X$, $k$): top k eigenvectors

1: X = N × m matrix data,

2: … respectively data point xi = column vector, i=1...m

3: $\underline{X} = \frac{1}{m}\sum_{i=1}^{m} X_i$

4: X ← Deduct mean $\underline{X}$ from the respective column vector $X_i$ in $\underline{X}$

5: $\Sigma \leftarrow XX^T$ … covariance matrix of X

6: $\{\lambda_i , u_i \}i=1..N$ = eigenvectors/eigenvalues of $\Sigma$ ... $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$

7: Return $\{ \lambda_i , u_i \}i=1..k$

8: top k principal components

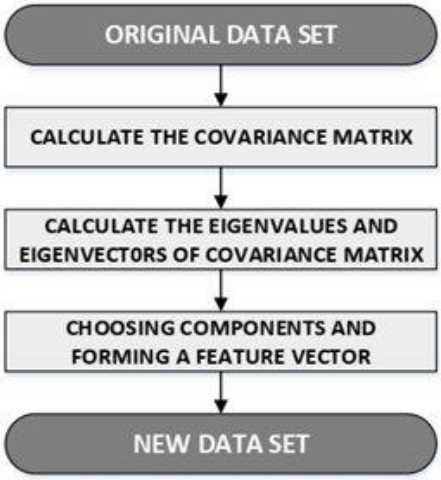



**Figure 2.6 Flowchart of the PCA**

**Source: Souza et al., (2019)**

### 2.5.3. Independent Component Analysis (ICA)

ICA is a valued extensive of PCA and conservative since, visor independent separation bases from linear grouping (Tan *et al.*, 2014). The fact of ICA possesses uncorrelation of overall PCA. Assembled *n* x *p* on information matrix X, with rows *ri (j=1…, n)* calculates to observed variables and columns $c_j$ *(j=1…, p*) are objects of consistent variables, ICA *X* model is as follows:

$$X = AS \qquad 2.5$$

With a comprehensive indication, *A = n x n* fusion matrix, where *S = n x p* is a basis matrix with the need of statistical independent conceivable. Independent components are original variables reserved in the rows S. Variables spotted are linearly composed of independent components. The independent components realized by learning the exact linear groupings of the observed variables, subsequently joining can be reversed as:

$$U = S = A - 1X = WX \qquad 2.6$$

**Algorithm 2.6. Independent Component Analysis (ICA) Source: Ke et al., (2015)**

1: Relate observed signs $x$ to remove its mean;

2: Diminish the detected signs;

3: Choose an initial value for $\mu$, generally let $\mu = 0$;

4: Take arbitrary preliminary vector $w$ of norm 1;

5: Update the Lagrange multiplier $\mu$;

6: Update the de-mixing vector $w$ $w \leftarrow w - \gamma \Delta w$: where $\gamma$ is the learning rate;

7: Normalize the $w$ by $w \leftarrow w / \|w\|$;

8: While detecting the second increment of $\mu$ or minus $D$, then restart the algorithm from Step 3 with a new initial $w$ by deflationary orthogonalization technique, figure 2.10 shows the ICA algorithm framework;

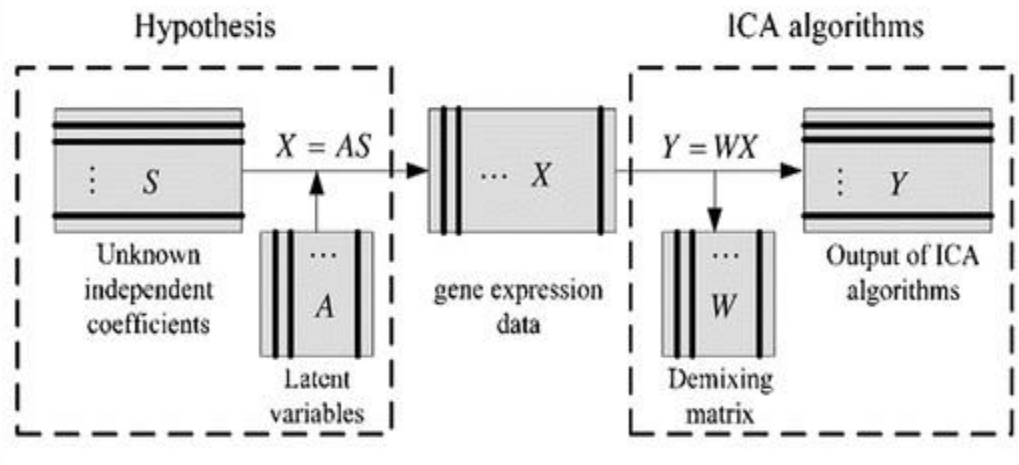



**Figure 2.7 ICA Algorithms Framework**

**Source: Kong *et al.*, (2018)**

### 2.5.4. Support Vector Machine (SVM)

SVM is a machine learning procedure presented by Vapnik in 1992 (Aydadenta & Adiwijaya, 2018). The algorithm works with points of discovering the top hyperplane isolating between the input space classes. SVM is a linear classifier; it is created to function with nonlinear difficulties by joining kernel ideas in high-dimensional workplaces. In non-linear issues, SVM utilizes a kernel in training the data to spread the dimension widely. When the dimensions are tweaked, SVM will look for the optimal hyperplane that can separate a class from different classes. As indicated by the adoption of Aydadenta and Adiwijaya (2018), the procedure to locate the best hyperplane utilizing SVM is as follows:

i. Let $y_i \in \{y_{1,} y_2, \dots, y_n\}, where\ y_i\ is\ the\ p - attributes\ and\ target\ class\ z_i \in \{+1, -1\}$

ii. Assuming the classes +1 and -1 can be separated by a hyperplane, as defined in equation 2.7:

$$v.y + c = 0 \qquad 2.7$$

From equation (2.7), Equations (2.8) and (2.9) are gotten as:

$$v.y + c \geq +1, for\ class\ +1 \qquad 2.8$$

$$v.b + c \leq -1, for\ class\ -1 \qquad 2.9$$

Where y is the input data, v is the ordinary level and c is the positive relation to the center coordinate fields.

SVM intends to discover hyperplanes that maximize margins between two classes. Intensifying boundaries is a quadratic program design problem resolved by discovering the nominal points. The benefit of SVM is its dimensions to achieve an extensive collection of classification difficulties in high dimensional data (Soofi & Awan, 2017).

Relating to supplementary methods of classification, SVM is prominent, with its brilliant classification competence (Baharudin et al., 2010). SVM is grouped into linear and non-linear separable. SVMs has kernel functions that change information into an advanced dimensional space, making it conceivable to accomplish separations. Kernel purposes are classes of procedures for design investigation or identification. Training vectors $x_i$ is recorded into higher space of dimension by capacity Φ. SVM discovers a linear unravelling hyperplane with maximal high-dimensional space. C > 0 is the consequence parameter of fault period.

Several SVM kernels exist like; the polynomial, linear, Sigmoid, Gaussian, String Kernels, Radial basis function (RBF), and so on. The result of a Kernel hinges on the present problem since its hinges on hat models are analyzed, a couple of kernel functions are admirably in for a wide assortment of applications (Suksut et al., 2019; Bhavsar & Panchal, 2012). The prescribed kernel function for this study is the SVM-Polynomial and Gaussian Kernel.

#### 2.5.4.1. SVM-Gaussian Kernel Functions

Gaussian kernel (Vanitha et al., 2015) is compared to a general smoothness supposition in all k-th order subordinates. Kernels coordinating a certain prior recurrence substance of the data can be developed to reflect earlier issues in learning. Each input vector x is mapped to

an interminable dimensional vector including all degree polynomial extensions of x's components.

#### 2.5.4.2. SVM Polynomial Kernel Functions

A polynomial kernel model features a combination to the directive of the polynomial. Radial basis functions permit circles in disparity with the linear kernel, which permits just selecting lines (or hyperplanes).

$$K(y_a, y_j) = (\gamma y_a^S y_b + q)^e, \gamma > 0 \qquad 2.10$$

#### 2.5.4.3. SVM-Linear Kernel Function

Linear is the least complex kernel function. Assumed by the inner produce (a,b) in addition to a discretionary constant K.

$$K(y_a, y_b) = y_a^S y_b \qquad 2.11$$

#### 2.5.4.4. SVM-RBF Kernel Function

In SVM kernel functions, a, **γ**, and b are kernel constraints, RBF is the fundamental kernel function due to the nonlinearly maps tests in developed dimensional space, compared to the linear kernel, it has reduced hyperparameters compared to the polynomial portion.

$$K(y_a, y_b) = exp(-\gamma||y_a, y_b||^{\ 2}), \gamma > 0 \qquad 2.12$$

### 2.5.5. Ensemble Classifier

An ensemble classifier trains unconnected subgroups of training data, varied constraints of the classifiers, having various feature subsets, in arbitrary subspace models (Nagi &

Bhattacharyya, 2013). The ensemble contains integrating results of diverse classifiers producing a conclusion. It is commonly used to gain very accurate outcomes. The ensemble classifiers are relatively common in machine learning complications and are dynamic in bioinformatic works. Classification decision is attained by integrating decisions of respective classifiers (Sheela & Rangarajan, 2018).

Ensemble method is a machine learning system that associates results to enhance the performance of the overall classification. Numerous relationships in reviews signifying similar suggestions have been revealed such as; the multi-strategy knowledge, combination, integration manifold classifiers, combination, committee, classifier synthesis, and so on. The ensemble as a classifier possesses overall enhanced performance than the discrete-based classifiers (Nti et al., 2020). The proficiency of ensemble methods are enormously reliant on nonconformity of faults dedicated by distinct learner. Ensemble methods performance center on the accuracy and variation of base learners, having mutual methods; the bagging and the boosting.

#### 2.5.5.1. Bagging

Bagging (bootstrap aggregating) uses the training samples by implicitly modifying specific *T*-training details by *N*- objects. Additional training sets are regarded as bootstrap, and it modified by using instances that do not occur while others occur multiple time. The concluding classifier $C^*(x)$ is made by combination of $Ci(x)$ where all $Ci(x)$ is an equal vote (Boutaba et al., 2018).

### 2.5.5.2. Adaptive Boosting

Adaptive Boosting (Ada-Boost) method possesses training information. Initially, algorithms allocate *xi* instances completely with equivalent weight. Each iteration *i*, procedure knowledge tries in diminishing weighted fault on trained set by yielding a classifier *Ci(x). Ci(x)* weighted error is designed, also beneficial in notifying weights on the *xi* training instances. The *xi* weight increases by giving effects on classifier's performance allotting high weightiness for misclassified *xi* with an insignificant weight for adequately classified *xi*. Definitive classifier *C*(x)* is done by weighted division of distinct *Ci(x)* interpreting accurateness made on weighted sets of training ( Tan & Gilbert, 2013).

Implementing Kowsari et al., (2019), boosting procedure for datasets, by training strategic multi-models (ensemble learning). These developments give rise to AdaBoost, by presuming $D_t$ construct such that $D_1(i) = \frac{1}{m}$ given $D_t$ and $h_t$:

$$D_{i+1}\{i\} = \frac{D_t(i)}{Z_t} X \begin{cases} e^{-\alpha_t} \text{ if} y_i = h_t(x_i) \\ e^{\alpha_t} \text{ if} y_i \neq h_t(x_i) \end{cases} \quad 2.13$$

$$= \frac{D_t(i)}{Z_t} \exp(-\alpha y_i h_t(x_i)) \quad 2.14$$

Where $Z_t$ positions to the control factor and $\alpha_t$ are stated as;

$$\alpha_t = \frac{1}{2} in(\frac{1-\epsilon_t}{\epsilon_t}) \quad 2.15$$

Ensemble classification methods namely: The Weighted Averaging, Maximum Vote and Average. The Maximum Vote exists (Guzman et al., 2015). Ensemble learning takes four

innovative grouping methods; Stacking, Blending, Bagging, and Boosting (Nisioti et al., 2018).

### 2.5.6. $K^{th}$-Nearest Neighbours (K-NN)

Gene information can be classified using K-NN algorithm, which is known as a supervised learning system, the product of innovative instance request can be classified grounded on mutual K-NN set as shown in Algorithm 2.4. K-NN system utilizes classification locality as an approximation rate of new instance query (Li et al., 2012).

The determination of the K-NN algorithm is classifying new entities grounded on training and attribute samplings. Classifiers do not use appropriate templates but only memory-based ones. Selected functions are considered to be module inputs. $K^{th}$ (sum of nearest neighbours) values are picked contiguous to query idea. Distance is measured among the query-instance and training models, sorted, and the nearest neighbours based on $K^{th}$ lowest distance is discovered. Group *Y* is assembled from the nearest neighbours, and the modest common grouping of the nearest neighbours as a prediction rate of the query instance is utilized. Short bonds may be fragmented randomly (Bose et al., 2016).

**Algorithm 2.4: K-Nearest Neighbour for Source: Wagner et al., (2017)**

```
Input:
 b, the sum of genetic factor.
 a, the sum of cells.
            X, ap×a matrix comprising of the sums for genetic factor.
 k, the sum of neighbours to train.
Output:
            S, ap×n matrix comprising of the sum of genes.
1: process K-NN (b, a, X, k)
2: S= Duplicate(X)
3: phases = [log2(k+1)]
4: for t=1 to phases do
5:          M= Median-Regulate(S) // original b×a matrix
6:          F= Freeman-Tukey-Transmute (M) // original b×a matrix
7:          D= Pairwise-Distance (F) // new a×a matrix
8:          A= Argsort-Rows (D) // new a×a matrix
9:          k step = Min ({2 t−1, k})
10:         for j=1 to a do // blank matrix S
11:               for i =1 to b do
12:                     Sij =0
13:               end
14:         end
15:         for j= 1 to a do // go through all gene cells
16:         for v= 1 to k step + 1 do // go through nearest neighbors1
17:                     u = Ajv
18: for i = 1 to b do // cumulate unique sums for each gene
19:                     Sij = Sij + Xiu
20:                          end
21:                   end
22:               end
23:         end
24:         return S
25:  end process
```

### 2.5.7. Decision Trees

The decision tree is a classifier that divides the instance space recursively through hyperplanes orthogonally to the axes. The model is constructed from origin node representing attributes. The instance space fragmented is grounded on attribute value functions (the split standards are selected inversely for various algorithms), often using its standards. Each original subspace of information is divided into new subspaces repetitive till termination condition, the terminal nodes or leaf nodes are given class labels each representing classification result (class of the instances delimited in subspace). Configuring objective end standard is significant since trees too huge are overfitted, little trees are under fitted and losses inaccuracies. Most processes have in-built mechanisms that handle overfitting called pruning. Individual new instances are classified by crossing them from the tree root down to the leaf, giving the test results and path (Polaka et al., 2010). Even though decision trees yield knowledgeable models and unbalanced – if the training sets of data varies slightly, the resultant models may be dissimilar for two sets. Owing to this fact, decision trees are frequently utilized in classifier ensembles.

## 2.6. Evaluation Measures

Executing malaria vector data analysis in data mining system by utilizing classification algorithms requires, getting the evaluation measures of the output results of a classification confusion matrix, which comprises of the metrics utilized in assessing the classified models, where the model predicts the classes and results (True Positive TP, True Negative TN, False Positive FP, and False Negative FN) (Balamurugan et al., 2017):

True positive (TP): the output for which fully realize appropriately the positive class.

True negative (TN): the product of the model accurately calculates negative class.

False positive (FP): the product of the system wrongly predicted to be positive class.

False negative (FN): the outcome of the system wrongly predicts negative class.

**Accuracy:** The understanding of restrained parameters to the criterion or identified rate is labelled as accuracy. Otherwise, it is quantified as a weighted calculation means of exactness and recall as shown in equation 2.16

$$\frac{TP+TN}{TP+TN+FP+FN} \qquad 2.16$$

**Sensitivity:** Sensitivity is a True Positive Rate known as a fraction of positives which are fittingly predictable, as shown in equation 2.17.

$$\frac{TP}{TP+FN} \qquad 2.17$$

TP signifies the number of true positives appropriately classified; the FN denotes the number of false negatives wrongly classified.

**Specificity:** A true negative evaluates the portion of negatives fittingly predictable, as shown in equation, as shown in equation 2.18.

$$\frac{TP}{TN+FP} \qquad 2.18$$

TN characterizes the number of true negatives correctly classified in a standard percentage; FP denotes the number of false positives incorrectly classifies.

**Precision:** Precision is called the percentage of the recovered case that is correlated to the interrogation and termed for the positive predictive rate (PPR), as shown in equation 2.19.

$$\frac{\text{TP}}{\text{TP+FP}} \qquad 2.19$$

**Recall:** The recall is the amount of the percentage of improved and appropriate instances, also called sensitivity, as shown in equation 2.20.

$$\frac{\text{TP}}{\text{TP+FN}} \qquad 2.20$$

**F-Measure:** The F1 Value is an indication of the performance of a test, identified as relational measures of accuracy and recall, as shown in equation 2.21.

$$2X\frac{Precision\ X\ Recall}{Precision+\ Recall} \qquad 2.21$$

**TP, TN, FP, FN**

The TP, TN, FP, and FN constraints decides and uses the confusion matrix for classifying normal labels in a given dataset.

## 2.7. Related Work

Computational approaches are based on a large inherited dataset of individuals living or not living with ailments, and mutations may be found responsible for the presence of the ailments. Differentially Expressed Genes (DEG) are recognized through various measures. Machine Learning (ML) functions are necessary for identifying variations between genes extracted from the human genome. Many methods have been utilized to examine and classify gene expression profiles of innumerable ailments. Necessity for gene expression profiling approaches of numerous studies of machine learning are deliberated. Quite a lot of investigational approaches by scholars in the aspect of gene expression analysis and machine learning approach discourse, current investigation limitations identified in investigating gene expressions (Karthik & Sudha, 2018).

Bartenhagen *et al. (2010)* worked on the relative analysis of dimensionality reduction techniques for gene expression data investigation. They evaluated PCA method performance with six non-linear dimensionality reduction approaches, such as the Kernel-PCA, LDA, LLE, Diffusion Maps, ISOMAP, Maximum Variance Unfolding and Laplacian Eigenmaps, using visualization of gene expression data.

Tan and Gilbert (2013) proposed an ensemble classification algorithm for classifying cancerous gene expression data, using C4.5, bagged and boosted ensemble classifiers for cancer, using seven carcinogenic gene expression data and correlated the classification performances. They noticed bagged ensemble learning and boosted decision trees enhanced more efficiently than the C4.5 decision tree classifier. Their findings suggested

that classification approach such as ensemble learning performs better than decision trees with 92% accuracy. Their study suggested single supervised machine learning approaches.

Xintao & Dongmei, (2014), proposed an effectual dimensionality reduction approach for sampling sizes and high dimensionality data model. Validating and simulating practical information demonstrated that the innovative dimensionality reduction procedures could be efficiently useful for examining and modelling atmospheric decomposed data. The feature selection fetches for optimal feature subset, and the feature extraction technique retains the unique form, categorizes data, and the integrity of the data. Their study proposed a comprehensive information dimensionality reduction result which efficiently explains the high-dimensional unimportant model data difficulties. Their study handled very small high dimensional dataset.

Pierson and Yau (2015) developed a zero-inflated single-cell gene expression dimensionality reduction data study using Zero Inflated Factor Analysis (ZIFA). It modelled the dropout uniqueness. it advances the performance on simulation and biological sets of data. They tested the interpretation of ZIFA compared to PCA, Probabilistic-PCA (P-PCA), Factor Analysis (FA) then non-linear methods containing the Stochastic Neighbor Embedding (t-SNE), ISOMAP, and Multi-dimension Scaling (MDS). Simulated sets of P-PCA-FA data model was generated with dropout models. Their study proves to be highly effective yet requires potential applicable correlated feature algorithms.

Simmons *et al. (2015)* worked on the discovery of what dimensionality reduction talks about RNA-Seq data, by introducing a fusion method component selection using mutual information (CSUMI) which is a statistical approach to re-explain the outcomes of PCA in

an expressive biological method. They applied CSUMI on RNA-seq data gotten from the GTEx repository. Their method allows unveiling earlier concealed association among principal components (PCs) and fundamental genetic and methodological derivation of difference across models. They also worked on how muscle type distresses PCs and developed principled means of selecting PCs to study when discovering the information. They supplemented and applied their method to RNA-seq brain information and demonstrated that most biological revealing PCs are high-dimension PCs; for example, PCs can distinguish the basal-ganglia from added muscles. They used CSUMI to discover how procedural objects disturb the comprehensive data structure, confirming preceding outcomes and indicating how their technique can be seen as an authentication outline for distinguishing undetected preconceptions in evolving skills and relate CSUMI to two correlated-based methods, outperformed both using python execution accessible online on the CSUMI internet site. Their results prove a better performance using traditional PCA methods with principal components, yet requires a structured data.

Pamukçu et al., (2015). proposed an innovative hybrid dimensional reduced method for small protein sequence cancer data expression complexity criterion classification by addressing limitations of Probabilistic-PCA (P-PCA) by presenting and evolving an innovative and new method with determined entropy covariance matrix hybridizing smooth estimators. Reducing the dimensional data and choosing numbers of probabilistic-PCs (P-PCs) retained and proceeded to present and advanced renowned Akaike Information Criterion (AIC), Information Theoretical Measure of Complexity (ICOMP) principle of Bozdogan and Consistent Akaike Information Criterion (CAIC). Six publicly accessible small-scale sets of data were investigated to display the usefulness, flexibility,

and adaptability of their method with hybrid smooth covariant matrix estimator, that does not deviate the performance of P-PCA to diminish dimension and perform supervise high dimensional cancer group classification. The suggested technique can resolve innovative difficulties and tasks existing in Next Generation Sequence data analysis in bioinformatics applications.

Arowolo et al., (2016) worked on ANOVA feature selection procedure for classifying gene expression data, by combining the algorithm; to diminish high data dimension of feature spaces and SVM classification algorithms for reducing computational complexity and effectiveness. Noises and computational burden arising from redundant and irrelevant features are eliminated. It reduces gene expression data, which can drop the cancer testing cost significantly. The proposed approach selects the most revealing subset of features for classification to obtain a high-performance.

Lenz et al., (2016) worked on PCA and stated low fundamental dimension of gene expression data. They reassessed the method and demonstrated that the fundamental linear dimension of the total record is more complex than earlier described. Also, they examined cases where PCA failed to spot applicable biological data and indicated researchers to approaches that overwhelmed the limitations. Their outcomes improved the present general structure with understanding of the gene expression space and demonstrated that PCA hinge on curtly on the result dimensions of genetic signs and segment of signal samples.

Song et al., (2016) proposed the design of an investigative ensemble classification method for cancer gene expression data by combining Recursive Feature Elimination (RFE) with

Ada-boost process to pick significant features for classification with the enhanced outcome. The gene subset obtained has strong classification discriminative capability. To a certain degree, the ensemble approach increases the efficiency of the SVM classifiers. Feature subset selection with the features is extracted having terminal outcome on the gene classification issue. It was observed that the output of the Adaboost grounded on the SVM is improved than the Adaboost based on decision-trees through the ROC curve. However, some positive results were accomplished with the SVM-based ensemble approach. But those outcomes are so poor. If the SVM's output on some data is better, then the ensemble is useless.

Wenyan *et al.* (2017) worked on feature selection for cancer classification for disease utilizing microarray data expression. This study used information on gene expression level to decide marker genes pertinent to sort of malignancy. They researched on separation-based element choice strategy for two-gathered grouping issue. To choose marker genes; Bhattacharyya separation was actualized in quantifying uniqueness in gene expressed levels. They used SVM for classifier with particular marker genes. Execution of marker genes selection and classification were represented in recreation study and genuine information analyses by proposing an innovative gene selection technique for SVM classification. They firstly ranked every gene according to the importance of Bhattacharyya distance among indicated classes. Optimal gene subsets were chosen to accomplish the least SVM misclassification rate ensuing from a forward selection procedure. 10-fold cross-validation was connected in locating optimal SVM parameters with the concluding optimal gene subsets. Subsequently, the classification model was trained and constructed. The test set prediction estimated the classification model. The execution of the suggested

B-SVM technique with SVM-RFE and SWKC-SVM gives normal 1% misclassification rate and 96% high normal recovery rate.

Tarek et al., (2017), proposed a cancer gene expression data classification approach using an operational ensemble classifier that raises the performance of the classifier and the result poise. Ensemble classification results were less dependent on the originalities of individual train sets. The reasons other than using ensemble classifiers are that results are less dependent on the intricacies of a specific training set and that the ensemble approach outperforms the performance of the better class label in the ensemble. The following method resulted in a quick and sufficient process that outdoes the ensemble solution offered. The K-NN classification model was also used as a core member to boost the consistency of the result, cover more types of cancer, and minimize the effect of over-fitting in this study. Ensemble classifiers may be used in future work and can be extended to other multi-class datasets benchmarks.

Zhengyan & Chi, (2017) worked on the classification of lungs glandular cancer and squamous cell carcinoma RNA-Seq data. They used a gene expression profile to discriminate NSCLC patient's subtype. They leveraged RNA-Seq information from the-cancer-genome-atlas (TCGA) and randomly split the data into training and testing subsets. Constructing classification based on data training, we considered three methods were considered: logistic regression on Principal Components (PCR), logistic regression through LASSO reduction (LASSO), and Kth Nearest Neighbors (K-NN). Performances of classifiers were estimated and equated based on the testing data. Results: All gene expression-based classifiers show high accuracy in discriminating LUSC and LUAD. The classifier obtained by LASSO has the smallest overall misclassification rate of 3.42% (95%

CI: 3.25%-3.60%) when using 0.5 as the cutoff value for the predicted probability of belonging to a subtype, followed by classifiers obtained by PCR (4.36%, 95% CI: 4.23%-4.49%) and K-NN (8.70%, 95% CI: 8.57%-8.83%). The LASSO classifier also has the highest average areas under the receivers operating characteristics curves of 0.993, compared to PCR (0.987) and K-NN (0.965). Their results suggest that mRNA expressions are highly informative for classifying NSCLC subtypes and may potentially be used to assist clinical diagnosis.

Balamurugan *et al.,* (2017) worked on Alzheimer's ailment analysis using a reduced dimensional technique based on K-NN classification algorithm. An innovative dimensionality reduction-based on K-NN algorithm for Alzheimer ailment and minor cerebral damage currently in the datasets were presented. The unvarying dataset is used to analyze the clinical and statistical evidence. Their result gives more accuracy, sensitivity and specificity percentages.

Usman et al., (2017) used PCA and Factor Analysis (FA) for dimensional reduction of gene expression leukaemia dataset, and the number of features were reduced. A study was carried out on reducing the number of features using PCA and FA. PCA was used on the data with nine selected components. FA was used in extracting significant features found to be significant attributes.

Zararsiz *et al.* (2017) proposed a broad simulation RNA-Seq data classification learning by comparing multiple classifiers with P-LDA having and lacking power transformation, NB-LDA, single-SVM, bagging-SVM, CART, and Random-Forests. They observed results of numerous constraints like; over-dispersion, sampling sizes, numbers of genetic

factor, numbers of modules, differential expression amount, and transformation methods on performance models were carried out, and the outcomes were reported. Their results showed cumulative sample sizes, differential expression rates and reducing dispersion parameters with group numbers resulting in increased accuracy in classification. Comparable with differential expression training, RNA-Seq data classification needs careful consideration in conducting data over-dispersion. They concluded that a count-based classifier control changed P-LDA and gene expression-based classifier and changed RF and SVM classifiers can be better classification option. Investigations were done based on scarce P-LDA classifiers and proves to be the best genes subset used in classification. The scarce P-LDA classifies subsequently when a power revolution is made accurately in all dispersal settings. Extending the NB-LDA into sparse classification process improves the classification performance by picking the most important features of the genes. Furthermore, an alternate choice is bringing the data closer to gene expression profiles and its classifiers. Their results discovered RF, SVM and Bagging-SVM gives accurate results, and its efficiency is enhanced evidently with cumulative sample size.

Rostom et al., (2017) worked on computational approaches to interpret sc-RNA-Seq data, by considering genetic queries for RNA - seq data, at cell with the gene levels. They defined the available tools for evaluations. A fast-developing field, in which clustering, pseudo time extrapolation, splitting inferences and analysis of gene- level are predominantly revealing computation analysis aspects. As the single cell information remains to produce a unique stride and the data generation that accompanies it. It is imperious to build tools and arithmetical approaches to interpret the information in the greatest conceivable technique, removing relevant and informative genetic sense.

Lin et al., (2017) used neural networks approach to reduce dimensions of sc-RNA-Seq data comprising numerous innovative computational challenges, including inquiries about best approaches for grouping sc-RNA-Seq data, identifying a unique cluster of experimental cells and determining the state of precise cells based on their expression profile. They developed and tested their experiment using neural networks (NN) algorithm for analyzing and retrieving sc-RNA-Seq data, with integrated prior genetic data, for obtaining reduced dimensional illustration of the single-cell gene expression data. They showed that NN technique advances over preceding approaches, the capability of appropriately grouping cells in tests not conducted in the training and capability to appropriately gather cell types by interrogating thousands of single cells profiling databases, enabling researchers to improve in characterizing cells when investigating diverse sc-RNA-Seq samples. The technologically advanced and tested results using deep neural networks using numerous NN architectures, with designs inhibited by prior biological knowledge. NN achieved relevant data classification performance and improved the prior approaches used in clustering the sets of data from researches, not training. They did an efficient analysis of extremely weighted nodes for individual cell type and presented that NN is labelled as a learning system.

Oh et al., (2017) proposed an evaluation of Autism range disease gene expression machine learning approach in identifying transcriptions used in classification with an RNA data. They used rank cluster investigation moderately. SVMs and K-NN classifiers were adopted to confirm the results of the data in a complete class estimate accuracy of 94%.

Jolliffe & Cadima, (2016) worked on the review and recent developments in PCA as a method for reducing RNA-Seq data dimensionality, for growing the interpretability and

diminishing information damage by generating innovative uncorrelated variables that continuously exploit variances. Discovering innovative variables, principal components lessen to resolving an eigenvalue, and innovative variables are distinct by the dataset used and not by apriori, it makes PCA an adaptive analytical method. Since variations of the method have been advanced that are channeled to numerous diverse data type structures, their study discussed concepts of PCA and described some variations and their application. The relative investigations among current NN classification procedures with K-NN classification demonstrated that high metric performance furthermore diminishes the information dimensionality and multifaceted computational nature. Their impending work stated that the feature extraction and classification improve classification performance. They reviewed recent ongoing advancements in PCA as an approach for diminishing the RNA-Seq data dimension, for increasing interpretability and yet preventive data disaster by creating innovative uncorrelated features that increasingly exploit variances. Their study offered an important opinion for PCA.

Wang & Gu, (2018) proposed an sc-RNA-Sequential data using a deep variation autoencoder using an unsupervised feature extraction model. The VASC models the dropout and fetches non-linear ranked feature representation of high dimensional data. Their result was tested on 20 sets of data. The VASC showed a better performance with broader compatibility features.

Lee *et al.,* (2018) worked on transcript training of malaria infections by studying general host-pathogen relations and reviewing contributions of transcript training to understand the infection of malaria, which is a bloodsucking virus retaining a key impact on human development and remains a cause of enormous worldwide disease problem. They studied

the malaria model for transcriptomic evaluation of general host-pathogen connections in human, due to most host-pathogen communication happens in the blood, a voluntarily tested section of the body. They demonstrated training learnt from malaria transcript studies and guides for studying host-pathogen connections in some other transmittable infections. They proposed that the latent of transcriptomic training in improving malaria understanding as a disease remains comparatively unexploited because of restrictions in learning designs relatively than consequences of scientific limitations. Additional developments will entail the combination of transcriptomic information with diagnostic methods from further methodical corrections, with epidemiology and scientific modelling.

Becht et al., (2018) investigated on the dimension reduction model for imagining single-cell data with non-linear dimension reduced method unvarying multiple approximation and prediction (UMAP) conventional for investigating high dimension data. They applied UMAP to biological data, with mass cytometry and Sc-RNA-seq datasets. Associating UMAP with some other tools, they discovered that UMAP delivers a faster run time, higher reproducibility and better expressive cell cluster organizations. Their work demonstrated UMAP usage for enhanced concept and single-cell data clarification.

Ding *et al.,* (2018) studied an interpretable dimensional reduced model of single-cell transcriptomic information through deep propagative models by working on a robust model called the SCVIS. This captured and showed the lower dimensional structure in the single-cell expression of gene data. A simulated demonstration of the lower dimensional data was presented, which preserved the limited and global structures in the data. They used savings in analyzing four Sc-RNA-Seq datasets, demonstrating interpretable two-dimension illustrations of the high dimension Sc-RNA-seq data.

Wenric & Shemirani, (2018) proposed a supervised learning method for an assortment of RNA-Seq genetic factor by grading huge gene ensembles with RNA-Seq, using variable rank procedures generated by random forest classifier and distinct the EPS (extreme pseudo-samples) frequency, with variation autoencoder and regressor to abstract ranks of RNA-Seq data samples. Their outcomes showed the hidden supervised learning gene selection methods in RNA-Seq training and determined the need for using gene selection methods on gene expression analysis.

Reid et al., (2018), proposed an RNA-Seq exposing unseen transcriptions in malaria flies by unfolding the discrepancy of RNA-seq process in deconvoluting transcript differences for about 500 different pests and malaria in human. They revealed hidden distinct transcript signatures during the pathogenic portion of the life cycle, indicating the over-expression progress is not as continuous as is widely assumed. We find novel, sex-specific functions in transmission stages for the differential expression of families of contingency genes that are typically related with insusceptible elusion and pathogenesis.

Xu et al., (2018) worked on feature selection of genes using supervised learning LLE and correlated coefficient approach for gene expression data classification. Collection of feature genetic factor through high recognition skill from gene profiling has increased with great meaning in biology. Most existing approaches have high time complication, and classification presentation is poor. This study proposed an operative dimensionality reduction technique, termed a supervised LLE and spearman rank correlation coefficient (SLLESC2), based on LLE and correlation coefficient systems. S-LLE took into justification class marker statistics and recovered the classification, and the spearman rank correlation coefficients eliminated co-expression genes. Their research outcomes got four

public cancer gene expression data illustrations that their technique is effective and achievable.

Alquicira-Hernandez et al., (2019) suggested a gene expression data classification with a supervised model by presenting a generalized method through much accurate single cell's classification, using combined unbiassed feature selection algorithm from dimension reduced space, also machine learning evaluation procedure. They used sc-Pred on RNA-seq data from a mononucleate cell, pancreas tissue, colorectal lump surgeries, and circular dendritic cell. They presented sc-Pred discrete cells having higher classification accuracies.

Cui et al., (2019) proposed a machine learning RNA-DNA investigation indicating low stated genomes that mutually influences PAH virus utilizing an unconventional feature selection and improved machine learning technique to classify an unrelated set of helpful genes. Results presented a cluster of small gene expression revealing prediction and unique transformed PAH.

Mohan & Nagarajan (2019) proposed an enhanced tree classification model, using an ensemble-based feature selection approach with random trees and a feature selection wrapper-based method to improve the classification. It initiates a subclass with bagging-ensemble, wrapper scheme, and random tree. Their approach removed unrelated features and picked the best features for classification with a probable weighting value. The feature selection procedure is assessed using Random Forest, SVM, and Naïve Bayes assessments and related their performance with GA-SVMb, GA-NBb, FS-NBb, FS-SVMb, and GA-RFb approaches. The method achieves a classification 92% accuracy.

Shon *et al.*, (2019) suggested a classification gene expression stomach malignancy data with CNN classifier. He demonstrated its gene data expression contracted from abdominal malignant patients were evaluated by PCA, heatmaps, and CNN algorithms. RNA-seq gene expression data scrutinized genetic factors and evaluated through CNN deep learning procedure with an accuracy of 96% and 51% achieved.

Kowsari *et al.*, (2019), performed a text algorithm classification survey, on several text dimensionality reduction methods, they presented several classification algorithms approaches, and evaluations. This study includes numerous extractions of text attributes, methods of dimensionality reduction, current procedures with systems of evaluation. Finally, it addresses drawbacks of individual practices with their application to practicable issues.

Chen et al., (2019) proposed Sc-RNA-Seq knowledge and its relating computational analysis. In their review, they provided an outline of presently obtainable single-cell procedures and discussed several techniques for several RNA-Seq Data analysis such as their gene expressions, mapping, cell clustering, imputation, normalization, feature selection, feature extraction, among others.

Luecken & Theis, (2019), studied on the current practices in Sc-RNA-Seq analysis, by formulating present best- training endorsements for stages based on self-determining assessment studies. They combined these training references into a workflow, applied to a free dataset to demonstrate its training. This review serves as a plan lesson for innovative participants, and aid conventional users keep informed their investigation conduits.

Townes et al., (2019) worked on feature selection and dimensionality reduction model for single-cell RNA-Seq model by proposing a modest polynomial method, with comprehensive-PCA used for abnormal dispersals, and feature selection with unconventionality. These approaches outdo the present training in a clustering evaluation using real datasets. Simple multinomial methods for non-normal distributions, with general principal component analysis (GLM-PCA), and selection of features using deviance have been suggested. In a down-stream clustering test with pulverized fact data, these techniques outperform the current norm. They were not equivalent to genes with high dropouts.

Howick et al., (2019) worked on profiling single-cell transcriptomes of different parasites, originating from the initial high-resolution transcript charts of the whole *plasmodium* life sequence. They used the charts to exactly describe evolving phases of single cells from three diverse malaria parasite types, with parasites secluded straight from diseased persons. The Malaria cell charts offer both a wide-ranging understanding of gene practice in eukaryotic parasite also an open-access situation data for learning malaria parasites.

Sun *et al.,* (2019) investigated on performance metrics of dimensionality reduction approaches for Sc-RNA-Seq investigation, by offering a comparative overview of several widely used methods of dimension reduction for sc-RNA-seq studies. In specific, on 30 freely available sc-RNA-seq datasets covering varieties of sequencing methods and sample sizes, they compared 18 different dimensionality reduction approaches were compared. They assessed the efficiency of various neighborhood-preserving dimensionality reduction approaches in relations of capability to restore features of the innovative expression matrix, and their precision and robustness for cell clustering and extraction reform. They also test

the statistical scalability of various methods of reducing dimensionality by documenting their computational cost.

Tamim et al., (2019) worked on a comparative analysis of programmed cell ID approaches for sc-RNA-Seq data. They benchmarked 22 classification approaches that repeatedly allot cell characteristics with single-cell-specific and all-purpose classifiers. The presentation of the approaches was assessed with 27 openly accessible single-cell RNA-Seq data of diverse dimensions, knowledge, classes, and stages of intricacy. They used two experimental configurations to approximate the performance of each system based on precision, the proportion of uncategorized cells and calculation time for dataset projections (intra-dataset) and datasets (inter-dataset). They tested the sensitivity of the approaches to input features, several cells per populace and their output over numerous annotation dataset levels. Huge datasets with overlying groups or deep annotations, noted classifiers work fine on datasets with reduced accuracy. The general-purpose benefit SVM classifier has the highest overall results over the various experiments.

Qi *et al.*, (2020) investigated the clustering and classification models for Sc-RNA-Seq data. In their study, they analytically reviewed combined approaches and tools, stressing the benefits and ploys of respective methods. They paid attention to clustering and classification methods as well as discussing approaches that have happened recently as predominant substitutes, with non-linear and linear methods and reducing dimension approaches. They emphasized on clustering and classification approaches for sc-RNA-seq data, combined approaches, and deliver a complete account of sc-RNA-seq information and URLs.

Feng et al., (2020) worked on dimensionality reduction and clustering for RNA-Seq data, by conducting reviews on conventional dimension reduction approaches and clustering representations. Four types of research were achieved on two huge RNA-seq datasets with 20 models. Feature selection technique contributed to sparse high-dimensional RNA-seq data. Feature extraction approaches help clustering performance but not ceaselessly unchallengeable. ICA did fine on tight feature spaces, PCA was securer than other approaches. ICA remained non perfect for fuzzy C-means clustering approach in analyzing RNA-seq datasets. K-means clustering was shared with feature extraction approaches to realize better outcomes.

Several techniques have been suggested in literature for dimensionality reduction and classification. Several limitations have been addressed as earlier revealed in Table 2.1 and Table 2.2.

From the related work reviews, it has been observed that several approaches have been exploited by researchers to develop dimensionality reduction models for gene expression analysis; these include machine learning approaches such as the clustering, dimensionality reduction, classification and hybrid approaches. The hybrid approaches have become a trend in recent time (Songyot, 2019). Many applications have proved that using more than a single dimensionality reduction approach in gene expression analysis task can lead to an important enhancement of the performance of the overall system. Proposals for improving gene expression analysis such as the combination of multiple dimensionality reduction models (Almugren & Alshamlan, 2019) or hybridization of classifiers (Singh, 2018) delivers desired results of the systems.

Since the combination of multiple dimensionality reduction techniques is justified, it was considered that dimensionality could be further reduced using optimization feature selection method such as Genetic Algorithm with feature extraction approaches such as the PCA and ICA, as better approaches to advance classification performances, in terms of computational metrics among other beneficial performance metrics.

### 2.7.1 Summary

This chapter presented the essential ideas of this study; basically, machine learning approaches for RNA-Seq gene expression, dimensionality reduction and classification approaches with prevailing reviews in the literature to determine the gaps for the study.

Machine learning is a key aspect of several classification algorithms for solving problems. Numerous literatures aim to eliminate redundancy and relevance from the data. Integration of dimensionality reduction algorithms is the better way to find dataset accuracy, to solve the class imbalance problems, two techniques are normally implemented in the literature, the first is to analyze specific genes, another is to find an optimum solution subset for good classification accuracy and solve the high-dimensional data problem. This study finds the dataset classification performance of malaria vector by combining dimensionality reduction algorithms, to fetch relevant information that will enhance the criteria and eliminate redundancy as well as irrelevant genes. The next chapter gives a detail report on the methodology implemented.

# CHAPTER THREE

## 3.0 METHODOLOGY

This chapter discusses the research methodology process and strategies that outlines how the study is to be conducted, including the identification of methods to be used. The tools and technologies used in the process of dimensionality reduction and classification algorithms are presented. The features of the model are shown with necessary procedures.

## 3.1. The Dataset

In this study, the RNA-Seq dataset for the malaria vector, *Anopheles gambiae* is publicly available. It was retrieved from a biomedical data repository for malaria vector (Bonizzoni et al., 2015). The data contains 2457 significant genes instances between field-caught resistant and susceptible mosquitoes from Western Kenya in 2010 and 2012. The data comprises of 7 genes attributes Tests, Genes, Locus, Sample Resistants, Sample Susceptible, and Status (Bonizzoni et al., 2015). The first is the predictors while the rest are the labels. Figure 3.1 highlights the data sample details. The training, testing, modelling and developments were achieved using MATLAB 2015a.

| | A | B | C | D | E | F | G | H | I | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Additional File 4A. List of the 2457 genes significantly DE between field-caught resistant and susceptible mosquitoes | | | | | | | | | |
| 2 | test_id | gene_id | gene | locus | sample_1 | sample_2 | status | value_1 | value_2 | R/S |
| 3 | XLOC_007931 | XLOC_007931 | ECH | 3L:3546074-3546412 | Resistant | Susceptible | OK | 0 | 1.07269 | 7 |
| 4 | XLOC_008163 | XLOC_008163 | CPFL2 | 3L:12824716-12825469 | Resistant | Susceptible | OK | 0 | 0.647051 | 7 |
| 5 | XLOC_009575 | XLOC_009575 | AGAP008752 | 3R:17088639-17092062 | Resistant | Susceptible | OK | 0.64351 | 82.1675 | -127.6864 |
| 6 | XLOC_003479 | XLOC_003479 | AGAP001970 | 2R:12992452-12993988 | Resistant | Susceptible | OK | 1.38726 | 122.932 | -88.61496 |
| 7 | XLOC_010757 | XLOC_010757 | CPLCG14 | 3R:10894980-10895533 | Resistant | Susceptible | OK | 0.179707 | 15.7186 | -87.46793 |
| 8 | XLOC_002148 | XLOC_002148 | CPR23 | 2L:24621231-24621964 | Resistant | Susceptible | OK | 1.04442 | 76.6002 | -73.34233 |
| 9 | XLOC_011617 | XLOC_011617 | CPR83 | 3R:49131809-49132540 | Resistant | Susceptible | OK | 0.252442 | 17.6994 | -70.11273 |
| 10 | XLOC_009418 | XLOC_009418 | CPLCG15 | 3R:10897682-10898268 | Resistant | Susceptible | OK | 1.23697 | 52.8 | -42.68494 |
| 11 | XLOC_005818 | | | 2R:26667141-26668059 | Resistant | Susceptible | OK | 0.0696753 | 3.76306 | -42.19511 |

Explore more content

2457 RvS | 182 constitutive DE genes | 55 candidate resistance genes

13071_2015_1083_MOESM4_ESM.xlsx (394.48 kB)

MD5: 2744095f111bde11002c216b6db36438

**Figure 3. 1 Data Sample for Mosquito *Anopheles gambiae***

## 3.2. Research Design

Machine learning is an aspect of computer science, it requires applications to advance innumerable computer approaches learned from trained data. In this thesis, a hybrid dimensionality reduction model is suggested for classifying malaria vector data. Dimensionality reduction entails the feature selection and feature extraction approaches. The feature selection fetches out relevant information from the huge dimensional data utilizing an optimized Genetic Algorithm approach. The feature extraction algorithm realizes the latent components from the reduced data using the PCA and ICA learning approach individually, to differentiate the efficiencies of linearity and non-linearity learning methods respectively (Bhattacharyya & Kalita, 2013). The knowledge contingent from these studies is carried out to classify the unidentified individuals (test individuals) consequently by using four types of classifiers known as the SVM, K-NN, Decision Tree and Ensemble procedures.

The main goal of applying machine learning approach into RNA-seq is the accurate class label predictions of given samples, based on their expression profile. Thus, RNA-seq is an excellent field for applying machine learning.

The anopheles' dataset is an online resource consisting of gene profiles of infections for a total of 2457 gene samples of RNA sequence studies of humans containing a certain number of samples based on its experiment. The raw sequencing data are processed for ease of analysis. These information deliver a rich means that researchers can adopt for detection validation, imitation, or method development. This study analyzes this dataset to identify suitable studies to assess the performance of classifications. Relevant criteria are

considered to retain several classes: for most of them, this study attempted to have more than two classes, after the clear identification of relevant biological classes.

### 3.2.1. Research Design Layout

High dimensional dataset of RNA-Seq malaria vector is used in this study as an experiment in MATLAB environment. The dataset is passed into a feature selection technique (Genetic Algorithm) to realize relevant information in the dataset, the feature extraction method which helps in extracting the optimal information in the pre-processed data by using PCA or ICA feature extraction methods on the selected features in the dataset before results are generated by classification algorithms (SVM, K-NN, Decision Tree and Ensemble). Based on the nature of classifiers selected, the parameters of each of the classifiers will be tuned, and the generated models will be tested accordingly. The result is analyzed to examine the effect of dimensionality reduction models on classifiers.

Four main steps were undertaken to evaluate and assess the validity of the approaches used in this study:

i. Database selection
ii. Pre-processing using dimensionality reduction procedures
iii. Classification
iv. Evaluation

### 3.2.2. Feature Selection

Feature Selection is the method of selecting specific features of the candidate subset of features that are uncorrelated (Parimala & Nallaswamy, 2011). It aids minimization of dimensions, and increase classification quality and accuracy (Chen et al., 2010). It recognizes the most important fields in predicting the certain result. In this work, the feature selection module uses a Genetic Algorithm on the Anopheles RNA-Seq datasets for training purpose, to skip unimportant attributes from the dataset. Genetic Algorithm is used to eliminate arbitrary features, prevent high data dimensionality and poor classification accuracy by using a genetic algorithm as a selector function. The selection of the features often attempts to select the optimum subset comprising of m characteristics chosen from the total of n characteristics. The dataset is transformed to the classifier framework after removing insignificant attributes and it is then divided into two parts: training and testing data, respectively. Selection of features is an important stage in the development of a machine-learning classification in innovations like RNA transcript for producing valuable discrete identifiers for transcript sequences for training and testing models. The selection of features allows to choose suitable essentials to be implemented in classification models and to remove unrelated and redundant features in order to diminish the dimensionality curse. It helps to make the learning process of the classification step efficient and reinforces the model of success. For example, the selection procedure for extensive data features; RNA-Seq data includes supervised and unsupervised learning of decision making. Rank features that confer relevance are critical for classification problems, and selecting the best will enhance the effectiveness of the prediction model.

Feature selection is classified into active methods recognized as the Filter, Wrapper and Embedded methods. To achieve this objective, this study identifies relevant features for RNA-Seq Malaria vector data prediction using Genetic Algorithm Optimization (Kleftogiannis et al., 2013).

### 3.2.2.1 Genetic Algorithm

GA is an iterative method for selecting wrapper-based features used to explore system optimization complications. In the retention of the fittest basis, GA can be based on real actions relevant to public genes. GA covers population size advance, fitness valuation, parent selection, crossover and mutation. (Soufan et al., 2015). Figure 3.2 depicts the traditional flowchart for the genetic algorithm.

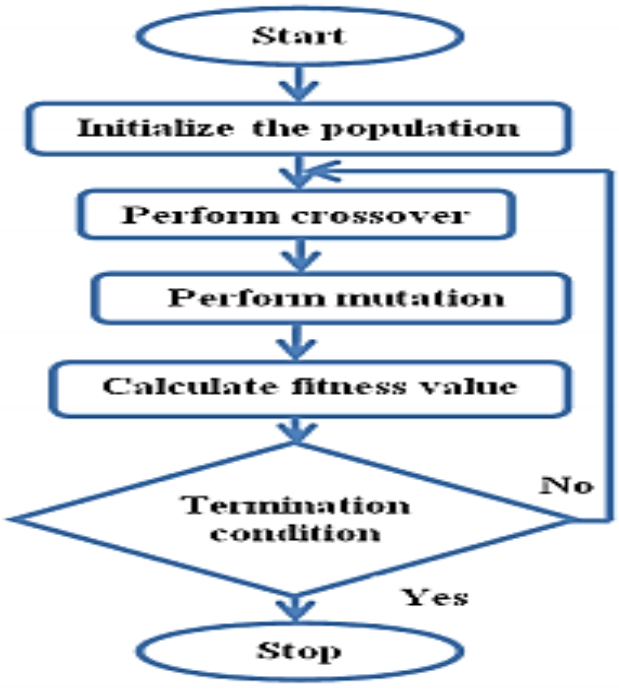



**Figure 3. 2 Flowchart Representation of Genetic Algorithm**

**Source: Asir et al., (2016)**

In enhancing the genetic algorithm as a wrapper-based feature selection approach, several operations have been performed. The basic steps taken for genetic algorithm is shown in algorithm 3.1.

**Algorithm 3.1. Genetic Algorithm**

```
Necessitate: Set parameters nPop = m, tmax, t = 0;
Confirm: Optimum feature subset with the maximum suitable
rate.
    1: while (t<=tmax) do
    2:    Create pop a, tmax;
    3:    For k = 1 to a do
    4:          Parents [a1, a2] = system selection (a, nPop)
    5:          Child = Xor[a1, a2]
    6:          M u = mutation [Child}
    7:    End for
    8:    Replace a with Child1, Child2, …, Childm
    9:    t = t+ 1;
    10:   End while
    11:   Save the Highest fitness value;
```

a = population size, r = random number 0 to 1, chrome = certain or non-certain feature through threshold δ, set value = 0.5, and α = threshold number of picked features. Selecting maximum fit features from the predictable datasets is the main problem of the GA technique.

In this study, four major phases are involved, and the experimental workflow is displayed in Figure 3.2.

To obtain objective one, feature selection using an Optimized Genetic Algorithm method is used on the Anopheles mosquito dataset containing 2457 instances and 7 attributes, the loaded data is depicted in Figure 3.4 which shows relevant information fetched from the dataset using the procedure in Figure 3.3.

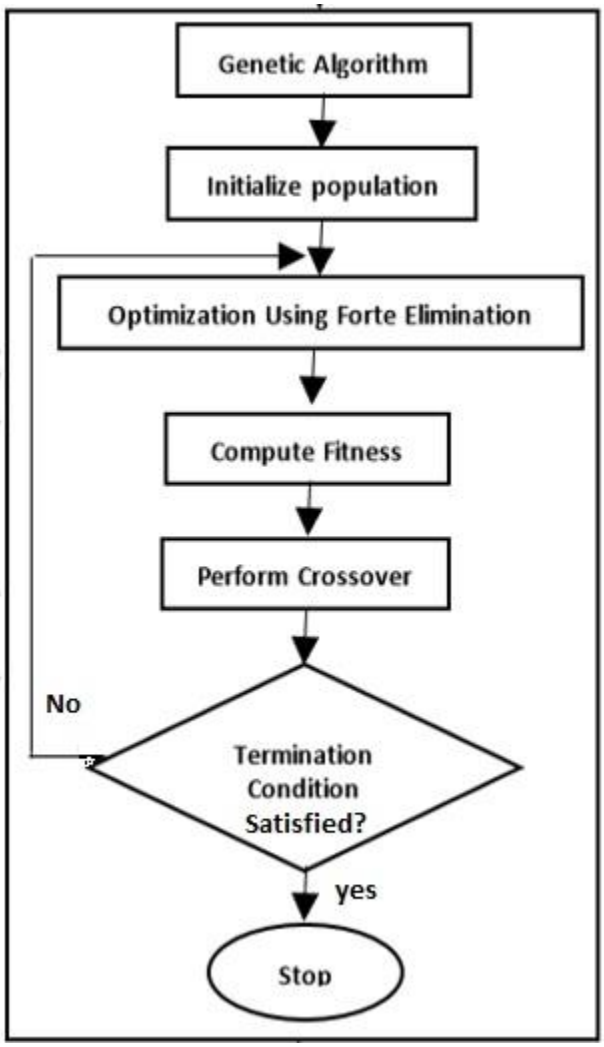



**Figure 3. 3 Flowchart for the proposed Feature Selection Technique**

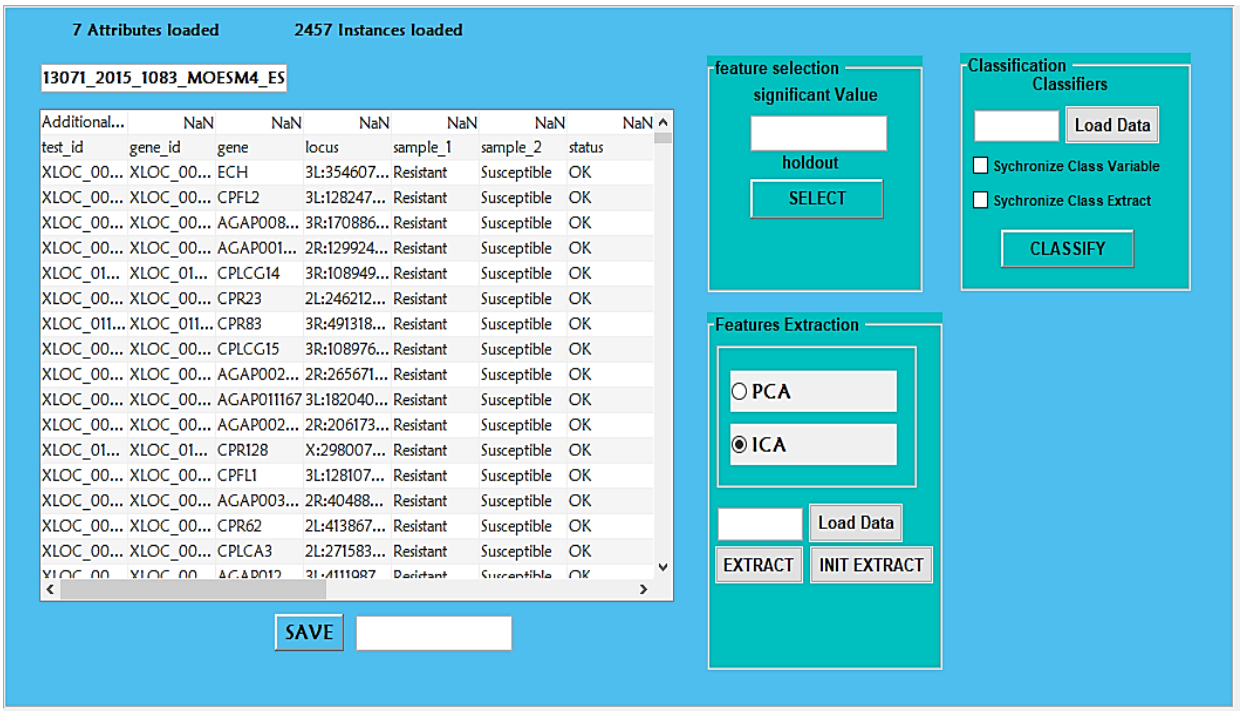
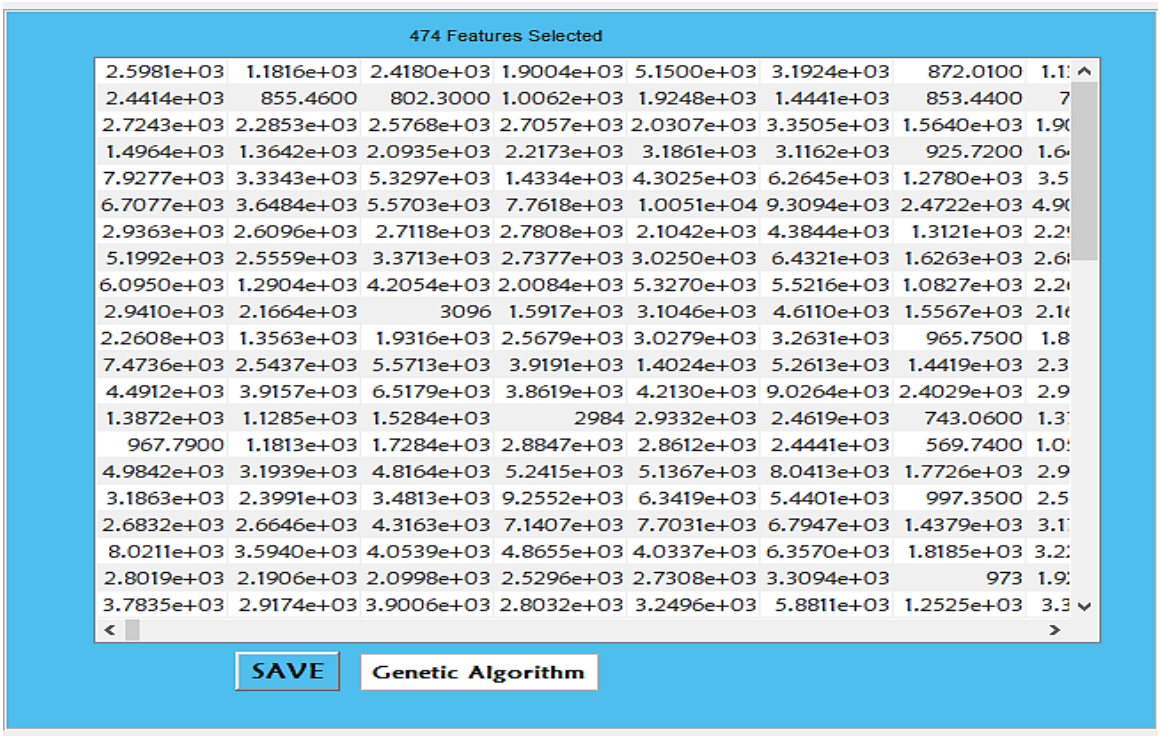



**Figure 3. 4 RNA-Seq Anopheles Dataset Attributes before Feature Selection**

In this experiment, an optimized genetic algorithm is developed as a feature selection technique to fetch relevant features from the original dataset and was mapped onto a reduced dimensionality space, by selecting the subset of the unique features, 474 features were selected based on the selection principles and shown in Figure 3.5.




| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2.5981e+03 | 1.1816e+03 | 2.4180e+03 | 1.9004e+03 | 5.1500e+03 | 3.1924e+03 | 872.0100 | 1.1 |
| 2.4414e+03 | 855.4600 | 802.3000 | 1.0062e+03 | 1.9248e+03 | 1.4441e+03 | 853.4400 | 7 |
| 2.7243e+03 | 2.2853e+03 | 2.5768e+03 | 2.7057e+03 | 2.0307e+03 | 3.3505e+03 | 1.5640e+03 | 1.9 |
| 1.4964e+03 | 1.3642e+03 | 2.0935e+03 | 2.2173e+03 | 3.1861e+03 | 3.1162e+03 | 925.7200 | 1.6 |
| 7.9277e+03 | 3.3343e+03 | 5.3297e+03 | 1.4334e+03 | 4.3025e+03 | 6.2645e+03 | 1.2780e+03 | 3.5 |
| 6.7077e+03 | 3.6484e+03 | 5.5703e+03 | 7.7618e+03 | 1.0051e+04 | 9.3094e+03 | 2.4722e+03 | 4.9 |
| 2.9363e+03 | 2.6096e+03 | 2.7118e+03 | 2.7808e+03 | 2.1042e+03 | 4.3844e+03 | 1.3121e+03 | 2.2 |
| 5.1992e+03 | 2.5559e+03 | 3.3713e+03 | 2.7377e+03 | 3.0250e+03 | 6.4321e+03 | 1.6263e+03 | 2.6 |
| 6.0950e+03 | 1.2904e+03 | 4.2054e+03 | 2.0084e+03 | 5.3270e+03 | 5.5216e+03 | 1.0827e+03 | 2.2 |
| 2.9410e+03 | 2.1664e+03 | 3096 | 1.5917e+03 | 3.1046e+03 | 4.6110e+03 | 1.5567e+03 | 2.1 |
| 2.2608e+03 | 1.3563e+03 | 1.9316e+03 | 2.5679e+03 | 3.0279e+03 | 3.2631e+03 | 965.7500 | 1.8 |
| 7.4736e+03 | 2.5437e+03 | 5.5713e+03 | 3.9191e+03 | 1.4024e+03 | 5.2613e+03 | 1.4419e+03 | 2.3 |
| 4.4912e+03 | 3.9157e+03 | 6.5179e+03 | 3.8619e+03 | 4.2130e+03 | 9.0264e+03 | 2.4029e+03 | 2.9 |
| 1.3872e+03 | 1.1285e+03 | 1.5284e+03 | 2984 | 2.9332e+03 | 2.4619e+03 | 743.0600 | 1.3 |
| 967.7900 | 1.1813e+03 | 1.7284e+03 | 2.8847e+03 | 2.8612e+03 | 2.4441e+03 | 569.7400 | 1.0 |
| 4.9842e+03 | 3.1939e+03 | 4.8164e+03 | 5.2415e+03 | 5.1367e+03 | 8.0413e+03 | 1.7726e+03 | 2.9 |
| 3.1863e+03 | 2.3991e+03 | 3.4813e+03 | 9.2552e+03 | 6.3419e+03 | 5.4401e+03 | 997.3500 | 2.5 |
| 2.6832e+03 | 2.6646e+03 | 4.3163e+03 | 7.1407e+03 | 7.7031e+03 | 6.7947e+03 | 1.4379e+03 | 3.1 |
| 8.0211e+03 | 3.5940e+03 | 4.0539e+03 | 4.8655e+03 | 4.0337e+03 | 6.3570e+03 | 1.8185e+03 | 3.2 |
| 2.8019e+03 | 2.1906e+03 | 2.0998e+03 | 2.5296e+03 | 2.7308e+03 | 3.3094e+03 | 973 | 1.9 |
| 3.7835e+03 | 2.9174e+03 | 3.9006e+03 | 2.8032e+03 | 3.2496e+03 | 5.8811e+03 | 1.2525e+03 | 3.3 |

**Figure 3. 5 Selected Features Using an Optimized Genetic Algorithm**

### 3.2.3. Feature Extraction

Feature extraction is a method being used to classify prominent attributes, features or attributes that are embedded in data. In a group of reports, examples of the procedure of feature extraction include the recognition of variations and the discovery of specific precedents. The use of feature extraction to deliver a similar understanding of its classification involves data with dimensional stacks. Function extraction enables innovative variables of selected features to reduce the existing curse of dimensionality. For extracting features, there are two large groups of algorithms, namely: linear (adopting data on a low-dimensional feature space such as PCA) and non-linear (adopting data on a low-dimensional feature space and characterized on a high dimensional feature for a non-linear assembly among features can be initiated, for example, ICA) (Liang et al., 2018).

To achieve objective 2, the feature extraction component employs PCA and ICA distinctly on the RNA-Seq reduced data subsequently transitory over the Genetic Algorithm feature selection to examine the difference of proficiency results. ICA is a method for modelling relationships among sets of independent variables employing inherent variables; aims to find uncorrelated linear transformations (latent components) of the initial predictor variables which covariate highly with the response variables. PCA is a method used to evaluate the primary variables in a multidimensional data set that describes the variations in the measurements and is also supportive for the simulation and interpretation of high dimensional data sets research. Extraction of functionality produces additional variables as variations of those to minimize the dimensionality of the chosen functions. This approach makes use of beneficial attributes while at the same time, minimizing negative attributes. It functions by replacing the initial variables (numeric) with new numeric variables; it captures the most defining feature (Santos et al., 2019).

#### 3.2.3.1. Principal Component Analysis (PCA)

PCA is a procedure that is unsupervised. It establishes normal metadata and diagonalizes the covariance matrix. A traditional inherent correlation coefficient attribute is transformed in to the linear predictor variables using orthogonal variation. Difficulties of linear dimensionality minimization procedures are the accumulation of irrelevant data information in a smaller portion of the dimension. PCA will screen examples and increase the chance to illustrate. PCA is a widely used tool for minimizing dimensionality, attribute extraction, data compression, visual analytics, respectively. (Barshan et al., 2011).

PCA was used in this analysis to extract gene expression features having variations between models. PCA defines the principal dimensions of the subspace, leveraging the volatility of static results. An example of the experimental value generates a function vector of the observed values in this principal subspace. The investigation mean $\bar{X}$ and the matrix S for data covariance are as follows:

$$\bar{X} = \frac{1}{N}\sum_{n=1}^{N} X_n \qquad 3.1$$

$$S = \frac{1}{N}\sum_{n=1}^{N} (X_n - \bar{X})(X_n - \bar{X})^T \qquad 3.2$$

Adopting equations (3.1) and (3.2), the component vector on the principal subspace that feats the variance of a specified data as follows.

$$Su_i = \lambda_i u_i u_i{}^T \; Su_i = \lambda_i \qquad 3.3$$

The vector maximizing the modification of the predictable information develops an eigenvector, $u_i$, of matrix *S*, and the maximal variance size in the path of the eigenvector develops the eigenvalue $\lambda_i$. Principal subspace collected for the principal component resultant from PCA comprises of eigenvectors with M bits of best eigenvalues for matrix *S* (Jolliffe and Cadima, 2016).

### 3.2.3.2. Independent Component Analysis (ICA)

ICA is a valued leeway of PCA with conservative layers, since the visor parting of independent bases from their linear grouping. The actual fact of ICA is the possessions of uncorrelation of the general PCA. Built a x b on data matrix P, whose rows ri (d=1…, a) reckon to observational variables and whose columns kd (d=1…, b) are the entities of the matching variables, the ICA model of P can be written as shown in equation 3.4:

$$P = AS \qquad 3.4$$

With complete overview, A is a a x a fusion matrix, where S is a a x b is a basis matrix below the need of statistically independent as conceivable. Independent components are original variables kept in rows of S, to wit, the variables detected are linearly composed independent components. The independent components achieved by learning the precise linear groupings of the experimental variables, since mixing can be inverted as shown in equation 3.5:

$$U = S = A\text{-}1P = WP \qquad 3.5$$

In this study, objective 2 uses the feature extraction (PCA and ICA) methods to select optimal latent components from the given reduced dataset.

### 3.2.4. Classification

To achieve objective 3, this study used the reduced features obtained from objective 2 to classify the selected features.

Classification models influence the final performance. Different classifiers contain diverse performance on the respective data set. The classification module uses the concept of four diverse classifiers (SVM, K-NN, Ensemble and Decision Tree). These are machine learning methods that can efficiently deal with small-scale sample difficulties with samples of huge dimensions. Classification is a predominant supportive process. It attaches and correlates class labels specified to existing information from a predetermined target class. Building a classifier is perform in two stages:

i. The learning stage, where identification model is built with a class label giving a collection of training data.

ii. The model is developed to predict target class for hidden information, when the classifier's accuracy is calculated.

This study develops a model and investigates the accuracy of existing classification algorithms in the prediction of insecticide resistance. Classification methods are employed using SVM, K-NN, Ensemble, and Decision Tree.

### 3.2.4.1. Support Vector Machine (SVM)

SVM is a machine learning algorithm presented by Vapnik in 1992. The procedure functions with point of discovering the fittest hyperplane that isolates between classes in the input space. SVM is a linear classifier; it is created to work with non-linear problems by joining the kernel ideas in high-dimensional workspaces. In non-linear issues, SVM utilizes a kernel in training the data to spread the dimension widely. When the dimensions are tweaked, SVM will look for the optimal hyperplane that can separate a class from different classes. As indicated by the adoption of Aydadenta and Adiwijaya (2018), the procedure to locate the best hyperplane utilizing SVM is as follows:

*iii.* Let $y_i \in \{y_1, y_2, \dots, y_n\}$, where $y_i$ is the p-attributes and target class $z_i \in \{+1, -1\}$

*iv.* Assuming the classes +1 and -1 can be separated by a hyperplane, as defined in equation 3.6 below:

$$v.y + c = 0 \qquad 3.6$$

Then equation (3.7, 3.8, and 3.9) are gotten:

$$v.y + c \geq +1, \text{ for class } +1 \qquad 3.7$$

$$v.b + c \leq -1, \text{ for class } -1 \qquad 3.8$$

Where *y* is the input data, *v* is the ordinary plane and *c* is the positive relation to the center field coordinates.

SVM discover hyperplanes that maximize margins between two classes, expanding the margins is a quadratic problem in programming that can be solved by reaching the minimum value. The benefit of SVM is its potential in high dimensional data is its capabilities of handling wide variety of classification problems. It is grouped into linear and non-linearly separable (Tharwat & Gabel, 2019).

SVM's consists of kernel functions that change data into a higher dimensional space making it conceivable to accomplish separations. Kernel functions are pattern analysis or recognition class of algorithms. Training variables xi by capability Φ are translated into higher dimensional space. In this subspace, it considers a linear separating hyperplane with the limit. The penalty parameter of the error term is C >0.

Several SVM kernels exist for instance; the polynomial kernel, Radial basis function (RBF), linear kernel, Sigmoid, Gaussian kernel, String Kernels, among others. The decision of a Kernel relies on on the existing issues at hand since it relies on what models are to be analyzed, a couple of kernel functions have been initiated to function admirably in for a wide assortment of applications. The prescribed kernel function for this study is the SVM-Polynomial Kernel and Gaussian Kernel (Deepika et al., 2019).

**SVM-Gaussian Kernel**

Gaussian kernel compares to a general smoothness supposition in all k-th order subordinates. Kernels coordinating a certain prior recurrence substance of the data can be developed to reflect earlier issues in learning. Each input vector $\underline{x}$ is mapped to an interminable dimensional vector including all degree polynomial extensions of *x's* components (Hassan et al., 2017; Chaeikar et al., 2020).

**SVM Polynomial Kernel**

For instance, a polynomial kernel model features combination up to the direction of the polynomial. Radial basis functions permit loops in contrast with linear kernel, that authorizes just selecting lines (or hyperplanes).

$$K(y_a, y_j) = (\gamma y_a^S y_b + q)^e, \gamma > 0 \qquad 3.9$$

**SVM-Linear Kernel Function**

For instance, the polynomial kernel is the least complex kernel function. It is specified by the inner invention *(a,b)* in addition to a discretionary constant K.

$$K(y_a, y_b) = y_a^S y_b \qquad 3.10$$

**SVM-RBF Kernel Function**

In SVM kernel functions, $\gamma$, *a,* and *b* are kernel parameters, RBF is the fundamental kernel function due to the nonlinearly maps tests in higher dimensional space unlike the linear kernel, it has less hyperparameters than the polynomial portion.

$$K(y_a, y_b) = \exp(-\gamma||y_a, y_b||^{\ 2}), \gamma > 0 \qquad 3.11$$

### 3.2.4.2. $K^{th}$ Nearest Neighbour

The data on the genes were categorized using the KNN algorithm. KNN is a supervised learning procedure; the product of a different instance query is graded based on general neighbor group K-nearest. KNN algorithm utilizes the classification of localities as the estimated rate of the original query case. This algorithm has the function of classifying an original entity based on features and training models. The classifiers use no matching model and are based on retention only. This module is given the chosen functions as an origin. The values K (nearest neighbor numbers) that are adjoining to the question point are chosen. Calculate the distance between the query-instance and all of the testing samples. Then the distance is sorted, and the closest neighbors are determined based on the minimum distance from Kth. The closest neighbors, Division Y is collected. The simple majority of nearest neighbor's division is used as the demand instance prediction factor. Any bonds can be broken by chance (An et al., 2019).

### 3.2.4.3. Ensemble Classification

Ensemble classifiers can be trained using on unrelated subsets of the training data, diverse parameters of the classifiers, or even with diverse subsets of features as in random subspace models.

Ensemble classifier comprises of integrating results of diverse classifiers to produce a concluding decision. It is frequently used for gaining precise outcomes. Ensemble classifiers are relatively common in machine learning complications and can be employed in the bioinformatics field. Classification decision is achieved by merging the decision of each classifier (Onan, 2015).

Ensemble techniques are processes for machine learning that combine decisions to enhance the efficiency of the general classification. In the literature, many words were found to denote similar interpretations such as; studying multi-strategy, aggregation, multiple integration classifiers, combination of classifiers, sorting, assembly, and so on.

Completely better efficiency can be obtained by the Ensemble classifier than by the discrete base classifiers. The efficacy of ensemble methods is highly dependent on the unconventionality of the discrete learner's error. Ensemble strategies rely on the accuracy and variety of the basic learners in terms of efficiency. Classification of the ensemble has traditional methods; bagging and boosting (Alfaro et al., 2018).

Bagging (**b**ootstrap aggregating) employs the training data by arbitrarily changing the unique *T* training information by *N* items. The additional training sets are called bootstrap duplicates with some instances not appearing while others seem more than once. The concluding classifier $C^*(x)$ is built by combining $Ci(x)$ where every $Ci(x)$ has an equivalent vote.

The training data is influenced by the AdaBoost (Adaptive Boosting) technique. Initially, the algorithm assigns equal weight to every instance xi. The information algorithm attempts to decrease the weighted error on the training set in each iteration *I* and yields a *Ci* classifier *(x)*. The weighted *Ci(x)* error is measured and helpful in reminding the weights of the xi training instances. The weight of xi increases, giving its impact on the efficiency of the classifier, allowing a large weight for a misclassified xi and a small weight for an acceptably classified xi. A weighted vote of the discrete *Ci(x)* rendering to its precision

based on the weighted training set is built by the final classifier *C*(x)* (Osareh & Shadgar, 2013).

Adopting Kowsari et al., (2019) they illustrated how a dataset boosting algorithm operates, then educated by multi-model designs (ensemble learning). Such inventions culminated in the AdaBoost (Adaptive Boosting). Assume to construct $D_t$ such that $D_1(i) = \frac{1}{m}$ given $D_t$ and $h_t$, as shown in equation 3.12 and equation 3.13:

$$D_{i+1}\{i\} = \frac{D_t(i)}{Z_t} X \begin{cases} e^{-\alpha_t} \text{ if } y_i = h_t(x_i) \\ e^{\alpha_t} \text{ if } y_i \neq h_t(x_i) \end{cases} \quad 3.12$$

$$\frac{D_t(i)}{Z_t} \exp(-\alpha y_i h_t(x_i)) \quad 3.13$$

Where $Z_t$ normalizes factors and $\alpha_t$ is as shown in equation 3.14;

$$\alpha_t = \frac{1}{2} in(\frac{1-\epsilon_t}{\epsilon_t}) \quad 3.14$$

Elementary ensemble classification procedures namely: The Weighted Averaging (WA); Max Voting (MV) and Averaging. Max Voting (MV) exists.

Ensemble learning has four advanced grouping methods; Stacking (STK); Blending (BLD); Bagging (BAG); Boosting (BOT).

### 3.2.4.4. Decision Tree

Decision tree classifiers use orthogonal hyperplane axes to iteratively partition the instance space. The system is constructed from a root node representing an attribute, as well as the space break instance is built on the feature of attribute values (the split values are chosen

for other algorithms), most often using their values. Every new information sub-space then is recursively split into new sub-spaces until a target threshold is satisfied and a class mark identifying the classified result is then allocated to the terminal nodes (leaf nodes) (the class of all or most of the instances in the sub-space). It is very important to set the correct end criteria since trees that are too large can be overfitted and small trees can be under fitted and in all cases experience a lack of precision. Most algorithms have built-in a function that deals with overfitting. It is called pruning. Each new instance is graded according to the outcome of the tests along the route by navigating them from the root of the tree down to a leaf. While decision trees generate effective models, they are unreliable if the training data sets vary only marginally, the resulting models for those two sets may be entirely different. Because of this, decision trees are frequently used in ensembles of classifiers.

To achieve objective 4, the experiment evaluates the performance in terms of Accuracy, Sensitivity, Specificity, Precision, Recall and F-Score. Using the classification confusion matrix, the True Positive, True Negative, False Positive and False Negative is considered as the outcomes where the model correctly predicts the classes for evaluation.

## 3.3. Proposed Model

Increase in biological data dimensionality is a challenging predictable investigation method. Using conventional methods for learning intricate designs at numerous layers stimulated from morphological operations receptive to processing is of the essence. Most standard techniques used for high dimensional data such as RNA-Seq data involves several complexities. Combining different dimensionality reduction methods can be of the essence, exploiting specific benefits, where the gene subset attained from one process is served as the

input to another. Commonly, feature extraction techniques can be used to aid feature selection effectively by using feature selection to select the initial gene subset or aid to eliminate redundant genes. Combination of numerous feature extraction systems can be useful to extract the preliminary feature subsets.

Several procedures have been proposed to attain better classification, finding an optimal set that can be clinically utilized is of essence. Figure 3.6 shows an existing hybridized model that requires improvement by introducing an optimized genetic algorithm as a feature selection technique with feature extraction methods for classification.

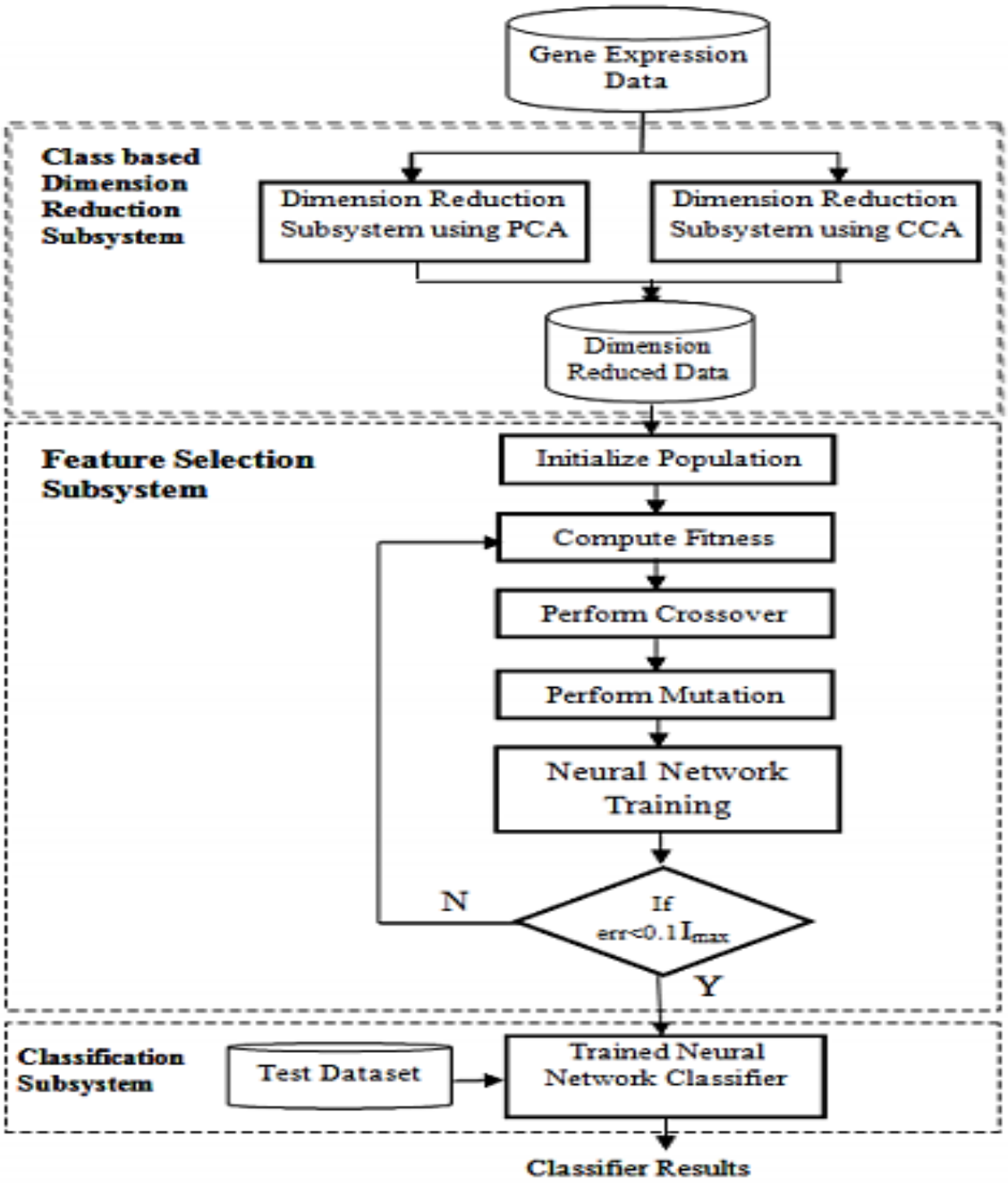



**Figure 3. 6 Existing Hybrid Architecture**

**Source: Susmi (2016)**

In the proposed experiment, an optimization is suggested for the genetic algorithm, Algorithm 3.2 shows the existing optimized genetic algorithm pseudocode, that requires further enhancement. Algorithm 3.3 shows the enhanced genetic algorithm optimization pseudocode required to be utilized with a hybrid approach for this study.

**Algorithm 3.2: Existing Pseudocode for Genetic Algorithm Optimization Parameter Source: Kuang et al., (2020)**

Step 1: initialize parameters $P_c$, $P_m$, $N$, $G$, $T$, and $Q$ and randomly generate the first swarm $pop$
Step 2: for $i < popsize$
Step 3: calculate the tangent value $\tan(x_{in}/x_{in+1})$ of the included angle between the two vectors in adjacent dimensions for each individual $pop(i)$
Step 4: if $\tan(x_{in}/x_{in+1}) < 4.3926(1/(1+e^{-D})) - 3.6072$, then update the value of the $n^{\text{-th}}$ dimension of the $i^{\text{-th}}$ individual to 0; otherwise, do not update the value
Step 5: return to step 2
Step 6: calculate the number of 0 elements in each dimension for the updated swarm $pop$; if it is above the critical value $Q$, then delete this dimension
Step 7: obtain the updated swarm $pop$
Step 8: calculate the fitness value $F(i)$ of each individual in the swarm $pop$
Step 9: initialize the new swarm $newpop$
Step 10: select two individuals from the swarm $pop$ according to the fitness using the proportional selection algorithm
Step 11: if $random(0, 1) < P_c$, then move on to step 12; otherwise, implement step 13
Step 12: apply the crossover operator according to the crossover probability $P_c$ on the two individuals
Step 13: if $random(0, 1) < P_m$, then move on to step 14
Step 14: apply the mutation operator according to the mutation probability $P_m$ on the two individuals
Step 15: add the two new individuals into the swarm $newpop$
Step 16: repeat this process until the $N^{\text{th}}$ generation is produced; otherwise, return to step 4
Step 17: replace $pop$ with $newpop$
Step 18: repeat this process until the number of generations exceeds $G$; otherwise, return to step 8
Step 19: end

In this study, a hybrid dimensionality reduction technique named HYDREC is implemented for the classification of malaria vector dataset, with a nominal set of predictor genes. The optimization-based technique was utilized in the feature selection stage, algorithm 3.3 shows the improved optimized genetic algorithm pseudocode.

**Algorithm 3.3: Improved Optimized Genetic Algorithm (GA-O) Pseudocode for the Proposed Study**

Step 1: initialize parameters $P\ and\ Q$ and randomly create the first population

Step 2: for $i < popsize$

Step 3: calculate the tangent value $\tan(x_{in}/x_{in+1})$ of the involved angle between the two vectors in adjacent dimensions for each pop($i$)

Step 4: if $tan(xin/xin+1) < 4.4(1/(1+e-D)) - 3.6$, then update the value of the $nth$ dimension of the $ith$ individual to $0$;

otherwise, do not update the value and continue in step 6

**Step 4.1: if $X > 1$**

**Step 4.2: calculate the compatible persistence with Euclidean distance $D$ between $Xi\ and\ Xj$**

**Step 4.3: calculate the comparation principle within the distance $L = |Xi - Xj| < D$**

**Step 4.4: if no similarity,**

**i. Remove individual fitness with the similarity of biallelic loci $SD(Xi, Xj)$ and average similarity $MSDi$**

**ii. Calculate the subpopulations $M(t+1)$;**

**otherwise**

***Merge* $N$ individuals in memory pool with subpopulation sorted by fitness in descending order**

**Step 4.5: compute the $subpopulation * threshold\ \delta\ (0.5)$**

Step 5: judge the convergence condition

Step 6: compute the number of 0 elements in each dimension for the updated pop; if it is above the critical value Q, then delete this dimension

Step 7: acquire the updated population

Step 8: calculate the fitness value F(i) of each individual in the population

Step 9: initialize the new population

Step 10: select two individuals from the population giving to the fitness using the relative selection algorithm

Step 11: if random (0, 1) < $P_c$, then move on to step 12; otherwise, implement step 13

Step 12: apply the crossover operator rendering to the crossover probability $P_c$ on the two individuals

Step 13: if random (0, 1) < $P_m$, then move on to step 14

Step 14: apply the mutation operator according to the mutation probability $P_m$ on the two individuals

Step 15: add the two new individuals into the new population

Step 16: reiterate the process until the $N^{-th}$ generation is generated; else, return to step 4

Step 17: replace the population with the new population

Step 18: reiterate this process until the number of generations tops $G$; otherwise, return to step 8

Step 19: end

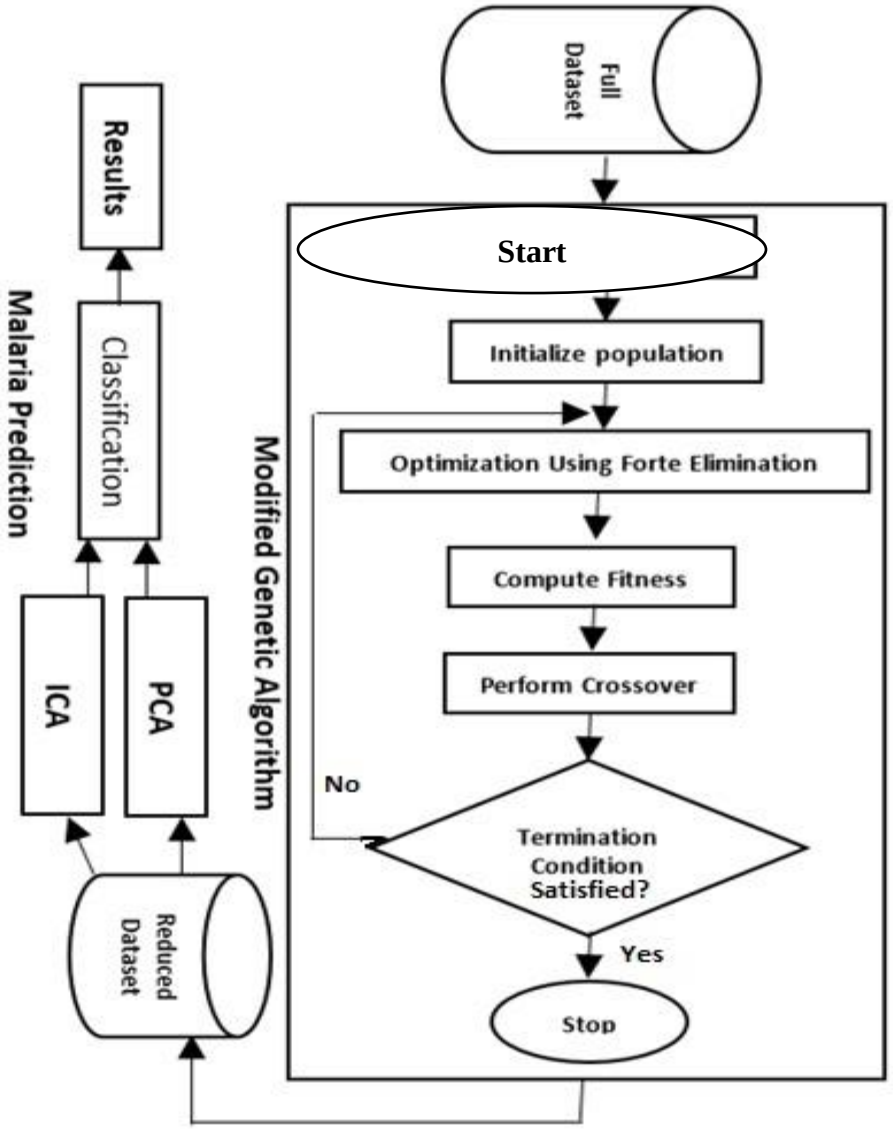



**Figure 3. 7 The Proposed Model for Improved Hybrid GA-O Dimension Reduced Classification Model, HYDREC Framework**

This study proposes an efficient hybrid dimensionality reduction technique for classifying Anopheles gambiae RNA-Seq gene expression data, Figure 3.7 shows the HYDREC framework and figure 3.8 shows the workflow. The proposed classification technique comprises of three phases, namely:

i. Feature selection
ii. Feature extraction
iii. Classification

The feature selection phase selects the relevant optimal subset feature of genes from the original data, without halting the dimensional space by using an Optimized genetic algorithm proposed for this study.

In the second phase, the feature extraction technique uses PCA and ICA individually on the selected features from the first phase.

The feature extraction technique transforms the high dimensional data into latent components, and further reduces the dimensionality. It is necessary to reduce the dimensionality of feature space for adequate classification. A singular dimensionality reduction is not sufficient enough for classification, because not all the newly selected features are helpful. This phase additionally makes the classification process more effective and efficient, by further eliminating redundancies that halters the classification performance. Classification approach is used in the third phase of this study, by applying SVM, KNN, Decision tree and Ensemble algorithms to predict the performance of the procedures. The results present the performance criteria used with the classifiers by evaluating the performance metrics and show the validity of the proposed model. Figure 3.8 highlights the workflow of the study.

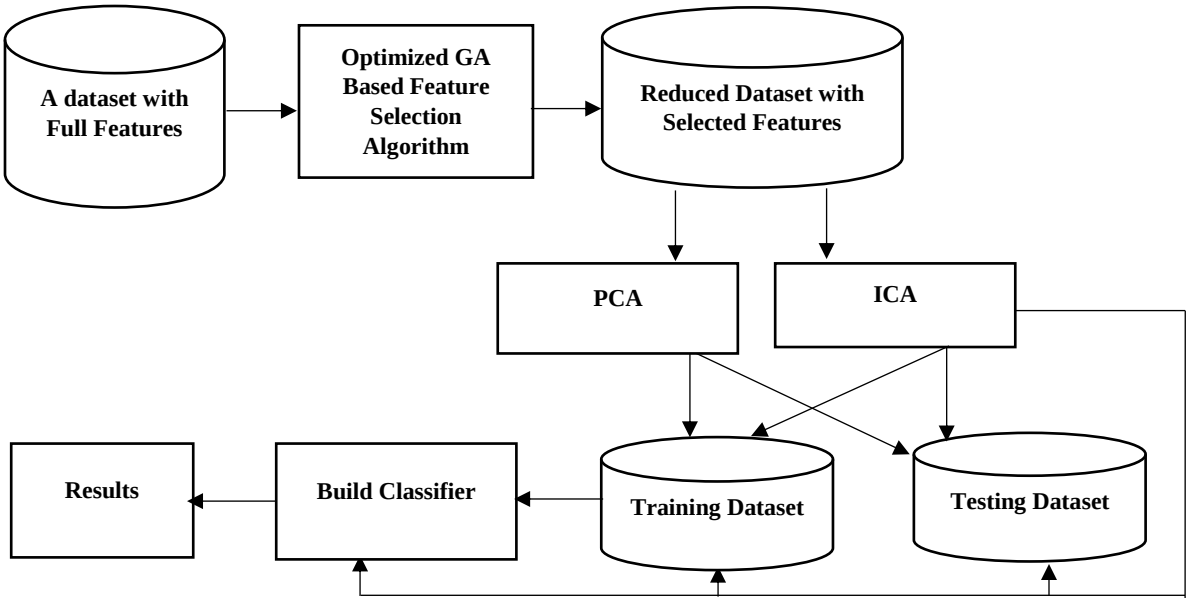



**Figure 3. 8 The HYDREC Model Workflow**

## 3.4. Performance Evaluation Metrics

This study analyzes the performance evaluation metrics of the classifier in terms of accuracy, sensitivity, specificity, precision, f-score and recall.

Assessing machine-learning algorithm efficiency needs specific validation metrics. The confusion matrix uses four characteristics for evaluating the classification models; True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). It discovers the examples categorized correctly and incorrectly from the data set sample given to test the model (Karthik & Sudha, 2018). Performance metrics with their formula are stated in equations 3.16, 3.17, 3.18, 3.19, 3.20, and 3.21 (Arowolo et al., 2016).

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \% \quad 3.16$$

$$Specificity = TN / (TN + FN) \% \quad 3.17$$

$$Sensitivity = TP / (TP + FN) \% \quad 3.18$$

$$Precision = TP / (TP + FP) \quad 3.19$$

$$Recall: TP/TP + FN \quad 3.20$$

$$F - Score: 2\ x\ (Recall\ x\ Precision) / (Recall + Precision) \quad 3.21$$

Where:

TP (True Positives) = correctly classified positive cases,

TN (True Negatives) = correctly classified negative cases,

FP (False Positives) = incorrectly classified negative cases,

FN (False Negatives) = incorrectly classified positive cases.

Accuracy is the possibility that an analytical test is correctly performed.

Specificity (true negative fraction) is the probability that a diagnostic test is negative, stating that the individual does not have the disease.

Sensitivity (true positive fraction) is the probability that a diagnostic test is positive, stating that the individual has the disease.

F score is the harmonic mean of Positive Predictive Value and sensitivity.

In this study, analyzing the RNA-Seq Anopheles gene expression datasets, was implemented on MATLAB. MATLAB is a powerful graphical, and computational tool used to answer comparatively complex science and engineering issues designed, developed and implemented.

# CHAPTER FOUR

## 4.0 RESULTS AND DISCUSSIONS OF FINDINGS

This chapter presents the results of the implementation performed and its evaluation. It also provides a detailed discussion of the results and findings of the proposed model. The result of the evaluation serves as justification for the performance of this study in line with the aim and objectives of the study.

### 4.1. Results and Discussions

Feature selection and feature extraction algorithms were implemented on MATLAB 2015 platform, thereafter, classification techniques were performed. Specifically, this section presents the results of the studies for the proposed model. Application and comparison of the methods were performed, using an optimized genetic algorithm (GA-O), with feature extraction (PCA and ICA). This study Implements a dimensionality reduction method with classification procedures using SVM, KNN, Decision Tree and Ensemble on a Mosquito Anopheles dataset. The data was normalized and consists of 7 attributes and 2457 gene expression levels. Figure 4.1 shows the integrated development area on MATLAB 2015 that was used for the simulation of the model.

**Figure 4. 1 MATLAB Integrated Development Environment 2015a**

The user interface consists of five functionalities. These includes; Menu, Load, feature Selection, feature extraction, and Classify, and are depicted in figure 4.2. The Load button is used to import the data set after which other procedures can be applied to the loaded datasets.

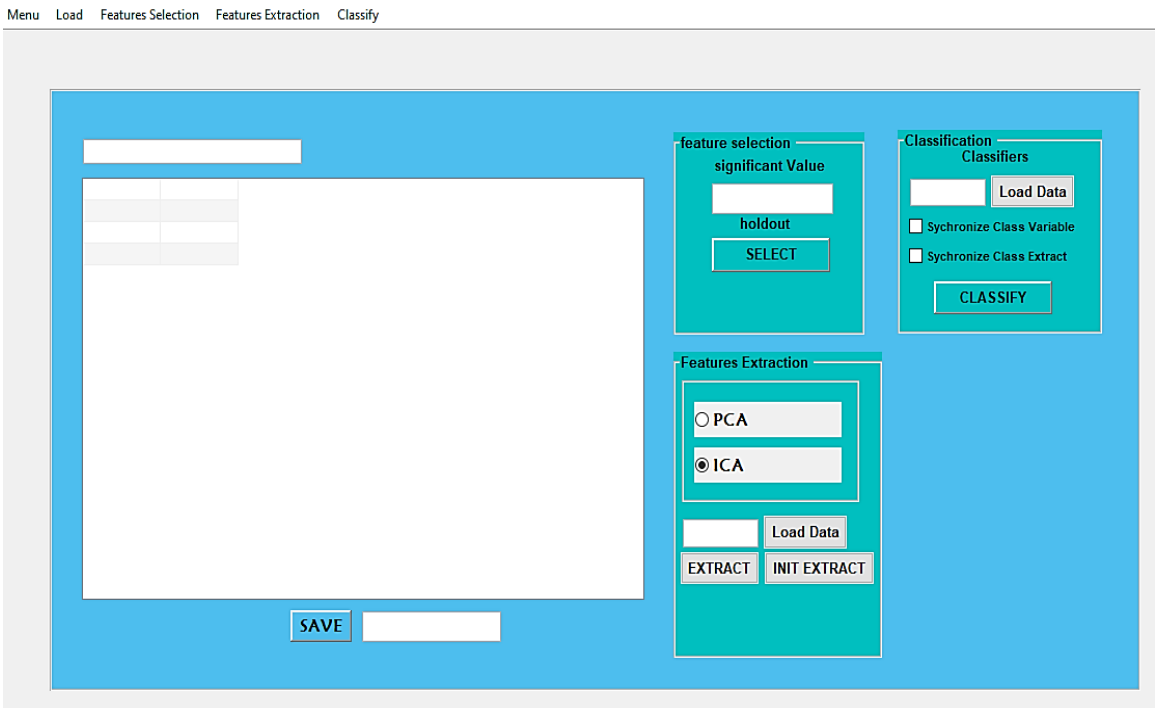



**Figure 4. 2 User Interface**

For better compilation and user-friendliness, the MATLAB software developer tools (GUIDE) was used to create an interactive environment for easy readability, formatting and interactivity. The GUI development was grouped into four major sections, namely:

i. Load
ii. Feature Selection
iii. Feature Extraction
iv. Classification

Figure 4.3 shows the MATLAB interface for loading the *Anopheles gambiae* gene expression data. The loaded gene expression dataset comprises of 7 attributes and 2457 instances. The data code is shown in Appendix B

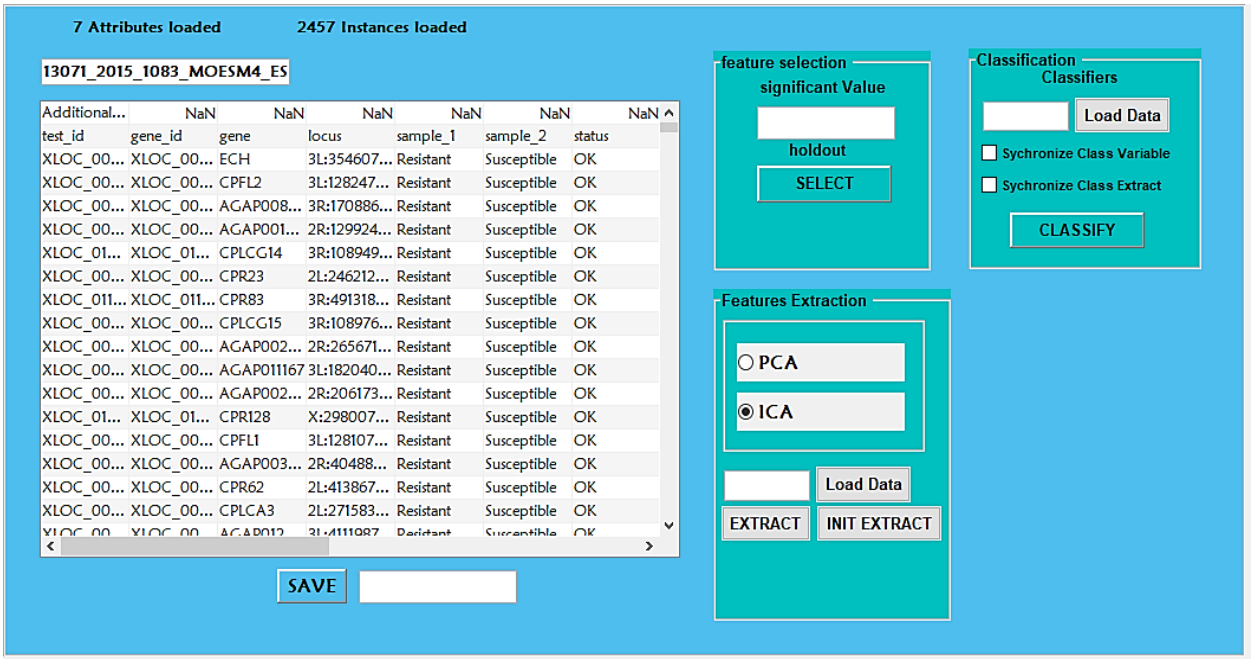



**Figure 4. 3 Graphical User Interface for Loading *Anopheles Gambiae* Dataset**

## 4.2. Feature Selection

The feature selection mode uses the GA-O to fetch relevant components from the huge data with the aid of threshold δ set value to be 0.5, and α is the threshold number of features selected. It is observed that the 2457 genes and their attributes were reduced using GA-O to eliminate irrelevant features in the data; 474 features were carefully selected in 26.6592sec as a subset in the data., figure 4.4 shows the loaded data and the computational time for the feature selection technique. The detailed figure is shown in Appendix C.

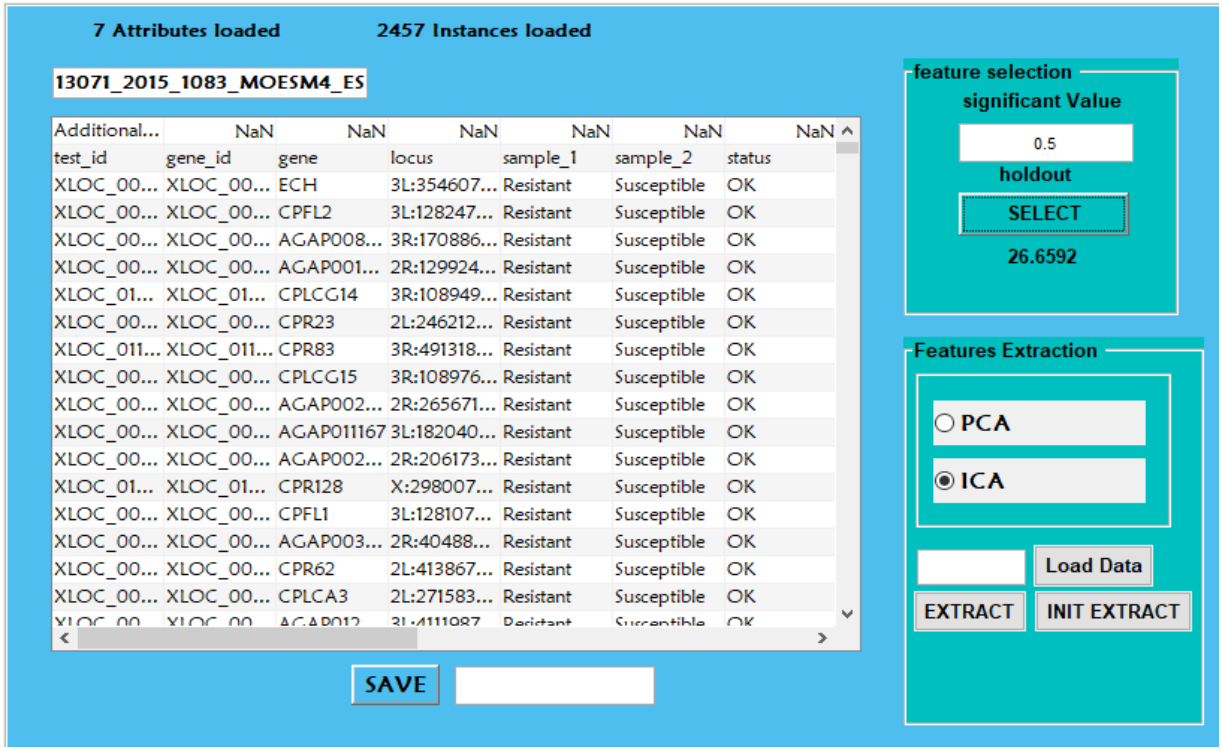



**Figure 4. 4 Feature Selection Using a Genetic Algorithm- Optimized Timing**

### 4.2.1. Genetic Algorithm Optimized Feature Selection Approach with Classifiers

This study explores the RNA-Seq gene expression datasets, carrying 2457 instances of mosquitoes Anopheles Gambiae. Optimized Genetic algorithm (GA-O) is used to diminish the curse of dimensionality, the selected features are shown in Figure 4.5. GA-O function dimensionality reduction selects the optimum data subset and reduces uncorrelated attributes (variables) to assess the highest variance from a reduced number of variable subset functions. GA-O is added to evidence from the Anopheles mosquito, which offers valuable gene knowledge that is useful for further study. Classification algorithms uses SVM, KNN, Ensemble and Decision tree and their confusion matrices as shown in figure 4.7 to 4.12. The classification performance metrics are also shown in Tables 4.1 to 4.3. Using GA-O as a selection tool for dimension reduction with a threshold of 0.5 as shown

in figure 4.6, 474 ideal subset features of genes were selected as important insecticidal target genes and are shown in figure 4.5.

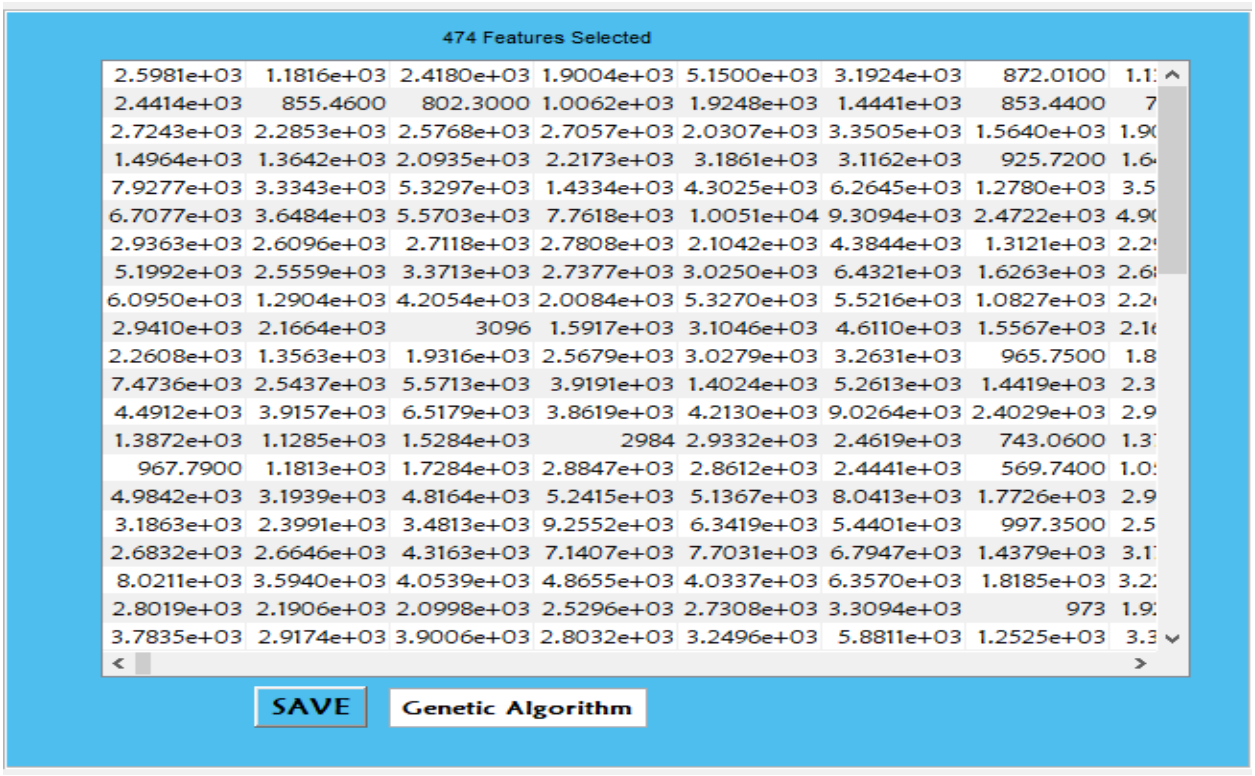



**Figure 4. 5 Selected 474 Anopheles Insecticide Target Genes**

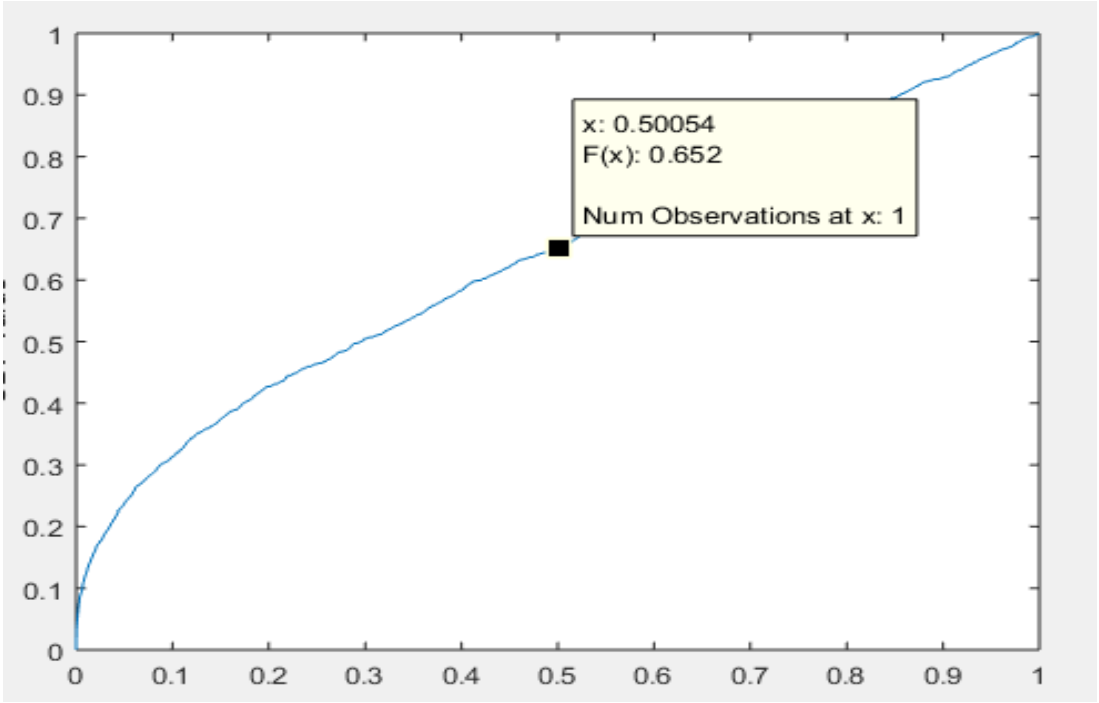



**Figure 4. 6 GA-O Threshold at 0.5**

The classification algorithms (SVM, KNN, Decision tree and Ensemble) uses 10-fold cross-validation to measure the performance of classification models, using 0.5 parameter holdout of training data and 5% for validations. The evaluation result mentioned is focused on the calculation time and efficiency metrics (Accuracy, Specificity, Sensitivity, Precision, F-score and Recall).

This analysis contrasts the model classification efficiency using GA-O with L-SVM, RBF-SVM, Decision Tree, KNN, Ada-boost and Bagged Ensemble classifications. The performance values and confusion matrix are shown in Figures 4.7 to 4.12. This analysis uses GA-O to extract related components from the data loaded in Figure 4.5. The chosen features are classified, and the results are tabulated. The performance matrix gives the output metrics a solution. Using the L-SVM classification kernel, this study achieves an

accuracy of 93.3%, RBF-SVM kernel classification method achieved an accuracy of 95% essentially, other performance metrics are shown in tabulated form in the Table 4.1, 4.2 and 4.3. The confusion matrix with 10-fold cross-validation results shows the selected features with samples correctly classified and used for evaluating the performance metrices. The tables shows the evaluation performances of the performance metrics evaluations of all the achieved results.

### 4.2.2. Genetic Algorithm Optimized Feature Selection Approach with SVM Classifiers

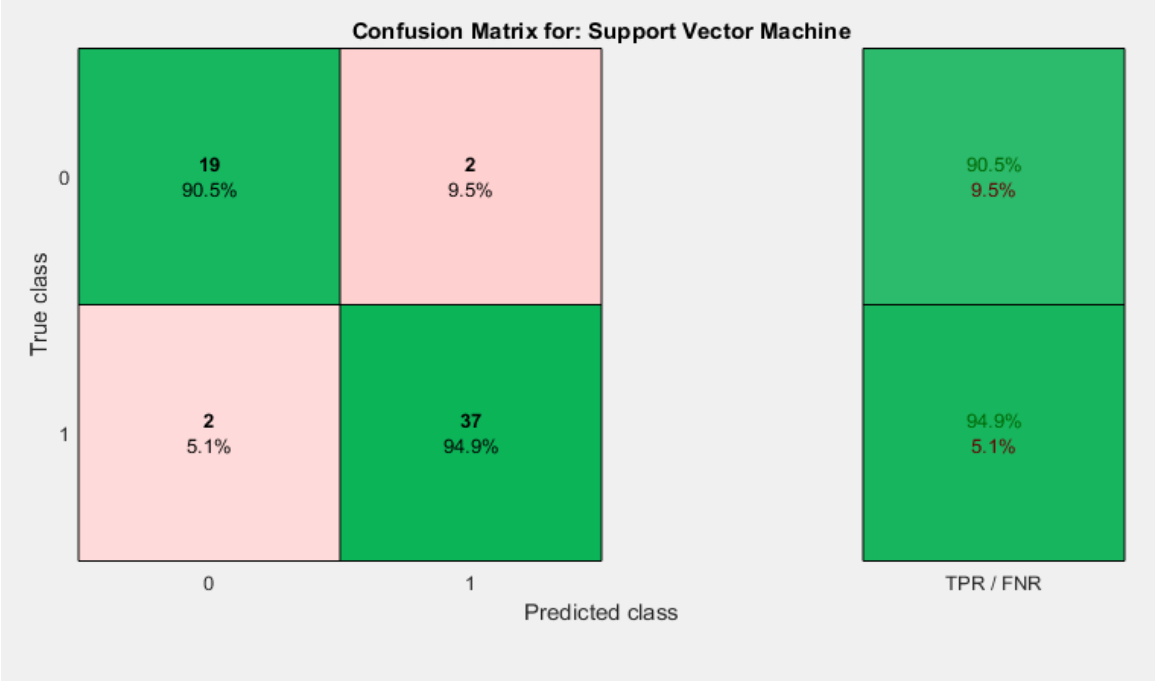



**Figure 4. 7 Confusion Matrix for GA-O with L-SVM Classification Model. TP=37; TN=19; FP=2; FN=2**

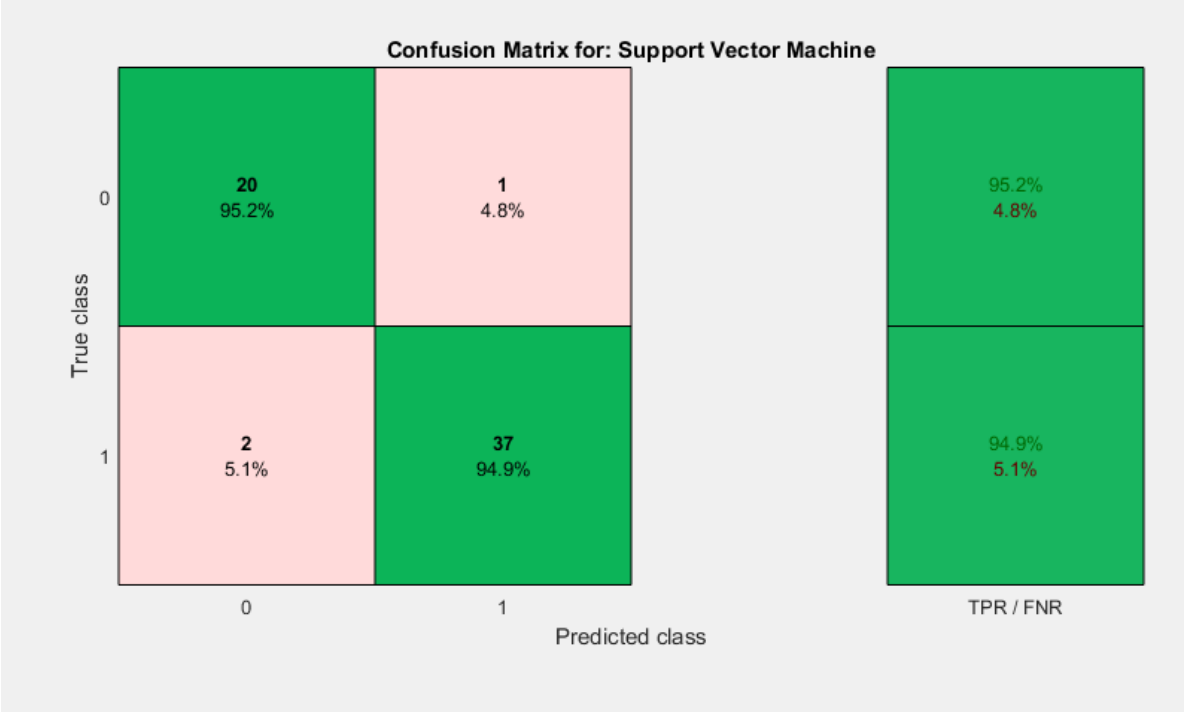



**Figure 4. 8 Confusion Matrix for GA-O with RBF-SVM Classification Model**

**TP=37; TN=20; FP=1; FN=2**

**Table 4. 1 Performance Metrics Table for the GA-O with SVM Classifier**

| Performance Metrics | GA-O + L-SVM Classification | GA-O+RBF-SVM Classification |
|---|---|---|
| Accuracy (%) | 93.3 | 95.0 |
| Sensitivity (%) | 94.9 | 94.9 |
| Specificity (%) | 90.5 | 95.2 |
| Precision (%) | 94.9 | 97.4 |
| Recall (%) | 94.9 | 94.9 |
| F-Score (%) | 95.0 | 96.13 |

### 4.2.3. Genetic Algorithm Optimized Feature Selection Approach with Decision Tree Classifier

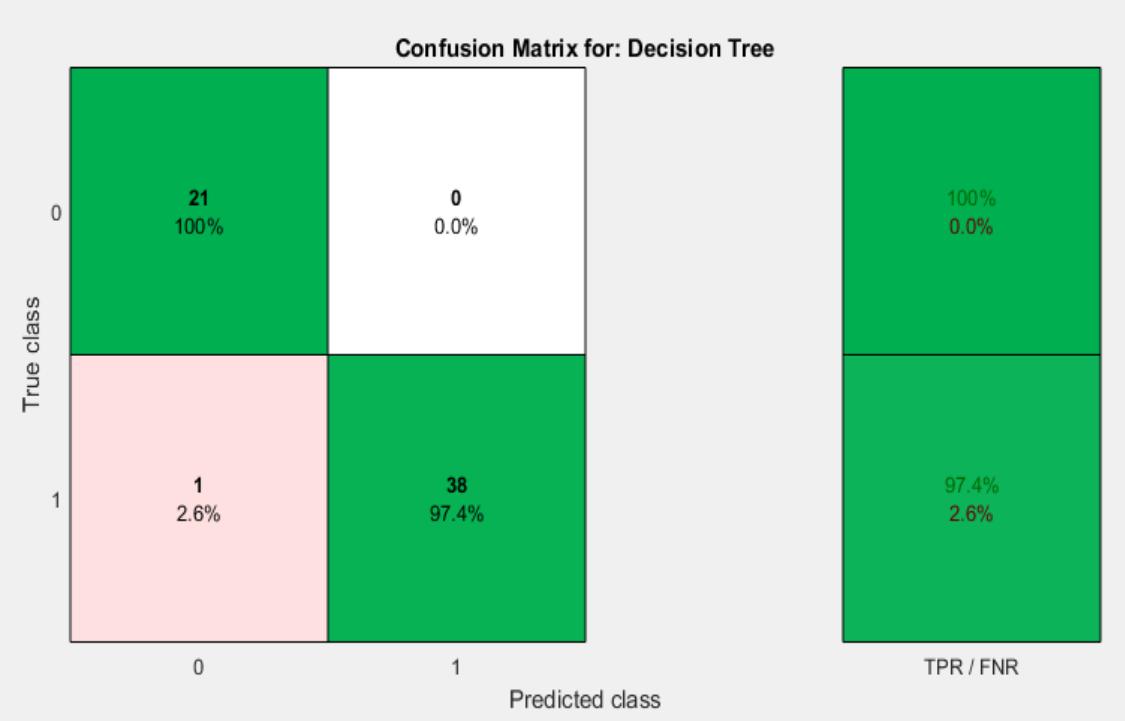



**Figure 4. 9 Confusion Matrix for GA-O with Decision Tree**

**TP=38; TN=21; FP=0; FN=1**

### 4.2.4. Genetic Algorithm Optimized Feature Selection Approach with KNN Classifier

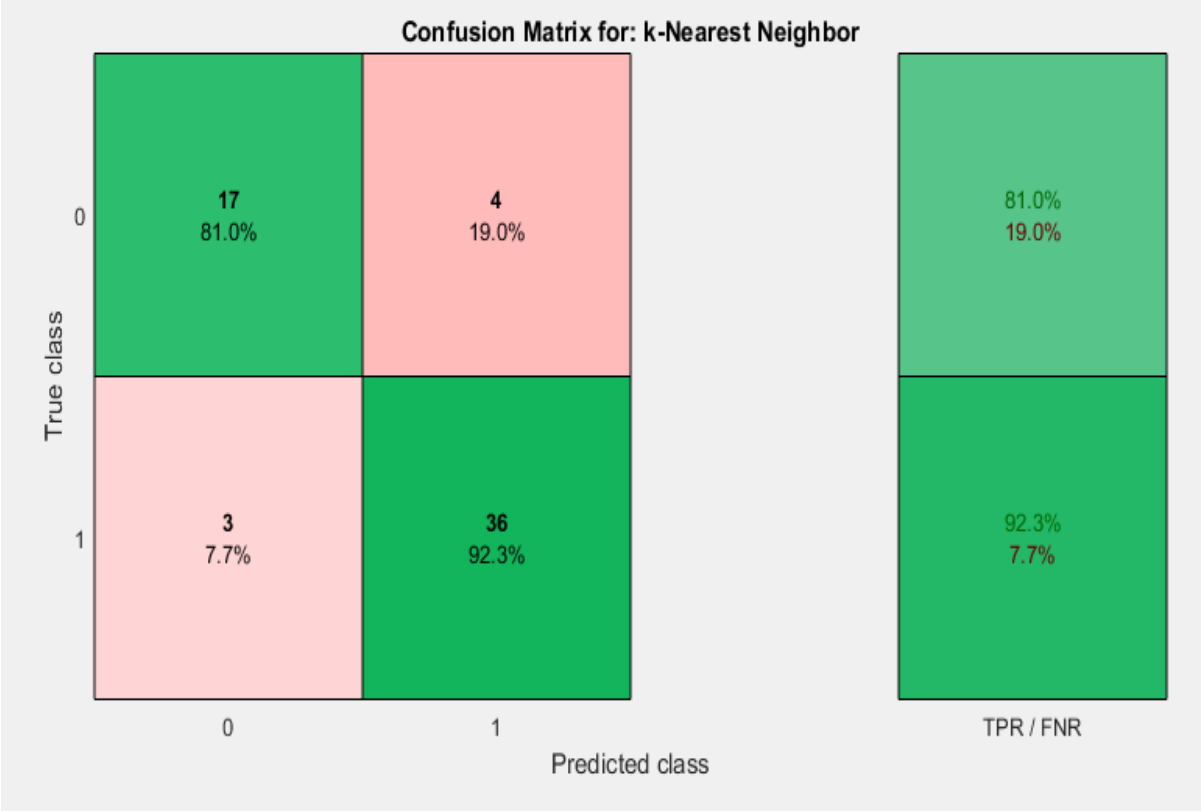



**Figure 4. 10 Confusion Matrix for GA-O with K-NN**

**TP=36; TN=17; FP=4; FN=3**

**Table 4. 2 Performance Metrics Table for the GA-O with K-NN Classifier**

| Performance Metrics | GA-O + Decision Tree Classification | GA-O+K-NN Classification |
|---|---|---|
| Accuracy (%) | 98.3 | 88.3 |
| Sensitivity (%) | 97.4 | 92.3 |
| Specificity (%) | 100 | 81.0 |
| Precision (%) | 100 | 90.0 |
| Recall (%) | 97.4 | 92.3 |
| F-Score (%) | 98.7 | 91.1 |

### 4.2.5. Genetic Algorithm Optimized Feature Selection Approach with Ensemble Classifiers

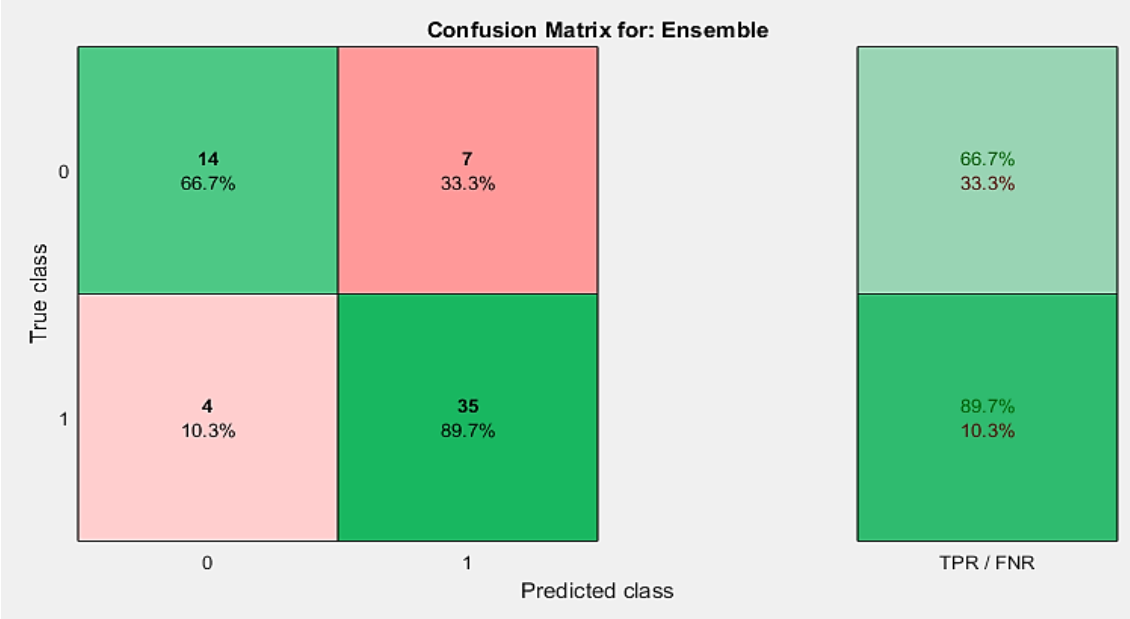



**Figure 4. 11 Confusion Matrix for GA-O with Ada-Boost Ensemble Classifier TP=35; TN=14; FP=7; FN=4**

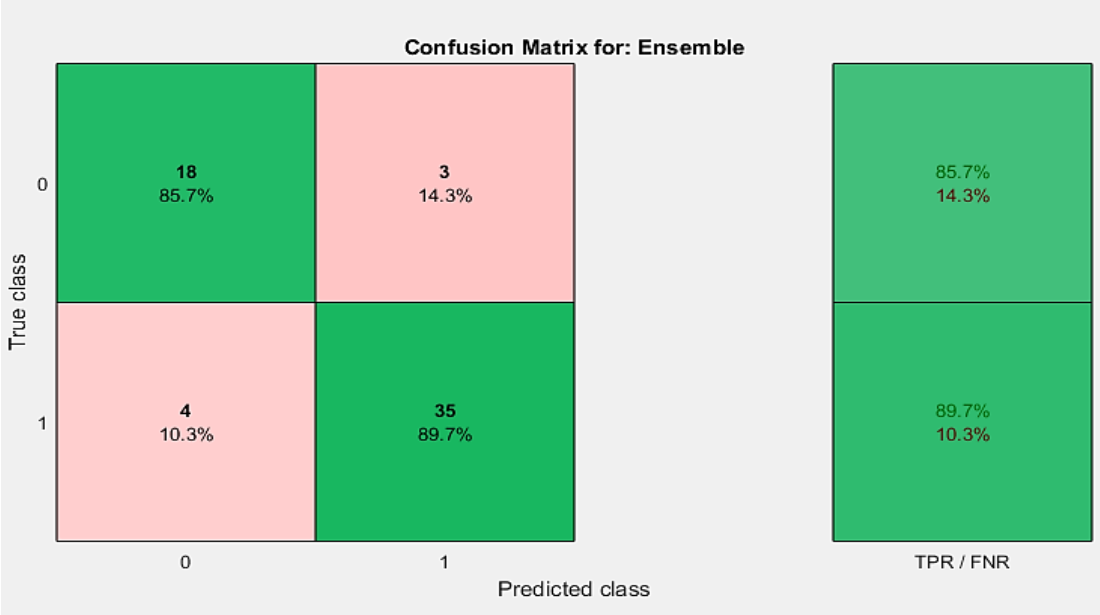



**Figure 4. 12 Confusion Matrix for GA-O with Bagged Ensemble Classifier**

**TP=35; TN=18; FP=3; FN=4**

**Table 4. 3 Performance Metrics Table for the GA-O with Ensemble Classifiers**

| Performance Metrics | GA-O + Ada-Boost Ensemble Classification | GA-O + Bagged Ensemble Classification |
|---|---|---|
| Accuracy (%) | 81.7 | 88.3 |
| Sensitivity (%) | 89.7 | 89.7 |
| Specificity (%) | 90.6 | 85.7 |
| Precision (%) | 83.3 | 92.1 |
| Recall (%) | 89.7 | 92.1 |
| F-Score (%) | 86.4 | 92.1 |

The feature selection algorithm using GA-O is used to select relevant features from an Anopheles gambiae dataset, the selected features are the classified using SVM, KNN, DT and Ensemble, with GA-O+ Decision tree outperforming other procedures with 98% as shown in Table 4.2.

## 4.3. Feature Extraction

The feature extraction in the current study uses PCA non-linear approach as well as ICA linear approach to extract subset latent components from the 2457 loaded data. Procedures and results achieved are discussed in the Figures 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, and 4.19. Table 4.4, 4.5, and 4.6 shows the performance evaluations for the PCA with classifiers.

### 4.3.1. PCA Feature Extraction Algorithm with Classification Approaches

This study explores RNA-Seq *Anopheles gambiae* Mosquitoes data, having susceptible and resistant genes. PCA algorithm which is a non-linear approach was executed on the data to reduce the curse of dimensionality, figure 4.13 shows the PCA procedure. PCA identifies and removes uncorrelated Attributes (Variables), to decide maximum variance with a smaller number of latent components. In this study, PCA is applied to the given data, to lessen the dimensionality issue and give significant gene information that is useful for further investigation. Classification algorithm applies SVM-Gaussian kernel and Polynomial kernel by utilizing MATLAB platform to implement the model. Using PCA as a dimensionality reduction method, 10 latent components were achieved in 11.6195 Seconds and displayed in figure 4.14. The extracted features are classified using the SVM, KNN, Decision Tree and Ensemble, the confusion matrix are shown in figures 4.15, to 4.19, the results of their performance metrics are also shown in tables 4.4 to 4.6. 10-folds cross-validation was used to evaluate the execution of the performance of the classification models, using 0.05 parameter holdout of data for training and 5% for testing to check the accuracy of the classifiers.

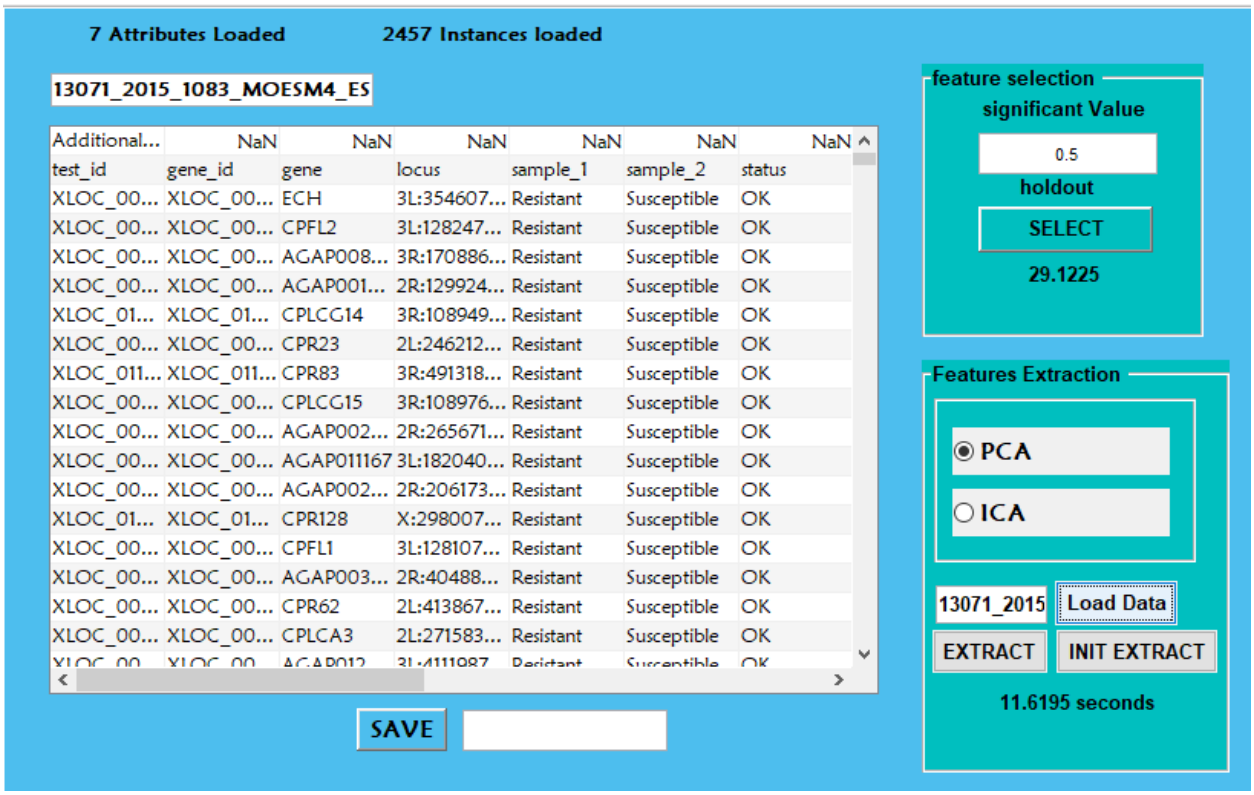
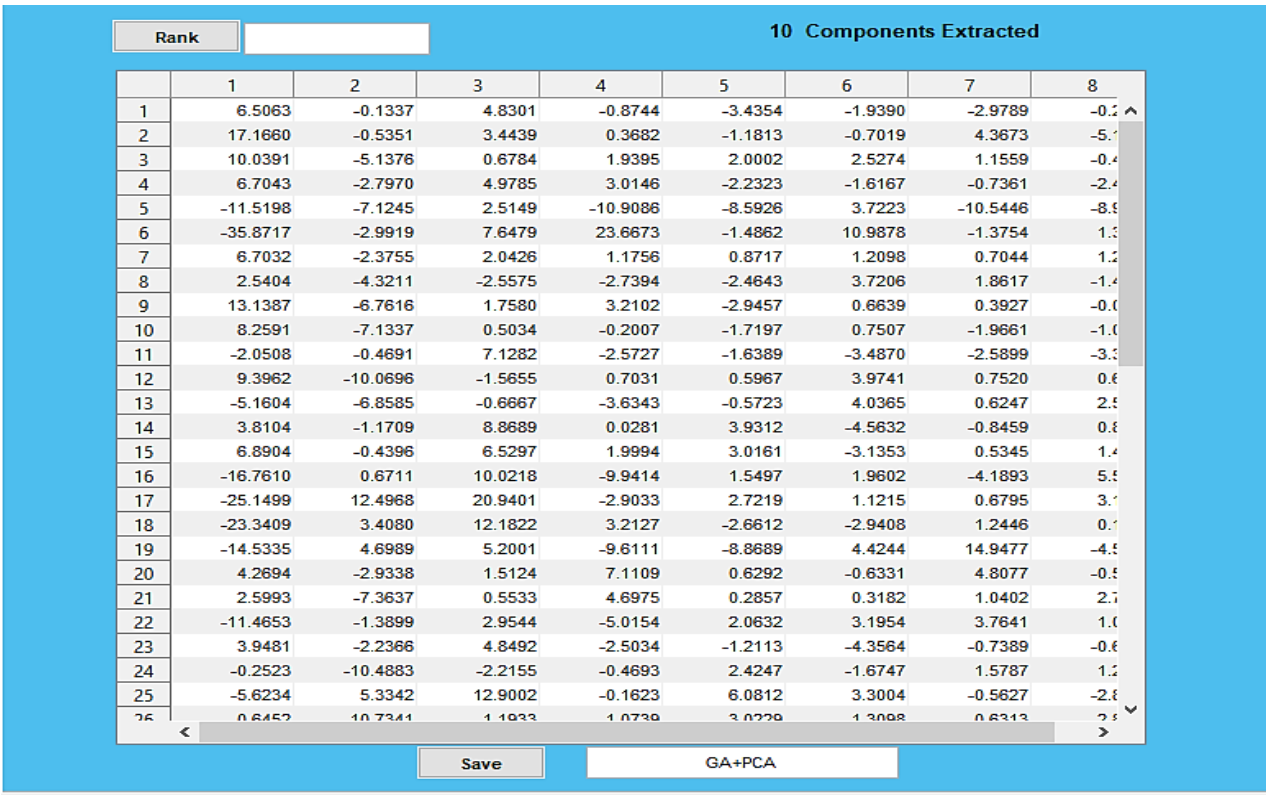



**Figure 4. 13 Feature Extraction Using PCA**




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 6.5063 | -0.1337 | 4.8301 | -0.8744 | -3.4354 | -1.9390 | -2.9789 | -0.2 |
| 2 | 17.1660 | -0.5351 | 3.4439 | 0.3682 | -1.1813 | -0.7019 | 4.3673 | -5.1 |
| 3 | 10.0391 | -5.1376 | 0.6784 | 1.9395 | 2.0002 | 2.5274 | 1.1559 | -0.4 |
| 4 | 6.7043 | -2.7970 | 4.9785 | 3.0146 | -2.2323 | -1.6167 | -0.7361 | -2.4 |
| 5 | -11.5198 | -7.1245 | 2.5149 | -10.9086 | -8.5926 | 3.7223 | -10.5446 | -8.9 |
| 6 | -35.8717 | -2.9919 | 7.6479 | 23.6673 | -1.4862 | 10.9878 | -1.3754 | 1.3 |
| 7 | 6.7032 | -2.3755 | 2.0426 | 1.1756 | 0.8717 | 1.2098 | 0.7044 | 1.2 |
| 8 | 2.5404 | -4.3211 | -2.5575 | -2.7394 | -2.4643 | 3.7206 | 1.8617 | -1.4 |
| 9 | 13.1387 | -6.7616 | 1.7580 | 3.2102 | -2.9457 | 0.6639 | 0.3927 | -0.0 |
| 10 | 8.2591 | -7.1337 | 0.5034 | -0.2007 | -1.7197 | 0.7507 | -1.9661 | -1.0 |
| 11 | -2.0508 | -0.4691 | 7.1282 | -2.5727 | -1.6389 | -3.4870 | -2.5899 | -3.3 |
| 12 | 9.3962 | -10.0696 | -1.5655 | 0.7031 | 0.5967 | 3.9741 | 0.7520 | 0.6 |
| 13 | -5.1604 | -6.8585 | -0.6667 | -3.6343 | -0.5723 | 4.0365 | 0.6247 | 2.5 |
| 14 | 3.8104 | -1.1709 | 8.8689 | 0.0281 | 3.9312 | -4.5632 | -0.8459 | 0.8 |
| 15 | 6.8904 | -0.4396 | 6.5297 | 1.9994 | 3.0161 | -3.1353 | 0.5345 | 1.4 |
| 16 | -16.7610 | 0.6711 | 10.0218 | -9.9414 | 1.5497 | 1.9602 | -4.1893 | 5.5 |
| 17 | -25.1499 | 12.4968 | 20.9401 | -2.9033 | 2.7219 | 1.1215 | 0.6795 | 3.1 |
| 18 | -23.3409 | 3.4080 | 12.1822 | 3.2127 | -2.6612 | -2.9408 | 1.2446 | 0.1 |
| 19 | -14.5335 | 4.6989 | 5.2001 | -9.6111 | -8.8689 | 4.4244 | 14.9477 | -4.5 |
| 20 | 4.2694 | -2.9338 | 1.5124 | 7.1109 | 0.6292 | -0.6331 | 4.8077 | -0.5 |
| 21 | 2.5993 | -7.3637 | 0.5533 | 4.6975 | 0.2857 | 0.3182 | 1.0402 | 2.7 |
| 22 | -11.4653 | -1.3899 | 2.9544 | -5.0154 | 2.0632 | 3.1954 | 3.7641 | 1.0 |
| 23 | 3.9481 | -2.2366 | 4.8492 | -2.5034 | -1.2113 | -4.3564 | -0.7389 | -0.6 |
| 24 | -0.2523 | -10.4883 | -2.2155 | -0.4693 | 2.4247 | -1.6747 | 1.5787 | 1.2 |
| 25 | -5.6234 | 5.3342 | 12.9002 | -0.1623 | 6.0812 | 3.3004 | -0.5627 | -2.8 |
| 26 | 0.6452 | 10.7341 | 1.1933 | 1.0739 | 3.0229 | 1.3098 | 0.6313 | 2.8 |

**Figure 4. 14 Using PCA on the Pre-processed Anopheles Gambiae RNA-Seq Dataset**

To each of the classifiers, a basic supervised learning assessment protocol is carried out. In particular, the training and testing stages are assessed as 10-fold cross-validation to eliminate the sampling bias. This protocol is implemented using MATLAB 2015 platform. The reported result of the assessment is based on the following performance metrics (Accuracy, Sensitivity, Specificity, F-score, Precision and Recall) (Nathan *et al.*, 2017). This study compares the classification performance of the models, using SVM, KNN, Decision tree and Ensemble using the confusion matrix to calculate the performance metrics.

### 4.3.2. PCA with SVM Classifiers

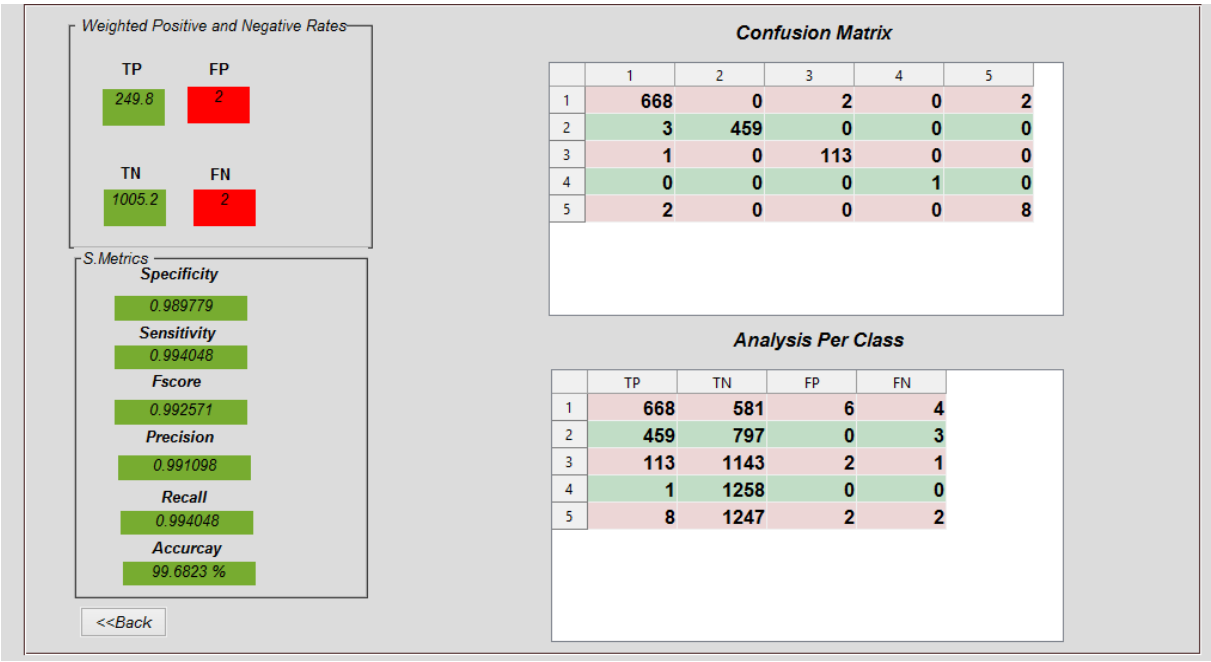



**Figure 4. 15 Confusion Matrix for PCA with SVM-Polynomial Kernel**

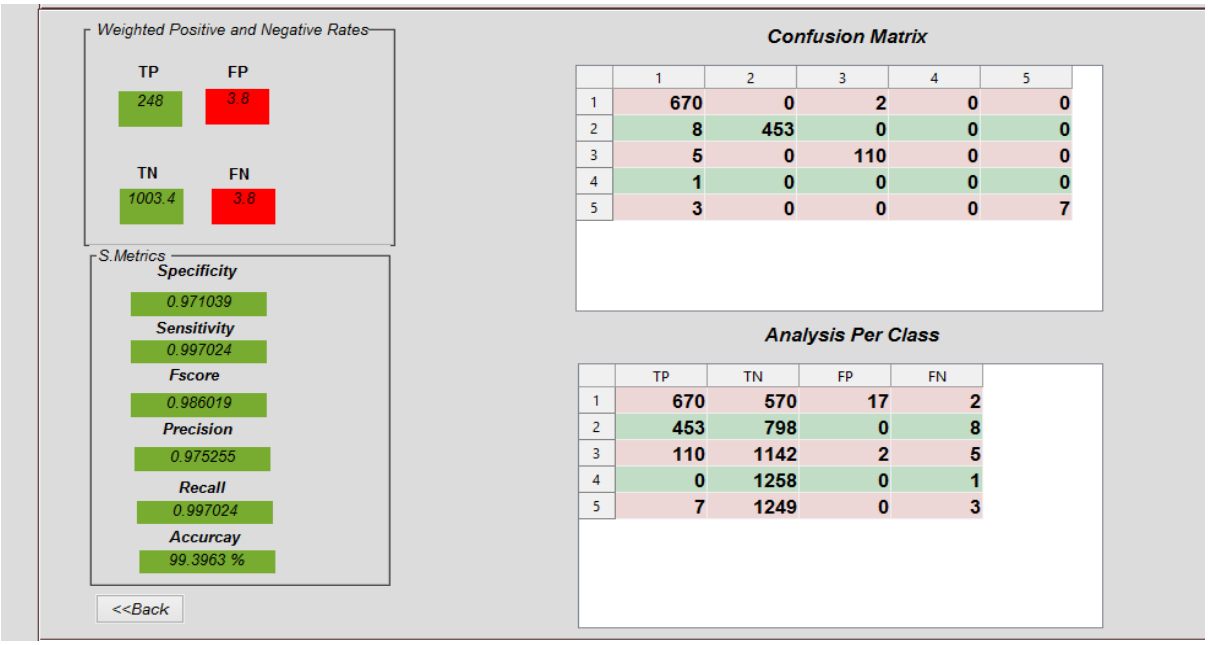



**Figure 4. 16 Confusion Matrix for PCA+SVM-Gaussian Kernel**

**Table 4. 4 Execution Results Table for PCA with SVM Classifiers**

| Performance Metrics | PCA+SVM-Polynomial Kernel | PCA+SVM-Gaussian Kernel |
|---|---|---|
| Accuracy (%) | 99.68 | 99.39 |
| Sensitivity (%) | 99.40 | 99.71 |
| Specificity (%) | 98.97 | 97.10 |
| F-Score (%) | 99.25 | 98.60 |
| Precision (%) | 99.10 | 97.52 |
| Recall (%) | 99.40 | 99.70 |

### 4.3.3. PCA with K-NN

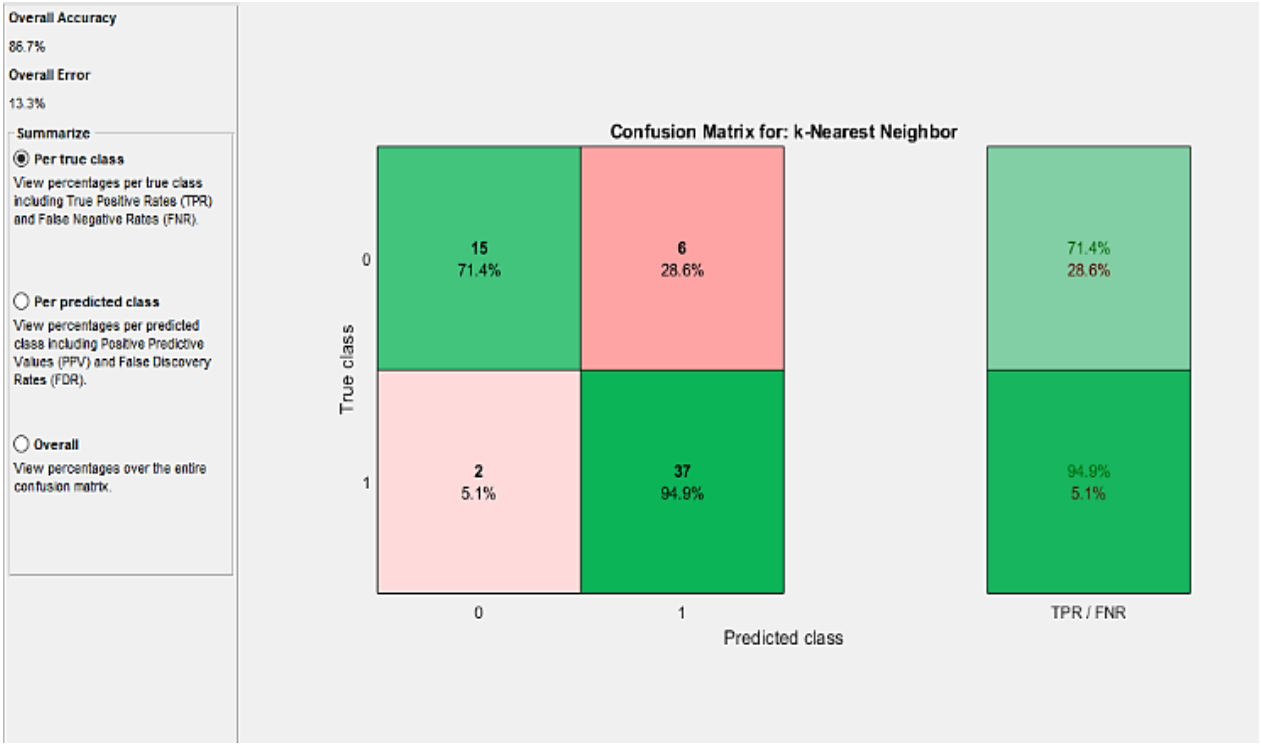



**Figure 4. 17 Confusion Matrix for PCA with K-NN. TP=37; TN=15; FP=6; FN=2**

### 4.3.4. PCA with Decision Tree

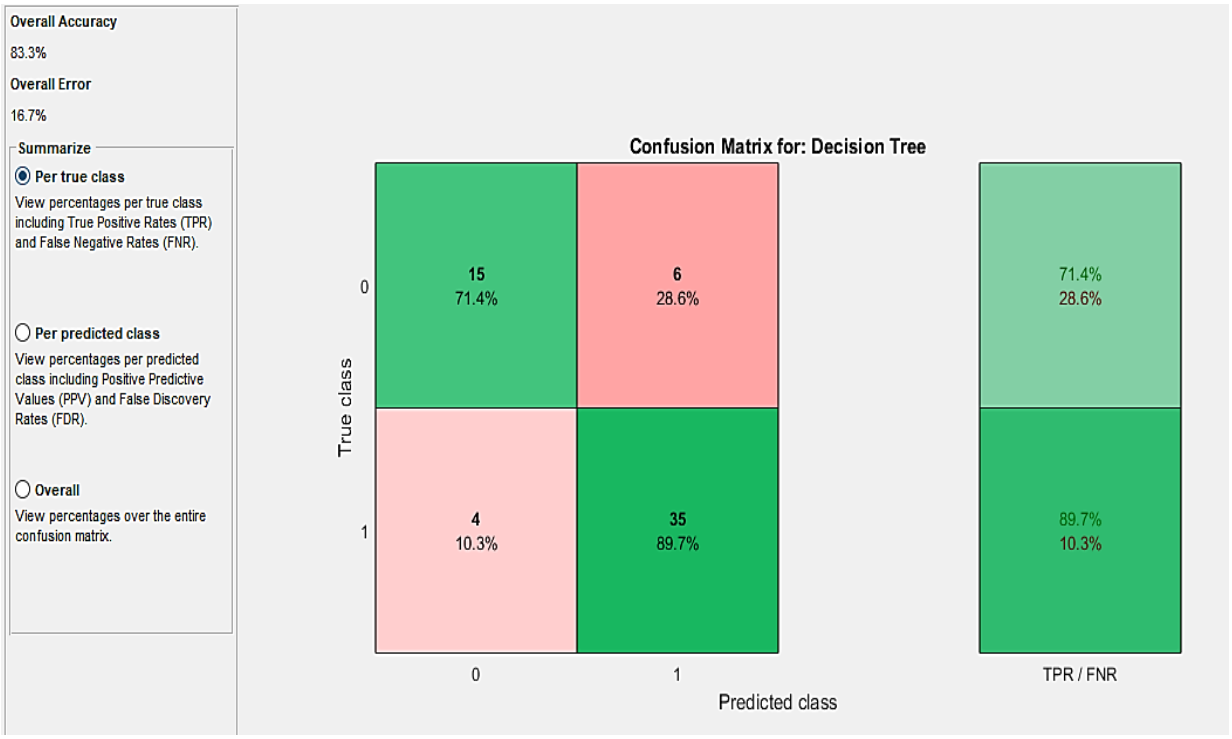



**Figure 4. 18 Confusion Matrix for PCA with Decision Tree Classifier.**

**TP=35; TN=15;FP=6; FN=4**

**Table 4. 5 Performance Metrics Table for the PCA with K-NN and PCA with Decision Tree Classifiers**

| Performance Metrics | PCA+K-NN Classification | PCA + Decision Tree Classification |
|---|---|---|
| Accuracy (%) | 86.7 | 83.3 |
| Sensitivity (%) | 94.9 | 89.7 |
| Specificity (%) | 71.4 | 71.4 |
| Precision (%) | 86.1 | 85.4 |
| Recall (%) | 94.9 | 89.7 |
| F-Score (%) | 90.3 | 87.5 |

### 4.3.5. PCA with Ensemble Classification Approach

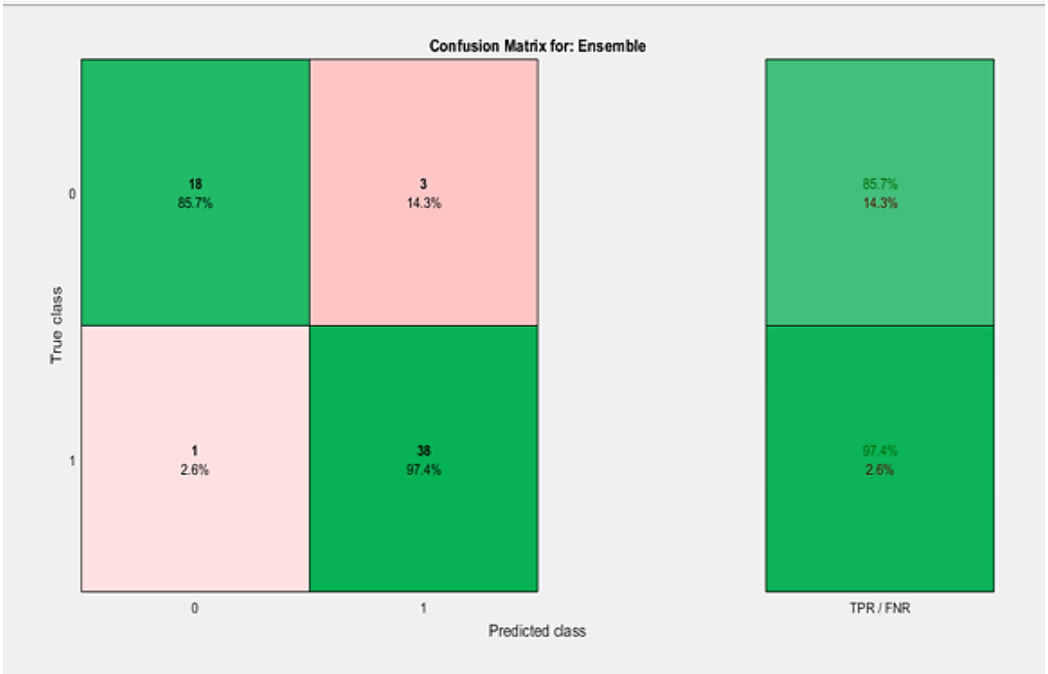



**Figure 4. 19 Confusion Matrix for PCA with Ensemble Classification.**

**TP=38; TN=18;FP=3; FN=1**

**Table 4. 6 Performance Metrics Table for the PCA with Ensemble Classifier**

| Performance Metrics | PCA+ Ensemble Classification |
|---|---|
| **Accuracy (%)** | **93.3** |
| **Sensitivity (%)** | **97.4** |
| **Specificity (%)** | **85.7** |
| **Precision (%)** | **92.7** |
| **Recall (%)** | **97.4** |
| **F-Score (%)** | **93.7** |
| **ROC Curve (%)** | **99.6** |
| **Training Time (sec)** | **11.6195** |

PCA feature extraction was used to fetch relevant latent components from the *Anopheles gambiae* dataset was classified and PCA with SVM outperformed other methods as shown in Table 4.4.

## 4.4. ICA classifications

The *Anopheles gambiae* dataset uses ICA as a linear feature extraction approach to reduce the curse of dimensionality, the ICA algorithm was applied as a dimension reduction extraction to identify and eliminate uncorrelated attributes (variables) to determine the overall variance for a smaller number of individual components and fetched out a subset in 2.0509 seconds, as shown in Figure 4.20.

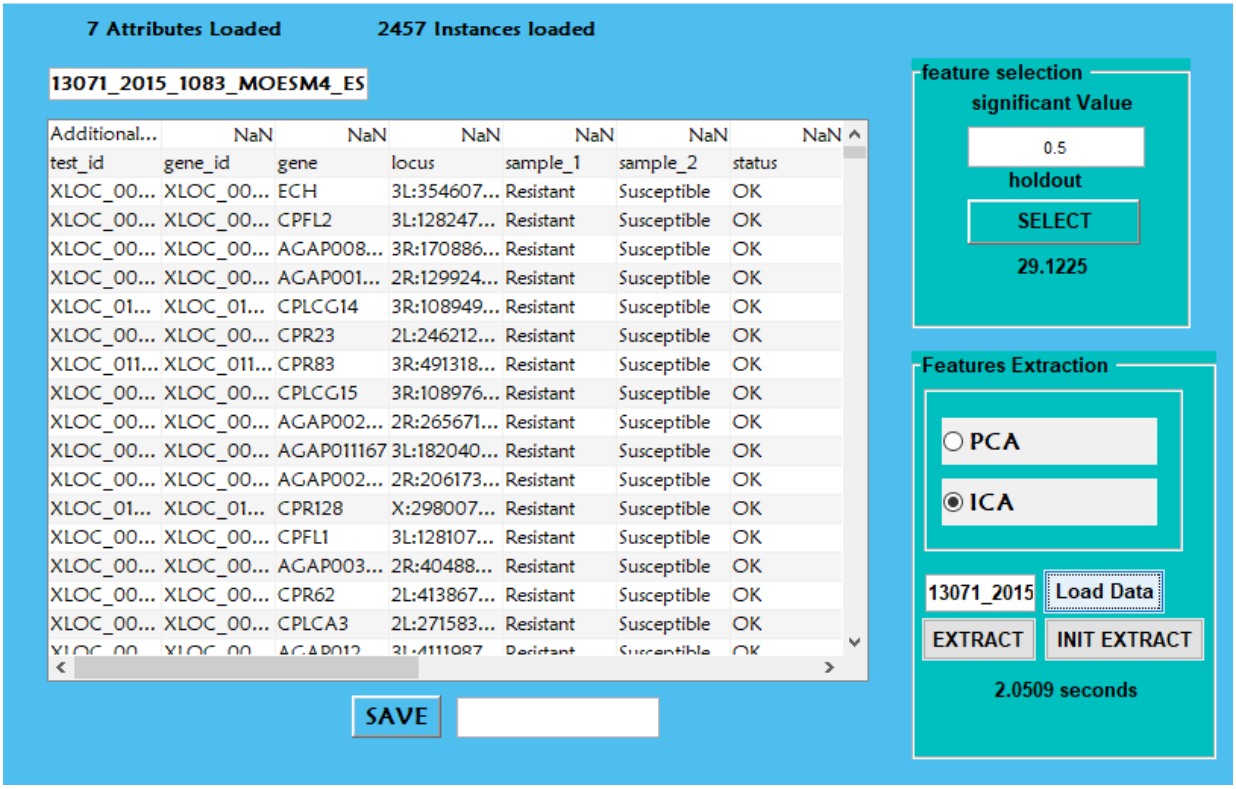
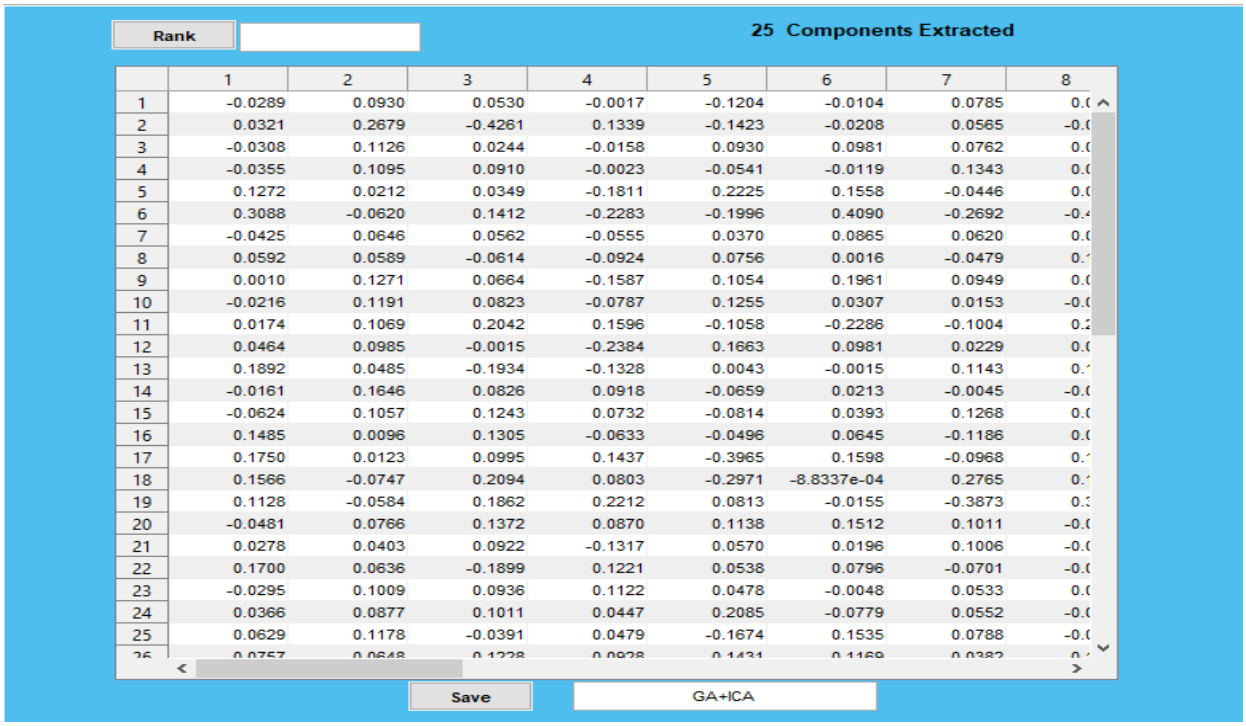



**Figure 4. 20 Feature Extraction Using ICA Feature Extraction Algorithm**

### 4.4.1. ICA Feature Extraction Algorithm with Classifications Approaches

Here, ICA is added to data from Mosquito Anopheles, which offers important gene knowledge that is useful for further studies. Classification algorithms use SVM kernels to execute the model using the MATLAB function. Using ICA as a reduction tool for extraction dimensionality, 25 latent component features of genes were important and shown in figure 4.21.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | -0.0289 | 0.0930 | 0.0530 | -0.0017 | -0.1204 | -0.0104 | 0.0785 | 0.0 |
| 2 | 0.0321 | 0.2679 | -0.4261 | 0.1339 | -0.1423 | -0.0208 | 0.0565 | -0.0 |
| 3 | -0.0308 | 0.1126 | 0.0244 | -0.0158 | 0.0930 | 0.0981 | 0.0762 | 0.0 |
| 4 | -0.0355 | 0.1095 | 0.0910 | -0.0023 | -0.0541 | -0.0119 | 0.1343 | 0.0 |
| 5 | 0.1272 | 0.0212 | 0.0349 | -0.1811 | 0.2225 | 0.1558 | -0.0446 | 0.0 |
| 6 | 0.3088 | -0.0620 | 0.1412 | -0.2283 | -0.1996 | 0.4090 | -0.2692 | -0.4 |
| 7 | -0.0425 | 0.0646 | 0.0562 | -0.0555 | 0.0370 | 0.0865 | 0.0620 | 0.0 |
| 8 | 0.0592 | 0.0589 | -0.0614 | -0.0924 | 0.0756 | 0.0016 | -0.0479 | 0.1 |
| 9 | 0.0010 | 0.1271 | 0.0664 | -0.1587 | 0.1054 | 0.1961 | 0.0949 | 0.0 |
| 10 | -0.0216 | 0.1191 | 0.0823 | -0.0787 | 0.1255 | 0.0307 | 0.0153 | -0.0 |
| 11 | 0.0174 | 0.1069 | 0.2042 | 0.1596 | -0.1058 | -0.2286 | -0.1004 | 0.2 |
| 12 | 0.0464 | 0.0985 | -0.0015 | -0.2384 | 0.1663 | 0.0981 | 0.0229 | 0.0 |
| 13 | 0.1892 | 0.0485 | -0.1934 | -0.1328 | 0.0043 | -0.0015 | 0.1143 | 0.1 |
| 14 | -0.0161 | 0.1646 | 0.0826 | 0.0918 | -0.0659 | 0.0213 | -0.0045 | -0.0 |
| 15 | -0.0624 | 0.1057 | 0.1243 | 0.0732 | -0.0814 | 0.0393 | 0.1268 | 0.0 |
| 16 | 0.1485 | 0.0096 | 0.1305 | -0.0633 | -0.0496 | 0.0645 | -0.1186 | 0.0 |
| 17 | 0.1750 | 0.0123 | 0.0995 | 0.1437 | -0.3965 | 0.1598 | -0.0968 | 0.1 |
| 18 | 0.1566 | -0.0747 | 0.2094 | 0.0803 | -0.2971 | -8.8337e-04 | 0.2765 | 0.1 |
| 19 | 0.1128 | -0.0584 | 0.1862 | 0.2212 | 0.0813 | -0.0155 | -0.3873 | 0.3 |
| 20 | -0.0481 | 0.0766 | 0.1372 | 0.0870 | 0.1138 | 0.1512 | 0.1011 | -0.0 |
| 21 | 0.0278 | 0.0403 | 0.0922 | -0.1317 | 0.0570 | 0.0196 | 0.1006 | -0.0 |
| 22 | 0.1700 | 0.0636 | -0.1899 | 0.1221 | 0.0538 | 0.0796 | -0.0701 | -0.0 |
| 23 | -0.0295 | 0.1009 | 0.0936 | 0.1122 | 0.0478 | -0.0048 | 0.0533 | 0.0 |
| 24 | 0.0366 | 0.0877 | 0.1011 | 0.0447 | 0.2085 | -0.0779 | 0.0552 | -0.0 |
| 25 | 0.0629 | 0.1178 | -0.0391 | 0.0479 | -0.1674 | 0.1535 | 0.0788 | -0.0 |
| 26 | 0.0757 | 0.0648 | 0.1228 | 0.0928 | 0.1431 | 0.1169 | 0.0382 | 0.1 |

**Figure 4. 21 Using ICA on the Pre-processed Anopheles Gambiae RNA-Seq Dataset**

The classification technique uses 10-folds cross-validation using 0.05 parameter holdout data for training, and 5 percent for testing to verify the accuracy of classifiers, the efficiency of the classification models was evaluated. The extracted features are passed into the classifiers, and the results of the confusion matrix are shown in figures 4.22 to 4.27, the confusion matrix gives a solution to the performance metrics and are shown in Tables 4.7 to 4.9.

### 4.4.2. ICA with SVM Classifiers

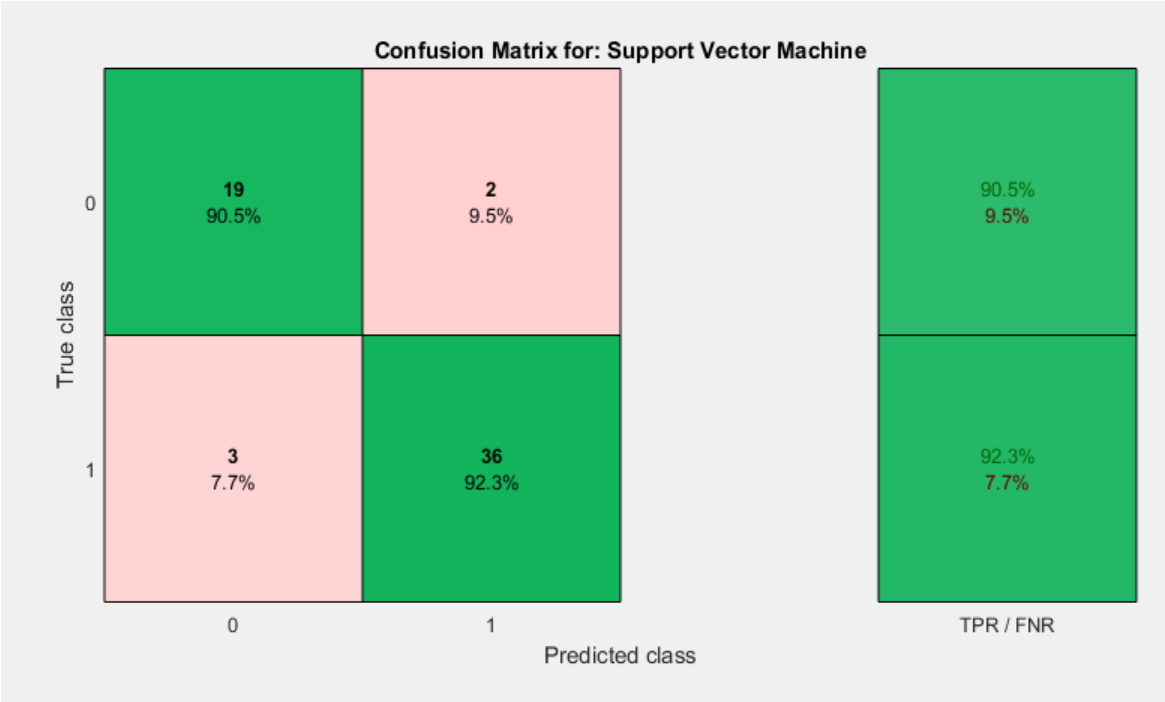



**Figure 4. 22 Confusion Matrix for the ICA with Linear-SVM (L-SVM) TP=36; TN=19; FP=2; FN=3**

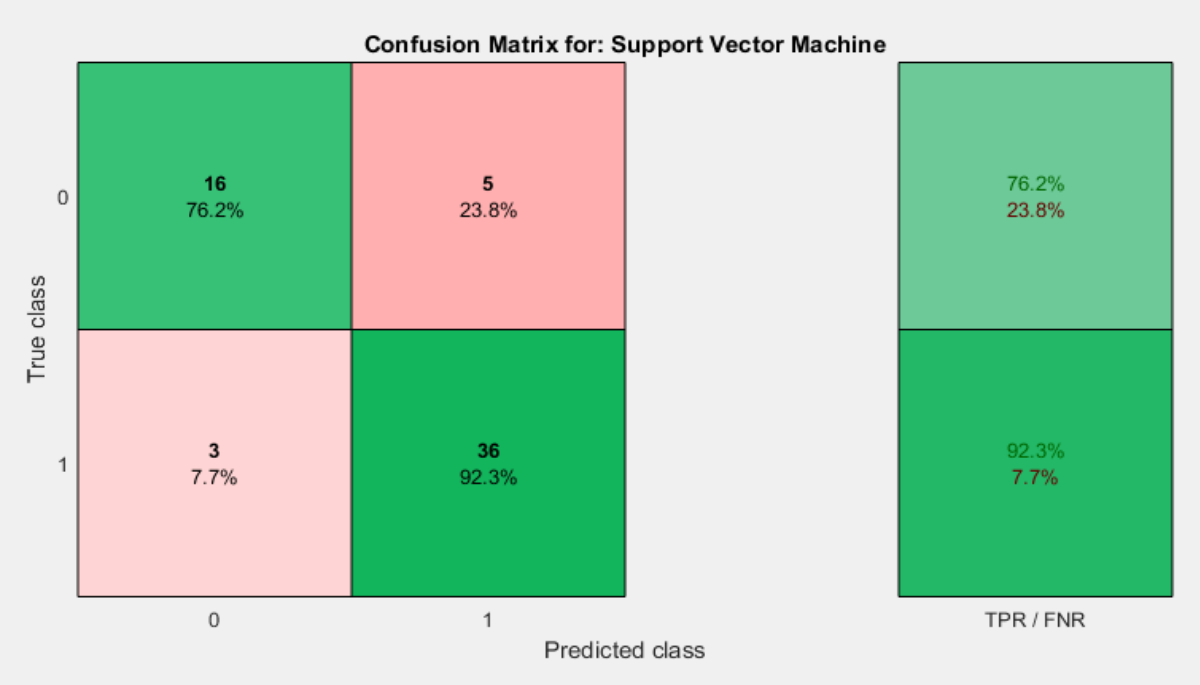



**Figure 4. 23 Confusion Matrix for ICA with Radial Basis Function - SVM (RBF-SVM) TP=36; TN=16; FP=5; FN=3**

**Table 4. 7 Performance Metrics Table for the ICA with SVM Classifiers**

| Performance Metrics | ICA+L-SVM Classification | ICA+RBF-SVM Classification |
|---|---|---|
| Accuracy (%) | 91.7 | 86.7 |
| Sensitivity (%) | 92.3 | 92.3 |
| Specificity (%) | 90.5 | 76.2 |
| Precision (%) | 94.7 | 87.8 |
| Recall (%) | 92.3 | 92.3 |
| F-Score (%) | 93.5 | 90.0 |

### 4.4.3. ICA with K-NN Classifier

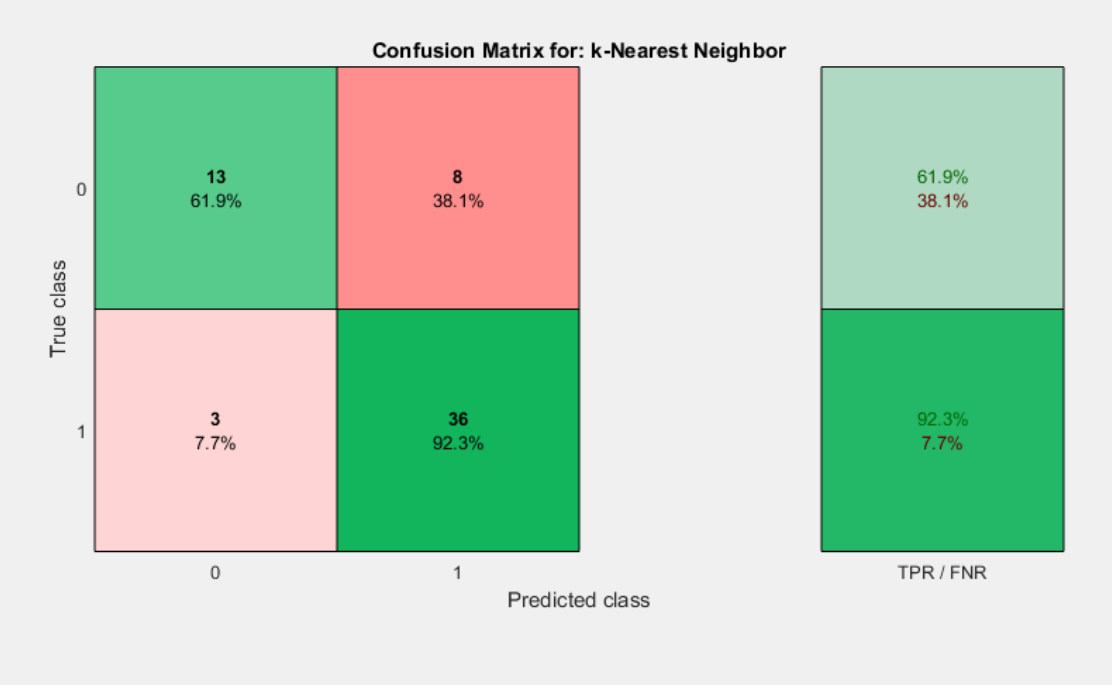



**Figure 4. 24 Confusion Matrix for ICA with K-NN**

**TP=36; TN=13; FP=8; FN=3**

### 4.4.4. ICA with Decision Tree Classification

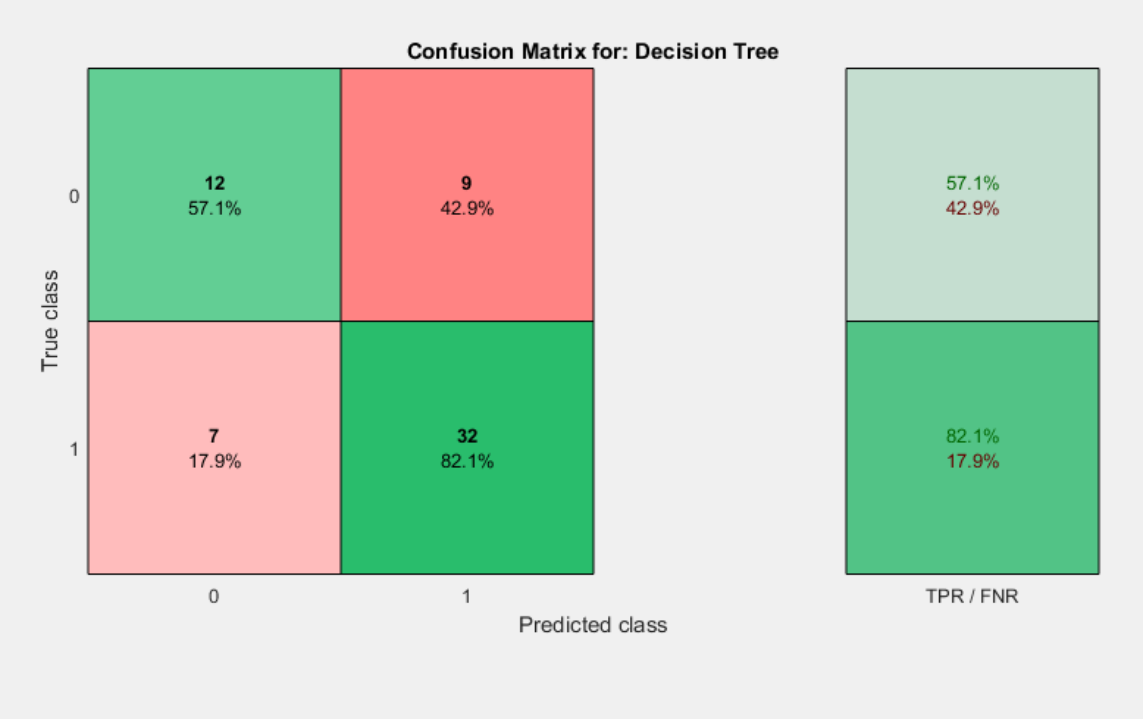



**Figure 4. 25 Confusion Matrix for the ICA with Decision Tree.**

**TP=32; TN=12; FP=9; FN=7**

**Table 4. 8 Performance Metrics Table for the ICA-K-NN and ICA with Decision Tree Classifiers**

| Performance Metrics | ICA-K-NN Classification | ICA-Decision Tree Classification |
|---|---|---|
| Accuracy (%) | 81.7 | 73.3 |
| Sensitivity (%) | 92.3 | 82.1 |
| Specificity (%) | 62.0 | 57.1 |
| Precision (%) | 81.8 | 78.1 |
| Recall (%) | 92.3 | 82.1 |
| F-Score (%) | 86.7 | 80.1 |

### 4.4.5. ICA with Ensemble Classification Approaches


**Figure 4. 26 Confusion Matrix for ICA with Ensemble (Boosted) Subspace Discriminant Classification TP=38; TN=18; FP=3 FN=1**

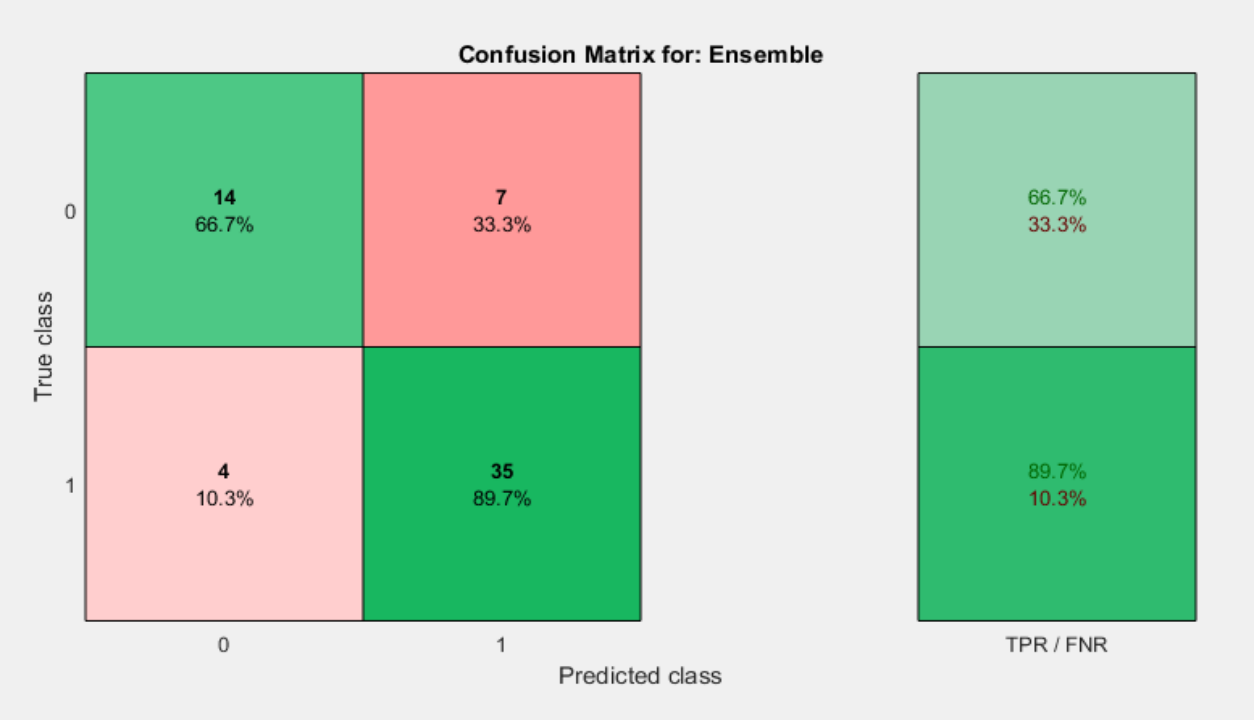



**Figure 4. 27 Confusion Matrix for ICA with Ensemble Bagged Tree Classification TP=35; TN=14; FP=7; FN=4**

**Table 4. 9 Performance Metrics Table for the ICA with Ensemble Classifiers**

| Performance Metrics | ICA + Ensemble Subspace Discriminant Classification | ICA + Ensemble Bagged Tree Classification |
|---|---|---|
| Accuracy (%) | 93.3 | 81.7 |
| Sensitivity (%) | 97.4 | 89.7 |
| Specificity (%) | 85.7 | 66.7 |
| Precision (%) | 92.7 | 83.3 |
| Recall (%) | 97.4 | 89.7 |
| F-Score (%) | 95.0 | 86.4 |

Traditional methods such as the Genetic algorithm, PCA, and ICA methods have been widely used as a first standard stage in dimension reduction and classification. They suffer from numerous limitations, such as class imbalance and undersized samplings. They inherently have distinctive difficulties in evaluating small samples in high dimensions, due to singularity of the covariance matrix. These difficulties have not been satisfactorily addressed in the literature. Since sample covariance matrix degenerates and becomes singular, computational cost, making the classification accuracies high, as seen in the results of single dimension reduction procedure. It has been observed that these procedures exhibit noises still, and needs proper elimination. The single approach has proven in this experiment to fetch for solutions by sacrificing totality to increase efficiency in a reasonable time, but not good enough for solving the problems at hand.

Hybrid approach has been proposed in recent time by numerous authors using the advantages of the dimensionality reduction methods. It consists of two steps majorly to identify best relevant features, by using different criteria at different stages to further improve the efficiency of the classification and further reduced the noises and computational cost becomes acceptable. Consequently, a hybrid dimensionality approach is carried out in the current work.

## 4.5. Hybridized models

The hybrid dimensionality reduction models developed uses GA-O-PCA as a non-linear approach and GA-O-ICA as a linear approach, the GA-O selects the relevant features, PCA and ICA further selects optimal latent components from the reduced data. SVM, KNN, Decision Tree and Ensemble classifiers were used in the hybridized approach. Scattered plots were used to depict the relationship amongst variables were achieved as shown in figures 4.28 to 4.41. The classification achieves confusion matrices as shown in Figures 4.29 to 4.41. also, the performance metrics table were analyzed and shown in Tables 4.9 to 4.13.

### 4.5.1. The GA-O with PCA with SVM Results

The extracted features are passed into the SVM classification algorithm using 10-folds cross-validation and the confusion matrix of L-SVM and Medium Gaussian SVM classification algorithms are evaluated.

In order to represent values for two separate numeric variables, a scatter plot utilizes dots. Each dot's location on the horizontal and vertical axis shows the values for the individual data point. For analyzing relationships between variables, scatter plots are used.

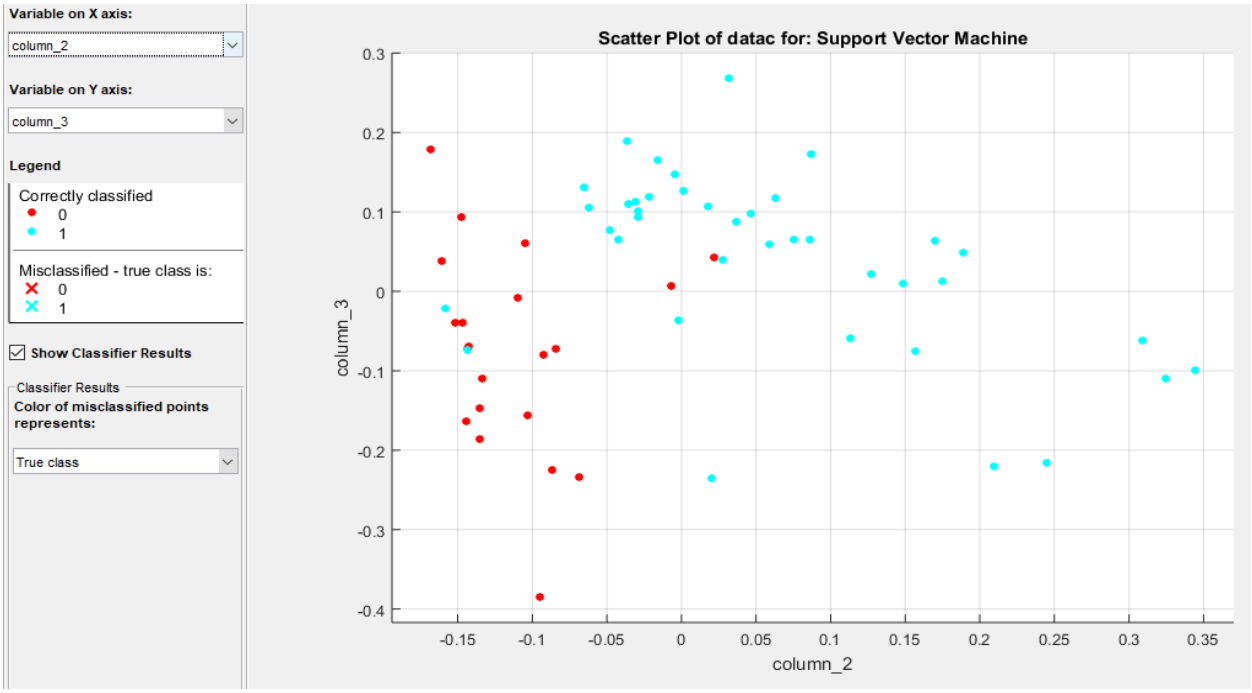



**Figure 4. 28 A scatter plot of the SVM attributes to show effects of the Variables.**

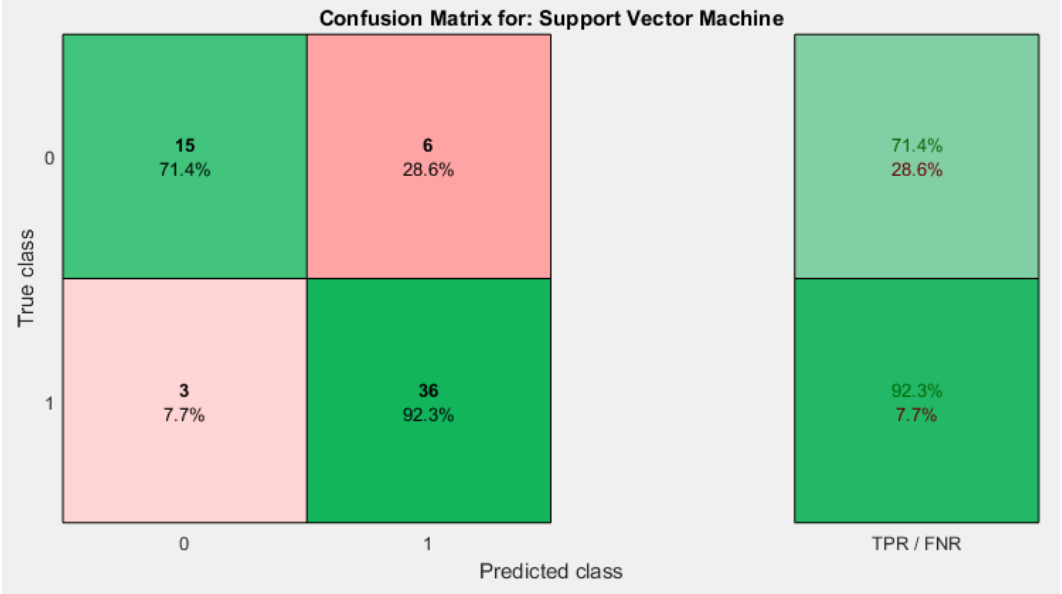



**Figure 4. 29 Confusion Matrix for GA+PCA+ SVM-RBF**

**TP= 36; TN= 15; FP= 6; FN= 3.**

### 4.5.2. The GA-O with ICA with SVM Results

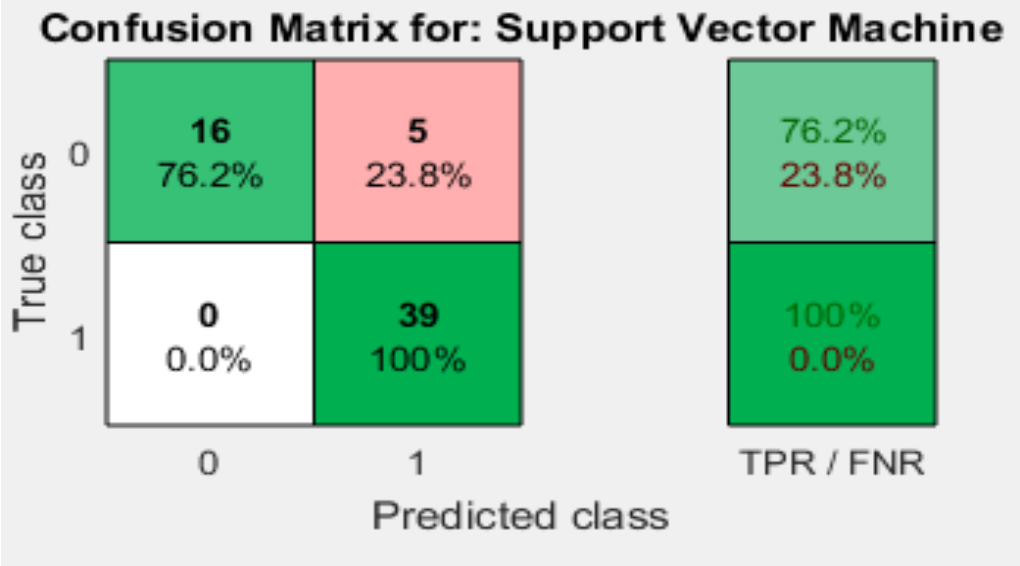



**Figure 4. 30 Confusion Matrix for GA-O+ICA+ SVM-RBF**

**TP= 39; TN= 16; FP= 5; FN= 0.**

**Table 4. 10 Performance Metrics Table for the GA-O with PCA and SVM, GA-O with ICA and SVM classifiers**

| Performance Metrics (%) | GA-O +PCA+ SVM-RBF | GA-O +ICA+SVM-RBF |
|---|---|---|
| **Accuracy** | **85.0** | **91.7** |
| **Sensitivity** | **92.3** | **100** |
| **Specificity** | **71.4** | **76.2** |
| **Precision** | **85.7** | **88.6** |
| **Recall** | **92.3** | **100** |
| **F-score** | **88.9** | **94** |

### 4.5.3. The GA-O with PCA with K-NN Results

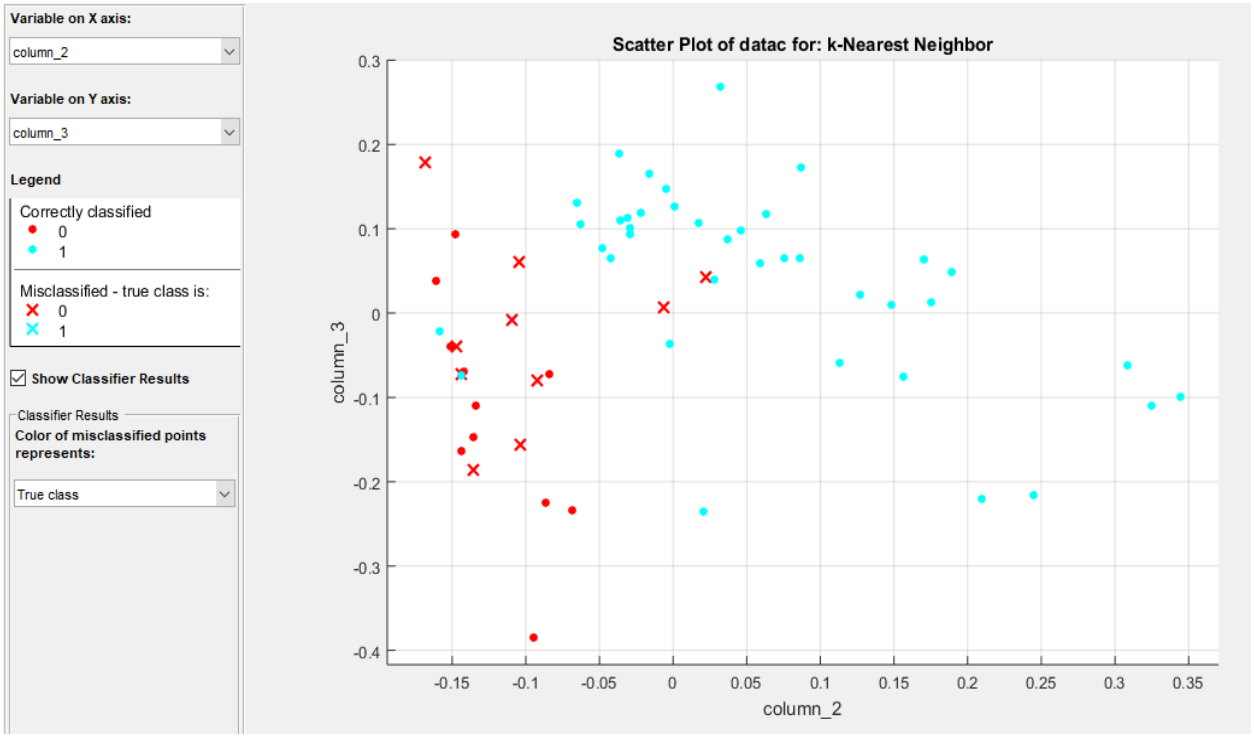



**Figure 4. 31 A Scatter Plot of The Attributes For K-NN to Show Effects of The Variables.**

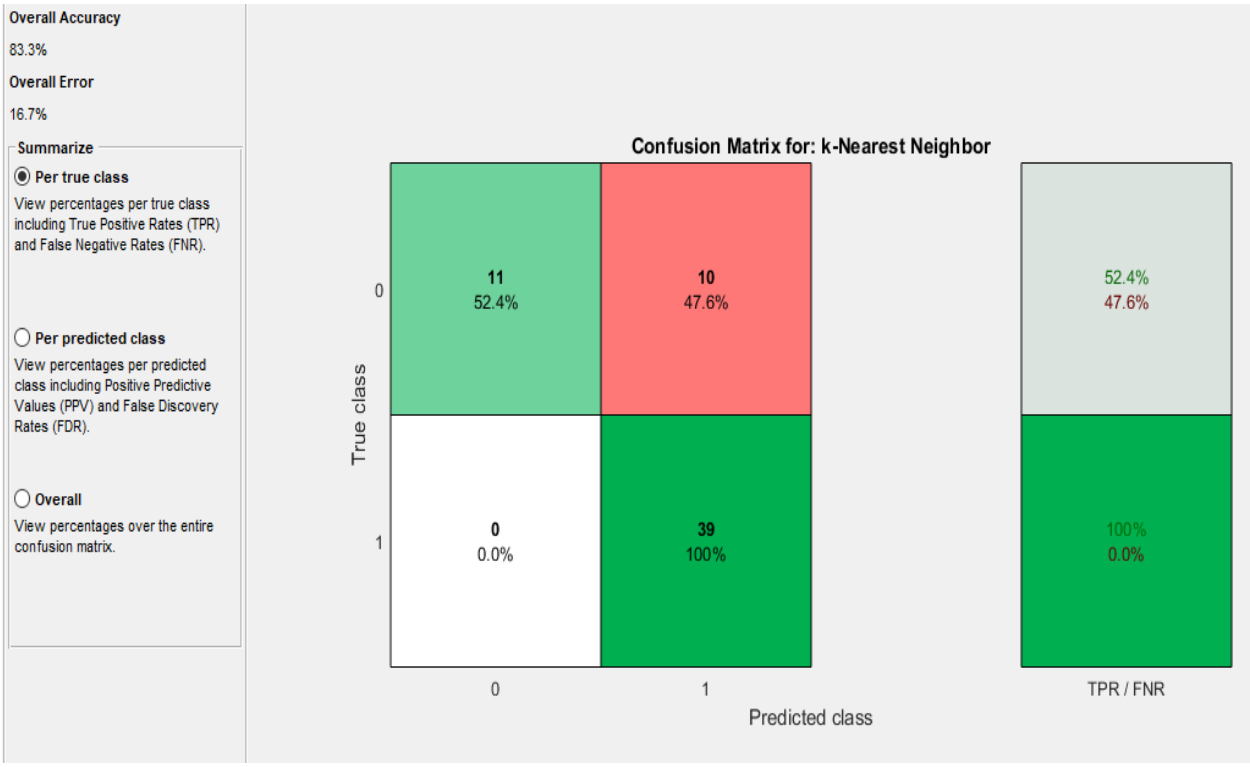



**Figure 4. 32 Confusion Matrix for GA-O+PCA+K-NN.**

**TP= 39; TN= 11; FP= 10; FN= 0.**

### 4.5.4. The GA-O with ICA with K-NN Results

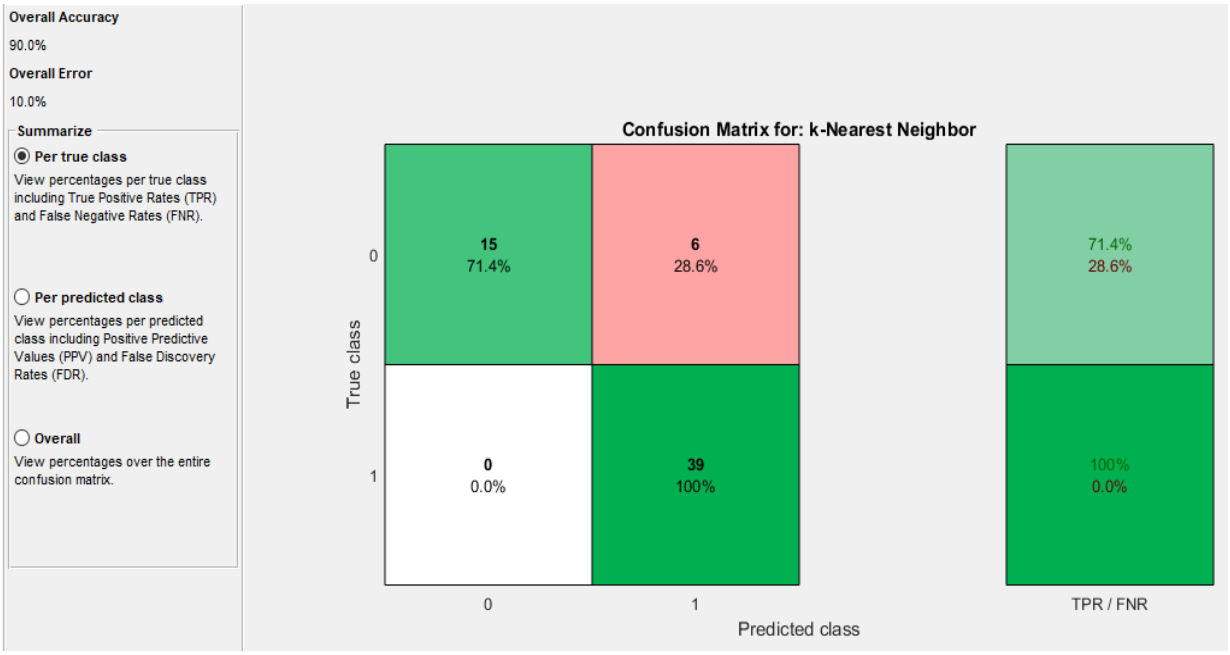



**Figure 4. 33 Confusion matrix for GA-O+ICA+K-NN TP= 39; TN= 15; FP= 6; FN= 0.**

**Table 4. 11 Performance Metrics Table for the GA-O+PCA+K-NN and GA-O+ICA+KNN Classification**

| Performance Metrics (%) | GA-O+PCA+K-NN | GA-O+ICA+K-NN |
|---|---|---|
| Accuracy | 88.3 | 90 |
| Sensitivity | 100 | 100 |
| Specificity | 52.4 | 52.4 |
| Precision | 79.6 | 86.7 |
| Recall | 100 | 100 |
| F-score | 88.6 | 92.88 |

### 4.5.5. The Ensemble Classification Results

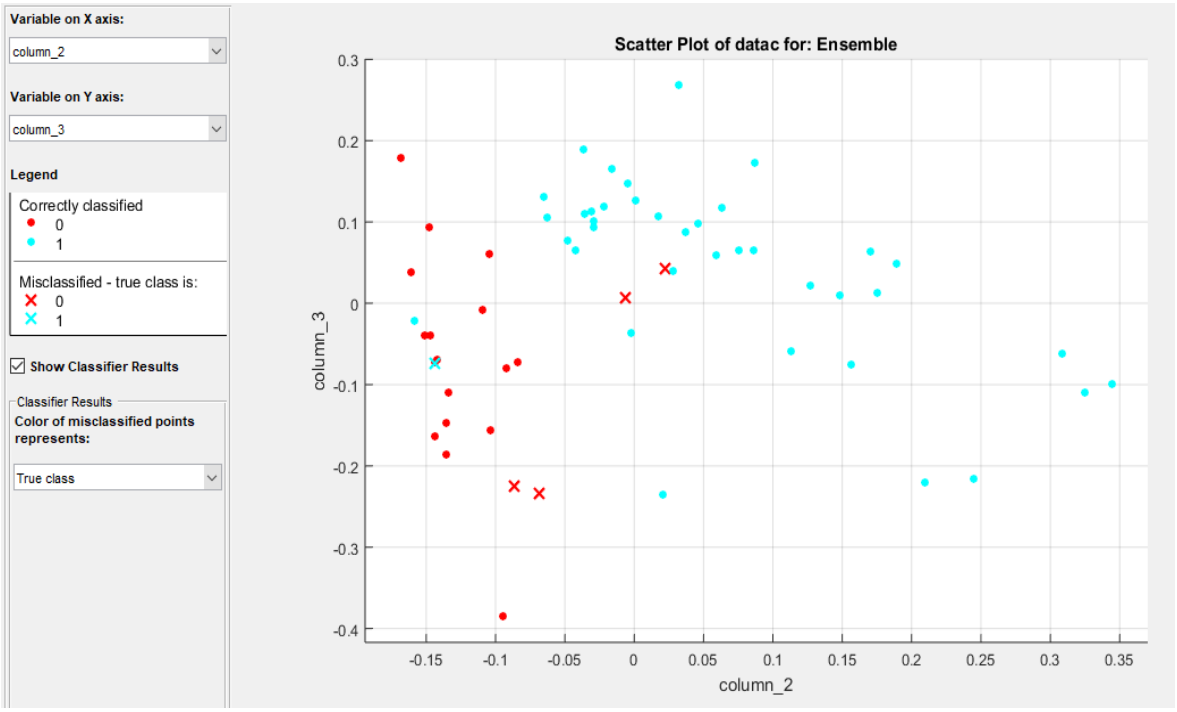



**Figure 4. 34 A Scatter Plot of The Attributes Ensemble to Show Effects of The Variables.**

### 4.5.6. The GA-O with PCA with Ensemble Approaches

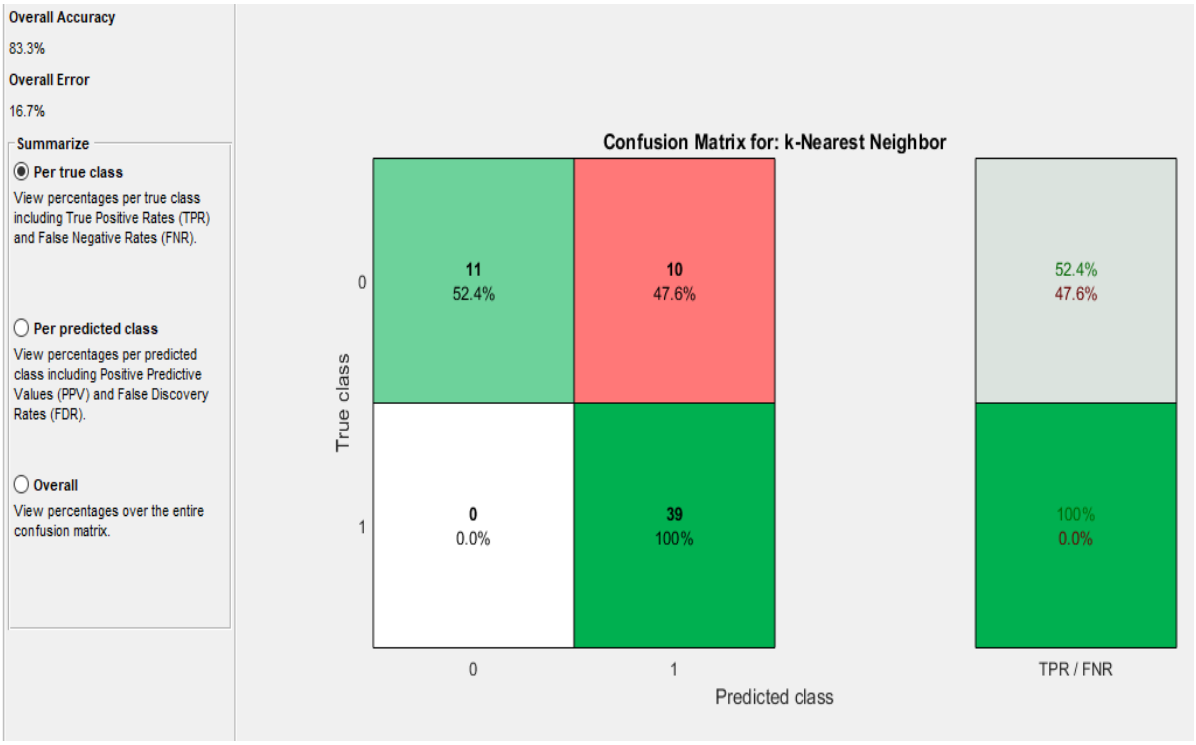



**Figure 4. 35 Confusion Matrix for GA-O+PCA+ Ensemble (boosted)**

**TP= 39; TN= 11; FP= 10; FN= 0.**

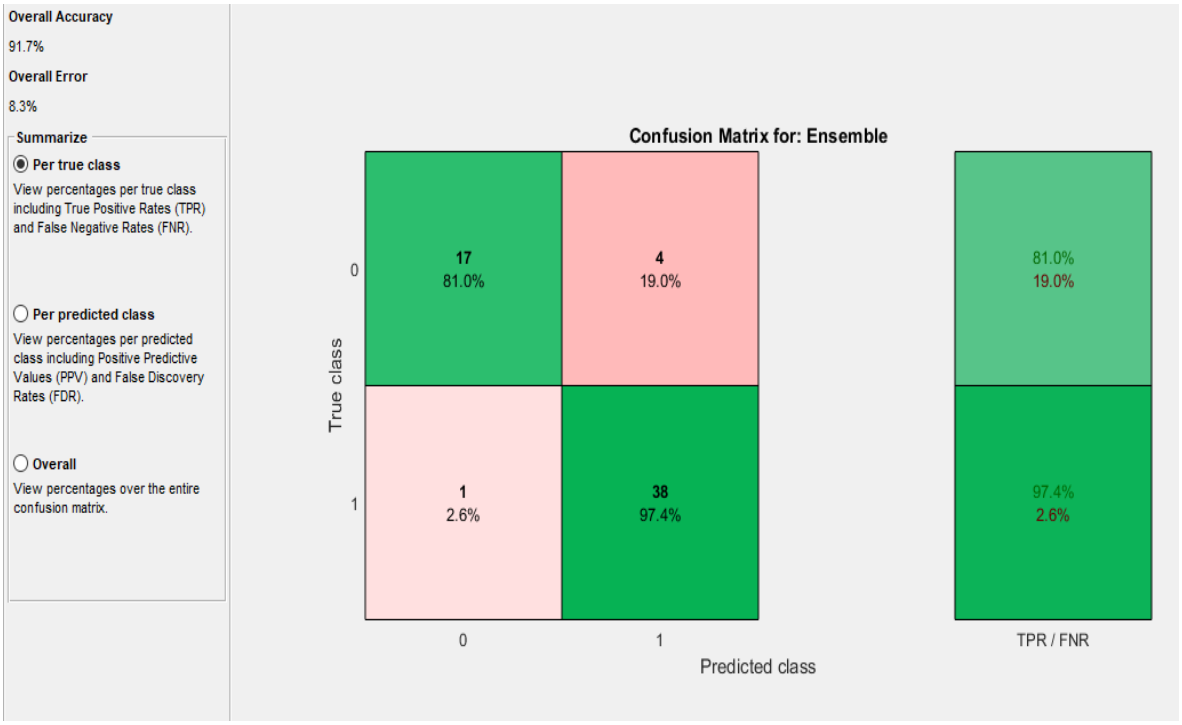



**Figure 4. 36 Confusion Matrix for GA-O+PCA+ Ensemble (bagged)**

**TP= 38; TN= 17; FP= 4; FN=1.**

### 4.5.7. The GA-O with ICA with Ensemble Approaches

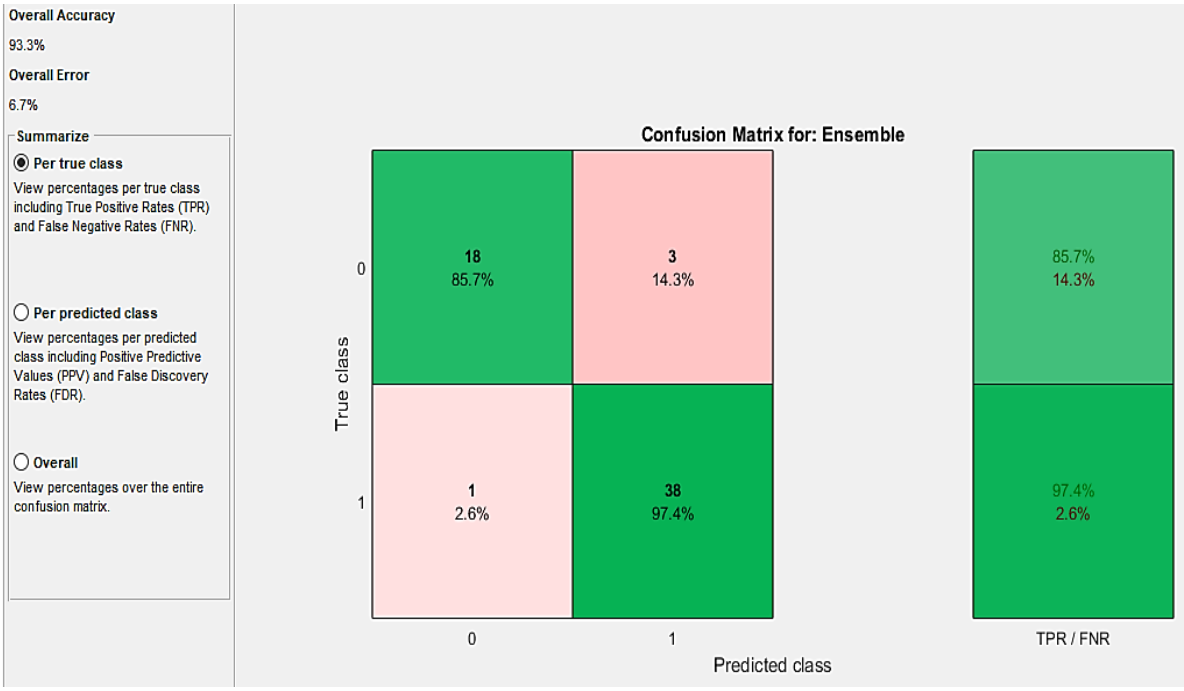



**Figure 4. 37 Confusion Matrix for GA-O + ICA + Ensemble (boosted)**

**TP= 38; TN= 18; FP= 3; FN= 1.**

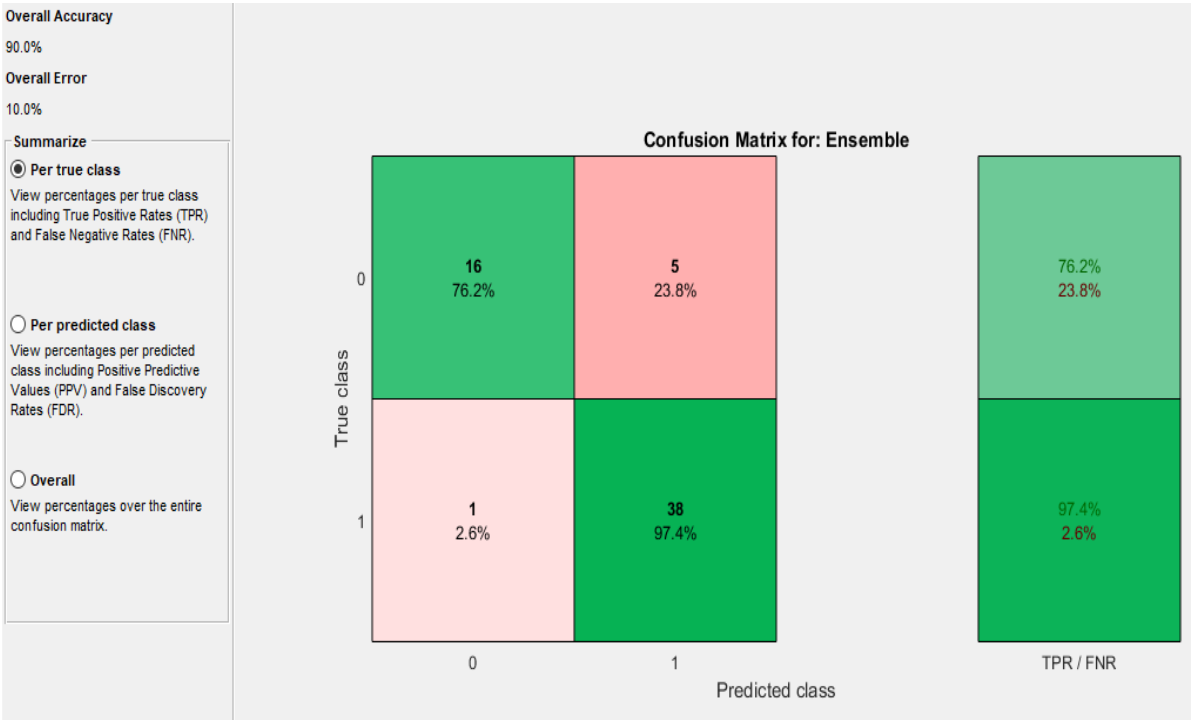



**Figure 4. 38 Confusion Matrix for GA-O + ICA + Ensemble (bagged)**

**TP= 38; TN= 16; FP= 5; FN= 1**

**Table 4. 12 Performance Metrics Table for the GA-O + PCA + Ensemble Classification**

| Performance Metrics (%) | GA-O+PCA + Ensemble (boosted) | GA-O+PCA + Ensemble (Bagged) | GA-O+ICA+ Ensemble (boosted) | GA-O+ICA+ Ensemble (Bagged) |
|---|---|---|---|---|
| Accuracy | 83 | 91.7 | 93.0 | 90.0 |
| Sensitivity | 100 | 97.4 | 97 | 97.4 |
| Specificity | 52.0 | 81.0 | 81 | 85.7 |
| Precision | 80.0 | 91.0 | 90.5 | 88.4 |
| Recall | 100 | 97.0 | 97.4 | 97.0 |
| F-score | 89.0 | 94.0 | 93.8 | 92.5 |

### 4.5.8. The Decision Tree Results

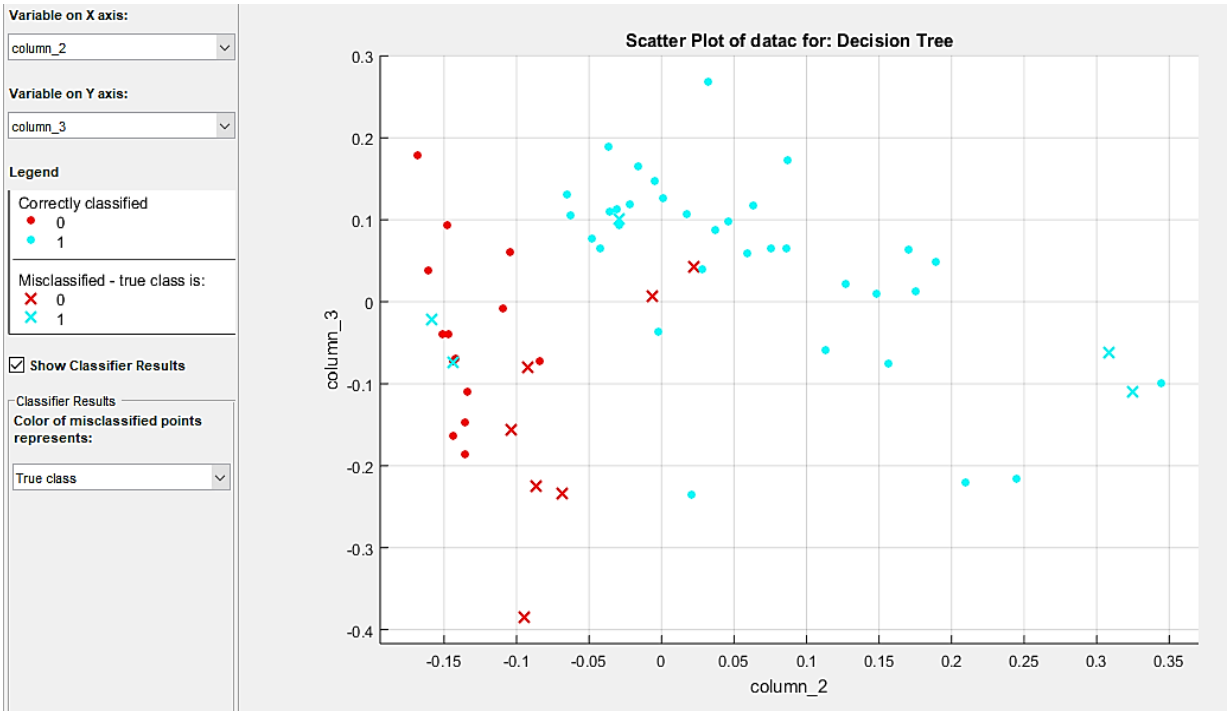



**Figure 4. 39 A Scatter Plot of The Attributes Decision Tree.**

### 4.5.9. The GA-O with PCA with Decision Tree Approach

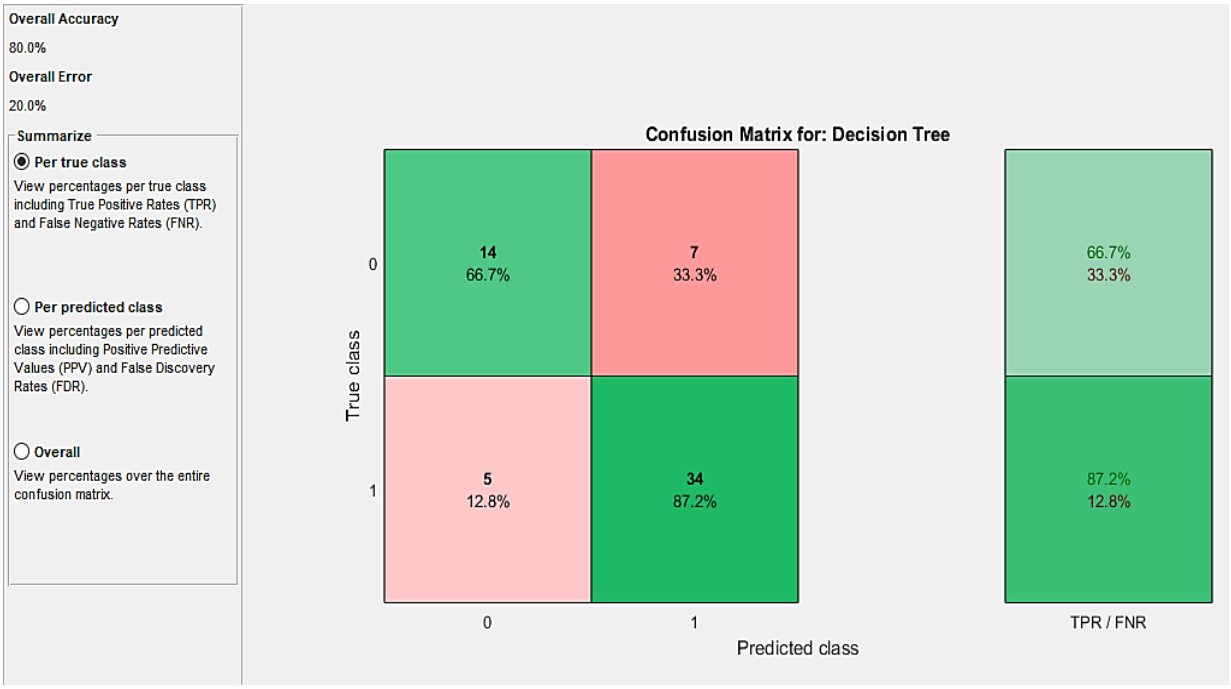



**Figure 4. 40 Confusion matrix for GA-O + PCA + Decision Tree**

**TP= 34; TN= 14; FP= 7; FN= 5**

### 4.5.10. The GA-O with ICA with Decision Tree

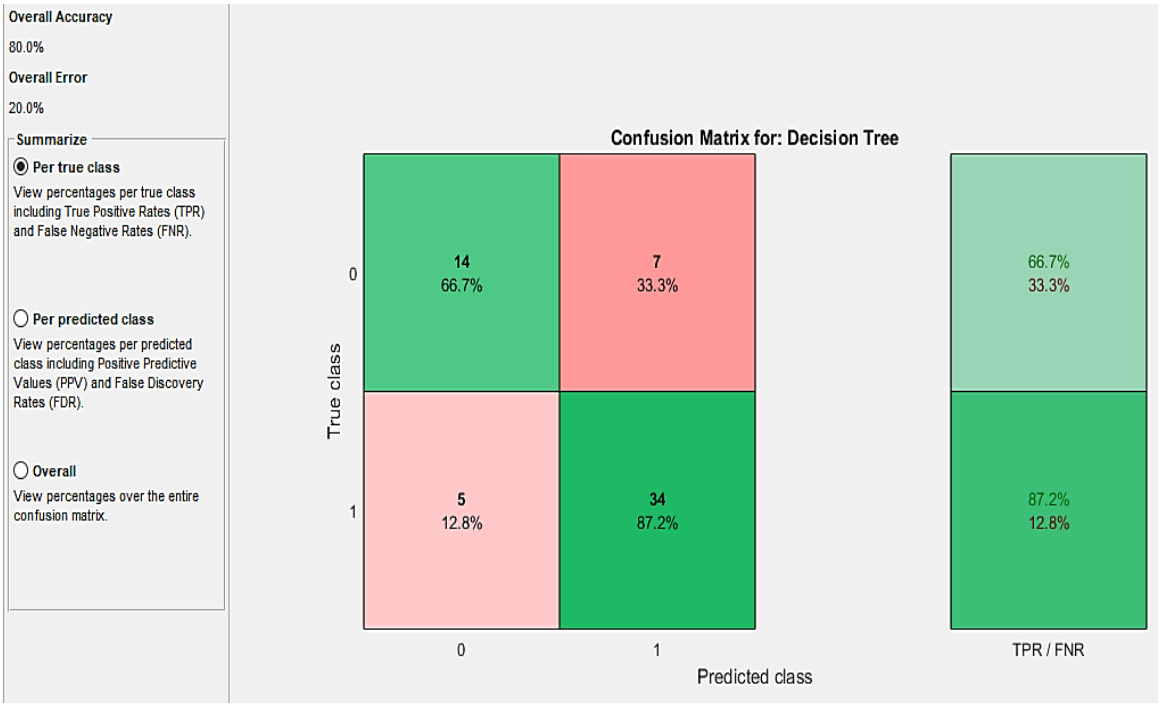



**Figure 4. 41 Confusion matrix for GA-O + ICA + Decision Tree.**

**TP= 34; TN= 14; FP= 7; FN= 5**

**Table 4. 13 Performance Metrics Table for the GA-O + PCA + Decision Tree Classification**

| Performance Metrics (%) | GA-O+PCA+ DECISION TREE | GA-O+ICA+ DECISION TREE |
|---|---|---|
| **Accuracy** | **80** | **80** |
| **Sensitivity** | **87.2** | **87** |
| **Specificity** | **66.7** | **67.0** |
| **Precision** | **83.0** | **82.9** |
| **Recall** | **87.0** | **87.2** |
| **F-score** | **84.9** | **85.0** |

## 4.6. Validation of Result

Experiments have been performed using a hybrid dimensionality reduction using an optimized genetic algorithm with feature extraction (PCA and ICA) was carried out for the classification of RNA-Seq *Anopheles gambiae* dataset. The classifier employs SVM, K-NN, Ensemble and Decision Tree algorithms for implementation, the output of the accuracy for the performance metrics of the hybrid approach is tabulated in table 4.14 with GA-O + ICA + Ensemble outperforming other approaches with 93% accuracy.

**Table 4. 14 Accuracy Metrics for the Hybridized Technique of the Study**

| **Hybridized Approach (HYDREC)** | **Accuracy (%)** |
|---|---|
| **GA-O+PCA+SVM** | **85** |
| **GA-O +ICA+SVM** | **91.7** |
| **GA-O +PCA+K-NN** | **88.3** |
| **GA-O +ICA+K-NN** | **90** |
| **GA-O + PCA + Ensemble (Boosted)** | **83** |
| **GA-O + PCA + Ensemble (Bagged)** | **91.7** |
| **GA-O + ICA + Ensemble (Boosted)** | **93** |
| **GA-O + PCA + Decision Tree** | **80** |
| **GA-O + ICA + Decision Tree** | **80** |

A significant application of the data obtained from RNA-Seq dimensions is the classification of resistance, tolerant and susceptible genes which are either regulated up or down when samples are exposed to pyrethroid class of insecticide to determine insecticide resistance control. To achieve such classifications, a variety of algorithms have been proposed. These algorithms typically require optimization of parameters to get detailed results. It is very important to find an optimal collection of markers among controlled genes which can be used clinically. In this study the anopheles' dataset was used in the experiment, 474 features were relevant using the Genetic Algorithm optimization procedure. PCA and ICA were used to fetch for latent components in the reduced data, 10 and 25 latent components were observed. With GA-O, PCA and ICA algorithms, several experiments were observed, even though there was a necessity for efficient approaches to be implemented, yet optimization is required to obtain accurate results by finding the optimal set of markers among genes that can clinically be utilized for building assays for diagnosis and prognosis of malaria infections. This optimization-based method has been proved as efficient for classification and can allow clinicians to diagnose and follow the progression of malaria infections in human.

The hybrid dimensionality reduction model developed in this study outperformed existing models in terms of accuracy with GA-O + ICA + Ensemble achieving 93% as shown in Table 4.15. Several publications have emanated from this work as shown in Appendix A.

**Table 4. 15 Comparative Table Showing Performance Measures of Other Techniques**

| **Authors** | **Techniques** | **Accuracy (%)** |
|---|---|---|
| Susmi, 2016 | PCA-GA & CCA-GA+NN | 88 |
| Shreem et al., 2016 | Symmetrical Uncertainty+ Harmony Search Algorithm + Naïve Bayes | 87 |
| Lu et al., 2017 | Mutual Information Maximization-GA+SVM | 83 |
| Aziz et al., 2017b | ICA+ABC + Naïve Bayes | 92 |
| Dashtban & Balafar, 2017 | Laplacian-GA and Fisher-GA+KNN | 91 |
| Salem et al., 2017 | Information Gain-GA + Genetic Programming | 85 |
| Mashhour et al., 2018 | Firefly + Chi-square + KNN | 80 |
| Dashtban et al., 2018 | Fisher + Bat Algorithm + SVM | 85 |
| **Proposed Hybrid Model** | **GA-O + ICA + Ensemble (Boosted)** | **93** |

# CHAPTER FIVE

## 5.0 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

### 5.1. Summary

In this study, a hybrid dimensionality reduction model was developed by combining feature selection and feature extraction technique, the reduced data are then classified. The system combines an optimized Genetic Algorithm with PCA and ICA separately. The reduced data are then classified using K-NN, DT, SVM and Ensemble algorithms to improve the accuracy of the system and to reduce dimensionality. Genetic algorithm used in the system handles the correlation of the data more efficiently. PCA and ICA were adopted to extract outliers which simplify the data for classification. K-NN, DT, SVM and Ensemble classification were used to classify the Anopheles data and results were obtained. The developed method improves the performance of the gene expression RNA-Seq data. This system performs dimensionality reduction in a simpler way by reducing the complexity of the system when compared with a conventional model. It achieved good accuracy, and the Receiver Operating Characteristics Curve is decreased. Thus, the dimensionality reduction can be made more efficient using the newly developed system. This experiment shows that not all the top components of PCA and ICA are useful for classification. Also, the tail components contain discriminative information, so it is of great necessity to combine feature selection with feature extraction to analyze high dimensional problems.

## 5.2. Conclusions

This research work contributes to knowledge by employing hybridized genetic algorithm optimizer as a feature selection and feature extraction algorithms as pre-processing stages in mining RNA-Seq dataset. It also developed a prediction model using RNA-Seq Anopheles dataset, which can provide clinicians and researchers with a prediction of insecticidal classification and designs.

Further, a novel method was introduced to resolve the inherent problems in high dimensionality of gene expression data, the hybridized dimensionality reduction method was used to guarantee positive definiteness of the relevant data. An optimized genetic algorithm with PCA and ICA algorithms was developed, on a malaria vector benchmark dataset. The results are unique, using SVM, KNN, Ensemble and Decision tree classifiers. This study has proven that the conventional techniques such as the classical GA, to reduce the dimensionality in gene expression datasets do not work well enough and they degenerate. This point has been overlooked in the literature. There is a need for a new and novel method for fetching relevant information that will help in decision making from the original data.

Although this study has analyzed and demonstrated its results on publicly available benchmarked Anopheles RNA-Seq gene expression dataset, the novel methodology is useful for the analysis of high quality of genomic data obtained from high throughput experiments and state-of-the-art technologies due to its openings for quality new genomic data. There is a need for new and novel methods such as the ones presented in this study to

analyze and interpret the genomic data better with high dimensions. For example, the identification of new insecticide resistance and susceptible genes may provide new therapeutic targets and improve the predictive abilities of genetic testing. This will help clinical sequencing of patients suffering from disease and may eventually guide diagnosis and treatment decisions in personalized medicine. The proposed method can be used to solve new problems and challenges present in the analysis of transcriptomics data in bioinformatics and other biomedical applications.

The use and introduction of the optimization approach for dimension reduction in GA model is confined to dimension reduction. It has many other applications in predictive computational modeling of physical and biological diverse materials. In the literature, several classifiers have been used, the framework, unique holdout and cross validation of 10 with one predictor have been used to carry out the analysis.

## 5.3. Recommendations

The developed system is therefore recommended to the entire body of knowledge and to researchers for carrying out feature selection, feature extraction and classification in bioinformatics, for classifying other diseases and in data mining in general.

The proposed future work is to build a framework that efficiently identifies unlabeled data with less time, more work can be done to train and validate the system with various datasets of diseases and organisms, such as cancer data, HIV data and even Corona virus data. Also, extension of the results of this work to cover Linear Discriminant Analysis (LDA),

classification optimizer, probabilistic independent component analysis (PICA), Ant Colony Optimizer (ACO) and Cluster analysis problems as well as choosing the best subset of the genes can be used to compare their performances with other strategies of algorithms.

The classification accuracy of GAO + ICA + Ensemble has shown improvement when compared to other techniques. Hence, combining more than one method yields insightful classification accuracy and aids in identification of relevant genes.

## 5.4. Contribution to Knowledge

The main input of this study to the body of knowledge will be the development of a hybrid dimensionality reduction techniques, by evolving an optimized genetic algorithm, combined with feature extraction algorithms (PCA and ICA) respectively, for the classification of malaria vector RNA-Seq gene expression data. A model will be developed, for mosquito Anopheles gambiae in RNA-Seq gene expression data for the prediction of malaria infection control and transmission.

This study provides clinicians with classifications approaches which reduce computational load and detect subtypes of genes and proteins as target. This work provides an improved technique for better understanding and interpretation of redundancy elimination in RNA-Seq data; a new approach is created for finding a better representation of RNA-Seq data.

# REFERENCES

Agarwal, A., Koppstein, D., Rozowsky, J., Sboner, A., Habegger, L., Hillier, L. W., Sasidharan, R., Reinke, V., Waterston, R. H., & Gerstein, M. (2010). Comparison and calibration of transcriptome data from RNA-Seq and tiling arrays. *BMC Genomics*, *11*(1), 383. https://doi.org/10.1186/1471-2164-11-383

Alanni, R., Hou, J., Azzawi, H., & Xiang, Y. (2019). Deep gene selection method to select genes from microarray datasets for cancer classification. *BMC Bioinformatics*, *20*(1), 608. https://doi.org/10.1186/s12859-019-3161-2

Alelyani, S., Tang, J., & Liu, H. (2018). Feature Selection for Clustering: A Review. In *Data Clustering* (pp. 29–60). Chapman and Hall/CRC. https://doi.org/10.1201/9781315373515-2

Alfaro, E., Gámez, M., & García, N. (2018). Ensemble classifiers methods. Ensemble Classification Methods with Applications in R. *Preprint*, 31–59. https://doi.org/10.1002/9781119421566.ch3

Almasri, A., Celebi, E., & Alkhawaldeh, R. S. (2019). EMT: Ensemble Meta-Based Tree Model for Predicting Student Performance. *Scientific Programming*, *2019*, 1–13. https://doi.org/10.1155/2019/3610248

Almugren, N., & Alshamlan, H. (2019). A survey on hybrid feature selection methods in Microarray gene expression data for cancer classification. *IEEE Access*, *7*, 78533–

78548. https://doi.org/10.1109/access.2019.2922987.

Alquicira-Hernandez, J., Sathe, A., Ji, H. P., Nguyen, Q., & Powell, J. E. (2019). ScPred: Accurate supervised method for cell-type classification from single-cell RNA-SEQ data. *Genome Biology*, *20*(1). https://doi.org/10.1186/s13059-019-1862-5.

Alzubi, R., Ramzan, N., Alzoubi, H., & Amira, A. (2018). A Hybrid Feature Selection Method for Complex Diseases SNPs. *IEEE Access*, *6*, 1292–1301. https://doi.org/10.1109/ACCESS.2017.2778268

An, Y., Xu, M., & Shen, C. (2019). Classification Method of Teaching Resources Based on Improved KNN Algorithm. *International Journal of Emerging Technology in Learning*, *14*(4).

Angermueller, C., Pärnamaa, T., Parts, L., & Stegle, O. (2016). Deep learning for computational biology. *Molecular Systems Biology*, *12*(7), 878. https://doi.org/10.15252/msb.20156651

Ariga, K. (2014). Barry Carter and Grant Norton: Ceramic Materials, 2nd Edition. *Journal of Inorganic and Organometallic Polymers and Materials*, *24*(6), 1110–1111. https://doi.org/10.1007/s10904-014-0070-8

Arowolo, M. O., Abdulsalam, S. O., Saheed, Y. K., & Salawu, M. D. (2016). A Feature Selection Based on One-Way-ANOVA for Microarray Data Classification. *Al-Hikmah Journal of Pure and Applied Sciences.*, *3*, 30-35.

Arul, V. K., & Elavarasan, U. N. (2014). A Survey on Dimensionality Reduction Technique. *International Journal of Emerging Trends and Technology in Computer Science (IJETTCS)*, *3*(6), 36–42.

Asir, A. G. S., Leavline., Priyanka, R., & Priya, P. P. (2016). Genetic algorithm approach for gene expression classification. *International Journal of Intelligent Systems and Applications.*, *1*(1), 67-73.

Aydadenta, H., & Adiwijaya. (2018). On the classification techniques in data mining for microarray data classification. *Journal of Physics: Conference Series*, *971*, 012004. https://doi.org/10.1088/1742-6596/971/1/012004

Aziz, R., Verma, C. K., & Srivastava, N. (2017a). Dimension reduction methods for microarray data: a review. *AIMS Bioengineering*, *4*(2), 179–197. https://doi.org/10.3934/bioeng.2017.2.179

Aziz, R., Verma, C. K., & Srivastava, N. (2017b). A novel approach for dimension reduction of microarray. *Computational Biology and Chemistry*, *71*, 161–169. https://doi.org/10.1016/j.compbiolchem.2017.10.009

Baharudin, B., Lee, L. H., & Khan, K. (2010). A Review of Machine Learning Algorithms for Text-Documents Classification. *Journal of Advances in Information Technology*, *1*(1). https://doi.org/10.4304/jait.1.1.4-20

Balamurugan, M., Nancy, A., & Vijaykumar, S. (2017). Alzheimer's disease diagnosis by using dimensionality reduction based on Knn classifier. *Biomedical and*

*Pharmacology Journal*, *10*(4), 1823–1830. https://doi.org/10.13005/bpj/1299.

Barshan, E., Ghodsi, A., Azimifar, Z., & Zolghadri Jahromi, M. (2011). Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, *44*(7), 1357–1371. https://doi.org/10.1016/j.patcog.2010.12.015.

Bartenhagen, C., Klein, H.-U., Ruckert, C., Jiang, X., & Dugas, M. (2010). Comparative study of unsupervised dimension reduction techniques for the visualization of microarray gene expression data. *BMC Bioinformatics*, *11*(1), 567. https://doi.org/10.1186/1471-2105-11-567

Becht, E., McInnes, L., Healy, J., Dutertre, C., Kwok, I. W., Ng, L. G., Ginhoux, F., & Newell, E. W. (2018). Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, *37*(1), 38–44. https://doi.org/10.1038/nbt.4314

Bell, J. (2014). *Machine Learning*. John Wiley & Sons, Inc. https://doi.org/10.1002/9781119183464

Bhattacharyya, D. K., & Kalita, J. K. (2013). *Network Anomaly Detection*. Chapman and Hall/CRC. https://doi.org/10.1201/b15088

Bhavsar, H., & Panchal, M. H. (2012). A Review on Support Vector Machine for Data Classification. *International Journal of Advanced Research in Computer Engineering and Technology*, *1*(2), 185–189.

Bonizzoni, M., Ochomo, E., Dunn, W. A., Britton, M., Afrane, Y., Zhou, G., Hartsel, J., Lee, M.-C., Xu, J., Githeko, A., Fass, J., & Yan, G. (2015). RNA-seq analyses of changes in the Anopheles gambiae transcriptome associated with resistance to pyrethroids in Kenya: identification of candidate-resistance genes and candidate-resistance SNPs. *Parasites & Vectors*, *8*(1), 474. https://doi.org/10.1186/s13071-015-1083-z

Bose, J. S. C., Changalesetty, S. B., Badawy, A. S., Ghribi, W., Baili, J., & Bangali, H. (2016). A Hybrid GA/K-NN/SVM Algorithm for Classification of Data". Bio. *House Journal of Computer Science.*, *2*(2), 5–11.

Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., & Caicedo, O. M. (2018). A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities. *Journal of Internet Services and Applications*, *9*(1), 1–11. https://doi.org/10.1186/s13174-018-0087-2

Chaeikar, S., Manaf, A. A., Alarood, A. A., & Zamani, M. (2020). PFW: Polygonal fuzzy weighted—An SVM kernel for the classification of overlapping data groups. *Electronics*, *9*(4), 615. https://doi.org/10.3390/electronics9040615

Chattopadhyay, N., Chattopadhyay, A., Gupta, S. S., & Kasper, M. (2019). Curse of Dimensionality in Adversarial Examples. *2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary*, 1–8. https://doi.org/10.1109/IJCNN.2019.8851795

CHATURVEDI, R., PATHIK, B., & KUMAR, S. (2018). Intrusion Detection Using Data Mining Along Fuzzy Logic & Genetic Algorithms. *JOURNAL OF COMPUTER AND INFORMATION TECHNOLOGY*, *09*(01), 9–13. https://doi.org/10.22147/jucit/090102

Chen, G., Ning, B., & Shi, T. (2019). Single-cell RNA-SEQ technologies and related computational data analysis. *Frontiers in Genetics, 10*. https://doi.org/10.3389/fgene.2019.00317.

Chen, L.-P. (2019). Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar: Foundations of machine learning, second edition. *Statistical Papers*, *60*(5), 1793–1795. https://doi.org/10.1007/s00362-019-01124-9

Chen, T.-C., Hsieh, Y.-C., You, P.-S., & Lee, Y.-C. (2010). Feature selection and classification by using grid computing based evolutionary approach for the microarray data. *2010 3rd International Conference on Computer Science and Information Technology*, 85–89. https://doi.org/10.1109/ICCSIT.2010.5564986

Chen, Y., Li, Y., Narayan, R., Subramanian, A., & Xie, X. (2016). Gene expression inference with deep learning. *Bioinformatics*, *32*(12), 1832–1839. https://doi.org/10.1093/bioinformatics/btw074

Cheng, X., Cai, H., Zhang, Y., Xu, B., & Su, W. (2015). Optimal combination of feature selection and classification via local hyperplane based learning strategy. *BMC Bioinformatics*, *16*(1), 219. https://doi.org/10.1186/s12859-015-0629-6

Chowdhary, M., Rani, A., Parkash, J., Shahnaz, M., & Dev, D. (2016). Bioinformatics: an overview for cancer research. *Journal of Drug Delivery and Therapeutics*, *6*(4). https://doi.org/10.22270/jddt.v6i4.1290

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M. W., Gaffney, D. J., Elo, L. L., Zhang, X., & Mortazavi, A. (2016). Erratum to: A survey of best practices for RNA-seq data analysis. *Genome Biology, 17*(1), 181. https://doi.org/10.1186/s13059-016-1047-4

Costa-Silva, J., Domingues, D., & Lopes, F. M. (2017). RNA-Seq differential expression analysis: An extended review and a software tool. *PLOS ONE*, *12*(12), e0190152. https://doi.org/10.1371/journal.pone.0190152

Cui, S., Wu, Q., West, J., & Bai, J. (2019). Machine learning-based microarray analyses indicate low-expression genes might collectively influence PAH disease. *PLOS Computational Biology, 15*(8), e1007264. https://doi.org/10.1371/journal.pcbi.1007264.

Dagliyan, O., Uney-Yuksektepe, F., Kavakli, I. H., & Turkay, M. (2011). Optimization Based Tumor Classification from Microarray Gene Expression Data. *PLoS ONE*, *6*(2), e14579. https://doi.org/10.1371/journal.pone.0014579

Dashtban, M., & Balafar, M. (2017). Gene selection for microarray cancer classification using a new evolutionary method employing artificial intelligence concepts. *Genomics, 109*(2), 91–107. https://doi.org/10.1016/j.ygeno.2017.01.004

Dashtban, M., Balafar, M., & Suravajhala, P. (2018). Gene selection for tumor classification using a novel bio-inspired multi-objective approach. *Genomics*, *110*(1), 10–17. https://doi.org/10.1016/j.ygeno.2017.07.010

Deepika, K., Bodapati, J. D., & Srihitha, R. K. (2019). An efficient automatic brain tumor classification using LBP features and SVM-based classifier. *Proceedings of International Conference on Computational Intelligence and Data Engineering*, 163–170. https://doi.org/10.1007/978-981-13-6459-4_17.

Ding, J., Condon, A., & Shah, S. P. (2018). Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature Communications*, *9*(1), 2002. https://doi.org/10.1038/s41467-018-04368-5

Dudoit, S., Fridlyand, J., & Speed, T. P. (2002). Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. *Journal of the American Statistical Association*, *97*(457), 77–87. https://doi.org/10.1198/016214502753479248

Duncan, J. S., Insana, M. F., & Ayache, N. (2020). Biomedical Imaging and Analysis in the Age of Big Data and Deep Learning [Scanning the Issue]. *Proceedings of the IEEE*, *108*(1), 3–10. https://doi.org/10.1109/JPROC.2019.2956422

Duval, B., & Hao, J.-K. (2010). Advances in metaheuristics for gene selection and classification of microarray data. *Briefings in Bioinformatics*, *11*(1), 127–141. https://doi.org/10.1093/bib/bbp035

Feng, C., Liu, S., Zhang, H., Guan, R., Li, D., Zhou, F., Liang, Y., & Feng, X. (2020). Dimension reduction and clustering models for single-cell RNA sequencing data: A comparative study. *International Journal of Molecular Sciences*, *21*(6), 2181. https://doi.org/10.3390/ijms21062181.

Frank, M., Drikakis, D., & Charissis, V. (2020). Machine-Learning Methods for Computational Science and Engineering. *Computation*, *8*(1), 15. https://doi.org/10.3390/computation8010015

Gachelin, G., Garner, P., Ferroni, E., Verhave, J. P., & Opinel, A. (2018). Evidence and strategies for malaria prevention and control: a historical analysis. *Malaria Journal*, *17*(1), 96. https://doi.org/10.1186/s12936-018-2244-2

Ghareb, A. S., Bakar, A. A., & Hamdan, A. R. (2016). Hybrid feature selection based on enhanced genetic algorithm for text categorization. *Expert Systems with Applications*, *49*, 31–47. https://doi.org/10.1016/j.eswa.2015.12.004

Greene, C. S., Tan, J., Ung, M., Moore, J. H., & Cheng, C. (2014). Big Data Bioinformatics. *Journal of Cellular Physiology*, *229*(12), 1896–1900. https://doi.org/10.1002/jcp.24662

Griffith, M., Walker, J. R., Spies, N. C., Ainscough, B. J., & Griffith, O. L. (2015). Informatics for RNA Sequencing: A Web Resource for Analysis on the Cloud. *PLOS Computational Biology*, *11*(8), e1004393. https://doi.org/10.1371/journal.pcbi.1004393

Guia, J. M. De, Devaraj, M., & Vea, L. A. (2018). Cancer Classification of Gene Expression Data using Machine Learning Models. *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, 1–6. https://doi.org/10.1109/HNICEM.2018.8666435

Guzman, E., El-Haliby, M., & Bruegge, B. (2015). Ensemble methods for app review classification: An approach for software evolution (N). *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE).* https://doi.org/10.1109/ase.2015.88.

Han, Y., Gao, S., Muegge, K., Zhang, W., & Zhou, B. (2015). Advanced Applications of RNA Sequencing and Challenges. *Bioinformatics and Biology Insights*, *9s1*, BBI.S28991. https://doi.org/10.4137/BBI.S28991

Hasan, A., & Md. Akhtaruzzaman Adnan. (2012). High dimensional microarray data classification using correlation based feature selection. *2012 International Conference on Biomedical Engineering (ICoBE)*, 319–321. https://doi.org/10.1109/ICoBE.2012.6179029

Hassan, A. K., Moinuddin, M., Al-Saggaf, U. M., & Shaikh, M. S. (2017). On the kernel optimization of radial basis function using Nelder mead simplex. *Arabian Journal for Science and Engineering*, *43*(6), 2805–2816. https://doi.org/10.1007/s13369-017-2888-1

Hazrati Fard, S. M., Hamzeh, A., & Hashemi, S. (2013). Using reinforcement learning to find an optimal set of features. *Computers & Mathematics with Applications*, *66*(10), 1892–1904. https://doi.org/10.1016/j.camwa.2013.06.031

Hien, A. S., Sangaré, I., Coulibaly, S., Namountougou, M., Paré-Toé, L., Ouédraogo, A. G., Diabaté, A., Foy, B. D., & Dabiré, R. K. (2017). Parasitological Indices of Malaria Transmission in Children under Fifteen Years in Two Ecoepidemiological Zones in Southwestern Burkina Faso. *Journal of Tropical Medicine*, *2017*, 1–7. https://doi.org/10.1155/2017/1507829

Hira, Z. M., & Gillies, D. F. (2015). A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data. *Advances in Bioinformatics*, *2015*, 1–13. https://doi.org/10.1155/2015/198363

Howick, V. M., Russell, A., Andrews, T., Heaton, H., Reid, A. J., Natarajan, K. N., Butungi, H., Metcalf, T., Verzier, L. H., Rayner, J., Berriman, M., Herren, J., Billker, O., Hemberg, M., Talman, A., & Lawniczak, M. (2019). The malaria cell atlas: A comprehensive reference of single parasite transcriptomes across the complete plasmodium life cycle: *File S1*. https://doi.org/10.1101/527556

Huang, Y., & Lowe, H. J. (2007). A Novel Hybrid Approach to Automated Negation Detection in Clinical Radiology Reports. *Journal of the American Medical Informatics Association*, *14*(3), 304–311. https://doi.org/10.1197/jamia.M2284

Jagga, Z., & Gupta, D. (2014). Classification models for clear cell renal carcinoma stage

progression, based on tumor RNAseq expression trained supervised machine learning algorithms. *BMC Proceedings*, *8*(S6), S2. https://doi.org/10.1186/1753-6561-8-S6-S2

Jain, D., & Singh, V. (2018). Feature selection and classification systems for chronic disease prediction: A review. *Egyptian Informatics Journal*, *19*(3), 179–189. https://doi.org/10.1016/j.eij.2018.03.002

Jiang, X., Peery, A., Hall, A. B., Sharma, A., Chen, X.-G., Waterhouse, R. M., Komissarov, A., Riehle, M. M., Shouche, Y., Sharakhova, M. V, Lawson, D., Pakpour, N., Arensburger, P., Davidson, V. L. M., Eiglmeier, K., Emrich, S., George, P., Kennedy, R. C., Mane, S. P., … Tu, Z. (2014). Genome analysis of a major urban malaria vector mosquito, Anopheles stephensi. *Genome Biology*, *15*(9), 459. https://doi.org/10.1186/s13059-014-0459-2

Jindal, P., & Kumar, D. (2017). A Review on Dimensionality Reduction Techniques. *International Journal of Computer Applications*, *173*(2), 42–46. https://doi.org/10.5120/ijca2017915260

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065). https://doi.org/10.1098/rsta.2015.0202.

Jovic, A., Brkic, K., & Bogunovic, N. (2015). A review of feature selection methods with

applications. *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1200–1205. https://doi.org/10.1109/MIPRO.2015.7160458

Karthik, S., & Sudha, M. (2018). A Survey on Machine Learning Approaches in Gene Expression Classification in Modelling Computational Diagnostic System for Complex Diseases. *International Journal of Engineering and Advanced Technology.*, *8*(2), 182–191.

Kashyap, H., Ahmed, H. A., Hoque, N., Roy, S., & Bhattacharyya, D. K. (2016). Big data analytics in bioinformatics: architectures, techniques, tools and issues. *Network Modeling Analysis in Health Informatics and Bioinformatics*, *5*(1), 28. https://doi.org/10.1007/s13721-016-0135-4

Ke, Q., Zhang, J., Srivastava, H. M., Wei, W., & Chen, G.-S. (2015). Independent Component Analysis Based on Information Bottleneck. *Abstract and Applied Analysis*, *2015*, 1–8. https://doi.org/10.1155/2015/386201

Keerthi Vasan, K., & Surendiran, B. (2016). Dimensionality reduction using Principal Component Analysis for network intrusion detection. *Perspectives in Science*, *8*, 510–512. https://doi.org/10.1016/j.pisc.2016.05.010

Kleftogiannis, D., Korfiati, A., Theofilatos, K., Likothanassis, S., Tsakalidis, A., & Mavroudi, S. (2013). Where we stand, where we are moving: Surveying computational techniques for identifying miRNA genes and uncovering their

regulatory role. *Journal of Biomedical Informatics*, *46*(3), 563–573. https://doi.org/10.1016/j.jbi.2013.02.002

Kodratoff, Y., & Michalski, R. S. (2014). Machine Learning: An Artificial Intelligence Approach. *Elsevier Science.*, *3*, 138.

Kong, W., Vanderburg, C. R., Gunshin, H., Rogers, J. T., & Huang, X. (2008). A review of independent component analysis application to microarray gene expression data. *BioTechniques*, *45*(5), 501–520. https://doi.org/10.2144/000112950

Kowsari, Jafari, M., Heidarysafa, Mendu, Barnes, & Brown. (2019). Text classification algorithms: A survey. *Information*, *10*(4), 150. https://doi.org/10.3390/info10040150.

Kratz, A., & Carninci, P. (2014). The devil in the details of RNA-seq. *Nature Biotechnology*, *32*(9), 882–884. https://doi.org/10.1038/nbt.3015

Kuang, T., Hu, Z., & Xu, M. (2020). A Genetic Optimization Algorithm Based on Adaptive Dimensionality Reduction. *Mathematical Problems in Engineering*, 1–7. https://doi.org/10.1155/2020/8598543.

Kuhn, M., & Johnson, K. (2013). Measuring Predictor Importance. In *Applied Predictive Modeling* (pp. 463–485). Springer New York. https://doi.org/10.1007/978-1-4614-6849-3_18

Kumar, M., Rath, N. K., Swain, A., & Rath, S. K. (2015). Feature Selection and

Classification of Microarray Data using MapReduce based ANOVA and K-Nearest Neighbor. *Procedia Computer Science*, *54*, 301–310. https://doi.org/10.1016/j.procs.2015.06.035

kumar, N. M. S., Eswari, T., Sampath, P., & Lavanya, S. (2015). Predictive Methodology for Diabetic Data Analysis in Big Data. *Procedia Computer Science*, *50*, 203–208. https://doi.org/10.1016/j.procs.2015.04.069

Kumar, V. (2014). Feature Selection: A literature Review. *The Smart Computing Review*, *4*(3). https://doi.org/10.6029/smartcr.2014.03.007

Lachmann, A., Torre, D., Keenan, A. B., Jagodnik, K. M., Lee, H. J., Wang, L., Silverstein, M. C., & Ma'ayan, A. (2018). Massive mining of publicly available RNA-seq data from human and mouse. *Nature Communications*, *9*(1), 1366. https://doi.org/10.1038/s41467-018-03751-6

Lavanya, C., Nandihini, M., Niranjana, R., & Gunavathi, C. (2014). Classification of Microarray Data Based on Feature Selection Method. *International Conference on Engineering Technology and Science. International Journal of Innovative Research in Science, Engineering and Technology*, *3*(1), 1261–1264.

Lee, H. J., Georgiadou, A., Otto, T. D., Levin, M., Coin, L. J., Conway, D. J., & Cunnington, A. J. (2018). Transcriptomic Studies of Malaria: a Paradigm for Investigation of Systemic Host-Pathogen Interactions. *Microbiology and Molecular Biology Reviews*, *82*(2). https://doi.org/10.1128/MMBR.00071-17

Lenz, M., Müller, F., Zenke, M., & Schuppert, A. (2016). Principal components analysis and the reported low intrinsic dimensionality of gene expression microarray data. *Scientific Reports, 6*(1). https://doi.org/10.1038/srep25696.

Li, C., Zhang, S., Zhang, H., Pang, L., Lam, K., Hui, C., & Zhang, S. (2012). Using the k-nearest neighbor algorithm for the classification of lymph node metastasis in gastric cancer. *Computational and Mathematical Methods in Medicine*, 1–11. https://doi.org/10.1155/2012/876545

Li, G.-Z., Bu, H.-L., Yang, M., Zeng, X.-Q., & Yang, J. Y. (2008). Selecting subsets of newly extracted features from PCA and PLS in microarray data analysis. *BMC Genomics, 9*(Suppl 2), S24. https://doi.org/10.1186/1471-2164-9-S2-S24

Liang, H., Sun, X., Sun, Y., & Gao, Y. (n.d.). Correction to: Text feature extraction based on deep learning: a review. *EURASIP Journal on Wireless Communications and Networking*, *1*. https://doi.org/10.1186/s13638-018-1056-y

Lin, C., Jain, S., Kim, H., & Bar-Joseph, Z. (2017). Using neural networks for reducing the dimensions of single-cell RNA-SEQ data. *Nucleic Acids Research, 45*(17), 156–166. https://doi.org/10.1093/nar/gkx681.

Liu, C., Zhou, Q., Li, Y., Garner, L. V., Watkins, S. P., Carter, L. J., Smoot, J., Gregg, A. C., Daniels, A. D., Jervey, S., & Albaiu, D. (2020). Research and Development on Therapeutic Agents and Vaccines for COVID-19 and Related Human Coronavirus Diseases. *ACS Central Science, 6*(3), 315–331.

https://doi.org/10.1021/acscentsci.0c00272

Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y., & Gao, Z. (2017). A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing*, *256*, 56–62. https://doi.org/10.1016/j.neucom.2016.07.080

Lucas, A. (2013). *Package 'amap.* http://cran.r-project.org/web/ packages/amap/vignettes/amap.pdf.

Luecken, M. D., & Theis, F. J. (2019). Current best practices in single- cell RNA- seq analysis: a tutorial. *Molecular Systems Biology*, *15*(6). https://doi.org/10.15252/msb.20188746

Luo, K., Wang, G., Li, Q., & Tao, J. (2019). An Improved SVM-RFE Based on $F$ - Statistic and mPDC for Gene Selection in Cancer Classification. *IEEE Access*, *7*, 147617–147628. https://doi.org/10.1109/ACCESS.2019.2946653

Maldonado, S., & Weber, R. (2009). A wrapper method for feature selection using Support Vector Machines. *Information Sciences*, *179*(13), 2208–2217. https://doi.org/10.1016/j.ins.2009.02.014

Mashhour, E. M., El Houby, E. M. F., Wassif, K. T., & Salah, A. I. (2018). Feature Selection Approach based on Firefly Algorithm and Chi-square. *International Journal of Electrical and Computer Engineering (IJECE)*, *8*(4), 2338. https://doi.org/10.11591/ijece.v8i4.pp2338-2350

Mohamed, E., M., E., Tawfik, K., & Ibrahim, A. (2016). Survey on different Methods for Classifying Gene Expression using Microarray Approach. *International Journal of Computer Applications*, *150*(1), 12–21. https://doi.org/10.5120/ijca2016911441

Mohan, C., & Nagarajan, S. (2019). An improved tree model based on ensemble feature selection for classification. *Turkish Journal of Electrical Engineering & Computer Sciences*, 1290–1307. https://doi.org/10.3906/elk-1808-85.

Momeni, Z., & Saniee Abadeh, M. (2019). MapReduce-Based Parallel Genetic Algorithm for CpG-Site Selection in Age Prediction. *Genes*, *10*(12), 969. https://doi.org/10.3390/genes10120969

Moreno, M., Pavón, J., & Rosete, A. (2009). *Testing in Agent Oriented Methodologies* (pp. 138–145). https://doi.org/10.1007/978-3-642-02481-8_20

Nagi, S., & Bhattacharyya, D. K. (2013). Classification of microarray cancer data using ensemble approach. *Network Modeling Analysis in Health Informatics and Bioinformatics*, *2*(3), 159–173. https://doi.org/10.1007/s13721-013-0034-x

Nalband, S., Sundar, A., Prince, A. A., & Agarwal, A. (2016). Feature selection and classification methodology for the detection of knee-joint disorders. *Computer Methods and Programs in Biomedicine*, *127*, 94–104. https://doi.org/10.1016/j.cmpb.2016.01.020

Nandhini, & Porkodi. (2019). The Novel Gravitational Mass Weighted PCA Technique for Feature Extraction in Hyperspectral Data Classification. *International Journal of*

*Engineering and Advanced Technology*, *8*(5S3), 250–255. https://doi.org/10.35940/ijeat.E1056.0785S319

Nekrutenko, A., & Taylor, J. (2012). Next-generation sequencing data interpretation: enhancing reproducibility and accessibility. *Nature Reviews Genetics*, *13*(9), 667–672. https://doi.org/10.1038/nrg3305

Nguyen, L. H., & Holmes, S. (2019). Ten quick tips for effective dimensionality reduction. *PLOS Computational Biology*, *15*(6), e1006907. https://doi.org/10.1371/journal.pcbi.1006907

Nisar, S., & Tariq, M. (2016). Intelligent feature selection using hybrid based feature selection method. *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, 168–172. https://doi.org/10.1109/INTECH.2016.7845025

Nisioti, A., Mylonas, A., Yoo, P. D., & Katos, V. (2018). From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods. *IEEE Communications Surveys & Tutorials, 20*(4), 3369–3388. https://doi.org/10.1109/comst.2018.2854724.

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. *Journal of Big Data, 7*(1), 20. https://doi.org/10.1186/s40537-020-00299-5

Oh, D. H., Kim, I. B., Kim, S. H., & Ahn, D. H. (2017). Predicting autism spectrum disorder using blood-based gene expression signatures and machine learning.

Clinical. *Psychopharmacology and Neuroscience*, *15*(1), 47–52. https://doi.org/10.9758/cpn.2017.15.1.47.

Oladipupo, T. (2010). Types of Machine Learning Algorithms. In *New Advances in Machine Learning*. InTech. https://doi.org/10.5772/9385

Onan, A. (2015). Classifier and feature set ensembles for web page classification. *Journal of Information Science, 42*(2), 150–165. https://doi.org/10.1177/0165551515591724

Osareh, A., & Shadgar, B. (2013). An efficient ensemble learning method for gene Microarray classification. *BioMed Research International*, 1–10. https://doi.org/10.1155/2013/478410

Oshlack, A., Robinson, M. D., & Young, M. D. (2010). From RNA-seq reads to differential expression results. *Genome Biology*, *11*(12), 220. https://doi.org/10.1186/gb-2010-11-12-220

Pamukçu, E., Bozdogan, H., & Çalık, S. (2015). A novel hybrid dimension reduction technique for undersized high dimensional gene expression data sets using information complexity criterion for cancer classification. *Computational and Mathematical Methods in Medicine.*, 1–14. https://doi.org/10.1155/2015/370640

Parimala, R., & Nallaswamy, R. (2011). A Study of Spam E-mail classification using Feature Selection package. *Global Journal of Computer Science and Technology, 11*(7), 1–11.

Parva, E., Boostani, R., Ghahramani, Z., & Paydar, S. (2017). The Necessity of Datamining in Clinical Emergency Medicine; A Narrative Review of the Current Literature. *Bulletin of Emergency and Trauma.*, *5*(2), 90–95.

Pavithra, D., & Lakshmanan, B. (2017). Feature selection and classification in gene expression cancer data. *2017 International Conference on Computational Intelligence in Data Science(ICCIDS)*, 1–6. https://doi.org/10.1109/ICCIDS.2017.8272668

Peng, Y., Wu, Z., & Jiang, J. (2010). A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics*, *43*(1), 15–23. https://doi.org/10.1016/j.jbi.2009.07.008

Pierson, E., & Yau, C. (2015). ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, *16*(1), 241. https://doi.org/10.1186/s13059-015-0805-z

Pinto da Costa, J. F., Alonso, H., & Roque, L. (2011). A Weighted Principal Component Analysis and Its Application to Gene Expression Data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *8*(1), 246–252. https://doi.org/10.1109/TCBB.2009.61

Polaka, I., Tom, I., & Borisov, A. (2010). Decision tree classifiers in bioinformatics. *Scientific Journal of Riga Technical University. Computer Sciences.*, *42*(1), 118-123. https://doi.org/10.2478/v10143-010-0052-4.

Poostchi, M., Silamut, K., Maude, R. J., Jaeger, S., & Thoma, G. (2018). Image analysis and machine learning for detecting malaria. *Translational Research*, *194*, 36–55. https://doi.org/10.1016/j.trsl.2017.12.004

Prathusha, P., & Jyothi, S. (2017). Feature Extraction Methods: A Review. *International Journal of Innovative Research in Science, Engineering and Technology.*, *6*(12), 22558–22577.

Qi, R., Ma, A., Ma, Q., & Zou, Q. (2020). Clustering and classification methods for single-cell RNA-sequencing data. *Briefings in Bioinformatics*, *21*(4), 1196–1208. https://doi.org/10.1093/bib/bbz062

Rao, M. S., Van Vleet, T. R., Ciurlionis, R., Buck, W. R., Mittelstadt, S. W., Blomme, E. A. G., & Liguori, M. J. (2019). Comparison of RNA-Seq and Microarray Gene Expression Platforms for the Toxicogenomic Evaluation of Liver From Short-Term Rat Toxicity Studies. *Frontiers in Genetics*, *9*. https://doi.org/10.3389/fgene.2018.00636

rasan, N. E., & Mani, D. K. (2015). A Survey on Feature Extraction Techniques. *International Journal of Innovative Research in Computer and Communication Engineering*, *03*(01), 52–55. https://doi.org/10.15680/ijircce.2015.0301009

Raut, S. A., Sathe, S. R., & Raut, A. (2010). Bioinformatics: Trends in gene expression analysis. *2010 International Conference on Bioinformatics and Biomedical Technology*, 97–100. https://doi.org/10.1109/ICBBT.2010.5479003

Reid, A. J., Talman, A. M., Bennett, H. M., Gomes, A. R., Sanders, M. J., Illingworth, C. J., Billker, O., Berriman, M., & Lawniczak, M. K. (2018). Single-cell RNA-SEQ reveals hidden transcriptional variation in malaria parasites. *ELife, 7*. https://doi.org/10.7554/elife.33105.

Rostom, R., Svensson, V., Teichmann, S. A., & Kar, G. (2017). Computational approaches for interpreting scRNA-SEQ data. *FEBS Letters*, *591*(15), 2213–2225. https://doi.org/10.1002/1873-3468.12684.

Sahu, B., Dehuri, S., & Jagadev, A. (2018). A Study on the Relevance of Feature Selection Methods in Microarray Data. *The Open Bioinformatics Journal*, *11*(1), 117–139. https://doi.org/10.2174/1875036201811010117

Salem, H., Attiya, G., & El-Fishawy, N. (2017). Classification of human cancer diseases by gene expression profiles. *Applied Soft Computing*, *50*, 124–134. https://doi.org/10.1016/j.asoc.2016.11.026

Santos, R. D., Gorgulho, B. M., Castro, M. A., Fisberg, R. M., Marchioni, D. M., & Baltar, V. T. (2019). Principal component analysis and factor analysis: Differences and similarities in nutritional epidemiology application. *Revista Brasileira de Epidemiologia*, *22*. https://doi.org/10.1590/1980-549720190041

Shon, H. S., Yi, Y., Kim, K. O., Cha, E.-J., & Kim, K.-A. (2019). Classification of stomach cancer gene expression data using CNN algorithm of deep learning. *Journal of Biomedical Translational Research*, *20*(1), 15–20.

https://doi.org/10.12729/jbtr.2019.20.1.015

Shreem, S. S., Abdullah, S., & Nazri, M. Z. A. (2016). Hybrid feature selection algorithm using symmetrical uncertainty and a harmony search algorithm. *International Journal of Systems Science*, *47*(6), 1312–1329. https://doi.org/10.1080/00207721.2014.924600

Shukla, A. K., Singh, P., & Vardhan, M. (2019). A New Hybrid Feature Subset Selection Framework Based on Binary Genetic Algorithm and Information Theory. *International Journal of Computational Intelligence and Applications*, *18*(03), 1950020. https://doi.org/10.1142/S1469026819500202

Simmons, S., Peng, J., Bienkowska, J., & Berger, B. (2015). Discovering What Dimensionality Reduction Really Tells Us About RNA-Seq Data. *Journal of Computational Biology*, *22*(8), 715–728. https://doi.org/10.1089/cmb.2015.0085

Singh, R. (2018). A gene expression data classification and selection method using hybrid meta-heuristic technique. *ICST Transactions on Scalable Information Systems*, *0*(0), 159917. https://doi.org/10.4108/eai.13-7-2018.159917.

Sivapriya, T. R., & Kamal, A. R. N. B. (2013). Hybrid feature reduction and selection for enhanced classification of high dimensional medical data. *2013 IEEE International Conference on Computational Intelligence and Computing Research*, 1–4. https://doi.org/10.1109/ICCIC.2013.6724237

Sofie, V. A. (2017). Comparative Review of Dimensionality Reduction Methods for

High-Throughput Single-Cell Transcriptomics. *Master's Dissertation Submitted To Ghent University to Obtain the Degree Of Master Of Science In Biochemistry and Biotechnology. Major Bioinformatics and Systems Biology.*, 1–88.

Song, N., Wang, K., Xu, M., Xie, X., Chen, G., & Wang, Y. (2016). Design and Analysis of Ensemble Classifier for Gene Expression Data of Cancer. *Advancement in Genetic Engineering.*, *5*(1), 1–7.

Songyot, N. (2019). A Hybrid Gene Selection Algorithm Based on Interaction Information for Microarray-based Cancer Classification. *PLoS One.*, *14*(2), 1-10.

Soofi, A., & Awan, A. (2017). Classification Techniques in Machine Learning: Applications and Issues. *Journal of Basic & Applied Sciences, 13*, 459–465. https://doi.org/10.6000/1927-5129.2017.13.76

Soufan, O., Kleftogiannis, D., Kalnis, P., & Bajic, V. B. (2015). DWFS: A wrapper feature selection tool based on a parallel genetic algorithm. *PLOS ONE, 10*(2), e0117988. https://doi.org/10.1371/journal.pone.0117988

Suksut, K., Kaoungku, N., Kerdprasop, K., & Kerdprasop, N. (2019). Improvement the Imbalanced Data Classification with Restarting Genetic Algorithm for Support Vector Machine Algorithm. *International Journal of Future Computer and Communication*, *8*(2), 63–67. https://doi.org/10.18178/ijfcc.2019.8.2.541

Sumathi, A., Santhoshkumar, S., & Sakthivel, N. K. (2012). Development of an Efficient Data Mining Classifier With Microarray Data Set For Gene Selection And

Classification. *Journal of Theoretical and Applied Information Technology*, *35*(2), 209-214.

Sun, L., Wang, J., & Xu, Y. (2020). Breast mass classification method based on convolutional neural networks. *The Journal of Engineering*, *2020*(13), 630–634. https://doi.org/10.1049/joe.2019.1136

Sun, S., Chen, Y., Liu, Y., & Shang, X. (2019). A fast and efficient count-based matrix factorization method for detecting cell types from single-cell RNAseq data. *BMC Systems Biology*, *13*(S2), 28. https://doi.org/10.1186/s12918-019-0699-6

Susmi, S. J. (2016). Hybrid Dimension Reduction Techniques with Genetic Algorithm and Neural Network for Classifying Leukemia Gene Expression Data. *Indian Journal of Science and Technology*, *9*(1), 1–8. https://doi.org/10.17485/ijst/2016/v9iS1/70384

T, S., & Rangarajan, L. (2018). An Approach to reduce the large feature space of Microarray Gene Expression Data by gene clustering for efficient sample classification. *INTERNATIONAL JOURNAL OF COMPUTER APPLICATION*, *3*(8). https://doi.org/10.26808/rs.ca.i8v3.01

Tamim, A., Lieke, M., Davy, C., Dylan, H., Hailaiang, M., Marcel, J. T. R., & Ahmed, M. (2019). A Comparison of Automated Cell Identification Methods for Single-Cell RNA Sequencing Data. *Genome Biology*, *20*(194), 1-14.

Tan, A. C., & Gilbert, D. (2013). Ensemble Machine Learning on Gene Expression Data

for Cancer Classification. *Cancer*, *2*, *No. 3*,(3), 75-83.

Tan, C. S., Ting, W. S., Mohamad, M. S., Chan, W. H., Deris, S., & Ali Shah, Z. (2014). A Review of Feature Extraction Software for Microarray Gene Expression Data. *BioMed Research International*, *2014*, 1–15. https://doi.org/10.1155/2014/213656

Tang, J., Alelyani, S., & Liu, H. (2014). Feature Selection for Classification: A Review. *Data Classification: Algorithm Applications*, *37*.

Tarek, S., Elwahab, R. A., & Shoman, M. (2017). Gene Expression Based Cancer Classification. *Egyptian Informatics Journal.*, *18*(3), 151–159. https://doi.org/10.1016/j.eij.2016.12.001.

Taveira De Souza, J., Carlos De Francisco, A., & Macedo, D. C. De. (2019). Dimensionality Reduction in Gene Expression Data Sets. *IEEE Access*, *7*, 61136–61144. https://doi.org/10.1109/ACCESS.2019.2915519

Tharwat, A., & Gabel, T. (2019). Parameters optimization of support vector machines for imbalanced data using social ski driver algorithm. *Neural Computing and Applications*, *32*(11), 6925–6938. https://doi.org/10.1007/s00521-019-04159-z

Townes, F. W., Hicks, S. S., Aryee, M. J., & Irizarry, R. A. (2019). Feature Selection and Dimensionality Reduction for Single-cell RNA-Seq Based on Multinomial Model. *BMC Genome Biology*, *20*(295), 1-21.

Usman, M., Ahmed, S., Ferzund, J., Mehmood, A., & Rehman, A. (2017). Using PCA

and Factor Analysis for Dimensionality Reduction of Bio-informatics Data. *International Journal of Advanced Computer Science and Applications*, *8*(5). https://doi.org/10.14569/IJACSA.2017.080551

Uysal, A. K., & Gunal, S. (2012). A novel probabilistic feature selection method for text classification. *Knowledge-Based Systems*, *36*, 226–235. https://doi.org/10.1016/j.knosys.2012.06.005

Vanitha, C. D. A., Devaraj, D., & Venkatesulu, M. (2015). Gene Expression Data Classification Using Support Vector Machine and Mutual Information-based Gene Selection. *Procedia Computer Science*, *47*, 13–21. https://doi.org/10.1016/j.procs.2015.03.178

Verma, N. K., Dixit, S., Sevakula, R. K., & Salour, A. (2018). Computational Framework for Machine Fault Diagnosis with Autoencoder Variants. *2018 International Conference on Sensing,Diagnostics, Prognostics, and Control (SDPC)*, 353–358. https://doi.org/10.1109/SDPC.2018.8664980

Wagner, F., Yan, Y., & Yanal, I. (2017). K-Nearest Neighbor Smoothing for High-Throughput Single-cell RNA-Seq Data. *BioRxiv Preprint*. https://doi.org/10.1101/217737.

Wang, D., & Gu, J. (2018). VASC: Dimension reduction and visualization of single-cell RNA-SEQ data by deep variational Autoencoder. *Genomics, Proteomics & Bioinformatics*, *16*(5), 320–331. https://doi.org/10.1016/j.gpb.2018.08.003

Wenric, S., & Shemirani, R. (2018). Using supervised learning methods for gene selection in RNA-SEQ case-control studies. *Frontiers in Genetics*, *9*. https://doi.org/10.3389/fgene.2018.00297

Wenyan, Z., Xuewen, L., & Jingjing, W. (2017). Feature Selection for Cancer Classification Using Microarray Gene Expression Data. *Biostatistics and Biometrics Journals.*, *1*(2), 1-7.

White, N. J. (1996). The Treatment of Malaria. *New England Journal of Medicine*, *335*(11), 800–806. https://doi.org/10.1056/NEJM199609123351107

Witten, D. M. (2011). Classification and clustering of sequencing data using a Poisson model. *The Annals of Applied Statistics*, *5*(4), 2493–2518. https://doi.org/10.1214/11-AOAS493

Wuest, T., Weimer, D., Irgens, C., & Thoben, K.-D. (2016). Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, *4*(1), 23–45. https://doi.org/10.1080/21693277.2016.1192517

Xia, Y. (2020). *Correlation and association analyses in microbiome study integrating multiomics in health and disease* (pp. 309–491). https://doi.org/10.1016/bs.pmbts.2020.04.003

Xintao, Q., & Dongmei, F. (2014). An Efficient Dimensionality Reduction Approach for Small-sample Size and High-dimensional Data Modeling. *Journal of Computers*,

*9*(3).

Xu, J., Mu, H., Wang, Y., & Huang, F. (2018). Feature genes selection using supervised locally linear embedding and correlation coefficient for Microarray classification. *Computational and Mathematical Methods in Medicine*, 1–11. https://doi.org/10.1155/2018/5490513

Yong, Z., Dun-wei, G., & Wan-qiu, Z. (2016). Feature selection of unreliable data using an improved multi-objective PSO algorithm. *Neurocomputing*, *171,* 1281–1290. https://doi.org/10.1016/j.neucom.2015.07.057

Zahoor, J., & Zafar, K. (2020). Classification of Microarray Gene Expression Data Using an Infiltration Tactics Optimization (ITO) Algorithm. *Genes, 11*(7), 819. https://doi.org/10.3390/genes11070819

Zararsız, G., Goksuluk, D., Korkmaz, S., Eldem, V., Zararsiz, G. E., Duru, I. P., & Ozturk, A. (2017). A comprehensive simulation study on classification of RNA-Seq data. *PLOS ONE, 12*(8), e0182507. https://doi.org/10.1371/journal.pone.0182507

Zhang, Z. H., Jhaveri, D. J., Marshall, V. M., Bauer, D. C., Edson, J., Narayanan, R. K., Robinson, G. J., Lundberg, A. E., Bartlett, P. F., Wray, N. R., & Zhao, Q.-Y. (2014). A Comparative Study of Techniques for Differential Expression Analysis on RNA-Seq Data. *PLoS ONE, 9*(8), e103207. https://doi.org/10.1371/journal.pone.0103207

Zhengyan, H., & Chi, W. (2017). Classifying Lung Adenocarcinoma and Squamous Cell Carcinoma using RNA-Seq Data. *Cancer Studies and Molecular Medicine. Open*

*Journal.*, *3*(2), 27–31. https://doi.org/10.17140/ CSMMOJ-3-120.

Zhou, Z.-H., Chawla, N. V., Jin, Y., & Williams, G. J. (2014). Big Data Opportunities and Challenges: Discussions from Data Analytics Perspectives [Discussion Forum]. *IEEE Computational Intelligence Magazine*, *9*(4), 62–74. https://doi.org/10.1109/MCI.2014.2350953

Zhu, M., Xia, J., Yan, M., Cai, G., Yan, J., & Ning, G. (2015). Dimensionality Reduction in Complex Medical Data: Improved Self-Adaptive Niche Genetic Algorithm. *Computational and Mathematical Methods in Medicine*, *2015*, 1–12. https://doi.org/10.1155/2015/794586

Zielezinski, A., Vinga, S., Almeida, J., & Karlowski, W. M. (2017). Alignment-free sequence comparison: benefits, applications, and tools. *Genome Biology*, *18*(1), 186. https://doi.org/10.1186/s13059-017-1319-7

# APPENDICES

# APPENDIX A

## LIST OF PUBLICATIONS FROM THE WORK

1. Arowolo, M.O., Adebiyi, M.O., & Adebiyi, A.A. (2019). A Dimensional Reduced Model for the Classification of RNA-Seq Anopheles Gambiae Data, *Journal of Theoretical and Applied Information Technology*, **97(23), 3487-3496** (***Scopus indexed***).
2. Arowolo, M.O., Adebiyi, M.O., & Adebiyi, A.A. (2020). An Efficient PCA Ensemble Learning Approach for Prediction of RNA-Seq Malaria Vector Gene Expression Data Classification. *International Journal of Engineering Research and Technology*, 13(1)**, 163-169** (***Scopus indexed***).
3. Arowolo, M. O., Adebiyi, M., Adebiyi, A., & Okesola, O. (2020). PCA model for RNA-SEQ malaria vector data classification using KNN and decision tree algorithm. *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*. https://doi.org/10.1109/icmcecs47690.2020.240881 (***Scopus indexed***).
4. Adebiyi, M.O., Adebiyi, A.A., Okesola, O., & Arowolo. M.O. (2020). ICA Learning Approach for Predicting RNA-Seq Data Using KNN and Decision Tree Classifiers. *International Journal of Advanced Science and Technology*. **29(3), 12273 – 12282 *(Scopus indexed).***
5. Arowolo, M.O., Adebiyi, M.O., & Adebiyi, A.A. (2020). A Hybrid Heuristic Dimensionality Reduction Methods for Classifying Malaria Vector Gene Expression

Data. *IEEE Access.* ***8: 182422-182430.*** *https://doi.org/10.1109/access.2020.3029234.* ***(Scopus indexed).***

**6.** Arowolo, M.O., Adebiyi, M.O., Adebiyi, A.A., & Aremu, C. (2020). An ICA-Ensemble Learning Approaches for Prediction of RNA-Seq Malaria Vector Gene Expression Data Classification. *International Journal of Electrical and Computer Engineering.* **11(2). *(Scopus indexed).***

**7.** Arowolo, M.O., Adebiyi, M.O., Adebiyi, A.A., & Okesola, O.J. (2020). A Genetic Algorithm Approach for Prediction of RNA-Seq Malaria Vector Gene Expression Data Classification Using Ensemble Algorithms. *International Journal of Electrical Engineering and Computer Sciences.* ***21(2.) (Scopus indexed).***

**8.** Arowolo, M.O., Adebiyi, M.O., & Adebiyi, A.A. (2020). A Genetic Algorithm Approach for Predicting RNA-Seq Data Classification Using KNN and Decision Tree. *Telkomnika.* ***19(1). (Scopus indexed)***

**9.** Adebiyi, M.O., Arowolo, M.O., & Olugbara, O. (2021). A genetic algorithm for prediction of RNA-Seq malaria vector gene expression data classification using SVM kernels. *Bulletin of Electrical Engineering and Informatics*, ***10(2).*** *10.11591/eei.v10i2.2769* ***(Scopus Indexed).***

**10.** Arowolo, M.O., Adebiyi, M.O., Adebiyi, A.A., & Olugbara, O. (2021). Optimized hybrid investigative based dimensionality reduction methods for malaria vector using KNN classifier. *Journal of Big Data*, ***8(29),*** *1-14. 10.1186/s40537-021-00415-z* ***(Scopus Indexed).***

**11.** Arowolo, M.O., Adebiyi, M.O., & Adebiyi, A.A. (2021). A Survey of Dimension Reduction and Classification Methods for RNA-Seq Data on Malaria Vector. *Journal of Big Data, 8**(50),** 1-17. 10.1186/s40537-021-00441-x **(Scopus Indexed).***

**12.** Adebiyi, M.O., Arowolo, M.O., Adebiyi, A.A., & Olatunji, O. (2021). ICA Learning Approach for Predicting of RNA-Seq Malaria Vector Data Classification Using SVM Kernel Algorithms. *JESTEC **(In-Press).***

**13.** Arowolo, M.O., Adebiyi, M.O., & Adebiyi, A.A. (2021). Enhanced Dimensionality Reduction Methods for Classifying Malaria Vector Dataset Using Decision Tree. *Sains Malaysiana. **(In-Press).***

**14.** Arowolo, M.O., Adebiyi, M.O., Adebiyi, A.A., & Olugbara, O. (2021). Optimized Hybrid Heuristic Based Dimensionality Reduction Methods for Malaria Vector Using Ensemble Classifier. *Heliyon. **(In-Press).***

# APPENDIX B

# DATA CODE

```
function varargout = Data(varargin)
% DATA MATLAB code for Data.fig
%      DATA, by itself, creates a new DATA or raises the existing
%      singleton*.
%      H = DATA returns the handle to a new DATA or the handle to
%      the existing singleton*.
%      DATA('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in DATA.M with the given input arguments.
%      DATA('Property','Value',...) creates a new DATA or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before Data_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to Data_OpeningFcn via varargin.
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help Data
% Developed by GUIDE v2.5 16-Oct-2019 21:12:10
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Data_OpeningFcn, ...
                   'gui_OutputFcn',  @Data_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
```

```
                'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before Data is made visible.
function Data_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Data (see VARARGIN)
% Choose default command line output for Data
handles.output = hObject;
guidata(hObject, handles);
set(handles.uipanel7,'visible','off');
set(handles.uipanel8,'visible','off');
set(handles.text6,'visible','off');
set(handles.text7,'visible','off');
set(handles.text8,'visible','off');
set(handles.text9,'visible','off');
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
set(handles.text13,'visible','off');
```

```
set(handles.uipanel9,'visible','off');
% Update handles structure
% UIWAIT makes Data wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = Data_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
%% Setup the GA
clear all; clc;
ff='testfunction'; % objective function
npar=2; % number of optimization variables
varhi=2; varlo=-1; % variable limits
%% II Stopping criteria
maxit=100; % max number of iterations
mincost=-999999; % minimum cost
%% III GA parameters
popsize=100; % set population size
mutrate=.01; % set mutation rate
selection=0.8; % fraction of population kept
Nt=npar; % continuous parameter GA Nt=#variables
keep=floor(selection*popsize); % #population members that survive
nmut=ceil((popsize-1)*Nt*mutrate); % total number of mutations
M=ceil((popsize-keep)/2); % number of matings // CEIL   Round towards plus infinity.
%% Create the initial population
iga=0; % generation counter initialized
```

```
par=(varhi-varlo)*rand(popsize,npar)+varlo; % random
Coords{1}=par;
cost=feval(ff,par); % calculates population cost using ff
[cost,ind]=sort(cost,'descend'); % min cost in element 1// SORT in ascending or
descending order.
par=par(ind,:); % sort continuous
minc(1)=max(cost); % minc contains max of
meanc(1)=mean(cost); % meanc contains mean of population
%% Iterate through generations (Main Loop)
while iga<maxit
iga=iga+1; % increments generation counter
% Pair and mate
M=ceil((popsize-keep)/2); % number of matings
prob=flipud([1:keep]'/sum([1:keep])); % weights chromosomes
odds=[0 cumsum(prob(1:keep))']; % probability distribution function
pick1=rand(1,M); % mate #1 (vector of length M with random #s between 0 and 1)
pick2=rand(1,M); % mate #2
% ma and pa contain the indices of the chromosomes that will mate
% Choosing integer k with probability p(k)
%
ic=1;
while ic<=M
for id=2:keep+1
if pick1(ic)<=odds(id) && pick1(ic)>odds(id-1)
ma(ic)=id-1;
end
if pick2(ic)<=odds(id) && pick2(ic)>odds(id-1)
pa(ic)=id-1;
end
end
```

```
ic=ic+1;
end
% Performs mating using single point crossover
ix=1:2:keep; % index of mate #1
xp=ceil(rand(1,M)*Nt); % crossover point
r=rand(1,M); % mixing parameter
for ic=1:M
xy=par(ma(ic),xp(ic))-par(pa(ic),xp(ic)); % ma and pa mate
par(keep+ix(ic),:)=par(ma(ic),:); % 1st offspring
par(keep+ix(ic)+1,:)=par(pa(ic),:); % 2nd offspring
par(keep+ix(ic),xp(ic))=par(ma(ic),xp(ic))-r(ic).*xy; % 1st
par(keep+ix(ic)+1,xp(ic))=par(pa(ic),xp(ic))+r(ic).*xy; % 2nd
if xp(ic)<npar % crossover when last variable not selected
par(keep+ix(ic),:)=[par(keep+ix(ic),1:xp(ic))
par(keep+ix(ic)+1,xp(ic)+1:npar)];
par(keep+ix(ic)+1,:)=[par(keep+ix(ic)+1,1:xp(ic))
par(keep+ix(ic),xp(ic)+1:npar)];
end % if
end
% Mutate the population
mrow=sort(ceil(rand(1,nmut)*(popsize-1))+1);
mcol=ceil(rand(1,nmut)*Nt);
for ii=1:nmut
par(mrow(ii),mcol(ii))=(varhi-varlo)*rand+varlo;
% mutation
end % ii
% The new offspring and mutated chromosomes are
% evaluated
cost=feval(ff,par);
% Sort the costs and associated parameters
```

```
[cost,ind]=sort(cost,'descend');
par=par(ind,:);
Coords{iga+1}=par;
% Plot function  26_11_16
figure (2)
  [X,Y] = meshgrid(-1:.02:2, -1:.02:2);
        Z =sin(4*pi*X).*X-sin(4*pi*Y+pi).*Y+1;
        %hold on;
        % sin(4*pi*xx).*xx-sin(4*pi*yy+pi).*yy+1
        surf(X,Y,Z)
        xlabel('x')
        ylabel('y')
        zlabel('f(x,y)')
       hold on;
  pcolor(X,Y,Z);
 % is really a SURF with its view set to directly above shading interp
% Plot offspring population at each generation
  plot3(Coords{iga+1}(:,1)',Coords{iga+1}(:,2)',cost, 'b*');
  hold off;
  %plot3(par(:,1)',par(:,2)',cost, 'b*');
  % Coords{iga+1}(:,1)'.*sin(4*pi*Coords{iga+1}(:,1)') -
Coords{iga+1}(:,2)'.*sin(4*Coords{iga+1}(:,2)'+pi)+1
    %axis([-1 2 -1 2]);
    pause(0.1)
% Do statistics for a single nonaveraging run
minc(iga+1)=max(cost);
meanc(iga+1)=mean(cost);
% Stopping criteria
if iga>maxit || cost(1)<mincost
break
```

```
end
[iga cost(1)];
end %iga
%% Displays the output
day=clock;
disp(datestr(datenum(day(1),day(2),day(3),day(4),day(5),day(6)),0))
disp(['optimized function is ' ff])
format short g
disp(['popsize=' num2str(popsize) ' mutrate=' num2str(mutrate) ' # par=' num2str(npar)])
disp(['#generations=' num2str(iga) ' best cost=' num2str(cost(1))])
disp('best solution')
disp(num2str(par(1,:)))
disp('continuous genetic algorithm')
figure(1)
iters=0:length(minc)-1;
plot(iters,minc,iters,meanc,'r');
xlabel('generation');ylabel('fitness');
title('Fitness function')
legend('Best individual','Mean of population','Location','east')
function Load_Callback(hObject, eventdata, handles)
% hObject    handle to Load (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename, pathname] = uigetfile({
  '*.xlsx;*.xls','excel files (*.xlsx,*.xls)'; ...
  '*.*',  'All Files (*.*)'}, ...
  'Pick a file');
columnformat={''}
set(handles.edit5,'string',filename);
filet=[pathname,'\',filename];
```

```
[n,t,raw]=xlsread(filet,'');
[ni,na]=size(n);
ni=num2str(ni);
na=num2str(na);
 set(handles.uitable1,'Data',raw,'ColumnFormat',columnformat);
 %% Loading data to table
pause(1)
set(handles.text13,'visible','on');
set(handles.text6,'visible','on');
pause(0.3);
set(handles.text7,'visible','on');
pause(0.3);
set(handles.text8,'visible','on');
pause(0.3);
set(handles.text9,'visible','on');
pause(0.3);
set(handles.text10,'visible','on');
pause(0.3);
set(handles.text11,'visible','on');
pause(0.3);
set(handles.text12,'visible','on');
pause(0.3);
set(handles.text7,'visible','off');
set(handles.text8,'visible','off');
set(handles.text9,'visible','off');
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
pause(0.3);
set(handles.text7,'visible','on');
```

```
pause(0.3);
set(handles.text8,'visible','on');
pause(0.3);
set(handles.text9,'visible','on');
pause(0.3);
set(handles.text10,'visible','on');
pause(0.3);
set(handles.text11,'visible','on');
pause(0.3);
set(handles.text12,'visible','on');
pause(0.3);
set(handles.text7,'visible','off');
set(handles.text8,'visible','off');
set(handles.text9,'visible','off');
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
pause(0.3);
set(handles.text7,'visible','on');
pause(0.3);
set(handles.text8,'visible','on');
pause(0.3);
set(handles.text9,'visible','on');
pause(0.3);
set(handles.text10,'visible','on');
pause(0.3);
set(handles.text11,'visible','on');
pause(0.3);
set(handles.text12,'visible','on');
pause(0.3);
```

```
set(handles.text7,'visible','off');
set(handles.text8,'visible','off');
set(handles.text9,'visible','off');
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
pause(0.3);
set(handles.text7,'visible','on');
pause(0.3);
set(handles.text8,'visible','on');
pause(0.3);
set(handles.text9,'visible','on');
pause(0.3);
set(handles.text10,'visible','on');
pause(0.3);
set(handles.text11,'visible','on');
pause(0.3);
set(handles.text12,'visible','on');
pause(0.3);
set(handles.text7,'visible','off');
set(handles.text8,'visible','off');
set(handles.text9,'visible','off');
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
pause(0.3);
set(handles.text7,'visible','on');
pause(0.3);
set(handles.text8,'visible','on');
pause(0.3);
```

```
set(handles.text9,'visible','on');
pause(0.3);
set(handles.text10,'visible','on');
pause(0.3);
set(handles.text11,'visible','on');
pause(0.3);
set(handles.text12,'visible','on');
pause(0.3);
set(handles.text7,'visible','off');
set(handles.text8,'visible','off');
set(handles.text9,'visible','off');
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
pause(0.3);
set(handles.text7,'visible','on');
pause(0.3);
set(handles.text8,'visible','on');
pause(0.3);
set(handles.text9,'visible','on');
pause(0.3);
set(handles.text10,'visible','on');
pause(0.3);
set(handles.text11,'visible','on');
pause(0.3);
set(handles.text12,'visible','on');
pause(0.3);
set(handles.text7,'visible','off');
set(handles.text8,'visible','off');
set(handles.text9,'visible','off');
```

```
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
pause(0.3);
set(handles.text7,'visible','on');
pause(0.3);
set(handles.text8,'visible','on');
pause(0.3);
set(handles.text9,'visible','on');
pause(0.3);
set(handles.text10,'visible','on');
pause(0.3);
set(handles.text11,'visible','on');
pause(0.3);
set(handles.text12,'visible','on');
pause(0.3);
set(handles.text7,'visible','off');
set(handles.text8,'visible','off');
set(handles.text9,'visible','off');
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
pause(0.3);
set(handles.text7,'visible','on');
pause(0.3);
set(handles.text8,'visible','on');
pause(0.3);
set(handles.text9,'visible','on');
pause(0.3);
set(handles.text10,'visible','on');
```

```
pause(0.3);
set(handles.text11,'visible','on');
pause(0.3);
set(handles.text12,'visible','on');
pause(0.3);
set(handles.text7,'visible','off');
set(handles.text8,'visible','off');
set(handles.text9,'visible','off');
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
pause(0.3);
set(handles.text7,'visible','on');
pause(0.3);
set(handles.text8,'visible','on');
pause(0.3);
set(handles.text9,'visible','on');
pause(0.3);
set(handles.text10,'visible','on');
pause(0.3);
set(handles.text11,'visible','on');
pause(0.3);
set(handles.text12,'visible','on');
%
pause(3)
msgbox('Data Succesfully Loaded');
set(handles.text13,'visible','off');
set(handles.text6,'visible','off');
set(handles.text7,'visible','off');
set(handles.text8,'visible','off');
```

```
set(handles.text9,'visible','off');
set(handles.text10,'visible','off');
set(handles.text11,'visible','off');
set(handles.text12,'visible','off');
% set(handles.pushbutton7,'visible','off');
nii=' Instances loaded';
naa=' Attributes loaded';
na1=strcat(na,naa);
ni1=strcat(ni,nii);
set(handles.text14,'string',na1);
set(handles.text15,'string',ni1);
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
b=get(handles.edit6,'string');
ext='.xlsx';
ex=[b,ext];
dt=get(handles.uitable1,'Data');
xlswrite(ex,dt);
msgbox('Data Saved succesfully');
% --- Executes on button press in pca.
function pca_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to pca (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of pca
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 global normilization
 if (normilization == 1)
 data = load('feature.mat');
data=data.feature;
fprintf('Orignal dimensions/features in dataset for each example\n')
size(data,2)
fprintf('implementing PCA ..\n')
[x_norm, mu, sigma] = featureNormalize(data);
[U,S,X_reduce]= pca(x_norm,10);
fprintf('Now the no. of dimesions/features in each instance of dataset is:\n')
size(X_reduce,2);
sz=size(X_reduce,2)
pcareduce=X_reduce;
save sz
save pcareduce
 elseif (normilization==2)
end
% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
b=get(handles.edit9,'string');
if isempty(b);
   errordlg('sig value cannot be null');
   return
else
group=xlsread('malaria.xlsx','A2:A63');
obs=xlsread('malaria.xlsx','B2:BXY63');
rng(80000,'twister');
 holdoutCVP = cvpartition(group,'holdout',2)
dataTrain = obs(holdoutCVP.training,:);
save dataTrain
grpTrain = group(holdoutCVP.training);
dataTrainG1 = dataTrain(grp2idx(grpTrain)==1,:);
dataTrainG2 = dataTrain(grp2idx(grpTrain)==2,:);
Significantvalue=str2num(b);
% [a1,a2,a3] = anova1(obs(:,1),group);
timing=tic;
[h,p,ci,stat] =
ttest2(dataTrainG1,dataTrainG2,'Vartype','unequal','alpha',Significantvalue);
save p
save grpTrain
figure,ecdf(p);
xlabel('P value');
ylabel('CDF value');
[~,featureIdxSortbyP] = sort(p,2); % sort the features
testMCE = zeros(1,54);
resubMCE = zeros(1,54);
nfs = 100:100:1000;
classf = @(xtrain,ytrain,xtest,ytest) ...
```

```
        sum(~strcmp(ytest,classify(xtest,xtrain,ytrain,'quadratic')));
resubCVP = cvpartition(length(group),'resubstitution');
collecter;
collecter1;
timesel=toc(timing);
timesel=num2str(timesel);
tm=timesel;
set(handles.text25,'string',tm);
remdata;
end
% for i = 1:54
%    fs = featureIdxSortbyP(1:nfs(i));
%    testMCE(i) = crossval(classf,obs(:,fs),group,'partition',holdoutCVP)...
%        /holdoutCVP.TestSize;
%    resubMCE(i) = crossval(classf,obs(:,fs),group,'partition',resubCVP)/...
%        resubCVP.TestSize;
% end
%  plot(nfs, testMCE,'o',nfs,resubMCE,'r^');
%  xlabel('Number of Features');
%  ylabel('MCE');
%  legend({'MCE on the test set' 'Resubstitution MCE'},'location','NW');
%  title('Simple Filter Feature Selection Method');
% testMCE(3)
% %%
% % These are the first 15 features that achieve the minimum MCE:
% featureIdxSortbyP(1:55)
% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
set(handles.pushbutton2,'enable','on');
[filename, pathname] = uigetfile({
  '*.xlsx;*.xls','excel files (*.xlsx,*.xls)'; ...
  '*.*',  'All Files (*.*)'}, ...
  'Pick a file');
columnformat={''}
set(handles.edit5,'string',filename);
filet=[pathname,'\',filename];
[n,t,raw]=xlsread(filet,'');
[ni,na]=size(n);
ni=num2str(ni);
na=num2str(na);
 set(handles.uitable1,'Data',raw,'ColumnFormat',columnformat);
function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes when selected object is changed in ftgroup.
function ftgroup_SelectionChangedFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in ftgroup
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global  normilization
value= get(eventdata.NewValue,'Tag')
switch value
   case 'pca';
     normilization =1
   case  'ica';
      normilization=2
```

```
end
% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.pushbutton13,'enable','on');
[filename, pathname] = uigetfile({
  '*.xlsx;*.xls','excel files (*.xlsx,*.xls)'; ...
  '*.*',  'All Files (*.*)'}, ...
  'Pick a file');
columnformat={''}
set(handles.edit14,'string',filename);
filet=[pathname,'\',filename];
[n,t,raw]=xlsread(filet,'');
[ni,na]=size(n);
ni=num2str(ni);
na=num2str(na);
at=' Attributes Loaded';
naa=strcat(na,at);
set(handles.text14,'string',naa)
 set(handles.uitable1,'Data',raw,'ColumnFormat',columnformat);
% --- Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global normilization value
extract=get(handles.edit14,'string');
deta=xlsread(extract);
```

```
 if (value== 'radiobutton13')
 %data = load('feature.mat');
timing=tic;
data=deta;
fprintf('Orignal dimensions/features in dataset for each example\n')
size(data,2)
fprintf('implementing ICA ..\n')
[x_norm, mu, sigma] = featureNormalize(data);
[U,S,X_reduce]= ica(x_norm,10);
fprintf('Now the no. of dimesions/features in each instance of dataset is:\n')
size(X_reduce,2);
sz=size(X_reduce,2)
pcareduce=X_reduce;
save sz
save icareduce
timing2=toc(timing);
timing2=num2str(timing2);
secc=' seconds';
tm=[timing2,secc];
set(handles.text24,'string',tm);
selectedf;
 elseif (value== 'radiobutton15')
   grp=load('grpTrain.mat','grpTrain');
   grp=grp.grpTrain;
   timing=tic;
   ncomp=25;
  [XL,YL,XS] = ICA(deta,grp,ncomp);
  save XS
  timing2=toc(timing);
  timing2=num2str(timing2);
```

```
secc=' seconds';
tm=[timing2,secc];
set(handles.text24,'string',tm);
  selectedf;
 end
% --- Executes when selected object is changed in uibuttongroup2.
function uibuttongroup2_SelectionChangedFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in uibuttongroup2
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global  normilization value
value= get(eventdata.NewValue,'Tag')
switch value
    case 'pca';
      normilization =1
    case  'ica';
       normilization=2
end
% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global value
if (value == 'radiobutton13')
dt=get(handles.uitable1,'Data');
 datac=dt;
 save datac
 cd=load('datac.mat');
 cd=cd.datac
```

```
 classificationLearner
 elseif (value =='radiobutton15')
 dt=get(handles.uitable1,'Data');
 datac=dt;
 save datac
 cd=load('datac.mat');
 cd=cd.datac
 classificationLearner
 end
% --- Executes on button press in pushbutton15.
function pushbutton15_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename, pathname] = uigetfile({
   '*.xlsx;*.xls','excel files (*.xlsx,*.xls)'; ...
   '*.*',  'All Files (*.*)'}, ...
   'Pick a file');
columnformat={''}
set(handles.edit17,'string',filename);
filet=[pathname,'\',filename];
n=xlsread(filet,'');
 set(handles.uitable1,'Data',n,'ColumnFormat',columnformat);
% --- Executes on button press in checkbox1.
function checkbox1_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of checkbox1
s=load('grpTrain.mat');
```

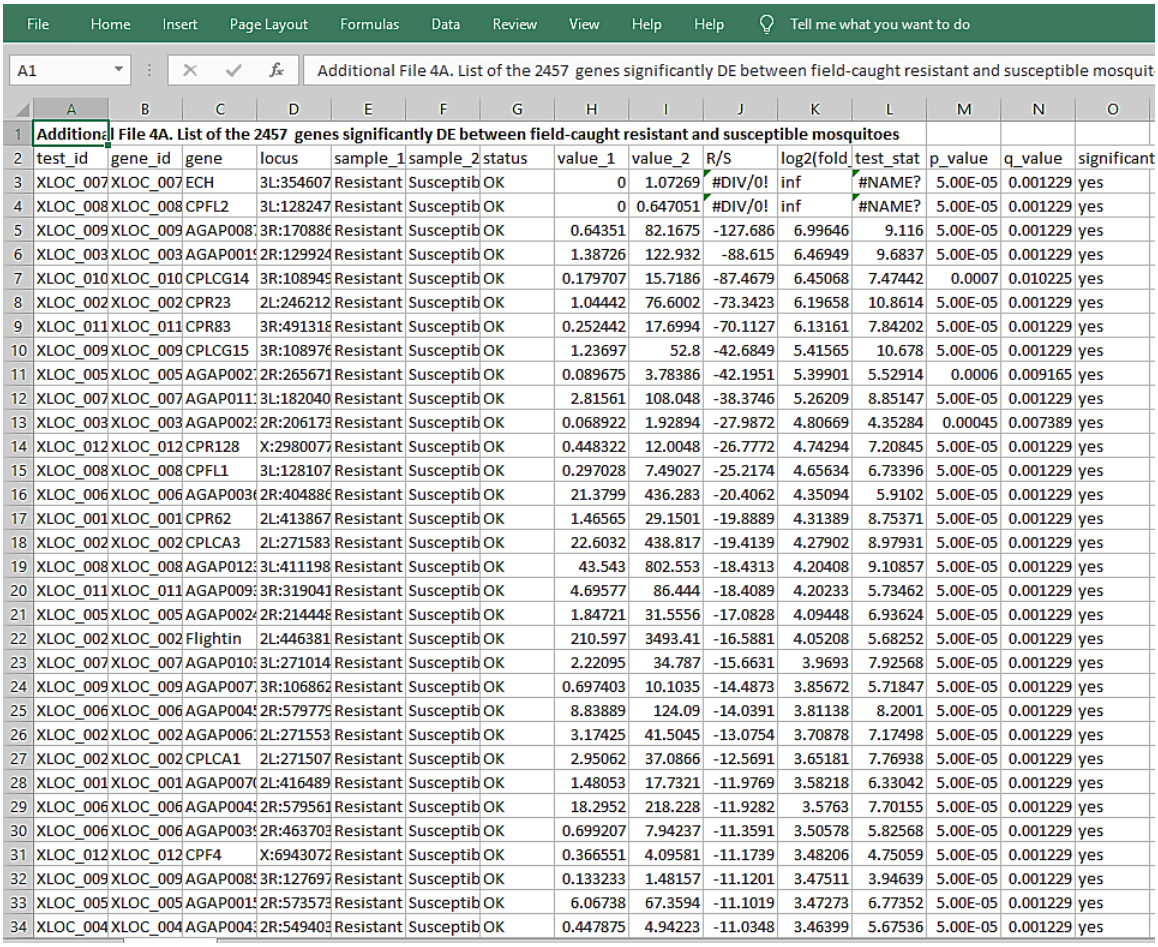
```
s=s.grpTrain;
d=get(handles.edit17,'string');
e=xlsread(d);
join=[s,e];
set(handles.uitable1,'Data',join);
% --- Executes on button press in pushbutton16.
function pushbutton16_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global normilization value
extract=get(handles.edit14,'string');
deta=xlsread(extract);
 if (value== 'radiobutton13')
 %data = load('feature.mat');
timing=tic;
data=deta;
deta=deta(:,2:end);
fprintf('Orignal dimensions/features in dataset for each example\n')
size(deta,2)
fprintf('implementing PCA ..\n')
[x_norm, mu, sigma] = featureNormalize(data);
[U,S,X_reduce]= pca(x_norm,10);
fprintf('Now the no. of dimesions/features in each instance of dataset is:\n')
size(X_reduce,2);
sz=size(X_reduce,2)
pcareduce=X_reduce;
save sz
save pcareduce
timing2=toc(timing);
```

```
timing2=num2str(timing2);
secc=' seconds';
tm=[timing2,secc];
set(handles.text24,'string',tm);
selectedf;
 elseif (value == 'radiobutton15')
     timing=tic;
 grp=deta(:,1);
 deta=deta(:,2:end);
 save grp;
 save deta;
   ncomp=20;
  [XL,YL,XS] = ica(deta,grp,ncomp);
  save XS
  timing2=toc(timing);
  timing2=num2str(timing2);
secc=' seconds';
tm=[timing2,secc];
set(handles.text24,'string',tm);
  selectedf;
 end
% --- Executes on button press in checkbox2.
function checkbox2_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of checkbox2
s=load('grp.mat','grp');
s=s.grp;
d=get(handles.edit17,'string');
```

```
e=xlsread(d);
join=[s,e];
set(handles.uitable1,'Data',join);
% --- Executes on button press in radiobutton15.
function radiobutton15_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton15
classes = [1,2,3]; % possible classes/labels
load CP-allGroups;
features_1 = features(labels==classes(1),:);
features_2 = features(labels==classes(3),:);
%% random subSampling
p = min(size(features_1,1), size(features_2,1));
idx = randsample(1:size(features_1,1),p);
features_1 = features_1(idx,:);
idy = randsample(1:size(features_2,1),p);
features_2 = features_2(idy,:);
features = [features_1;features_2];
%% binarize labels
labels = [];
labels(1:p,:) = 1;
labels(p+1:2*p,:) = 0;
labels = logical(labels);
```

# APPENDIX C

# LOADED DATA



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Additional File 4A. List of the 2457 genes significantly DE between field-caught resistant and susceptible mosquitoes | | | | | | | | | | | | | | |
| 2 | test_id | gene_id | gene | locus | sample_1 | sample_2 | status | value_1 | value_2 | R/S | log2(fold | test_stat | p_value | q_value | significant |
| 3 | XLOC_007 | XLOC_007 | ECH | 3L:354607 | Resistant | Susceptib | OK | 0 | 1.07269 | #DIV/0! | inf | #NAME? | 5.00E-05 | 0.001229 | yes |
| 4 | XLOC_008 | XLOC_008 | CPFL2 | 3L:128247 | Resistant | Susceptib | OK | 0 | 0.647051 | #DIV/0! | inf | #NAME? | 5.00E-05 | 0.001229 | yes |
| 5 | XLOC_009 | XLOC_009 | AGAP008 | 3R:170886 | Resistant | Susceptib | OK | 0.64351 | 82.1675 | -127.686 | 6.99646 | 9.116 | 5.00E-05 | 0.001229 | yes |
| 6 | XLOC_003 | XLOC_003 | AGAP001 | 2R:129924 | Resistant | Susceptib | OK | 1.38726 | 122.932 | -88.615 | 6.46949 | 9.6837 | 5.00E-05 | 0.001229 | yes |
| 7 | XLOC_010 | XLOC_010 | CPLCG14 | 3R:108945 | Resistant | Susceptib | OK | 0.179707 | 15.7186 | -87.4679 | 6.45068 | 7.47442 | 0.0007 | 0.010225 | yes |
| 8 | XLOC_002 | XLOC_002 | CPR23 | 2L:246212 | Resistant | Susceptib | OK | 1.04442 | 76.6002 | -73.3423 | 6.19658 | 10.8614 | 5.00E-05 | 0.001229 | yes |
| 9 | XLOC_011 | XLOC_011 | CPR83 | 3R:491318 | Resistant | Susceptib | OK | 0.252442 | 17.6994 | -70.1127 | 6.13161 | 7.84202 | 5.00E-05 | 0.001229 | yes |
| 10 | XLOC_009 | XLOC_009 | CPLCG15 | 3R:108976 | Resistant | Susceptib | OK | 1.23697 | 52.8 | -42.6849 | 5.41565 | 10.678 | 5.00E-05 | 0.001229 | yes |
| 11 | XLOC_005 | XLOC_005 | AGAP002 | 2R:265671 | Resistant | Susceptib | OK | 0.089675 | 3.78386 | -42.1951 | 5.39901 | 5.52914 | 0.0006 | 0.009165 | yes |
| 12 | XLOC_007 | XLOC_007 | AGAP011 | 3L:182040 | Resistant | Susceptib | OK | 2.81561 | 108.048 | -38.3746 | 5.26209 | 8.85147 | 5.00E-05 | 0.001229 | yes |
| 13 | XLOC_003 | XLOC_003 | AGAP002 | 2R:206173 | Resistant | Susceptib | OK | 0.068922 | 1.92894 | -27.9872 | 4.80669 | 4.35284 | 0.00045 | 0.007389 | yes |
| 14 | XLOC_012 | XLOC_012 | CPR128 | X:298007 | Resistant | Susceptib | OK | 0.448322 | 12.0048 | -26.7772 | 4.74294 | 7.20845 | 5.00E-05 | 0.001229 | yes |
| 15 | XLOC_008 | XLOC_008 | CPFL1 | 3L:128107 | Resistant | Susceptib | OK | 0.297028 | 7.49027 | -25.2174 | 4.65634 | 6.73396 | 5.00E-05 | 0.001229 | yes |
| 16 | XLOC_006 | XLOC_006 | AGAP003 | 2R:404886 | Resistant | Susceptib | OK | 21.3799 | 436.283 | -20.4062 | 4.35094 | 5.9102 | 5.00E-05 | 0.001229 | yes |
| 17 | XLOC_001 | XLOC_001 | CPR62 | 2L:413867 | Resistant | Susceptib | OK | 1.46565 | 29.1501 | -19.8889 | 4.31389 | 8.75371 | 5.00E-05 | 0.001229 | yes |
| 18 | XLOC_002 | XLOC_002 | CPLCA3 | 2L:271583 | Resistant | Susceptib | OK | 22.6032 | 438.817 | -19.4139 | 4.27902 | 8.97931 | 5.00E-05 | 0.001229 | yes |
| 19 | XLOC_008 | XLOC_008 | AGAP012 | 3L:411198 | Resistant | Susceptib | OK | 43.543 | 802.553 | -18.4313 | 4.20408 | 9.10857 | 5.00E-05 | 0.001229 | yes |
| 20 | XLOC_011 | XLOC_011 | AGAP009 | 3R:319041 | Resistant | Susceptib | OK | 4.69577 | 86.444 | -18.4089 | 4.20233 | 5.73462 | 5.00E-05 | 0.001229 | yes |
| 21 | XLOC_005 | XLOC_005 | AGAP002 | 2R:214448 | Resistant | Susceptib | OK | 1.84721 | 31.5556 | -17.0828 | 4.09448 | 6.93624 | 5.00E-05 | 0.001229 | yes |
| 22 | XLOC_002 | XLOC_002 | Flightin | 2L:446381 | Resistant | Susceptib | OK | 210.597 | 3493.41 | -16.5881 | 4.05208 | 5.68252 | 5.00E-05 | 0.001229 | yes |
| 23 | XLOC_007 | XLOC_007 | AGAP010 | 3L:271014 | Resistant | Susceptib | OK | 2.22095 | 34.787 | -15.6631 | 3.9693 | 7.92568 | 5.00E-05 | 0.001229 | yes |
| 24 | XLOC_009 | XLOC_009 | AGAP007 | 3R:106862 | Resistant | Susceptib | OK | 0.697403 | 10.1035 | -14.4873 | 3.85672 | 5.71847 | 5.00E-05 | 0.001229 | yes |
| 25 | XLOC_006 | XLOC_006 | AGAP004 | 2R:579775 | Resistant | Susceptib | OK | 8.83889 | 124.09 | -14.0391 | 3.81138 | 8.2001 | 5.00E-05 | 0.001229 | yes |
| 26 | XLOC_002 | XLOC_002 | AGAP006 | 2L:271553 | Resistant | Susceptib | OK | 3.17425 | 41.5045 | -13.0754 | 3.70878 | 7.17498 | 5.00E-05 | 0.001229 | yes |
| 27 | XLOC_002 | XLOC_002 | CPLCA1 | 2L:271507 | Resistant | Susceptib | OK | 2.95062 | 37.0866 | -12.5691 | 3.65181 | 7.76938 | 5.00E-05 | 0.001229 | yes |
| 28 | XLOC_001 | XLOC_001 | AGAP007 | 2L:416489 | Resistant | Susceptib | OK | 1.48053 | 17.7321 | -11.9769 | 3.58218 | 6.33042 | 5.00E-05 | 0.001229 | yes |
| 29 | XLOC_006 | XLOC_006 | AGAP004 | 2R:579561 | Resistant | Susceptib | OK | 18.2952 | 218.228 | -11.9282 | 3.5763 | 7.70155 | 5.00E-05 | 0.001229 | yes |
| 30 | XLOC_006 | XLOC_006 | AGAP003 | 2R:463703 | Resistant | Susceptib | OK | 0.699207 | 7.94237 | -11.3591 | 3.50578 | 5.82568 | 5.00E-05 | 0.001229 | yes |
| 31 | XLOC_012 | XLOC_012 | CPF4 | X:6943072 | Resistant | Susceptib | OK | 0.366551 | 4.09581 | -11.1739 | 3.48206 | 4.75059 | 5.00E-05 | 0.001229 | yes |
| 32 | XLOC_009 | XLOC_009 | AGAP008 | 3R:127697 | Resistant | Susceptib | OK | 0.133233 | 1.48157 | -11.1201 | 3.47511 | 3.94639 | 5.00E-05 | 0.001229 | yes |
| 33 | XLOC_005 | XLOC_005 | AGAP001 | 2R:573573 | Resistant | Susceptib | OK | 6.06738 | 67.3594 | -11.1019 | 3.47273 | 6.77352 | 5.00E-05 | 0.001229 | yes |
| 34 | XLOC_004 | XLOC_004 | AGAP004 | 2R:549403 | Resistant | Susceptib | OK | 0.447875 | 4.94223 | -11.0348 | 3.46399 | 5.67536 | 5.00E-05 | 0.001229 | yes |

https://figshare.com/articles/Additional_file_4_of_RNA-seq_analyses_of_changes_in_the_Anopheles_gambiae_transcriptome_associated_with_resistance_to_pyrethroids_in_Kenya_identification_of_candidate-resistance_genes_and_candidate-resistance_SNPs/4346279/1