CLASSIFICATION OF MEDICAL IMAGES USING METAHEURISTIC FEATURE SELECTION METHODS

A Thesis

by

KULADEEP ANAND KUMAR MADDULA

B.Tech, Velagapudi Ramakrishna Siddhartha Engineering College, 2014

Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

Texas A&M University-Corpus Christi
Corpus Christi, Texas

December 2019

December 2019

CLASSIFICATION OF MEDICAL IMAGES USING METAHEURISTIC FEATURE SELECTION METHODS

A Thesis

by

KULADEEP ANAND KUMAR MADDULA

This thesis meets the standards for scope and quality of
Texas A&M University-Corpus Christi and is hereby approved.

Dr. Scott King, PhD
Chair

Dr. Alaa Sheta, PhD
Co-Chair

Dr. Mamta Yadav, PhD
Committee Member

December 2019

ABSTRACT

Magnetic Resonance Imaging (MRI) is a popular non-invasive diagnostic tool for brain imaging. Accurate analysis of brain MRI images help in early detection of brain tumors and could save lot of lives. But accurate classification of the images as normal or pathological is a challenging task from the clinical as well as technology stand point. Brain MRI images consists of a large information set which contain redundancy in determining the condition of the brain. The redundant information would lead to increase in dimensionality of the data. Therefore, using a feature selection algorithm to find an optimum set of features would reduce the time and computation complexity of the classifiers for distinguishing the brain MRI images.

This work is to study the performance of feature selection with different meta-heuristic search algorithms with multiple fitness functions. The three meta-heuristic algorithms considered are Binary Genetic Algorithm, Binary Particle Swarm Optimization and Binary Grey Wolf Optimizer for selecting an optimal set of features out of the extracted features from brain MRI images. The feature selection is performed on the 13 statistical features extracted from the brain MRI images using Discrete Wavelet Transform, Principle Component Analysis and Grey Level Co-occurrence matrix. The performance of the feature selection algorithms are compared by applying 4 different sets of features from each algorithm to seven different test classifiers. Our results obtained show high performance using feature selection.

DEDICATION

To my mother, father and sister with love.

ACKNOWLEDGEMENTS

I would like to thank my committee members for their immense support in various stages of my thesis work, Dr. Scott King, Dr. Alaa Sheta and Dr. Mamta Yadav.

My sincere gratitude to Dr. Scott King and Dr. Alaa Sheta for guiding me throughout my research. I am heartily thankful to Dr. Hamza Turabieh for providing thoughts and insights in shaping the initial architecture of my work. Thanks to Hamid Kamangir for providing thoughts on visualizing my results. Thanks to Asha Nair and Ogwo Ogwo for helping me always.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES PAGE

FIGURES PAGE

LIST OF TABLES

CHAPTER I: INTRODUCTION

1.1 Problem Overview

Classification of brain images are done for the identification of the presence and type of brain tumors. Brain tumors are abnormal and uncontrolled proliferations of cells. Some originate in the brain itself, in which case they are termed primary. Others spread to this location from somewhere else in the body through metastasis, and are termed secondary. Primary brain tumors do not spread to other body sites, and can be malignant or benign. Secondary brain tumors are always malignant. Both types are potentially disabling and life threatening. Because the space inside the skull is limited, their growth increases intracranial pressure, and may cause edema, reduced blood flow, and displacement, with consequent degeneration, of healthy tissue that controls vital functions.

Early and accurate diagnosis of brain tumor is the key for implementing successful therapy and treatment planning. Traditionally the primary diagnosis tool for the detection of brain tumors by the neuro professional is using Magnetic Resonance Imaging (MRI). MRI is the viable option now for the study of tumor in soft tissues. The method clearly finds tumor types, size and location. MRI is a magnetic field which builds up a picture and has no known side effects related to radiation exposure. It has much higher details in soft tissues. However the Diagnosis is a very challenging task due to the large variance and complexity of tumor characterization in images, such as size, shape, location and intensities and can only be performed by professional neuroradiologists.[46]

Therefore, to overcome these challenges automatic computerized classification of brain MRI images for the presence and type of tumors is necessary. To classify any image it is required to extract some measurable property or characteristics from the image and which are called as features. The features required to classify the brain MRI images are numerous and out of which there exists redundancy. The redundancy can be removed by selecting some best set of features out of the available features in order to improve the classifier performance. In this work some statistical features have been extracted using some features extraction algorithms like Discrete Wavelet Transforma-

tion (DWT) and Principle Component Analysis (PCA) from the brain MRI images and used some famous and novel meta-heuristic search algorithms like Binary Genetic Algorithm, Binary Particle Swarm Optimization and Binary Grey Wolf optimizer for feature selection. The effectiveness of selected features from each algorithm is tested by applying six different classifiers like K-Nearest Neighbor (KNN), Naive Bayes classifier (NB), Discriminant Analysis (DA), Decision Tree (DT), Support Vector Machines (SVM) and Random Forest (RF).

1.2 Motivation

Brain tumors are, in fact, the second leading cause of cancer related deaths in children and young adults. It is one of the highly predominant health complications around the USA and across the world as well. According to the recent statistics from the year 2018, the American Brain Tumor Association predicted that there were around 80,000 new brain tumor cases are going to diagnosed in USA and out of which 32% of the cases were malignant. It also stated that that there were around 700,000 individuals across the USA suffering from both brain and central nervous system tumors and over 16,000 individuals are in the critical care getting treated for brain.[19]

Brain tumors are diagnosed by neuro physicians by performing an MRI imaging test and visually evaluating the radiography films. The type of brain tumors, whether benign or malignant can be evaluated by physicians by performing histopathology which is considered as the gold standard method for the tumor type detection. To perform histopathology on the tumor sample, the individual needs to be operated for removing the brain tumor. The cost of brain tumor operation from a hospital named Saint Elizabeth Medical Center from Nebraska would be up to 78,700 US dollars without considering the doctor fees. The brain tumor operations are always involved with additional side effects like weakness, dizzy spells, poor balance or lack of coordination, confusion, problems with speech and fits, which could be short term or long-term consequences. Therefore, early detection of type and presence of brain tumors would save a lot of lives.[56]

Classification of images requires extraction of information from the images which are called features. Numerous features can be extracted from an image which can be grouped into color,

texture, shape, position, dominant edges or regions from image. Image classification inherently deals with higher dimensional data and the result of increase in dimension would lead to increase in computation time and computation complexity. Therefore, feature selection is applied to select the best possible features that are unique and enough to classify the image. There are multiple feature selection methods that were applied in the past to classify the images and grouped into following three types.

- Filter Methods: The feature selection methods in this group select the subset of features based on the scores from various statistical tests to calculate the correlation of features with the dependent variable or the class label. The different techniques in this group are Pearson's correlation, Linear discriminant analysis, Analysis of variance, Chi-square. But the disadvantage of these methods is they do not remove multicollinear features and separate preprocessing need to be done for eliminating those.

- Wrapper Methods: These methods in common apply a subset of features for training the model and based on the feedback from the trained model they add or remove features from the subset. The different techniques in this group are Forward selection, Backward elimination, Recursive feature elimination. These methods are more likely like a search algorithm searching for a best feature subset. But these are computationally expensive and the ability to reach an optimum feature subset is no guaranteed.

- Embedded Methods: These methods are the combination of filter and wrapper methods. The different techniques involved are LASSO and RIDGE regression techniques. These techniques adds penalties to certain solutions in order to eliminate over fitting. But still these are unable to explore the complete search space and reach an optimal solution [28]

Therefore, considering the limitations of traditional feature selection algorithms, it is required to find an advanced algorithm which can search the whole search space in order to obtain an optimized subset of features. This leads us into choosing optimization algorithms which are not which are not bound to local optimal solution. Therefore, it is required to consider meta-heuristic

search algorithms which can explore a larger search space and effectively converge to a global optimal solution. The hierarchy of the optimization algorithms is displayed in Figure 1.1.

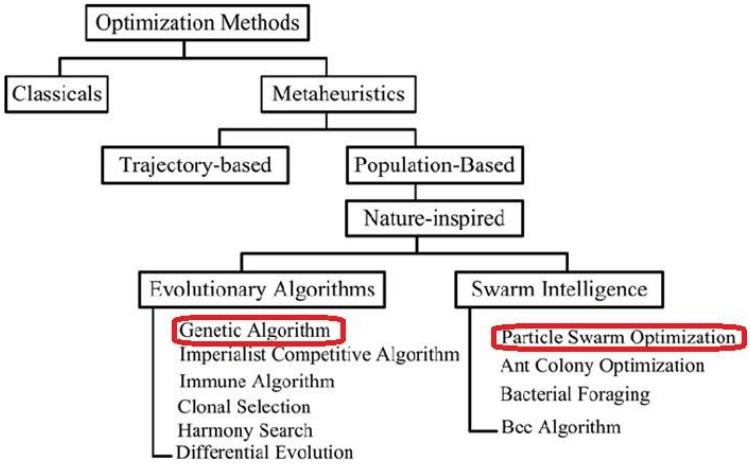


Figure 1.1: Hierarchy of Metaheuristic Optimization algorithms.

Some of the most popular meta-heuristic algorithms like Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), along with a novel swarm intelligence algorithm called Grey Wolf Optimizer (GWO) have been selected for feature selection in this research.

1.3 Purpose and Research Questions

The purpose of this research is to study the performance of different meta-heuristic feature selection algorithms with different variations of fitness functions in classifying the brain tumor images from the best features extracted.

In this work, two popular meta-heuristic algorithms namely *Binary Genetic Algorithm* (BGA) and Binary Particle Swarm Optimization (BPSO) are taken and a novel swarm intelligence-based algorithm called Binary Grey Wolf Optimizer (BGWO) is considered for feature selection. These algorithms were executed with three variations of fitness function namely K-Nearest Neighbor (KNN) classifier with two fold cross validation, KNN again with ten fold cross validation and Support Vector Machine (SVM) classifier with Gaussian Radial Basis function (RBF) kernel with ten fold cross validation. The research questions that will be addressed through this work are:

- How does the considered meta-heuristic search algorithms effect the classification performance of brain MRI images?
- How does different fitness functions affect the meta-heuristic search algorithms in finding an optimal solution?
- How the number of iterations is determined for the selected feature selection algorithms?
- What is the effect of computation time while using the meta-heuristic algorithms for feature selection?

1.4 Scope and Limitation

- This work compares the classification performance on the three feature subsets selected using Binary GA, Binary PSO, Binary GWO along with the complete feature set obtained from the brain MRI images.
- This work is confined with the Brain MRI images.

1.5 Outline

The outline of the remaining chapters is as follows: Chapter2 discusses the feature extraction, feature selection and classification algorithms that are considered in this work and also some of the previous work that is happening in this area of research, Chapter3 explains the data set, experimental setup, and methodology of the undergone research, Chapter4 explains the results obtained with feature selection algorithms, Chapter5 explains the analysis of results obtained using Classification algorithms, Chapter6 discusses the conclusion and future work for the research.

CHAPTER II: BACKGROUND

There is an increasing demand in developing advanced methods to automatically classify digital medical images. Extracting necessary and distinguishable information from the medical images is a crucial task in classifying them accurately. These measurable properties or information corresponding to an image are called features and are necessary in distinguishing the images into different categories. In general, the more information or features extracted, the more is the complexity and time involved in distinguishing the images. Therefore, there is an increase in demand in developing algorithms that select the optimal number of features, which could reduce the time and computation complexity of classification algorithms in classifying the images. In this research, brain MRI images are considered and different meta-heuristic feature selection algorithms are used to select the optimal set of features, and their performance is compared with different classification algorithms.

This chapter describes about the algorithms that were used for performing feature extraction, feature selection and classification in this research.

2.1 Feature Extraction

In this research the features from the brain MRI images are extracted using Discrete Wavelet Transformation, Principle Component Analysis and Grey Level Co-Occurrence matrix.

Discrete Wavelet Transformation

Wavelets are small localized basis functions with variable frequency and with finite or limited duration of time. They were initially introduced in [39], and a detail description of Wavelets is discussed in in [38]. They were also available in different sizes and shapes which are displayed in figure 2.1.

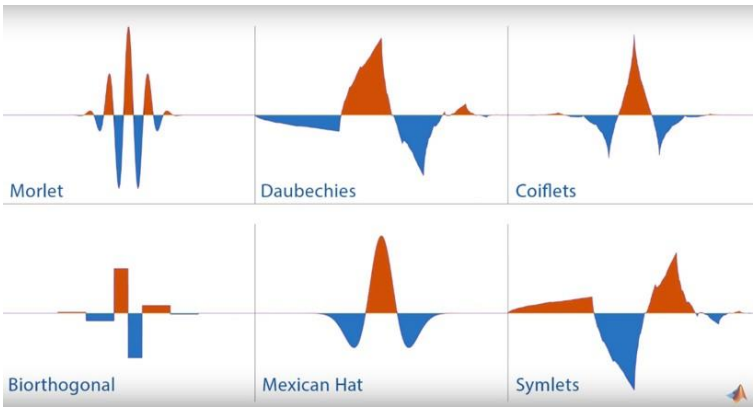


Figure 2.1: Different wavelets in use
[10]

The transformation of a signal or an image using wavelets is called Wavelet Transformation aka WT. In WT the original image or signal has been decomposed into basic wavelets with different resolutions which are called as mother wavelets. The transformation allows analysis of the image or signal in multiple resolution. Usually when seen in pattern recognition tasks, objects which are larger are well localized with small resolution levels and objects or details which are smaller are well localized with higher resolution levels [13]. Therefore, processing the image or signal at multiple resolutions provides the advantage of analyzing different characteristics of both the large and small objects with an image or signal.

In the image processing field, Discrete Wavelet Transformation aka DWT has been one of the popular multi-resolution transformation technique. The effectiveness of DWT for feature extraction of MRI brain images has been analyzed by several studies.

Basic mathematical notation of a wavelet decomposition is explained as follows.

For a given signal $x(t)$ which is a squared integral function, the continuous wavelet transformation of it relative to a real valued wavelet $\psi(t)$ is defined as in equation: 2.1

$$W_{\psi}(a,b) = \int_{-\infty}^{\infty} x(t)\,\psi_{a,b}(t)\,dx \tag{2.1}$$

The wavelet $\psi_{a,b}(t)$ is defined in equation 2.2

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left((t-a)/b\right) \tag{2.2}$$

and by performing dilation and translating on the mother wavelet ψ,the wavelet $\psi_{a,b}(t)$ is computed.

Where b is called as translaton parameter and a is called as dilation factor. Equation 2.1 can be discretized by constraining b and a to a discrete lattice ($a = 2b,\ a \in R_+,\ b \in R$) in order to obtain the DWT.

There are many different wavelets, and in this research, we use the Daubechies wavelet due to its balanced frequency response nature. It uses the overlapping windows due to which its high-frequency coefficient spectrum reflects all high-frequency changes. Therefore, this wavelet compresses and removes noise in signal and image processing.

DWT can be expressed as in equation 2.3

$$DW\,T_{x(n)} = \begin{cases} d_{j,k} = \Sigma x(n) h_j^*(n - 2jk), \\ a_{j,k} = \Sigma x(n) g_j^*(n - 2jk), \end{cases} \tag{2.3}$$

where, the parameters of DWT are explained below:

- The coefficients $d_{j,k}$ are referred as detail components in the input signal/image $x(n)$ and these are also correspond to wavelet function.
- The coefficients $a_{j,k}$ are referred as approximation components in the input signal/image $x(n)$.
- Function $h(n)$ is equation that represents the coefficient high pass filter.
- Function $g(n)$ is equation that represents the coefficient low pass filter.
- Parameter j represents the scale factor.
- Parameter k represents the translation factor.

The main characteristic of DWT is representing the function in multi scale. With the wavelets, any given function or image can be analyzed at different resolution levels. From the figure 2.2 the working of the DWT can be illustrated as follows. When a brain MRI image is passed to

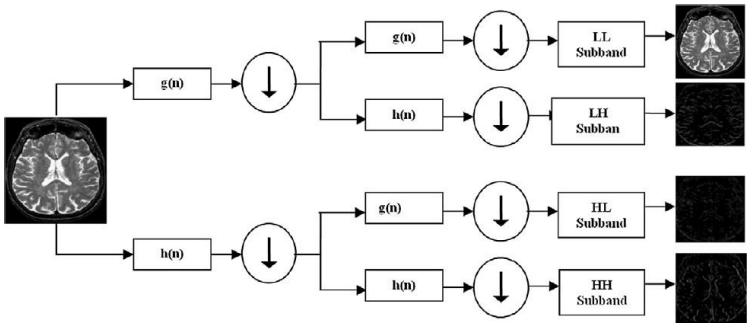


Figure 2.2: Schematic diagram of DWT process
[7]

DWT initially, it is processed along the *y* and *x* direction by $g(n)$ and $h(n)$ functions respectively. The resultant images that we obtain are called as Horizontal approximation and Horizontal details from the low and high pass filter respectively, and this is the row representation of the image. The approximation and detail images along the horizontal side are again passed through $g(n)$ and $h(n)$ functions and in result 4 sub band images (LL,LH,HH,HL) are obtained. The names of the sub band images are shown below:

- Image with LL sub band is called as Approximation image and is obtained when passed through two low pass filters.
- Image with LH sub band is called as Vertical detail and is obtained when passed through a low pass filter and then through a high pass filter.
- Image with HL sub band is called as Horizontal detail and is obtained when passed through a high pass filter and then through a low pass filter.
- Image with HH sub band is called as Diagonal detail and is obtained when passed through a high pass filter and then through a high pass filter.

The mentioned process is considered as a single level DWT, and in this research three levels of DWT is performed, where the low pass sub band represented by 'LL' from each end of level is passed on to another level [32].

Principal Component Analysis

Correlated features or attributes correspond to less significance in classifying the images. Therefore, transforming the feature space into a different dimension where the correlation among the features is least would greatly enhance the classification performance of the images. These operations can be performed a popular tool called Principal Component Analysis aka PCA.

PCA is referred to achieve valuable results from the applied linear algebra. It has been abundantly used for analysis in many fields, namely – from neurological sciences to computer graphics due to its simplicity and non-parametric approach of extracting most relevant information from high dimensional data set i.e. the data set with higher number of features. It requires a minimal effort in providing a road map on how to reduce a highly complex feature set into a low dimension level in order to reveal the hidden and simple dynamics that are often underlie in it [53].

PCA transforms the data orthogonally from a set of possibly correlated variables into a set of linearly uncorrelated variables, which are known as Principal Components. The transformation allows in retaining the most variations with in the feature set [23]. The algorithm of the PCA is explained in figure 2.3.

Grey Level Co-Occurrence Matrix

In medical image data texture information will constitute as one of the most important attribute or feature for identifying regions or areas of interest with an image. This information is crucial in classifying the medical images into different categories based on the nature of decease. Texture extraction can be done using a very robust method which involves the computation from a matrix called Grey Level Co-Occurrence Matrix aka GLCM.

GLCM has been one of the earliest methods for extracting texture information, and was proposed in 1973 by Haralick et.al. [17]. Since then GLCM has been used widely in the applications that require texture analysis.

GLCM approach which is also frequently called as spatial gray level dependence matrix approach

Let X be an input data set (X: matrix of dimensions $M \times N$).
Perform the following steps:

Step 1. *Calculate the empirical mean*: $u[m] = \frac{1}{N} \sum_{n=1}^{N} X[m,n]$.

Step 2. *Calculate the deviations from the mean and store the data in the matrix* $B[M \times N]$: $B = X - u \cdot h$, where h is a $1 \times N$ row vector of all 1's: $h[n] = 1$ for $n = 1, \ldots, N$.

Step 3. *Find the covariance matrix* C: $C = \frac{1}{N} B \cdot B^*$.

Step 4. *Find the eigenvectors and eigenvalues of the covariance matrix* $V^{-1}CV = D$: V – the eigenvectors matrix; D – the diagonal matrix of eigenvalues of C, $D[p,q] = \lambda_m$ for $p = q = m$ is the mth eigenvalue of the covariance matrix C.

Step 5. *Rearrange the eigenvectors and eigenvalues*: $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3 \geqslant \cdots \geqslant \lambda_N$.

Step 6. *Choosing components and forming a feature vector*: save the first L columns of V as the $M \times L$ matrix W,

$$W[p,q] = V[p,q], \quad \text{for } p = 1, \ldots, M,\ q = 1, \ldots, L \quad \text{where } 1 \leqslant L \leqslant M.$$

Step 7. *Deriving the new data set*: The eigenvectors with the highest eigenvalues are projected into space, this projection results in a vector represented by fewer dimension ($L < M$) containing the essential coefficients.

Figure 2.3: Principal Component Analysis Algorithm
[7]

aka SGLDM is mainly based on the studies of statistical distributions of pixel intensities. In practical applications statistics of a single pixel do no provide enough information regarding textures with in the image. Therefore, it is necessary to extract the second order statistical information, which can be obtained by considering the pairs of pixels with some spatial relationships to each other. This has brought the need to use co-occurrence matrices to express the probabilities or relative frequencies of the gray level pixel intensities [9]. The working of the approach is explained below 2.4

Consider an image with *NX N* pixels, then with the gray level intensities of each pixel a matrix has been developed for the image with an order of *NX N*. From the gray level pixel intensity matrix, the GLCM is calculated which is of the form $A(x, y|D, \vartheta)$. If x and y are two neighboring pixels in the image, then any of the element with the GLCM matrix *A* is represented by the relative frequency with which the neighboring pixels are separated with a pixel distance of *D*. This matrix shows the frequency of the appearance of different combination of gray levels with an image or segment of an image. The orientation ϑ for the two pixels to be in neighborhood can be considered

from the set (0°, 45°, 90°*and*135°), and in general the value for ϑ is considered as 45° and the value for pixel distance *D* is considered as 1 [63]. An example of the GLCM approach can be clearly understood form figure 2.4

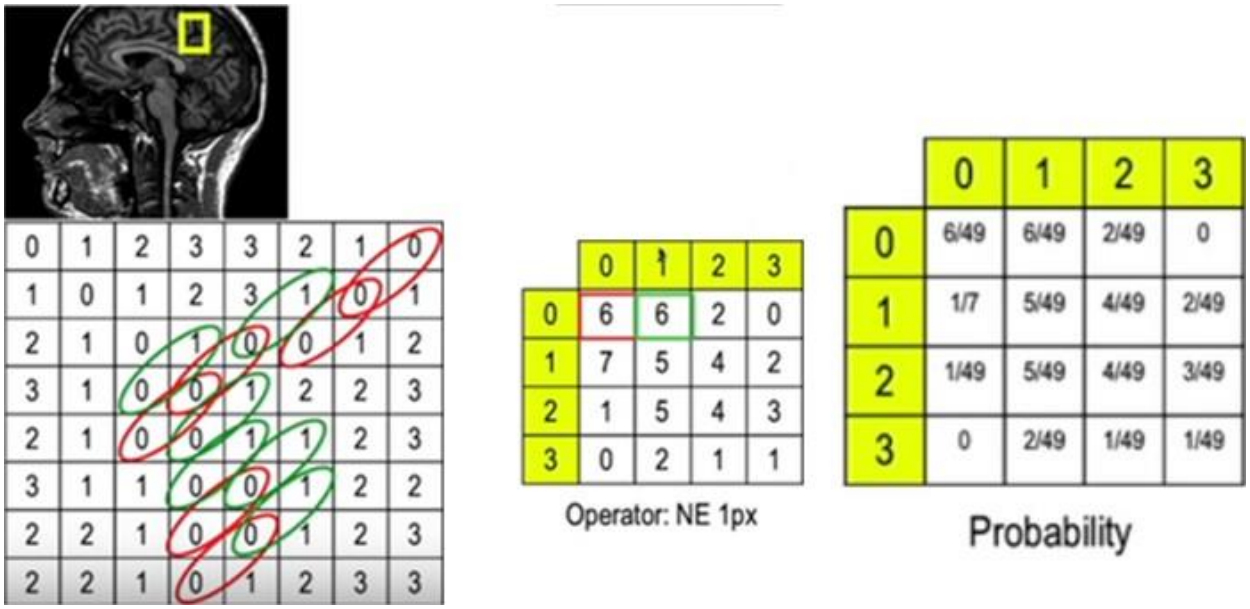


Figure 2.4: Computation of GLCM matrix

From the brain MRI image in the figure 2.4, a small 8*X* 8 pixel area has been selected. The gray level intensities of the pixels with in the selected area are in the range of 0*to*3, and a matrix with 8*X* 8 order has been populated with the gray level intensities of pixels. Later an intermediate matrix withe the frequency of neighborhood pixel intensity with a $\vartheta = 45°$ and $D = 1$ is calculated, and finally the GLCM matrix with the relative frequency of the intensity occurrence has been populated, then different texture properties like Homogeneity, Energy, Contrast and Correlation can be calculated for the respective image.

2.2 Feature Selection

Analysis of medical images deals with multitude of information which are called as attributes or features, which have to be considered to classify the deceased ones from the normal ones. Out of the available features there might be presence of irrelevant or redundant features that would

increase complexity in time and calculation for the classification algorithms. Therefore reducing the number of features by selecting a best subset from them is an efficient way in reducing the higher dimensionality and this approach is referred to Feature Selection.

A feature selection algorithm should search through a subset of total features and try to find the best one from the 2^N feature subsets by processing through an evaluation function or fitness function. But this happens to be an exhaustive approach as it is trying to find only one best one, and the computation can be highly complex for any mid sized feature set of size (N). There are other methods too which are based on heuristic search to reduce computation complexity, but these require a stopping criterion in order to prevent an exhaustive search within the subsets. In general, in any feature selection algorithm there are 4 main fundamental steps as show in in figure 2.5.

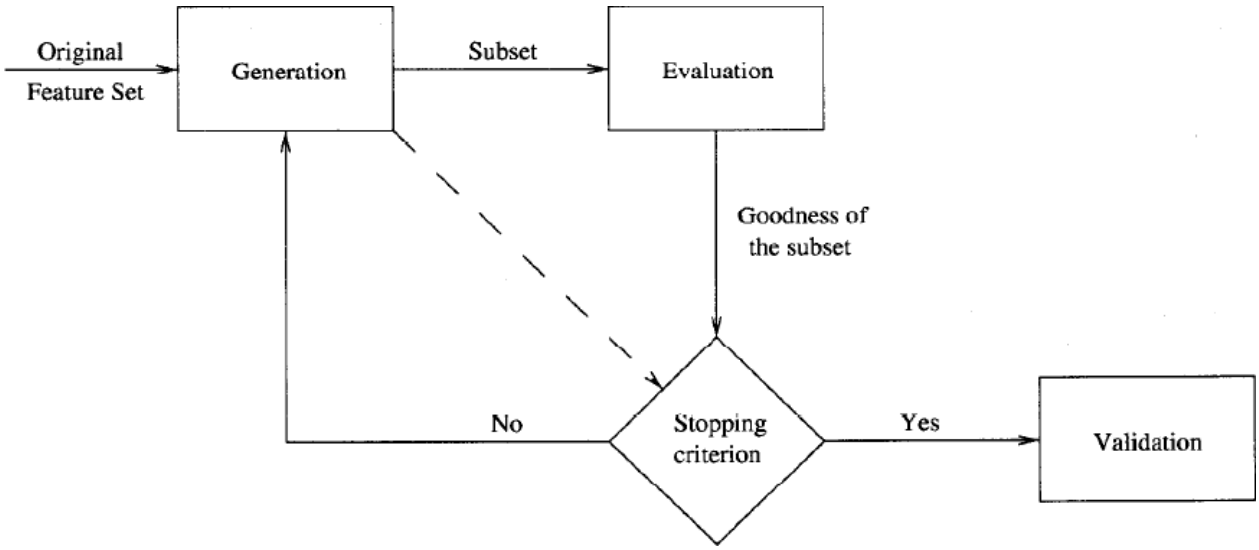


Figure 2.5: A typical Feature Selection algorithm
[8]

- Generation procedure for Feature Subsets: In this step the subset of features are generated for evaluation. The procedure can be started with any of the 3 following ways.
  - With zero number of features: This approach is generally considered by Forward selection methods.
  - With complete number of original features: This approach is generally considered by backward selection.

  - With random subset of features: In this approach features are either added or removed across the iterations or even randomly produced [35].

- Evaluation of feature subsets: In this step an the feature subset which is generated by some procedure is measured for goodness using an evaluation function, which is also referred to as fitness function.

- Stopping Criterion: In order to prevent an exhaustive search for feature subset a suitable stopping criterion is employed. The criterion can be chosen from following two ways.

  - Defined number of iteration has been reached
  - Defined number of features have been selected: This can be configured in two ways like - Addition or deletion of feature may not produce any improved feature subset, and An optimum feature set has been obtained according to the employed fitness function [54].

The traditional feature selection algorithms are grouped in to 3 main categories, which are namely Filter methods, Wrapper methods and Embedded methods, and a general working of each type is explained below.

- Filter Methods: A general working model of the algorithms that fall under this category is show in figure 2.6.

Figure 2.6: A generic algorithm for Filter Methods [28]

The approach used to select features in this method is independent of any kind of machine learning algorithm. Various statistical tests are performed to compare the correlation of the feature set with the class variable and based on the resulting scores the features subsets are selected.

- Wrapper Methods: The general mechanism of the algorithm that falls under this category is showing in figure 2.7.

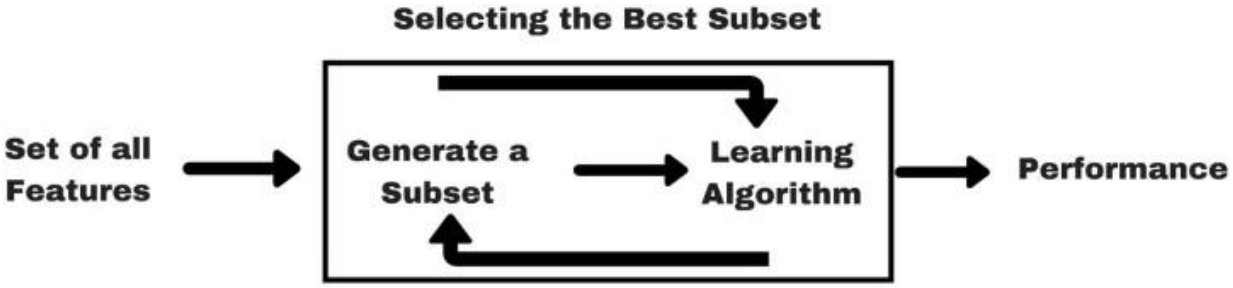


Figure 2.7: A generic algorithm for Wrapper Methods
[28]

In this method a feature subset is selected upon which a model is trained, and based on the inference of previous model the decision of adding or removing a feature to the original feature subset is performed. These algorithms are computationally expensive.

- Embedded Methods: These algorithms consists the characteristics of both Filter and Wrapper methods and built using algorithms with their own feature selection methods. A common representation of these algorithms are show in figure 2.8.

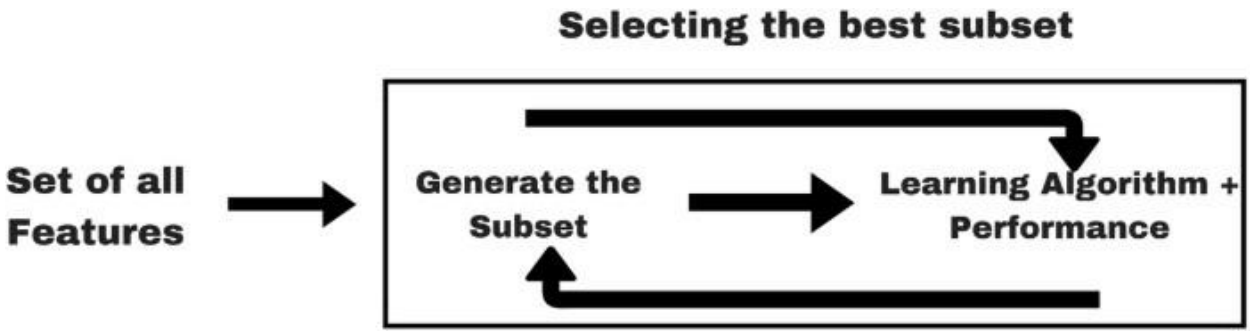


Figure 2.8: A generic algorithm for Embedded Methods
[28]

These feature selection algorithms select the feature subsets with a linear combination and due to which the search space for the feature subsets is less and hence result in providing a local optimal solution. The stochastic nature of meta-heuristic search algorithms find promising results

in performing feature selection. In this work 3 meta-heuristic feature selection algorithms are used namely Binary Genetic Algorithm, Binary Particle Swarm Optimization, Binary Grey Wolf Optimizer, and these algorithms will be introduced here.

Binary Genetic Algorithm

Genetic algorithm aka GA is one of the popular technique which belongs to the class of Evolutionary algorithms. It is a population based, randomized heuristic search technique which is developed based on the concept of natural selection, i.e. the nature's ability to preserve the fittest individuals []. It is introduced by Prof. John Holland in 1960 with the inspiration of Darwin's theory of evolution, and later extended by his student Prof. David E. Goldberg in 1989 [48]. The working of Genetic algorithm involves 6 main steps and are explained as follows.

- Initial Population: A high dimensional search space is generated with potential solutions, which are also called as chromosomes in GA. These individual chromosomes are represented as candidate solutions for the optimization problem being solved. There are many possible chromosome representations, namely matrices, bit vectors, LISP programs, etc., are used as encoding for the chromosomes. A typical representation of the chromosome is by n-bit binary vectors, which corresponds to an n-dimensional boolean search space.Therefore, in a features selection approach, each chromosome is represented as a possible feature subset.
- Fitness Evaluation: The quality of each individual solution is evaluated using an evaluation function often named as fitness function, and when applied provides a corresponding fitness value for each individual inside the population. In a feature selection problem, the chromosomes, which are a subset of total features are evaluated by a fitness function according to certain criteria like classification accuracy, cost of error from the classification using that feature subset, etc,. Depending upon the selected configuration of fitness measure, the problem becomes either as an increasing fitness or decreasing fitness approach.

- Selection: Using the probabilistic selection which is based on the fitness of individuals form the current population, are selected for reproduction to generate individuals for next generation. A number of selection methods are available for GA namely Roulette Wheel selection, Rank selection, Steady State selection, Tournament selection etc.,. In this research Roulette Wheel selection has been used and is explained as follows.

  – Roulette Wheel Selection is also know as fitness proportionate selection. Through this approach parent chromosomes are selected according to their fitness, and the better the fitness, the more chances to be selected. A general working of the selection procedure is explained in figure **??**.

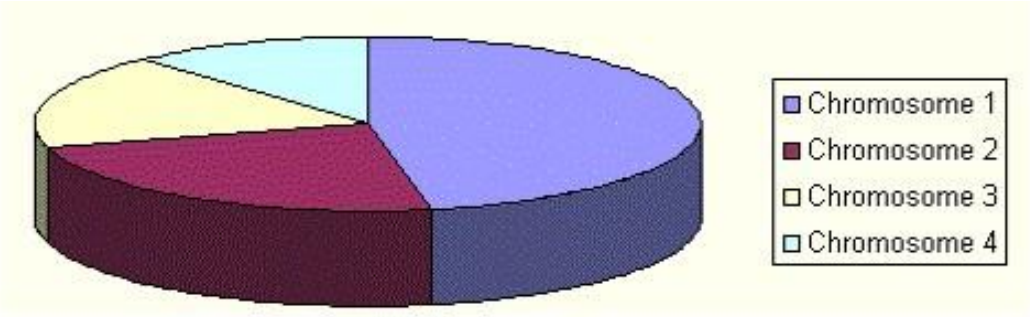


Figure 2.9: Roulette Wheel Selection Technique
[2]

Consider there are 4 chromosomes which constitute the population and based on the value of fitness the chromosomes occupy respective size of sectors on the disk. Therefore when the roulette wheel is moved, the probability of selecting the chromosomes with larger fitness is high [2].

- Crossover: This operation is performed to produce offspring which inherit the best characteristics of parents. Therefore, the two parents with higher fitness values, which are selected using Roulette Wheel selection as subjected to crossover operation. For example, consider

with a randomly chosen crossover position of 4, the two parents chromosomes 01101 and 11000 would produce offspring 01100 and 11001 respectively. This is performed to direct the individuals to a global optimal solutions, and is called exploitation. The crossover rate is one of the parameter which would determine the rate of performing crossover operation on chromosomes in the population.

- Mutation: It is performed on a single chromosome, which is in a binary string format, and in general alters a bit at random position inside the chromosome. For example, a chromosome represented with 11010 may change to 11110 as a result of mutation. Mutation is generally performed to explore the search space and help the population to reach a global optimal solution.

- Stopping Criterion: The end criterion for the algorithm can be met either by defining a limiting value for the average fitness of the population or either by defining a fixed number generations [42].

In this research the chromosome is encoded with binary string representation which makes the traditional GA called as Binary Genetic Algorithm aka BGA. The considered algorithm for BGA in this research is shown in figure 2.10.

The length of the chromosome considered in this research is equal to the total number of features, and the algorithm is executed using three different fitness functions namely K-Nearest Neighbors with 2 fold cross validation, K-Nearest Neighbors with 10 fold cross validation, and Support Vector Machine with Radial Basis Function kernel using 10 fold cross validation. The stopping criterion considered in this research is met using a fixed generation size of value 1000.

Binary Particle Swarm Optimization

Particle swarm optimization aka PSO is one of the parallel evolutionary algorithm introduced by Kennedy and Eberhart in the year 1995 [11]. This algorithms was developed with the simulation of multiple streamlined social models. It mimics the navigation and foraging of flock of birds and

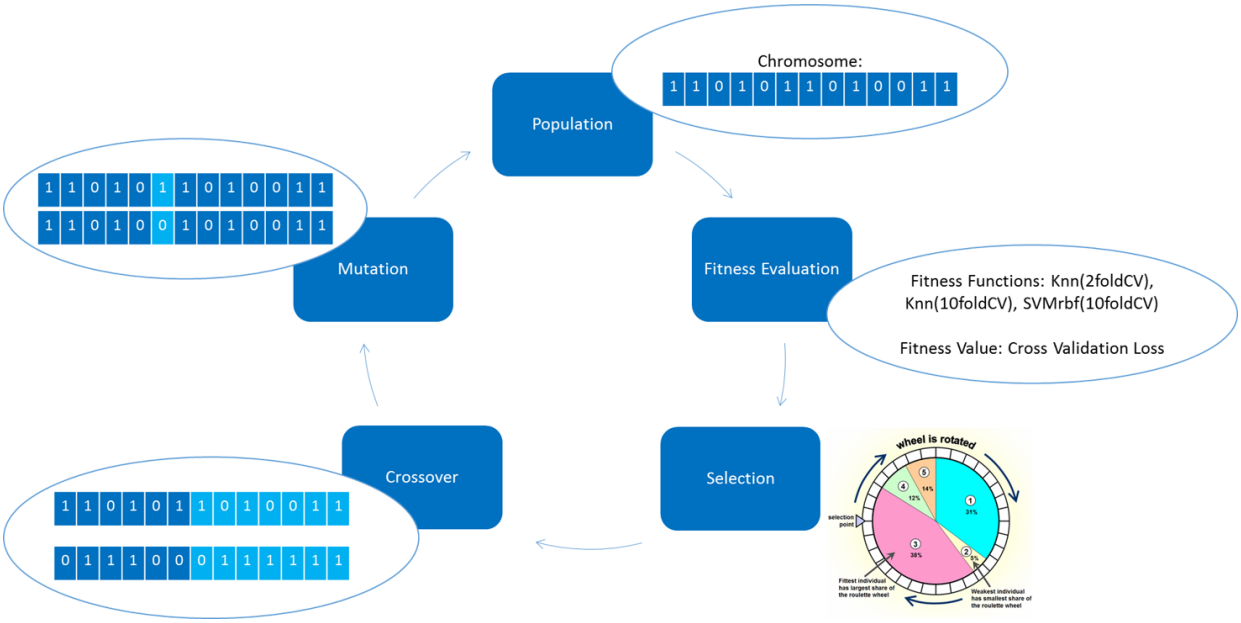


Figure 2.10: Binary Genetic Algorithm

school of fishes. The underlying concept of PSO is that the optimization of knowledge is obtained by social interactions in the population, where the decision is not exclusively taken by ones own personal cognition but with also considering a collective social information.

Similar to GA, each individual inside the population is a solution to the problem that is being analyzed, and each individual in PSO is considered as a particle inside the swarm. As the particles inside a swarm need to be moved from one position to another, each particle is associated with a position vector to represent its position and a velocity vector to represent its direction and speed of movement. In feature selection problem the representation of the particle is generally considered as an n-bit binary string for an n-dimensional search space. This make the algorithm represented as Binary Particle Swarm Optimization aka BPSO. The working of the PSO is summarized by the following steps.

- Initial Population: The algorithm is started with an initial population of randomly selected particles. Each particle is associated with a position vector say x_i, where x_i is position vector of particle i in the swarm and is represented as $x_i = (x_{i1}, x_{i2}, \ldots x_{iD})$. Where D is the search space dimension. And the velocity vector associated with each particle say v_i and is

represented as $v_i = (v_{i1}, v_{i2}, \ldots v_{iD})$

- Fitness Evaluation: Fitness of each individual particle inside the population is considered by the cost of error from classification, when performed on the feature subset represented by the particle.

- Updating Position and Velocity Vectors: During the iterations, the position and velocity vector of each particle in the population is updated using its personal experience and also considering with the global best results of the population. The is done by first storing the personal best position of the particle lets say in variable p_id, and storing the global best position of the whole population in lets say p_{gd}. Then with the following equations the position and velocity of each particle is updated.

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_1 * (p_{id} - x_{id}^{t}) + c_2 * r_2 * (p_{gd} - x_{id}^{t}) \quad (2.4)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2.5)$$

where d is the dimension index and $d \in D$, t is iteration number, w is called the inertial weight which is for controlling the significance of past velocity on current velocity, c_1, c_2 are the parameters for acceleration, r_1, r_2 are the random values which are uniformly distributed in the range [0,1].

- Stopping criterion: The algorithms is terminated either when fitness has reached the defined level or the maximum iterations have been passed [57].

The BPSO algorithm implemented in this research is shown in figure 2.11.

The representation of the particle is considered as the 13 bit binary string format, and the algorithm is executed using three different fitness functions namely K-Nearest Neighbors with 2 fold cross validation, K-Nearest Neighbors with 10 fold cross validation, and Support Vector

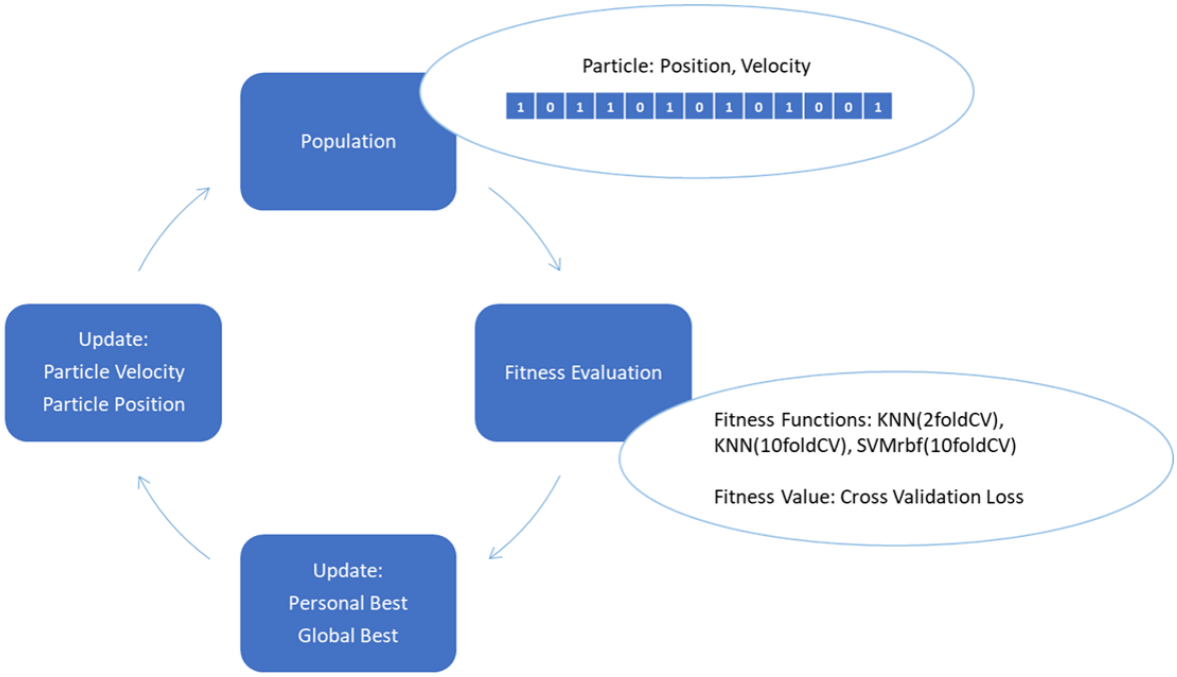


Figure 2.11: Binary Particle Swarm Optimization Algorithm

Machine with Radial Basis Function kernel using 10 fold cross validation. The stopping criterion considered in this research is met using a fixed generation size of value 1000.

Binary Grey Wolf Optimizer

Grey Wolf Optimization algorithm aka GWO is a recently developed swarm based stochastic search optimization technique. It was proposed in 2014 by an Australian scholar named Seyedali Mirjalili [41]. The algorithm mimics social dominant hierarchy and hunting strategy of grey wolves. Generally the grey wolves live in a pack of 5 to 12 and have a strict social dominant hierarchy. With the leadership of the head grey wolf, the pack performs the hunting mechanism trough o sequence of processes, first by surrounding the location of the prey, hunting and finally attacking. The social dominant hierarchy of the grey wolves is shown in figure2.12.

This social dominant hierarchy of grey wolves would lead to distributing the labour among the pack and help in effectively searching for prey. The α is the head wolf or leader of the pack, and is most capable and strongest individual in the pack. It is responsible for directing the food distribution, decision making, and predator searching activities of the pack. Where as the following

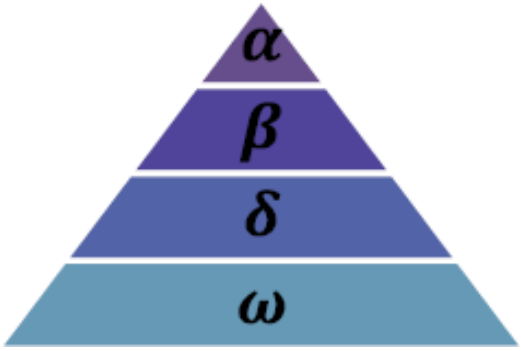


Figure 2.12: Social dominant hierarchy of grey wolves(Dominance decreases from top to down) [41]

two types of wolves β and δ are considered as second to α and mainly help in assisting the leader in managing pack activities.The last layer consists of the ω wolves which are the normal wolves that obey the instructions of the leader and second leaders of the pack. The hunting mechanism of the grey wolves is divided into 3 main steps which are shown in figure 2.13.

Figure 2.13: Grey wolf hunting mechansim: (A) Chasing, approaching and tracking prey, (B-D) Pursuing, harassing and encircling, (E) Attacking the prey [43]

Therefore from the figure 2.13, the hunting strategy is divided into three main steps as follows.

- Tracking, chasing and approaching the prey
- Pursuing, encircling, and harassing the prey till the movement of it is stopped.
- Performing attack towards the prey

The algorithm of GWO is divided into 3 preocesses: encircling,hunting and attacking, and are explained below:

- Encircling: The grey wolves begin to surround the prey after detecting its location. The mathematical equations explaining the process are:

$$D_p = |C \cdot X_p(t) - X(t)| \tag{2.6}$$

$$X(t+1) = X_p(t) - A \cdot D_p \tag{2.7}$$

D is the distance vector of gray wolf from prey, t is the iteration number, $X_p(t)$ is the position vector of prey in current iteration specified by the combined decision of α, β , δ , $X(t)$ is the position vector of gray wolf in current iteration, $X(t+1)$ is position vector of gray wolf in next iteration. And C, A are coefficient vectors, and are calculated as follows.

$$A = 2ar_1 - a \tag{2.8}$$

$$C = 2r_2 \tag{2.9}$$

r_1, r_2 are random vectors normally distributed in the range $[0, 1]$, and a is linearly decreased vector in the range $[2, 0]$ over the course of iterations. These coefficient vectors are responsible for determining the exploration and exploitation in the GWO algorithm.

- Hunting: The hunting operation is generally guided by the three dominant wolves in the pack α, β, δ. Partically, in the random genrated search space the location of prey is unknown, therefore the first three best solutions of the population (α, β, δ) are saved, and the rest of the ω wolves are to obliged by that decision. The pictorial representation of the hunting process is depticted in the figure 2.14.

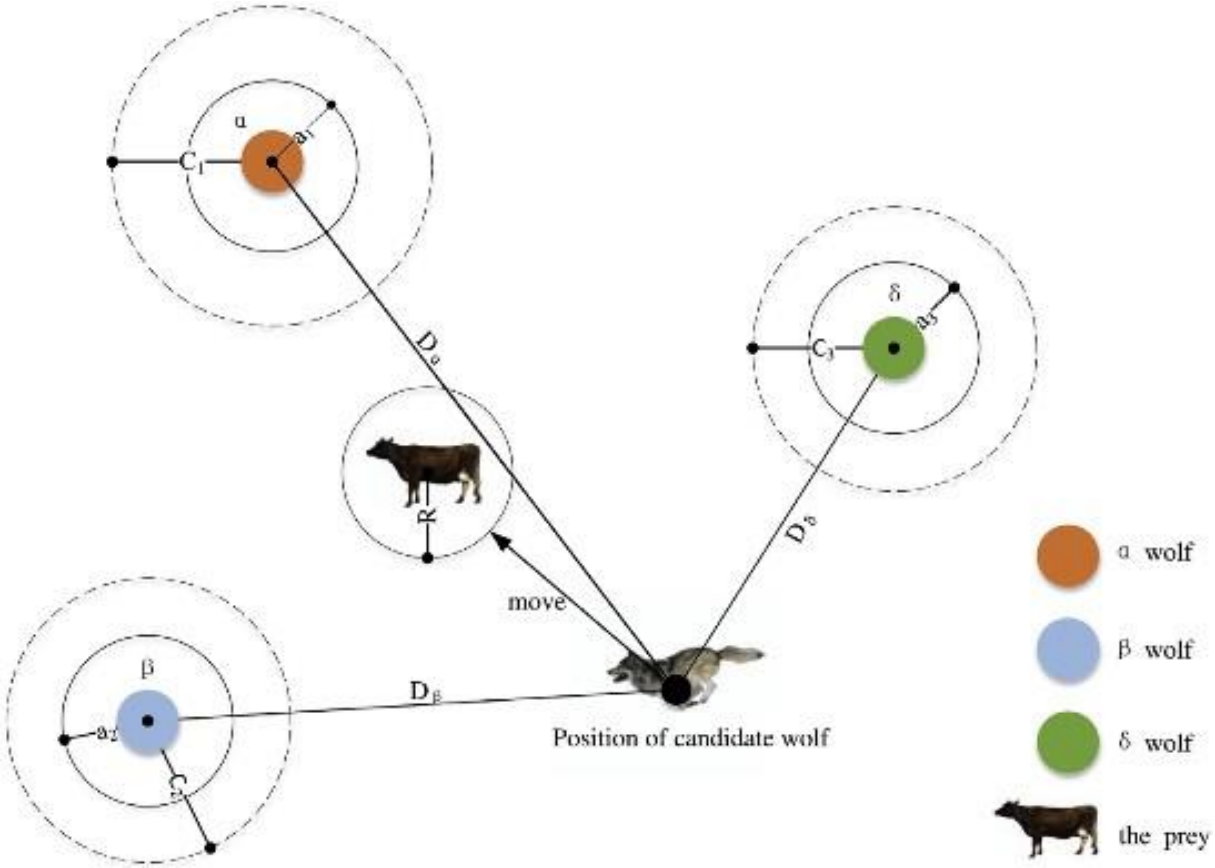


Figure 2.14: Updating position of gray wolf for hunting process [43]

The mathematical model representing the hunting process is shown below.

$$D_\alpha = |C_1 \cdot X_\alpha - X|, D_\beta = |C_2 \cdot X_\beta - X|, D_\delta = |C_3 \cdot X_\delta - X| \tag{2.10}$$

$$X_1 = X_\alpha - A_1 \cdot D_\alpha, X_2 = X_\beta - A_2 \cdot D_\beta, X_3 = X_\delta - A_3 \cdot D_\delta \tag{2.11}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{2.12}$$

Where the distance of α, β, δ wolves are calculated from the best average of previous positions and the current positions of respective wolves are updated accordingly. And based on the new average position all the other ω wolves are influenced.

- Attaching: Upon surrounding the prey, the grey wolves are ready to attack, i.e., the individuals are about to reach a solution. For mathematically modelling the process, the vector a is linearly decreased across the iterations. This results in the reduction of fluctuations of vector A. Depending on the value of a the vector A is in the range of $[-1, 1]$, and when $|A| > 1$ the wolves move away from converging and eventually at the end of iterations when $|A| < 1$ the wolves converge to a global optimal solution.

This algorithm is simple, easy to implement and has a very few parameters, due to which it has been an interest of research for many scholars today [36, 37, 14].

In this research the binary string representation of grey wolf is considered which makes it a Binary Grey Wolf Optimization algorithm aka BGWO, and the length of each grey wolf is 13, which is equal to the total number of features in the population.

2.3 Classification

The obtained feature subsets form the feature selection algorithms are used to perform classification. In this work 7 different classifiers with variations are used to to compare the performance of feature selection. These are discussed in this section.

K-Nearest Neighbors

K nearest neighbor aka KNN algorithm is one of the simplest and fundamental classification technique, with a minimal or no prior information on the distribution of the data [26]. The algorithm

performs classification of objects based on the neighbor hood of the training samples in the feature space.It is a kind of lazy learning or instance based learning where the computation is being deferred until classification [6].

In this method each object provided is classified based on the similarity of the surrounding samples. The distance of the surrounding known samples from unknown one is calculated using different methods namely Euclidean distance, Manhattan distance, Minkowski distance, Hamming distance, etc., out the available ones, Euclidean distance is most popularly used method. The algorithm with K=1 is in the simplest form, where for any unknown sample and the training set provided, the distance between all the samples and unknown sample are computed, and the unknown test sample is set to be belonging to the sample corresponding with shortest distance. From the figure 2.15, the model to the left is developed with K value 1, and to the right is developed with K value 4. When K is 1 the unknown test sample is compared with only nearest sample with shortest distance, where as when K is 4, 4 samples with shortest distance are computed and is the test sample is belong to class of nearest majority samples.

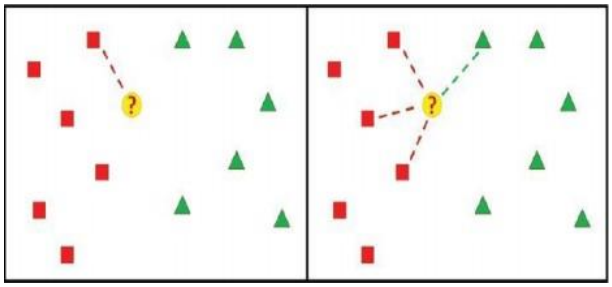


Figure 2.15: KNN classifier model with k=1(right),k=4(left)
[20]

Naive Bayes

The Naive Bayes classifier or simply Naive Bayes aka NB is a probabilistic classifier, which works on the principles of Bayes Theorem. It works by calculating the probabilities using the frequency counts and combination of different training values from the provided dataset [45]. The algorithm simplifies the computation, with the assumption of conditional independence between the feature

variables for a given class label or target value. To elaborate, the algorithm ignores the following possible dependencies among the features in the data set, namely - the correlations among the features of the dataset, there by decreasing the complexity of the classification problem from multivariate to uni variate.

The mathematical equations behind the NB classifier are shown below [21]:

$$P(a_1, a_2, ...a_n|v_j) = argmax_{v_j \in V} \prod_i P(a_i|v_j) \tag{2.13}$$

$$v_{NB} = argmax_{v_j \in V} P(v_j) \prod_i P(a_i|v_j) \tag{2.14}$$

Due to assumption of conditional independence between the feature, for the given feature set $a_1, a_2, ...a_n$ with known class label v_j, the probability of likelihood $P(a_1, a_2, ...a_n|v_j)$ is nothing but the probabilities of individual attributes. $P(v_j)$ is the prior probability information about the class labels of the each individual sample from data set. From the above equation, the product of distinct target values and the distinct attribute values is computed to predict the number of distinctive terms $P(a_i|vj)$ from the train data [21].

The variations of the algorithm can be obtained using different distribution types for the predictor variables, namely, normal or Gaussian , kernel , multivariate multinomial, multinomial distribution types. In the work normal and kernel variations of the algorithm are used for classification.

Linear Discriminant Analysis

Linear Discriminant Analysis is used for both regression and classification problems, but apart from the regression algorithms where a real valued output is provided, it provides the result with a predicted class label. In case of a regression problem, the algorithm provides a linear or curved line to separate the data set into different class categories. Where as in a tow dimensional or multidimensional data space, the algorithm generally develops a hyper plane to separate the data set. From the figure 2.16, the data belonging to two classes represented by blue and red dots can

be separated by linear or quadratic lines using this classification algorithm. The algorithm outputs and equation for classifying the test data samples [LDA1].

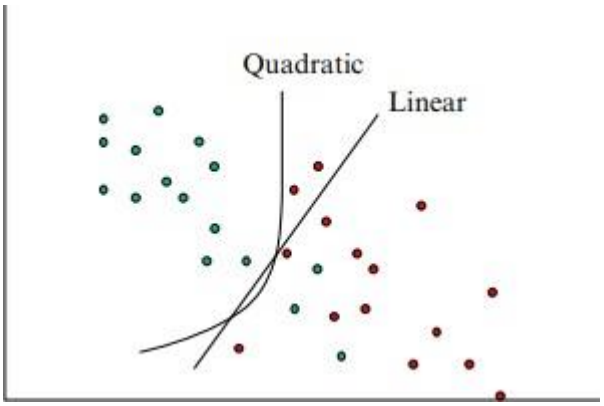


Figure 2.16: Quadratic and Linear functions by Linear Discriminant Analysis in separating the data set into categories
[16]

The algorithm performs classification by the linear combination of features or attributes of the data sample. This is performed with the assumption that the feature of the data are normally distributed, and also considering the equality of in-group co-variance matrices. Even with the avoidance of the assumptions on the data set, the algorithm is still robust in predicting the class labels. Therefore, this makes the algorithm simple and makes it a viable first decision for classification [16].

Decision Tree

Decision Tree is a decision support tool which uses a tree shaped graph with multiple decision branches, for classifying the data samples into categories. The algorithm contains conditional control statements which help in determining the class label of the data sample. This tree based learning algorithm is an accurate, stable and easily interpreted one, and is used for many supervised classification learning problems [3]. The algorithm is good at mapping non linear relationships than linear ones in the data set. It is adaptable in solving both regression and classification problem, which led to calling it as Classification and Regression trees aka CART.

Decision Tree follows the structure of a flow chart in building the model, where each node in the tree corresponds to an attribute or feature, each branch of the tree represents the outcome of the feature, and each leaf in the tree represents the class label. The node at the top of the tree is called root node, nothing but the feature or attribute selected to start building the tree. The path from root node to a leaf represents a classification rule. A simple classification problem of classifying the credit card customers into good and risk categories using the algorithm is shown in figure 2.17.

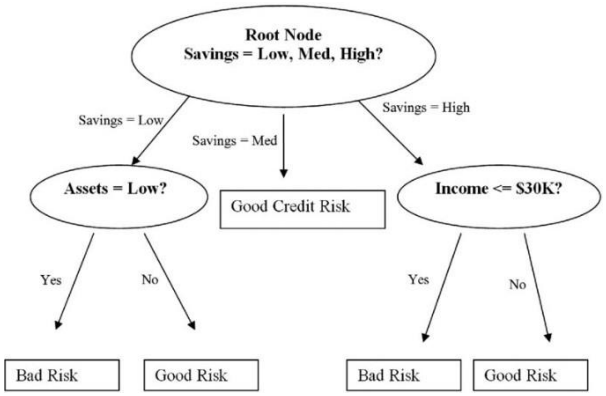


Figure 2.17: Decision Tree model structure for classifying credit card customers [3]

From the figure it is shown that the attributes Savings, Assets and Income are the features or the data set. A decision rule is shown with each branch starting connecting from root node represented with feature Savings to the any of the leaf. Decision trees and come with the limitation of overfitting and which can be solved using pruning process [3].

Random Forest

Random forest is an ensemble model for classification and regression problems. An ensemble algorithm is the one which combines more than one algorithms of different or same type to perform classification, and at the end considers a majority vote in predicting the class label [44].

Random forest is an ensemble model which uses multiple decision trees to perform classification. It creates a set of decision tress by randomly selecting subsets from the training data set, and at the end it considers an aggregate voting from all decision trees considered to predict the class

of the test data sample. A simple architecture of Random forest using multiple decision trees for classification is shown in figure 2.18.

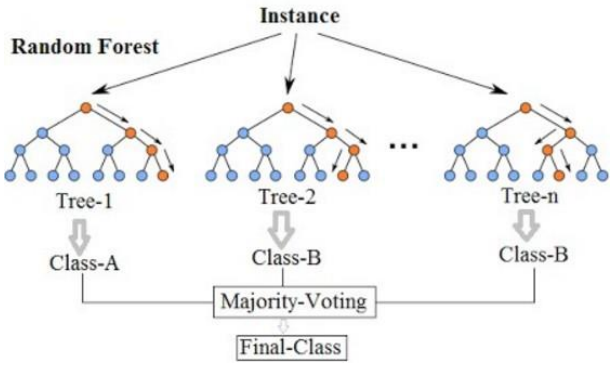


Figure 2.18: A simple architecture of Random Forest [33]

It is observed from the figure, that n number of decision trees are used for generating the random forest. And considered each decision tree provides a resulting binary class label of A or B, from which majority voting is considered from which a final class label is predicted [33].

Support Vector Machines

Support Vector Machine has been a powerful classification algorithm in the field of supervised machine learning applications since its introduction in 1995. The concept behind the algorithm is derived from the theory of statistics developed by Vapnick in 1982 [58]. It has been proved as a successful classification algorithm in numerous medical diagnostic applications [15, 62]. The algorithm can be used for classification and regression tasks, but have been widely implemented for classification problems [12]. The principle behind the algorithm is based on structural risk minimisation which has been derived from theory of statistical learning [29].

The algorithm's objective is to identify an optimal hyperplane for an N-dimensional feature space, which help in efficient separation of data members into different categories [30].

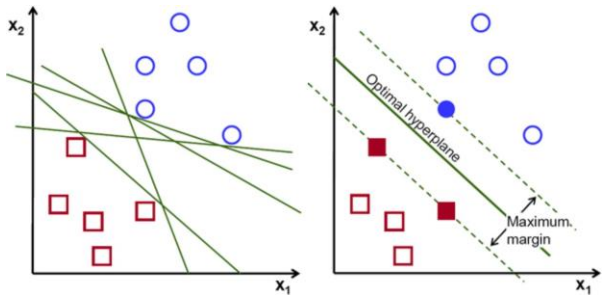


Figure 2.19: Support Vector Machine possible hyper planes
[12]

From figure 2.19, in order to separate the data samples into two distinct classes, there are numerous possibilities for choosing the hyper planes. The algorithm helps in choosing a particular hyper plane that divides the data samples with maximum margin, i.e., with the maximum distance possible between the samples. This helps in the classifying the future data with more confidence [12].

The hyper planes are separating vectors that divides the data in N-dimensional feature space. The figure 2.20 shows different dimensions of hyper planes depending on the dimension of feature space.

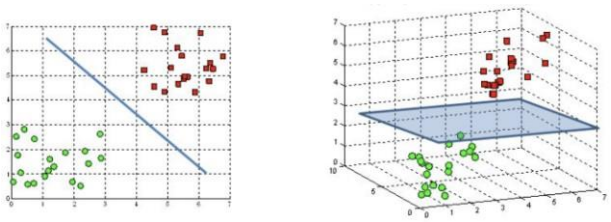


Figure 2.20: Support Vector Machine hyper planes with 2 and 3 dimensional feature space
[12]

From the above figure 2.20 it can be seen that the hyperplane is just a line for a two dimensional feature space, and the hyper plane is 2-D plane for a three dimensional feature space [12].

The support vectors are the data points which are closest to the hyper plane, and these are important in determining the slope and position of hyper plane. The margin is maximised with the help of support vectors and deleting of which will alter the slope and position of hyperplane.

As the non linearity of feature space increases, complexity of efficiently categorizing the data also increases. This requires the data to be transposed to higher dimension in order to make the data easily separable. The kernel functions are used to map the data on to a higher dimensional space, and there different variations of kernels available with SVM. The ones that are used in this work are shown below [29]:

- Linear Kernel:

$$K(x_i, x_j) = x_i \cdot x_j \tag{2.15}$$

- Polynomial kernel: Polynomial with degree 2 is used in this work. Which is called as quadratic kernel.

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{2.16}$$

- Radial Basis Function kernel:

$$K(x_i, x_j) = \exp\left(-\frac{|x_i - x_j|^2}{2\sigma^2}\right) \tag{2.17}$$

Artificial Neural Networks

Artificial Neural Network algorithm aka ANN has been a popular computational model for classification problems. It is inspired from the parallel working nature of human brain. The algorithm is a highly parallel interconnected network of neurons, which are the processing elements of the network. These neurons are inspired from the human biological nervous system [50].

The neurons in the network are divided into subgroup which are called layers. In a typical neural network there are three layers which are shown in figure 2.21.

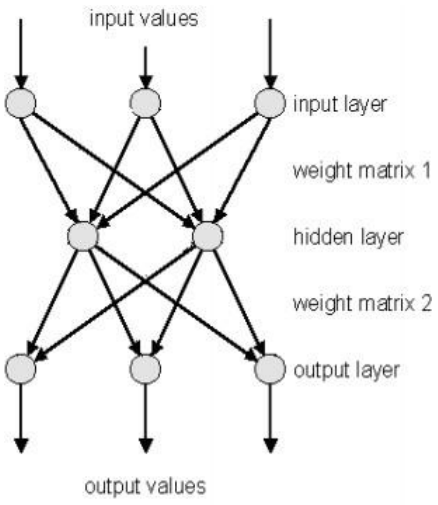


Figure 2.21: Typical Artificial Neural Network Structure
[50]

The three layers in a typical ANN are the input, hidden and output layers. Where generally the number of hidden layers can be varied. The network can be trained to perform different functions by altering the connection weights between the neurons. The algorithm has a wide span of implementations in both medical and scientific fields [24, 49, 5, 18]. Based on the nature of learning by the algorithm, the neural networks are divided into two groups namely supervised and unsupervised learning algorithms. In the former type the algorithm is trained by providing the both input and output values where as in later the algorithm by itself determines the class or group it belongs to from the provided input values. In this work supervised variation of ANN has been used.

There are different variation of supervised ANNs available out of which Feed Forward Artificial Neural Networks aka FFANN are most popularly used. These are used to solve many classification problems with complex and highly non linear relationship between input and output variables [27, 60]. This algorithm allows the information to be travelled only in forward direction, from input layer to output layer. The neurons in the hidden layer learn the pattern of input data during the training phase and map the relation between input and target class labels. To process the data that is received from input layer, neurons in the hidden layer uses a function called transfer function, and transfers the information after processing to the neurons in output layer [50].

But the learning mechanism of FFANN is not guaranteed with a global optimal solution. Different learning algorithms are used to train FFANN in order to avoid reaching a local optimal solution. In this work the popular back propagation learning algorithm is used to train the FFANN network. It has been widely used in many applications to train FFANN and is simple to implement [4].

2.4 Previous work

In recent times a lot of machine learning feature selection techniques are applied to reduce the redundant and unimportant features from brain MRI images. This is mainly done to improve the performance in classifying these images as tumorous and non-tumorous. Some of such studies which used feature selection to improve classification performance are discussed here.

Y. Zhang, and L. Wu, built a machine learning model able to classify MRI brain images as Normal and Abnormal, and incorporated dimensionality reduction technique in it. The data was collected from three open source databases such as from Harvard Medical School, OASIS dataset and ADNI dataset. The collected images as composed of T2 weighted MR brain images in axial plane with an in plane resolution of 256X256. The data corresponding to abnormal class consists of brain images that are deceased with Glioma, Meningioma, Alzheimer's disease, Alzheimer's disease plus visual agnosia, Pick's disease, sarcoma, and Huntington's disease. They performed a three level two dimensional Discrete Wavelet Transformation and extracted 1024 features out of the brain images. Later they used Principle Component Analysis (PCA) to reduce the dimensions of the data and selected the first 19 principle components. The selected components were treated as features and are fed into the Kernel Support Vector Machine where they used all the 3 type of kernels such as homogeneous, in homogeneous, and Gaussian kernels. The obtained classification performance has convincing results.[61]

A similar work is done by El-Sayed, Tamer and Salem. They collected 70 images of T2 weighted brain MRI images from Harvard Medical School. They created a model for classifying the brain MRI images into normal and abnormal classes. The feature extraction from raw brain

MRI images was performed using a single level 2 dimensional DWT, and later dimensionality reduction was done using PCA. The data with the reduced dimensions is fed into the Feed Forward Artificial Neural Network and K-Nearest Neighbors classifiers and the results were compared. It was stated that the results were convincing that the data with reduced dimensions has improved the classification performance.[7]

T. Suchada and N.P. Davies built a machine learning model to classify brain tumors in children. The data set was selected from Children's Cancer and Leukemia Group data base, which consisted of three different brain tumor types. The images were preprocessed using eddy current correction, skull stripping, Diffusion MR images reconstruction, registration, intensity normalization, segmentation and texture analysis. The texture features are extracted and upon which PCA and Maximum Relevance and Minimum Redundancy (mRMR) feed forward selection techniques are applied for feature selection. The SVM classifier with a linear kernel is applied to perform classification on selected and recombined features from mRMR and PCA respectively. The work stated that the better results were produced using texture features selected from mRMR feed forward selection technique than with PCA [55].

Another work where meta-heuristic algorithm is used for feature selection is done by Ahmed and Karim. They created a model to classify brain MRI images collected from Harvard Medical School into tumorous and non-tumorous classes. They used single level two dimensional DWT followed by spatial grey level dependence method (SGLDM) to extract 44 features. Later they used Binary GA as the feature selection technique with a binary encoded chromosome. The resultant feature were reduced to 5 and then used SVM with Gaussian radial basis function kernel to classify the dataset. The obtained results with feature selection was significantly improved compared to ones without using feature selection.[29]

Mehdi Jafari, Reza Shafaghi, performed a binary brain tumor classification using GA as feature selection. They collected MRI brain images from Harvard Medical School database and preprocessed the images to remove noise and enhance contrast. Later segmentation is done to remove nose, skull, eyes to extract portion of the brain. Four different categories of features are extracted

from the segmented images namely statistical features, features from three level DWT and PCA, features from the histogram of MRI brain images and combination of feature from all these sets. For feature selection GA is used with binary chromosome representation and 10 features are selected from 85 original features. The selected features are used to classify images into normal and abnormal with SVM classifier and the the stated results have improved the performance of classification[22].

K. Matsui, Y. Suganami has performed brain MRI classification, where classification is performed to differentiate the image into white and grey areas. They considered brain MRI images which are not related to tumors, and converted them to gray scale. Later 12 statistical features are extracted from the MRI image, and GA was applied to perform feature selection. Rather than using a classifier to evaluate the fitness of each chromosome inside population, they used vector quantified conditional class entropy (VQCCE) to evaluate the fitness of individuals. The best selected features from GA are provided to train the feed forward back propagation neural network to segment MRI image into white and gray areas. The results stated feature selection reduced the training time of neural networks to a greater extent [40].

A recent research was done by Arun Kumar, where he classified brain MRI images into normal and abnormal using a model with swarm intelligent based feature selection technique. He collected 354 images from BRATS-2015 dataset and performed de noising using biorth 3.7 wavelet filter. The improved images are segmented using Sobel edge detection and morphological operations for extracting the tumor part. These segmented images are used to extract 14 shape, intensity and texture statistical features. A well know and popular swarm intelligent algorithm called Binary PSO is used with a binary encoding of the particle to reduce the number of features to 6. He then used a simple SVM trained with a linear kernel to classify the images. He obtained a significant improvement in the performance of the classifier with the reduced feature set.[34]

V. Sheejakumari and B. S. Gomathi, worked on binary classification of brain MRI images into normal and pathological categories. They have manually collected MRI images from various diagnostic centers, and have segmented both the normal and abnormal images using skull stripping

method to extract the tissue part from the skull. Upon which they utilized gradient and histogram equalization methods on normal and abnormal images to segment the normal and edema portion of tissues respectively. In this work seven features with a combination of histogram, statistical and wavelet features are extracted from the brain image and an improved PSO is used to perform feature selection. They have modified the computation of velocity component of the particles by also considering the worst position of every particle. Using the selected features, feed forward artificial neural network with back propagation learning is used to perform classification of brain images. The results state that improved PSO has a faster convergence and improved accuracy and sensitivity values [51].

Another work using PSO as feature selection is performed by Atiq ur Rehman to perform binary classification of brain MRI images into normal and abnormal category. The data set consists of 20 brain MRI images collected from MRI Center at Ayub Medical Complex Abottabad Pakistan. The data set consists of 6 normal and 14 abnormal images. A total of 78 features using GLCM and DWT are extracted and PSO is used as feature selection for selecting best set of features. Two fitness functions namely SVM and KNN are used and 5 and 6 features are selected by PSO respectively. Using SVM with polynomial kernel with degree 5 and KNN is used for classification on both the selected features, and is observed that the feature selected using SVM fitness function is providing better results [47].

Research on feature selection using GWO in the field of MRI brain tumor classification is not much done, and hence, some of the less relevant literature where GWO has been used for feature selection in brain MRI classification is discussed here. In a most recent work by K. Nita and S. B. Kulkarni, GWO is used for optimising the extracted features to improve the brain hemorrhage classification. The considered data set for their work consists of 200 brain MRI images collected from SDM Medical College, Dharwad, and these belonged to 4 different hemorrhage categories. Preprocessing of images is performed by noise removal with edge and image enhancement. The images are segmented using multi level set algorithm and feature are extracted from local tetra patterns. The GWO is used for selecting the best subset of features and Relevance Vector Machine

(RVM) based classifier is used for classification of hemorrhages. The work stated that compromising results are produced with feature selection using GWO [25].

M. A. Ahmed and Y. Wei, performed feature selection using Chaotic Binary Grey Wolf Optimizer (CBGWO), for effectively diagnosing mild cognitive impairment (MCI) to prevent occurring of Alzheimer's decease (AD). The work used resting state functional MRI (rs-fMRI) brain images collected from the first affiliated hospital of Guangxi University of Chinese Medicine. The data set consists of 127 subsets out of which 62 belong to MCI class and 65 belongs to normal cognition (NC) class. Preprocessing of the brain images are performed by slicking, realigning, filtering, normalizing, smoothing and masking. The image segmentation is performed using fractional-ALFF method and PCA is applied to calculate the eigenvector matrix components, and the whole eigenvector is provided to CBGWO for feature selection. The chaos sequence maps are used to optimise the A and C parameters of GWO. Later using the selected features adaptive neuro fuzzy inference system (ANFIS) is used to perform classification of MCI/NC subjects using the optimised features selected. The results stated in the work have provided promising results with GWO [1].

CHAPTER III: METHODOLOGY

This chapter explains the adopted procedure in this research work. An brief description of the model considered for this research is shown in figure 3.1 and detail working explanation of each part is discussed later in this chapter.

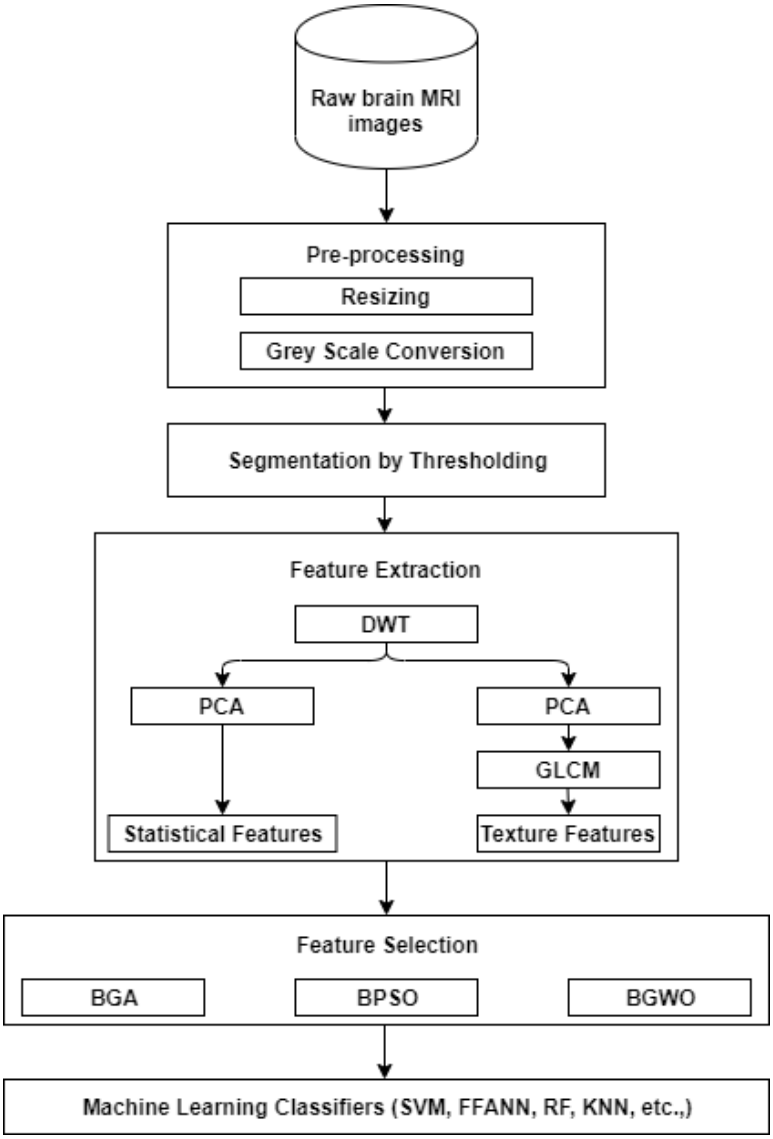


Figure 3.1: Brief Structure of model

Initially the data set consisting of the raw MRI brain images are taken and preprocessed. All the images are brought to a common resolution and are converted to grey scale. The gray scale images

are then segmented using the thresholding technique by applying a specific threshold luminescence value. It is done to extract the tumor part from the image. The segmented image is applied through feature extraction process where Discrete Wavelet Transformation is applied on to the image. The obtained feature matrix is thus needed to be compressed or reduced its dimension by applying Principle Component Analysis, after which 9 statistical features composed of are extracted from the resultant matrix and also GLCM is calculated from the PCA matrix and 4 texture features are calculated from it. The three meta-heuristic search algorithms, namely BGA, BPSO, and BGWO are used with different fitness functions and are applied on 13 different feature sets to select the best number of features. Later the selected feature subsets with different configuration of feature selection algorithms are used to perform binary classification using a set of 7 different classifiers and the results are compared with an evaluation criterion.

3.1 Dataset

The data set of brain MRI images for tumor detection is taken from the publicly available Kaggle data repository. The number of images are 253, out of which 155 are belongs to class ‘yes’, that means which consists of brain tumors and the other 98 belongs to the class ‘no’ which are healthy brain images. The image sizes are heterogeneous which range from 150X198 as the lowest up to 1920X1080 as the highest resolution, and these images are of different scales, some are of RGB and other are grey scale images. A sample of brain MRI images consisting of two different classes are shown in the figure below.

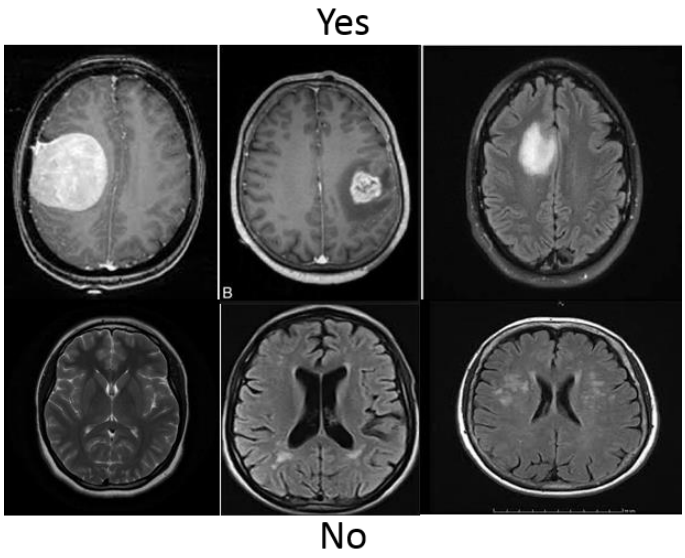


Figure 3.2: Sample images from dataset

3.2 Experimental Setup

The hyper parameters of the feature selection algorithms considered in this work are shown in table 3.1. The common configuration considered for all the feature selection algorithms are shown in table 3.2.Three configuration of fitness functions are used by the feature selection algorithms and are shown in table 3.3.

All the meta-heuristic feature selection algorithms are executed for 1000 iterations with an initial random population of 100 individuals and three different supervised classification algorithms namely KNN with two fold Cross validation, KNN with 10 fold cross validation and SVM using RBF kernel with 10 fold cross validation, are applied as fitness functions separately to evaluate the fitness of the individuals.

3.3 Methodology

The model is executed in three different phases as shown below. The research work can be explained in three steps as shown below.

Table 3.1: Hyper-parameters for feature selection algorithms

Binary Genetic Algorithm	
Crossover rate - CR	0.8
Mutation rate - MR	0.01
Binary Particle Swarm Optimization	
Cognitive Factor - C1	2
Social Factor - C2	2
W-min	0.4
W-max	0.9
V-min	-6
V-max	6
Binary Grey Wolf Optimizer	
a	Linearly Decreased from 2 - 0
Coefficient Vector - A	Random value in [-2a, 2a]
Coefficient Vector - C	2r2
Random vectors - r1,r2	Random value in [0 1]

Table 3.2: Common configuration of feature selection algorithms

Common Configuration	
Initial population size	100
Length of Individual	13
Encoding	Binary
Number of Iterations	1000
Number of Runs	33

Table 3.3: Fitness function configuration of feature selection algorithms

Fitness Functions		
KNN	k-fold	2
	k	5
KNN	k-fold	10
	k	5
SVM-rbf	k-fold	10

- Preprocessing and Feature Extraction
- Feature Selection
- Classification

The schematic diagram for the first phase of the work is displayed in Figure 3.3.

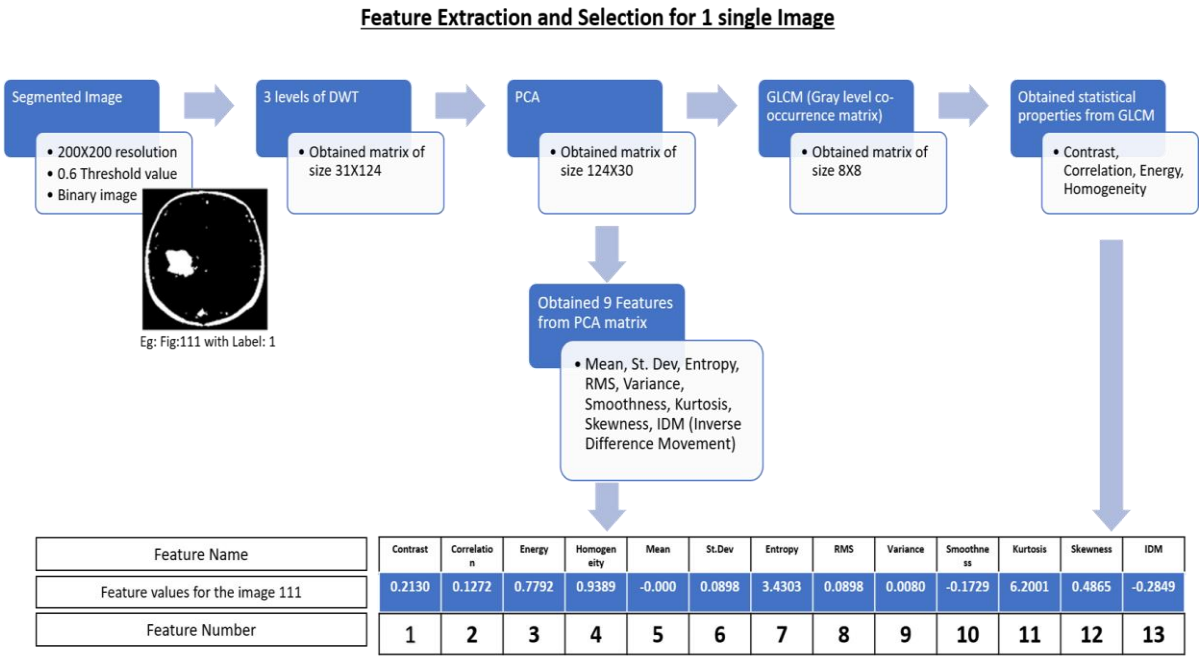


Figure 3.3: Preprocessing and Feature extraction for a single image

The figure describes the process that each image has to be passed through different stages for extracting the features out of it. Initially all the images are preprocessed in order to make them homogeneous.

Preprocessing

All the images are resized to a common resolution, and the modified resolution size is 200X200. Then all the images are converted to grey scale images and the segmentation is applied to the images. The thresholding technique is applied to the images, where a threshold luminescence value of '0.6' is applied to the converted grey scale images to differentiate the white and grey

matter of the brain MRI image. Later morphological operation is performed on the segmented image to clearly differentiate the tumor area from the other prats of the brain. The final image after segmentation can be observed from the above figure.

The preprocessing of the image is followed by the feature extraction, where necessary statistical information is extracted from the image. The feature extraction is divided into 3 steps and which are described in detail below.

Feature Extraction

Feature extraction is divided into 3 steps.

Discrete Wavelet Transformation

The transformation is applied to the image to reduce the resolution of the image or compressing the image by not losing the distinctive information from image. In this work Daubechies wavelet is used for performing the discrete wavelet transformation. When DWT is applied to an image 4 resultant images were obtained called as Horizontal approximation, and the details across 3 directions namely Vertical details, Horizontal details and Diagonal details are obtained which contain all the distinctive information form the image. This is called one level of DWT and this was performed 2 more times to extract the similar images, where the Horizontal approximation images obtained is passed to the next level of DWT. Finally the obtained 4 approximation and detail images are of size 31X31 individually and the 4 images with a combined resolution of 31X124 is passed to next step of feature extraction.

Principle component Analysis

The combined matrix for all 4 resultant images after 3 level DWT is passed to the PCA and a coefficient matrix for the input set is obtained. The resultant matrix from PCA is of size 124X30 and this is used for all the other calculations to extract the statistical features. From the obtained coefficient matrix following 9 features were extracted namely Mean, Standard deviation, Entropy,

Root mean square (RMS), Variance, Smoothness, Kurtosis, Skewness and Inverse Difference Moment (IDM).

Later the coefficient matrix is used to calculate the Grey Level Co-occurrence Matrix (GLCM) and some texture feature are extracted from the image.

Grey Level Co-occurrence Matrix

From the probability of joint occurrence between two different kinds of intensities at a point in picture the texture measurements are calculated. The joint occurrence means that two objects need to occur simultaneously in the image. Later the four properties of the GLCM which are nothing but the texture measurements are calculated from the image.

Property	Description	Formula
Contrast	Intensity contrast between a pixel and its neighbor	$\sum_{i,j} \lvert i-j \rvert^2 p(i,j)$
Correlation	Correlation between a pixel and its neighbor (μ denotes the expected value, σ denotes the standard variance)	$\sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)\, p(i,j)}{\sigma_i \sigma_j}$
Energy	Energy of the whole image	$\sum_{i,j} p(i,j)^2$
Homogeneity	Closeness of the distribution of GLCM to the diagonal.	$\sum_{i,j} \frac{p(i,j)}{1+\lvert i-j \rvert}$

Figure 3.4: Texture Features from Gray Level Co-occurrence Matrix

The ‘i’ and ‘j’ are the pixels which are being compared and p(i,j) is the joint occurrence of the pixels from the GLCM matrix.

These 13 features corresponding to an image is stored in a vector and this process is continued for all the 253 images and feature matrix is prepared as shown in the below figure.

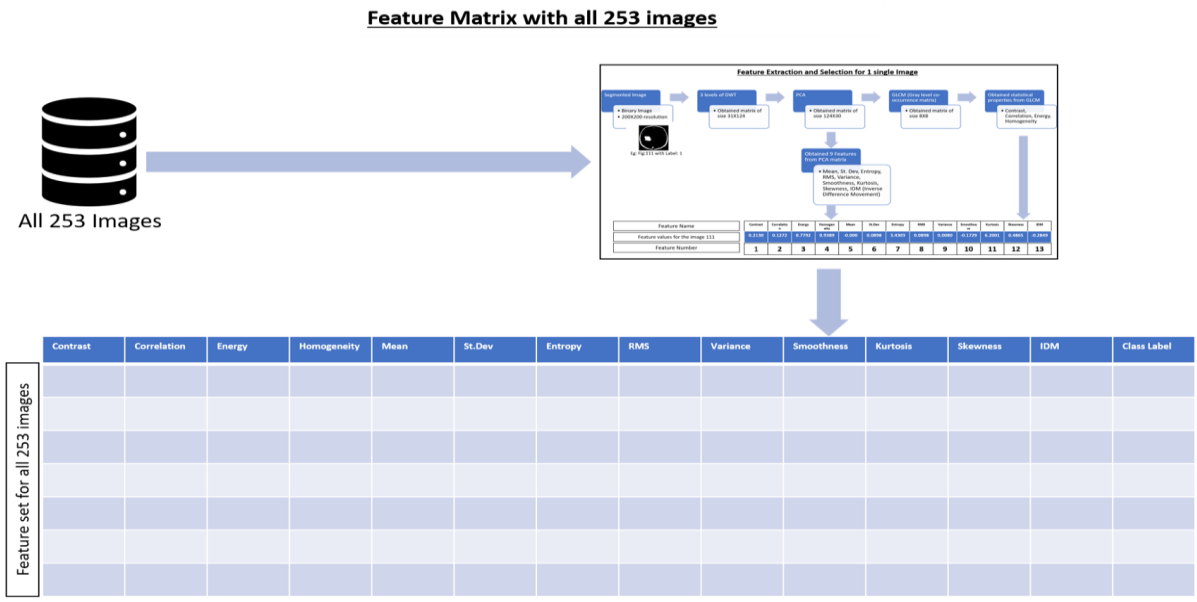


Figure 3.5: Accumulating all features into a feature set

Feature Selection

The schematic diagram of Feature Selection in the work is shown below.

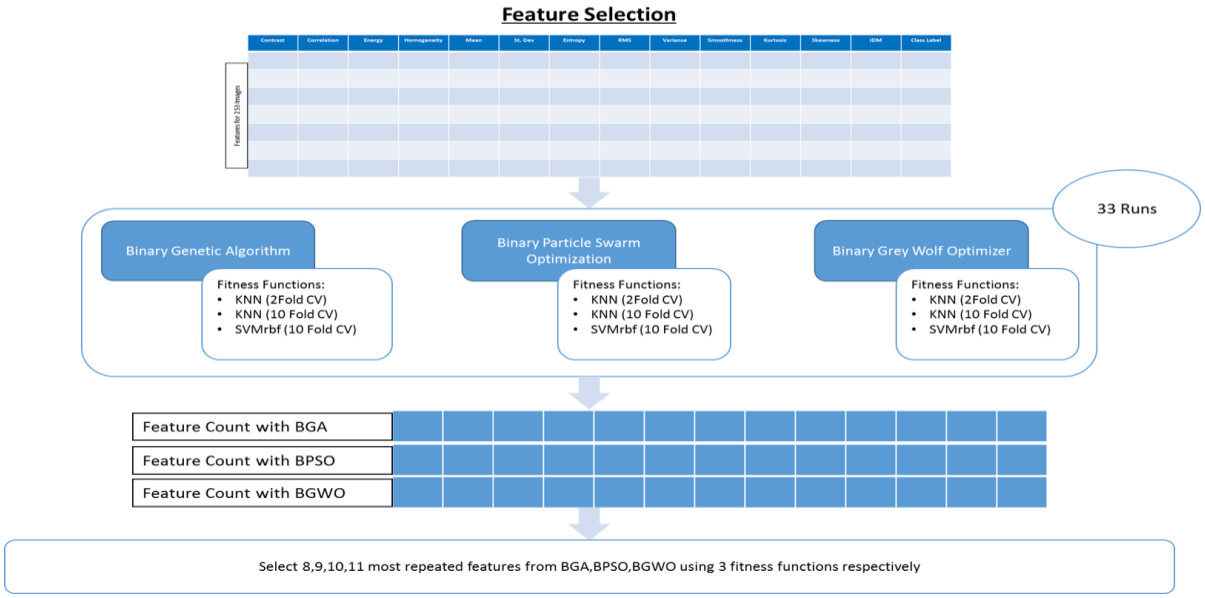


Figure 3.6: Feature Selection

Three meta-heuristic algorithms are applied on the extracted feature set with 13 features, to perform feature selection. These are:

- Binary Genetic Algorithm
- Binary Particle Swarm Optimization
- Binary Grey Wolf Optimizer

These algorithms are considered binary because of the binary encoded representations of the candidate solutions in the population. These feature selection algorithms are iterated for 1000 iterations with a randomly generated population size of 100. Over the course of iterations, the individual solutions in the population are updated to reach an optimal solution with a different best feature subset obtained for each algorithm. These algorithms are executed for 33 times each and a resultant vector is calculated at the end which holds the number of times each feature repeated across the 33 runs. This resultant vector is represented as Feature count vector for each BGA, BPSO, BGWO respectively in Figure:3.6. Therefore there will be 3 sets of such feature count vectors which were generated using different fitness functions, adding up to a total of 9 feature count vectors.
From these vectors 4 different subsets of features are calculated for each algorithm executed with 3 different variations of fitness functions. These feature sets are selected using 8,9,10 and 11 most repeated features from the feature count vectors. Therefore a total of 36 different feature sets were selected and along with the complete feature set, the classification algorithms were applied on them to compare the performances.

Classification

The schematic diagram of Classification part of the work is shown below.

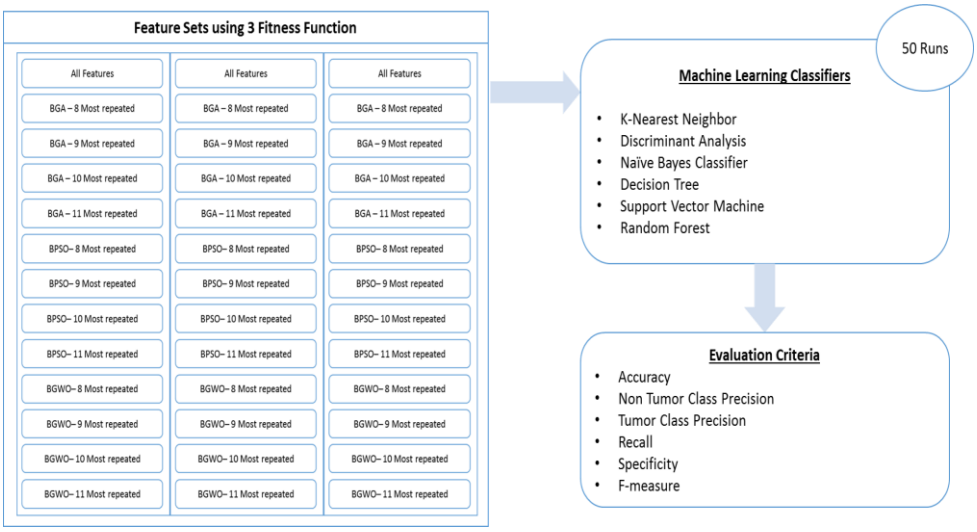


Figure 3.7: Classification

Classification on the full extracted feature set with 13 features and reduced feature sets obtained with feature selection is performed using six different classification techniques. The following are the classifiers with the variations used in this work:

- K-Nearest Neighbor
- Discriminant Analysis (Discriminant types: Pseudo Linear, Diag Linear)
- Naive Bayes classifier (Distribution types: Normal, Kernel)
- Decision tree
- Support Vector Machine (Kernel types: Linear, Quadratic,Gaussian Radial Basis Function (RBF))
- Random Forest

Binary classification is done where the images are classified into two different categories, one with containing tumors and other without tumors.

Evaluation Criteria

The following evaluation criteria is considered for measuring the performance of classifier.

- Accuracy: It is calculated as the number of all correct predictions divided by the total number data in the dataset.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3.18}$$

- Recall/Sensitivity: Defined as the total number of correct positive predictions divided by the total number of positives. In this work the positive class is the Non tumor class.

$$Recall = \frac{TP}{TP + FN} \tag{3.19}$$

- Specificity: Defined as the total number of correct negative predictions divided by the total number of negatives. In this work the negative class is the tumor class.

$$Specificity = \frac{TN}{TN + FP} \tag{3.20}$$

- Non tumor class precision (NTCP): It is calculated as the number of correct non tumor class predictions divided by the total number of non tumor class predictions.

$$Nontumorclassprecision = \frac{TP}{TP + FP} \tag{3.21}$$

- Tumor class precision (TCP): It is calculated as the number of correct tumor class predictions divided by the total number of tumor class predictions.

$$Tumorclassprecision = \frac{TN}{TN + FN} \tag{3.22}$$

- F-measure: It is defined as the harmonic mean of precision and recall.

$$F\_measure = \frac{2 \cdot NTCP \cdot Recall}{NTCP + Recall} \tag{3.23}$$

CHAPTER IV: FEATURE SELECTION RESULTS AND ANALYSIS

This chapter explain and analyze the results obtained using three meta-heuristic feature selection algorithms used in this work. Initially, a combination of 13 different intensity and texture features are extracted from the brain MRI medical images, using three levels of DWT and PCA and then applying GLCM. A snapshot of the feature set obtained is show in below table.

Table 4.4: A snap shot of extracted feature set

Contrast	Correlation	Energy	Homogeneity	Mean	St.Dev	Entropy	RMS	Variance	Smoothness	Kurtosis	Skewness	IDM	Class Label
0.344	0.101	0.802	0.941	0.005	0.090	2.263	0.090	0.008	0.949	15.080	1.459	0.254	0
0.334	0.100	0.782	0.937	0.005	0.090	2.811	0.090	0.008	0.952	19.909	1.924	-0.295	1
0.520	0.132	0.912	0.970	0.006	0.090	0.844	0.090	0.008	0.954	58.261	5.212	4.479	1
0.256	0.063	0.748	0.929	0.002	0.090	3.541	0.090	0.008	0.898	6.830	0.655	0.507	0
0.511	0.074	0.895	0.967	0.006	0.090	1.341	0.090	0.008	0.955	56.937	5.110	4.907	1
0.434	0.063	0.824	0.949	0.006	0.090	2.199	0.090	0.008	0.956	39.382	3.719	3.150	0
0.312	0.114	0.797	0.942	0.003	0.090	2.404	0.090	0.008	0.928	16.145	1.523	0.052	1
0.345	0.087	0.795	0.939	0.005	0.090	2.681	0.090	0.008	0.950	12.680	1.375	1.137	0
0.264	0.159	0.759	0.934	0.004	0.090	2.981	0.090	0.008	0.940	10.986	1.113	1.637	1
0.248	0.057	0.756	0.930	0.003	0.090	3.564	0.090	0.008	0.919	6.442	0.564	-0.078	1

Each row of the table represents a brain MRI image form the data set, and the last column in the table represents the class that the image belongs to. The records with class label 0, represents those images that belongs to non tumor class category, where as the records with class label 1, represents those images that belongs to tumor class category. Each column in the table other than class label represents the attribute or feature of the image that has been extracted.

Later, feature selection is performed on the complete feature set using 3 different fitness configurations by BGA, BPSO and BGWO algorithms. These algorithms are executed for 33 times and produced a feature count vector, which consists of the number of times each feature index is selected out of the 33 runs. Based on the frequency of feature indexes in the feature count vectors, feature subsets are selected with a length of 8, 9, 10 and 11. The following sections explain the results using feature selection.

Convergence Curves

The convergence curves obtained using BGA, BPSO, BGWO with the three fitness configurations namely - two configurations of KNN classifier with 2 fold and 10 fold cross validations, SVM classifier with RBF kernel using 10 fold cross validations, are shown in below figures.

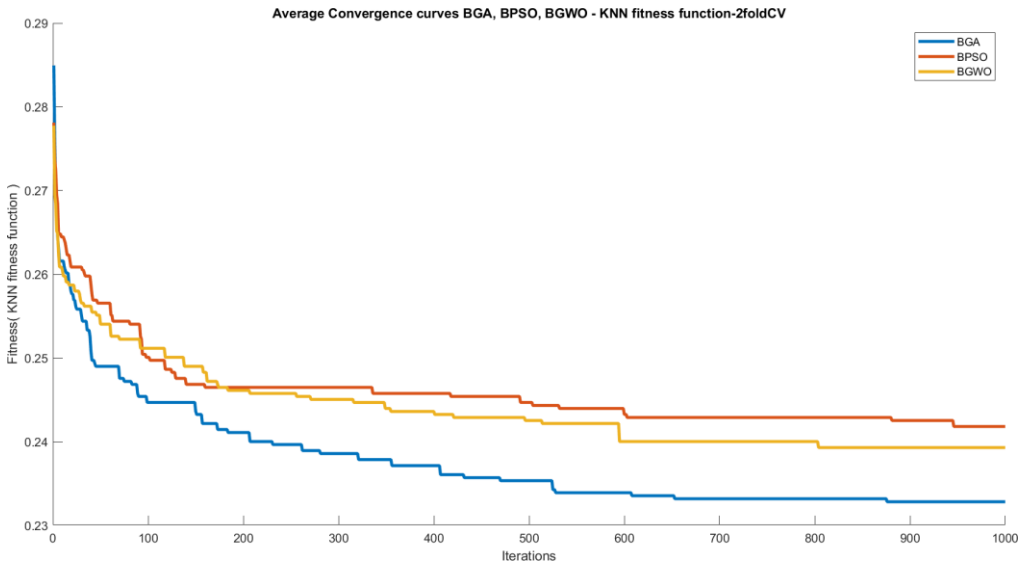


Figure 4.1: Convergence curves of BGA, BPSO, BGWO using KNN classifier with 2 fold cross validation as fitness function

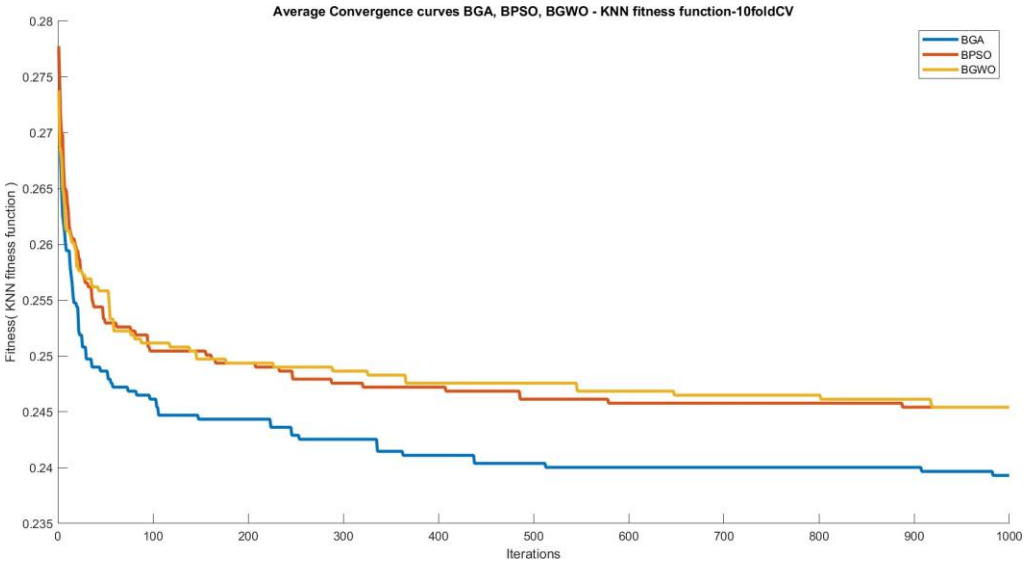


Figure 4.2: Convergence curves of BGA, BPSO, BGWO using KNN classifier with 10 fold cross validation as fitness function

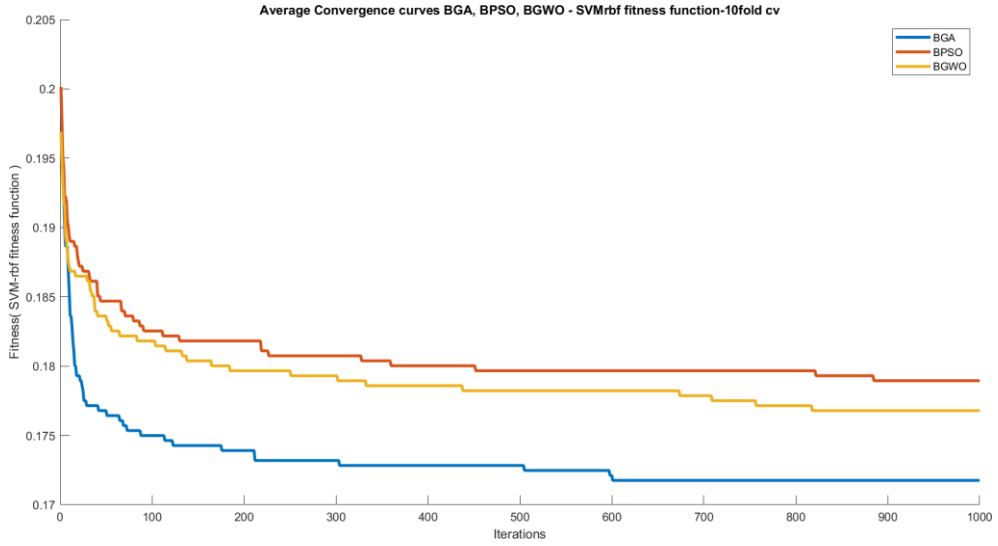


Figure 4.3: Convergence curves of BGA, BPSO, BGWO using SVM-RBF classifier with 10 fold cross validation as fitness function

Each of the feature selection algorithm is executed for 33 runs, where in each run the algorithm performs 1000 iterations. Figure 4.1 shows the average convergence curves for BGA, BPSO and BGWO represented in blue, red and yellow colors respectively. These algorithms used KNN clas-

sifier with 2 fold cross validation as fitness function. The x-axis represents the iteration number, and y-axis represents the average fitness value for the whole population at that particular iteration. The fitness value is KNN classification loss which corresponds to each individual or candidate solution in the population. The reduction in the fitness value over the course of iterations, signifies the approach to a near global optimal solution. In this case the minimum fitness obtained with BGA is less than BGWO which is in turn less than BPSO.

The figure 4.2 represents the convergence of BGA, BPSO and BGWO using the fitness function as KNN classifier with 10 fold cross validation. In this case, the minimum fitness obtained with BGA is lower than with BGWO and BPSO. The third figure 4.3 represents the convergence curves generate BGA, BPSO and BGWO with SVM classifier with RBF kernel and with 10 fold cross validation as fitness function. Here the minimum fitness obtained with BGA is lower than BGWO which is in turn lower than BPSO. Therefore, it is observed that the minimum fitness value obtained by all the meta-heuristic feature selection algorithms with SVM fitness configuration is lower than other fitness functions.

Feature Count Vectors

Upon executing the three feature selection algorithms, the feature count vector is obtained for each algorithm with each fitness function. The features are selected differently by each algorithm and the selection is also different based on the fitness function used. The distribution of the features selected by the algorithms with each fitness function is show in figure 4.4.

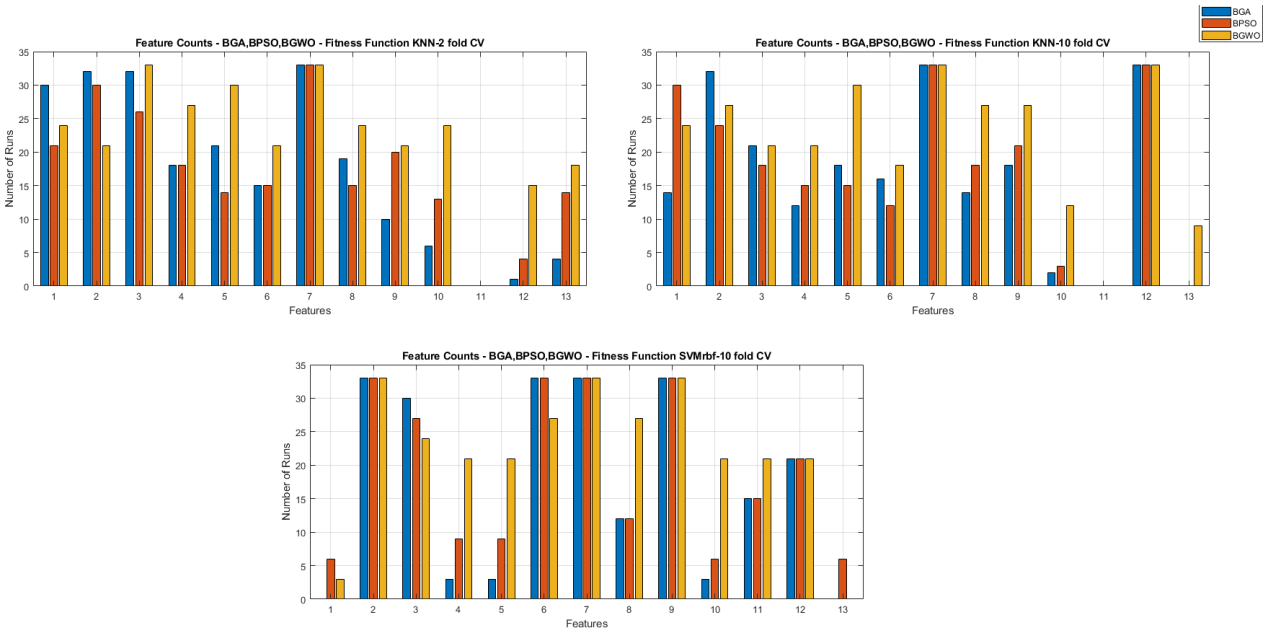


Figure 4.4: Distribution of features selected with BGA, BPSO, BGWO with three functions KNN with 2 fold CV, KNN with 10 fold CV, SVM-RBF with 10 fold CV respectively

From the figure 4.4, it is observed that, nearly 8 features are repeated for each algorithm for more than 50% of the times out of the 33 runs executed. Therefore, 8 most repeated features are selected and along with that 9,10 and 11 most repeated features sets are also selected from each feature count vector. The selected indexes form these algorithms are shown in table 4.5.

Table 4.5: Selected feature subsets using 3 fitness variations of feature selection algorithms

Fitness Function	Feature Subset	Selected Indexes										
knn2CV	**BGA-top8**	1	2	3	4	5	6	7	8			
	BGA-top9	1	2	3	4	5	6	7	8	9		
	BGA-top10	1	2	3	4	5	6	7	8	9	10	
	BGA-top11	1	2	3	4	5	6	7	8	9	10	13
	BPSO-top8	1	2	3	4	6	7	8	9			
	BPSO-top9	1	2	3	4	5	6	7	8	9		
	BPSO-top10	1	2	3	4	5	6	7	8	9	13	
	BPSO-top11	1	2	3	4	5	6	7	8	9	10	13
	BGWO-to8	1	2	3	4	5	7	8	10			
	BGWO-top9	1	2	3	4	5	6	7	8	10		
	BGWO-top10	1	2	3	4	5	6	7	8	9	10	
	BGWO-top11	1	2	3	4	5	6	7	8	9	10	13
knn10CV	**BGA-top8**	1	2	3	5	6	7	9	12			
	BGA-top9	1	2	3	5	6	7	8	9	12		
	BGA-top10	1	2	3	4	5	6	7	8	9	12	
	BGA-top11	1	2	3	4	5	6	7	8	9	10	12
	BPSO-top8	1	2	3	4	7	8	9	12			
	BPSO-top9	1	2	3	4	5	7	8	9	12		
	BPSO-top10	1	2	3	4	5	6	7	8	9	12	
	BPSO-top11	1	2	3	4	5	6	7	8	9	10	12
	BGWO-to8	1	2	3	5	7	8	9	12			
	BGWO-top9	1	2	3	4	5	7	8	9	12		
	BGWO-top10	1	2	3	4	5	6	7	8	9	12	
	BGWO-top11	1	2	3	4	5	6	7	8	9	10	12
SVM-rbf10CV	**BGA-top8**	2	3	6	7	8	9	11	12			
	BGA-top9	2	3	4	6	7	8	9	11	12		
	BGA-top10	2	3	4	5	6	7	8	9	11	12	
	BGA-top11	2	3	4	5	6	7	8	9	10	11	12
	BPSO-top8	2	3	6	7	8	9	11	12			
	BPSO-top9	2	3	4	6	7	8	9	11	12		
	BPSO-top10	2	3	4	5	6	7	8	9	11	12	
	BPSO-top11	1	2	3	4	5	6	7	8	9	11	12
	BGWO-to8	2	3	4	5	6	7	8	9			
	BGWO-top9	2	3	4	5	6	7	8	9	10		
	BGWO-top10	2	3	4	5	6	7	8	9	10	11	
	BGWO-top11	2	3	4	5	6	7	8	9	10	11	12

CHAPTER V: CLASSIFICATION RESULTS AND ANALYSIS

This chapter explains the results that were obtained using classification algorithms considered in this work. The extracted features and selected feature sets are used for binary classification, where the data is classified into two classes, one containing brain tumor MRI images and other containing normal brain MRI images.

For the binary classification seven different classifiers were used namely K-Nearest Neighbor (KNN), Discriminant Analysis (DA) with two discriminant types such as pseudo linear and diag linear, Decision Tree (DT), Naïve Bayes (NB) with two different distribution types such as normal and kernel type, Support Vector Machines (SVM) with 3 different kernels such as linear kernel, quadratic polynomial kernel and Gaussian radial basis kernel (RGB), Random Forest (RF) and Feed Forward Artificial Neural Network (FFANN) classifiers.

Data was partitioned using Hold out method with a configuration of 70% for training and 30% for testing to train and validate the classifiers. By applying the hold out method every time, the classifiers are executed for 50 times on the selected feature sets and the performance is compared with 6 evaluation metrics namely: Accuracy, Non Tumor Class precision, Tumor Class Precision, Recall, Specificity, and F-measure.

Initially the classifiers are applied on the complete feature set that are extracted and the corresponding average results for the 11 classifiers with 6 evaluation metrics are show in below table.

Table 5.6: Classification results on all the features

Classifier	Accuracy	Recall	Specificity	NT PR	T PR	F-measure
KNN	69.697	64.83	72.268	49.53	82.411	55.535
DA(pseudo-linear)	72.485	71.259	73.21	49.216	87.154	57.86
DA(diag-linear)	66.182	58.441	69.38	43.26	80.632	49.452
NB(Normal)	67.636	63.115	69.106	38.871	85.771	47.757
NB(Kernel)	67.758	61.36	70.138	43.574	83.004	50.698
DT	67.03	58.11	72.689	55.172	74.506	56.225
SVM(Linear)	69.867	81.936	68.625	31.094	94.412	42.877
SVM(RBF)	90.552	98.668	87.112	76.68	99.333	86.223
SVM(Quadratic)	54.392	48.666	56.122	46.904	59.133	44.619
RF	74.667	72.738	76.086	57.367	85.573	63.77
FFANN	63.281	60725	64.4999	31.72	83.522	40.279

From the above table it is observed that, the SVM classifier with the RGB kernel has higher performance in all the evaluation metrics compared to any other classifiers. A combination of linear and non linear classification algorithms are considered in this work and their performance on different feature subsets extracted using linear fitness function and non linear fitness function are discussed going on. The later sections explain the classifiers performance on different feature subsets using each evaluation metric.

5.1 Accuracy

Accuracy is an overall performance metric of the classifier. It shows how well it has performed in categorizing the samples in the data set. The classification algorithms considered in this work are executed for 50 times on each feature subset, and the resulting mean accuracies of each classifier on each feature subset in shown in below table.

Table 5.7: Accuracy of all classifiers with 13 feature subsets from BGA, BPSO, BGWO using 3 different fitness functions

Accuracy	AF	BGA8	BGA9	BGA10	BGA11	BPSO8	BPSO9	BPSO10	BPSO11	BGWO8	BGWO9	BGWO10	BGWO11
K-Nearest Neighbor													
knn2cv	67.3	66.8	66.8	66.7	66.9	66.8	66.8	67.8	66.9	66.7	66.7	66.7	66.9
knn10cv	67.3	69.7	69.7	69.7	70.0	69.7	69.7	69.7	70.0	69.7	69.7	69.7	70.0
svmrbf10cv	67.3	65.7	65.7	65.7	65.7	65.7	65.7	65.7	65.7	65.7	65.7	65.2	65.7
Discriminant Analysis - Pseudo Linear Discriminant type													
knn2cv	67.9	68.6	68.2	68.7	67.4	68.0	68.2	68.0	67.4	67.5	68.3	68.7	67.4
knn10cv	67.9	67.9	67.9	67.9	67.4	67.7	68.3	67.9	67.4	66.1	68.3	67.9	67.4
svmrbf10cv	67.9	67.0	67.6	67.8	68.7	67.0	67.6	67.8	67.5	69.8	70.3	68.9	68.7
Discriminant Analysis - Diag Linear Discriminant type													
knn2cv	64.2	64.2	64.6	64.6	64.2	64.6	64.6	63.7	64.2	64.6	64.4	64.6	64.2
knn10cv	64.2	63.9	63.9	64.4	64.4	64.7	64.8	64.4	64.4	64.6	64.8	64.4	64.4
svmrbf10cv	64.2	64.2	64.4	64.4	64.4	64.2	64.4	64.4	64.2	64.4	64.6	64.4	64.4
Naive Bayes- Normal Distribution													
knn2cv	67.2	65.1	66.2	64.3	65.5	65.9	66.2	67.0	65.5	66.2	64.5	64.3	65.5
knn10cv	67.2	65.4	65.0	65.0	64.2	64.3	65.2	65.0	64.2	65.9	65.2	65.0	64.2
svmrbf10cv	67.2	65.7	65.2	65.4	63.7	65.7	65.2	65.4	66.8	64.8	62.1	64.1	63.7
Naive Bayes - Kernel Distribution													
knn2cv	63.5	64.6	64.1	63.7	64.7	64.3	64.1	65.0	64.7	64.6	63.9	63.7	64.7
knn10cv	63.5	62.8	62.8	63.7	63.0	63.6	63.0	63.7	63.0	62.7	63.0	63.7	63.0
svmrbf10cv	63.5	62.1	62.8	63.0	63.4	62.1	62.8	63.0	62.5	65.4	64.8	63.7	63.4
Decision Tree													
knn2cv	66.1	67.3	67.5	67.5	66.8	69.6	67.5	66.8	66.8	67.3	67.3	67.5	66.8
knn10cv	66.1	68.3	68.3	65.9	65.9	63.4	65.9	65.9	65.9	68.3	65.9	65.9	65.9
svmrbf10cv	66.1	67.9	68.0	67.4	67.4	67.9	68.0	67.4	66.9	68.7	68.7	67.5	67.4
Support Vector Machine - Linear Kernel													
knn2cv	69.0	71.2	71.2	71.4	69.3	71.2	71.2	69.2	69.3	71.4	71.4	71.4	69.3
knn10cv	69.0	69.4	69.4	69.4	69.5	69.6	69.4	69.4	69.5	69.4	69.4	69.4	69.5
svmrbf10cv	69.0	68.8	68.8	68.8	69.0	68.8	68.8	68.8	68.8	70.3	70.3	67.7	69.0
Support Vector Machine - Radial Basis Function Kernel													
knn2cv	91.1	90.6	90.5	91.1	90.0	90.6	90.5	90.1	90.0	91.1	91.1	91.1	90.0
knn10cv	91.1	91.2	91.2	91.3	92.0	91.3	91.3	91.3	92.0	91.2	91.3	91.3	92.0
svmrbf10cv	91.1	91.6	91.6	91.6	91.6	91.6	91.6	91.6	91.6	90.3	90.1	90.6	91.6
Support Vector Machine - Quadratic Kernel													
knn2cv	60.2	64.1	65.8	60.4	73.7	56.6	65.8	72.7	73.7	56.0	71.7	60.4	73.7
knn10cv	60.2	72.2	72.2	71.0	74.0	73.9	68.9	71.0	74.0	74.2	68.9	71.0	74.0
svmrbf10cv	60.2	52.4	57.1	58.4	63.7	52.4	57.1	58.4	59.3	45.6	45.3	59.6	63.7
Random Forest													
knn2cv	73.5	72.8	72.2	74.5	74.0	71.3	72.2	73.4	74.0	73.0	73.4	74.5	74.0
knn10cv	73.5	72.3	74.0	73.1	74.2	70.8	74.6	73.1	74.2	73.9	74.6	73.1	74.2
svmrbf10cv	73.5	74.7	73.0	73.6	74.3	74.7	73.0	73.6	74.6	73.6	72.2	73.5	74.3
Feed Forward Artificial Neural Network													
knn2cv	63.3	66.0	63.0	64.7	59.8	65.1	65.2	62.8	64.6	64.6	60.8	61.4	60.6
knn10cv	63.3	64.2	64.1	64.7	62.7	65.3	63.9	65.9	63.4	63.8	65.7	62.9	64.9
svmrbf10cv	63.3	64.6	63.4	65.7	62.7	66.4	66.6	64.7	64.3	64.8	65.4	66.2	64.7

Each record in the above table consists of mean accuracies obtained using a specific classifier with 13 different feature subsets. The feature set with total number of extracted feature are also considered among the 13 subsets for comparison. The rest 12 feature subsets are obtained by the three meta-heuristic feature selection algorithms, with 3 different fitness functions. From the feature count vectors obtained with the feature selection algorithms, eight best features which repeated for more than 50% of the runs, and with adding subsequent best features namely nine,

ten and eleven best features are selected. These feature subsets are represented as BGA8, BGA9, BGA10, BGA11 i.e., the ones extracted using BGA algorithm. Similarly the rest are extracted using BPSO and BGWO algorithms. The fitness functions used in the research are KNN classifier with 2 fold cross validation (knn2cv), KNN classifier using 10 fold cross validation (knn10cv), SVM classifier with RBF kernel using 10 fold cross validation (svmrbf10cv).

To visualize the mean accuracy values of the classifiers, a scatter plot is plotted for each classifier. The scatter plots are divided into three different color groups for each classifier. The classifier accuracy obtained with feature subsets generated from knn2cv fitness function are represented in black, where the accuracy from knn10cv features are represented in green, and the ones from svmrbf10cv are represented in red. The mean accuracy scatters plots of all the classifiers are shown in the below figure.

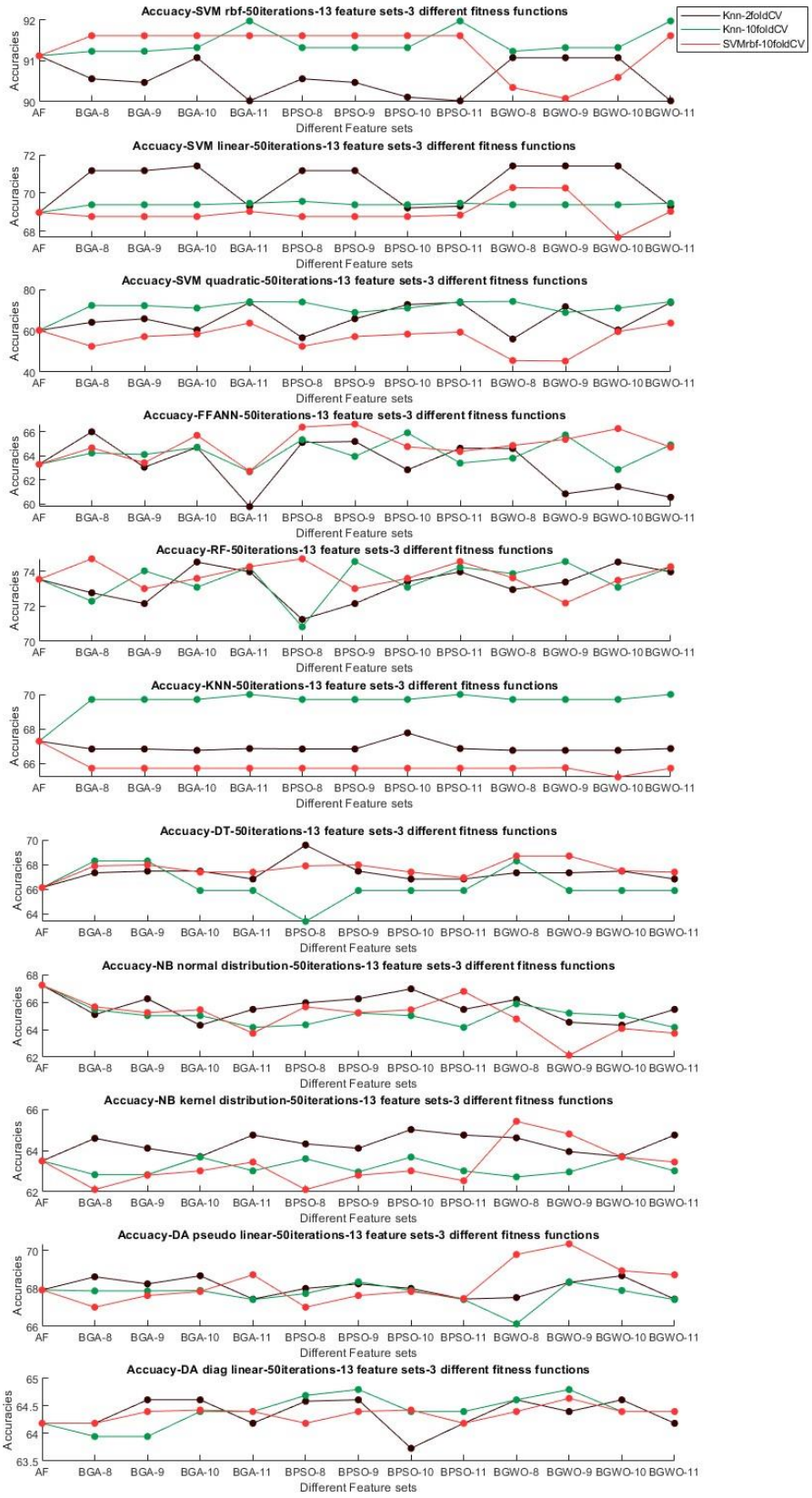
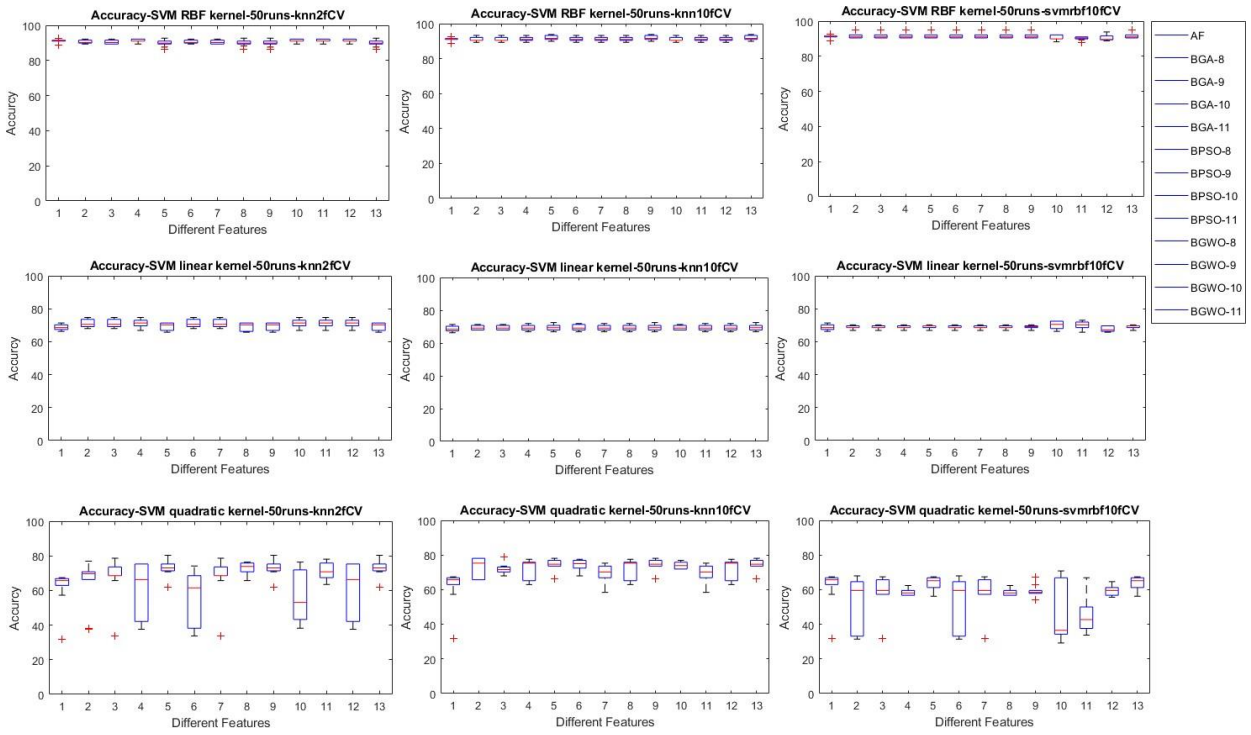


Figure 5.1: Scatter plot with mean accuracy of classifiers with 13 feature subsets from BGA,BPSO,BGWO using 3 different fitness functions

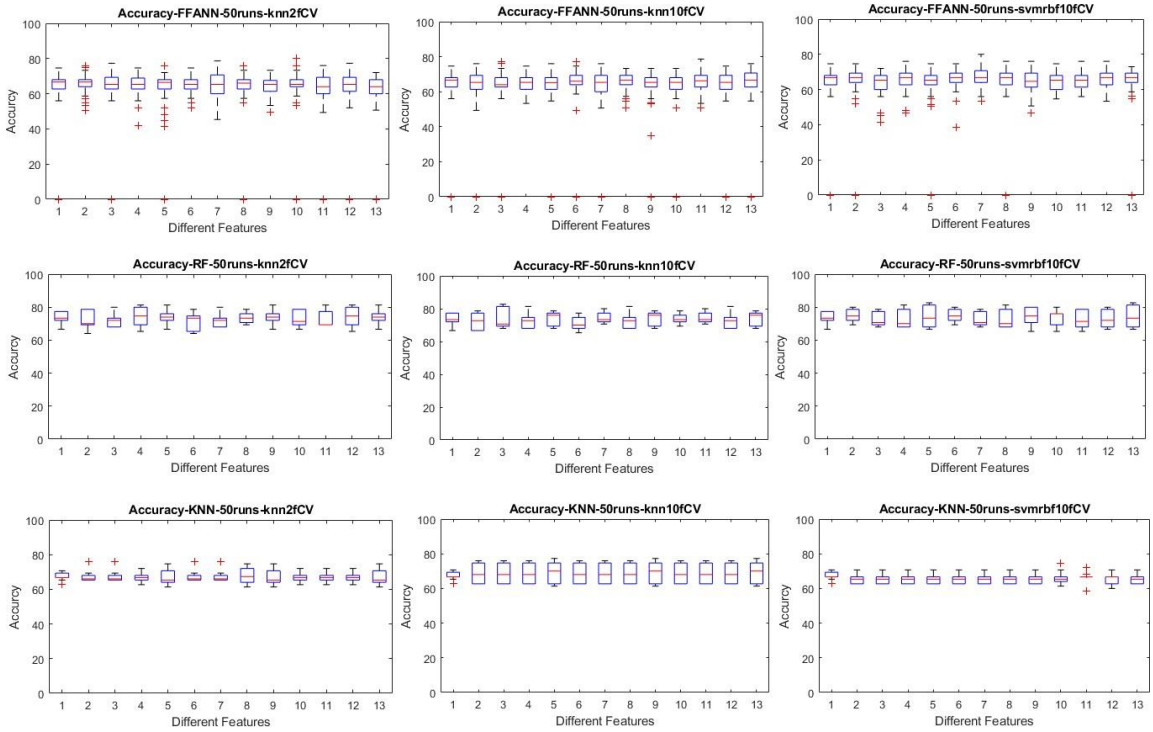
The lines connecting the scatter points in the figure 5.1 doesn't represented any continuous data. The lines are just plotted to observed the trend of the classifier on different feature subsets. The first point for each classifier is the mean accuracy obtained with all features, and the rest 12 points with same color are the mean classifier accuracies with selected feature subsets with different fitness functions.

The behaviour of the classifiers are different on the feature sets. The performance of the classifiers using features obtained with svmrbf10cv fitness function are better for non linear and statistical classifiers such as: SVM with RBF kernel, FFANN, RF, DT, NB with both data distribution types. Where as, for less quadratic and linear classifiers like KNN, SVM with linear and quadratic kernel, DA with both the discriminant types the performance of using feature obtained with knn10cv fitness function are better. The performance is also varied based on the number of feature selected by each feature selection algorithm. The performance of the SVM with quadratic kernel, RF, NB with kernel distribution, DA with pseudo linear discriminant classifiers is increased when higher number of features are selected using BGA and PSO algorithms. For the remaining classifiers the performance using the features from BGA and BPSO are mostly constant and with slight alternating variation. With the feature subsets selected using BGWO feature selection, non linear classifiers like SVM with linear and quadratic kernel, FFANN, RF and DT are showing increased performance with more number of selected features. But with the same feature subsets from BGWO the performance is either constant or decreasing using linear classifiers.

The classifiers are executed for 50 runs on each of the 13 different feature subsets from each feature selection algorithm. To observe the distribution of the obtained accuracy box plots are generated displaying the accuracy values for 50 runs for each classifier. The box plots for accuracy are shown in figure 5.2.

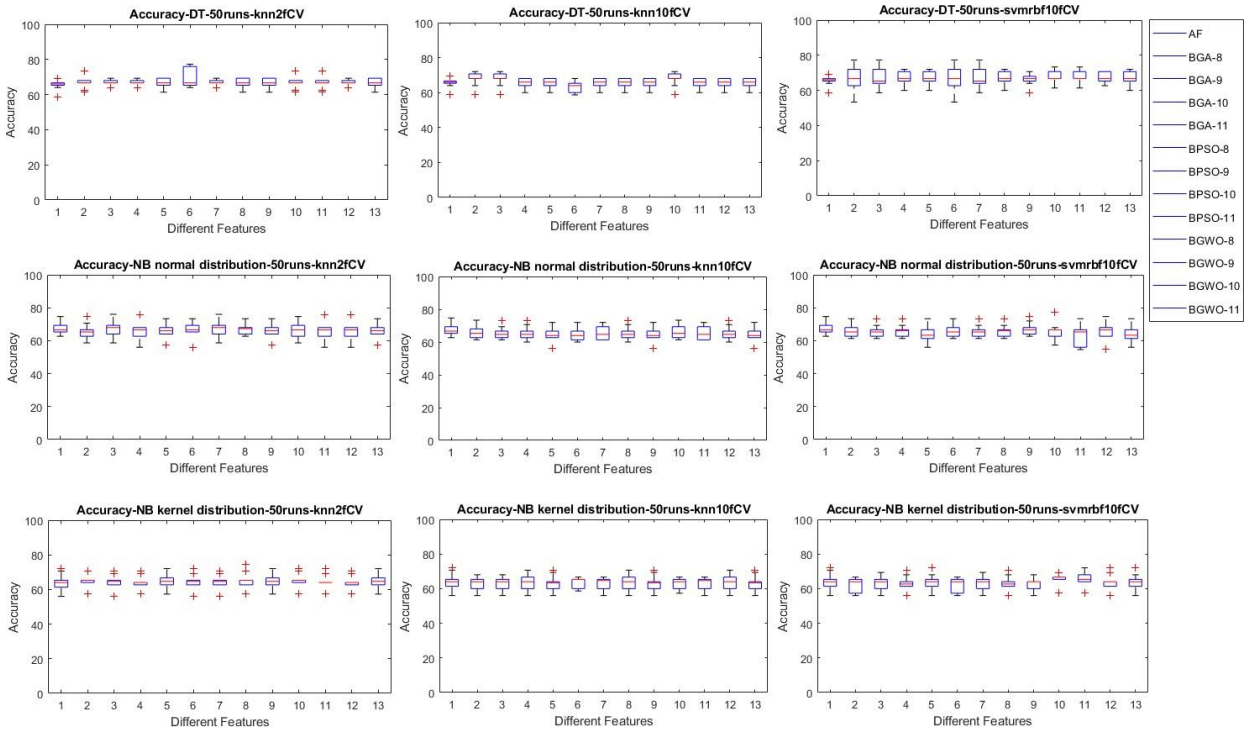


(a) Accuracy of SVM-rbf, SVM-linear, SVM-quadratic classifiers

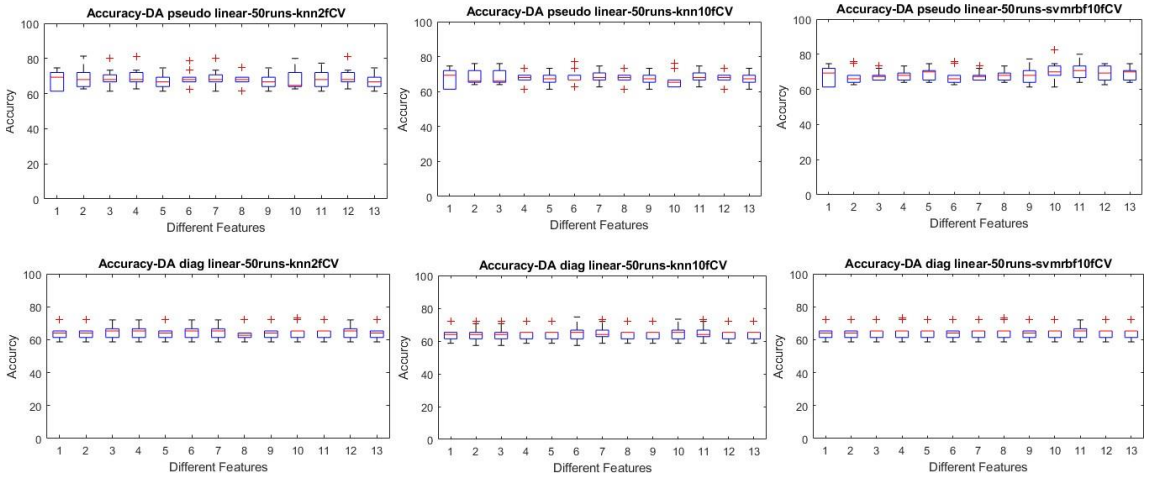


(b) Accuracy of FFANN, RF, KNN classifiers

Figure 5.2: Accuracy of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions



(c) Accuracy of DT, NB-nornal, NB-kernel classifiers



(d) Accuracy of DA pseudo linear,DA diag linear classifiers

Figure 5.2: Accuracy of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions

The classifiers provided good confidence results when using features subsets that are selected with svmrbf10cv fitness function. And the performance with features selected by BGA and BPSO are either altering or constant, but where as for most of the classifiers the values are more confident with the increase in size of feature subset selected by BGWO.

5.2 Non Tumor Class Precision

Non tumor class precision is a measure that tell us what proportion of images that are predicted as not having tumor, does actually not have tumor. In this work the images that belong to non tumor class category are represented as positive class.

The classifiers are iterated for 50 times using different feature sets selected by BGA, BPSO and BGWO with three different fitness functions, and the mean non tumor class precision obtained in each case is shown in table 5.8. The non tumor class is represented as NTCP in the table.

The values obtained with SVM classifier with RBF kernel are higher than any other classifier, but with in the classifier the performance when using the feature subsets selected from knn10cv fitness function is higher than the performs with all features, and for the rest the performance is slightly lower than with the all feature set.

To visualize the non tumor class precision results, scatter plots are plotted for the mean values shown in table 5.8. Figure 5.3 shows the mean non tumor class precision scatter plots for all the classifiers with the complete feature set as well as with the different selected feature subsets.

Table 5.8: Non Tumor Class Precision of all classifiers with 13 feature subsets from BGA, BPSO, BGWO using 3 different fitness functions

NTCP	AF	BGA8	BGA9	BGA10	BGA11	BPSO8	BPSO9	BPSO10	BPSO11	BGWO8	BGWO9	BGWO10	BGWO11
K-Nearest Neighbor													
knn2cv	47.6	48.2	48.2	47.6	49.2	48.2	48.2	49.2	49.2	47.6	47.6	47.6	49.2
knn10cv	47.6	54.7	54.7	54.7	53.7	54.7	54.7	54.7	53.7	54.7	54.7	54.7	53.7
svmrbf10cv	47.6	46.7	46.7	46.7	46.7	46.7	46.7	46.7	46.7	48.7	48.7	47.9	46.7
Discriminant Analysis - Pseudo Linear Descriminant type													
knn2cv	53.7	53.3	55.8	55.1	53.6	57.0	55.8	53.7	53.6	54.5	52.6	55.1	53.6
knn10cv	53.7	51.4	51.4	53.3	52.7	53.4	53.9	53.3	52.7	49.8	53.9	53.3	52.7
svmrbf10cv	53.7	49.9	52.8	53.4	53.4	49.9	52.8	53.4	52.2	55.9	55.7	54.3	53.4
Discriminant Analysis - Diag Linear Descriminant type													
knn2cv	41.9	43.1	44.2	44.2	43.1	43.6	44.2	41.9	43.1	43.7	43.7	44.2	43.1
knn10cv	41.9	42.5	42.5	43.7	43.7	43.7	43.5	43.7	43.7	43.0	43.5	43.7	43.7
svmrbf10cv	41.9	41.3	41.9	43.7	43.7	41.3	41.9	43.7	43.1	43.7	44.3	43.7	43.7
Naive Bayes - Normal Distribution													
knn2cv	37.2	40.8	44.3	46.0	37.8	43.6	44.3	37.2	37.8	45.9	44.3	46.0	37.8
knn10cv	37.2	37.2	37.8	38.3	40.6	36.7	37.8	38.3	40.6	38.9	37.8	38.3	40.6
svmrbf10cv	37.2	37.2	36.7	37.2	38.9	37.2	36.7	37.2	37.2	44.3	47.2	44.3	38.9
Naive Bayes - Kernel Distribution													
knn2cv	41.7	40.3	40.9	41.0	39.2	42.0	40.9	38.7	39.2	41.6	40.4	41.0	39.2
knn10cv	41.7	42.0	42.0	42.0	41.5	42.1	41.4	42.0	41.5	42.6	41.4	42.0	41.5
svmrbf10cv	41.7	41.5	39.9	40.4	41.0	41.5	39.9	40.4	42.1	40.3	40.4	41.0	41.0
Decision Tree													
knn2cv	53.5	58.3	54.7	54.7	54.6	57.9	54.7	54.6	54.6	58.3	58.3	54.7	54.6
knn10cv	53.5	60.3	60.3	53.6	53.6	51.4	53.6	53.6	53.6	60.3	53.6	53.6	53.6
svmrbf10cv	53.5	61.1	57.4	55.2	55.2	61.1	57.4	55.2	54.1	57.4	57.4	55.2	55.2
Support Vector Machine - Linear Kernel													
knn2cv	33.4	46.3	46.3	46.2	37.3	46.3	46.3	36.8	37.3	46.2	46.2	46.2	37.3
knn10cv	33.4	43.1	43.1	43.3	45.4	43.5	43.3	43.3	45.4	43.1	43.3	43.3	45.4
svmrbf10cv	33.4	37.7	37.7	37.7	38.4	37.7	37.7	37.7	37.7	45.7	45.2	39.3	38.4
Support Vector Machine - Radial Basis Function Kernel													
knn2cv	79.7	77.5	77.5	79.4	77.0	77.5	77.5	77.4	77.0	79.4	79.4	79.4	77.0
knn10cv	79.7	80.4	80.4	80.6	81.3	80.6	80.6	80.6	81.3	80.4	80.6	80.6	81.3
svmrbf10cv	79.7	79.6	79.6	79.6	79.6	79.6	79.6	79.6	79.6	76.6	77.2	77.7	79.6
Support Vector Machine - Quadratic Kernel													
knn2cv	46.8	59.8	58.4	63.0	46.7	54.4	58.4	52.0	46.7	47.2	63.5	63.0	46.7
knn10cv	46.8	48.2	46.6	51.2	53.8	52.5	51.4	51.2	53.8	51.4	51.4	51.2	53.8
svmrbf10cv	46.8	45.9	41.5	35.6	40.6	45.9	41.5	35.6	37.8	43.9	45.3	37.7	40.6
Random Forest													
knn2cv	53.4	56.6	55.7	60.1	55.2	53.5	55.7	55.1	55.2	55.6	56.7	60.1	55.2
knn10cv	53.4	54.8	57.2	55.1	58.2	52.8	56.2	55.1	58.2	57.4	56.2	55.1	58.2
svmrbf10cv	53.4	55.7	54.0	54.3	58.2	55.7	54.0	54.3	58.3	55.1	53.4	56.8	58.2
Feed Forward Artificial Neural Network													
knn2cv	31.2	35.9	34.5	34.5	31.2	36.8	38.8	34.6	35.7	37.6	33.4	33.5	31.9
knn10cv	31.2	37.6	35.4	37.2	34.3	35.4	35.2	36.0	35.2	32.2	37.8	35.6	35.9
svmrbf10cv	31.2	35.9	36.3	34.7	33.7	35.7	38.6	34.4	34.7	39.1	37.2	36.9	34.1

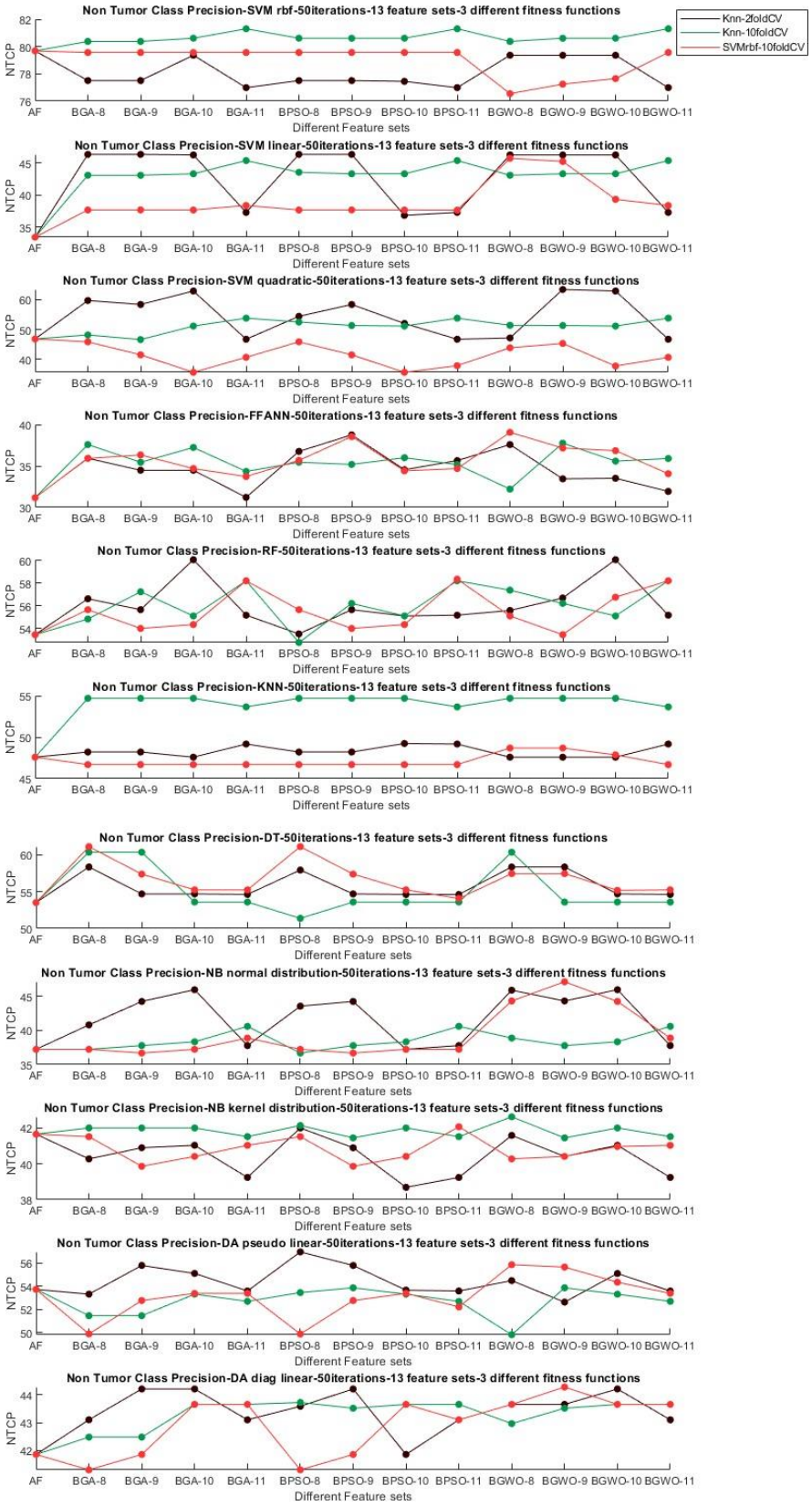


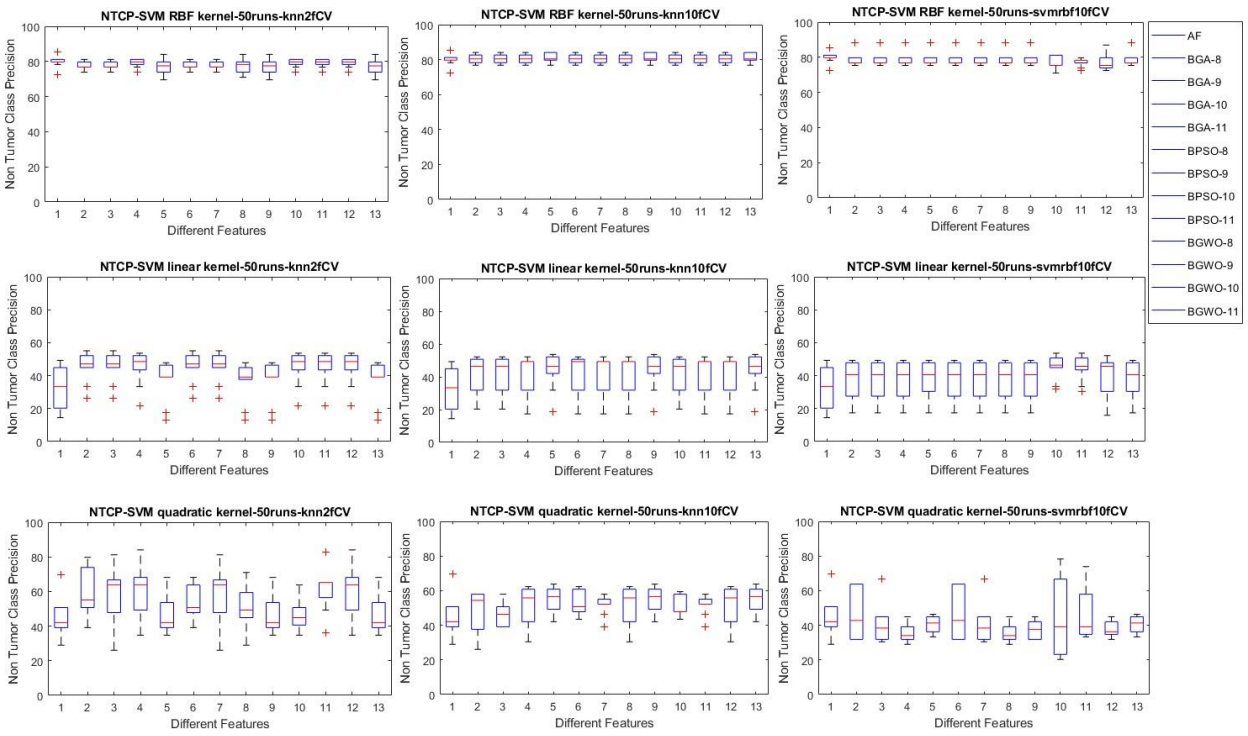
Figure 5.3: Scatter plot with mean non tumor class precision of classifiers with 13 feature subsets from BGA,BPSO,BGWO using 3 different fitness functions

From the figure 5.3 it is observed that, except NB and DA classifiers, the reset of the classifiers provided better performance on the features subsets that were selected using knn10cv fitness function than svmrbf10cv fitness function.

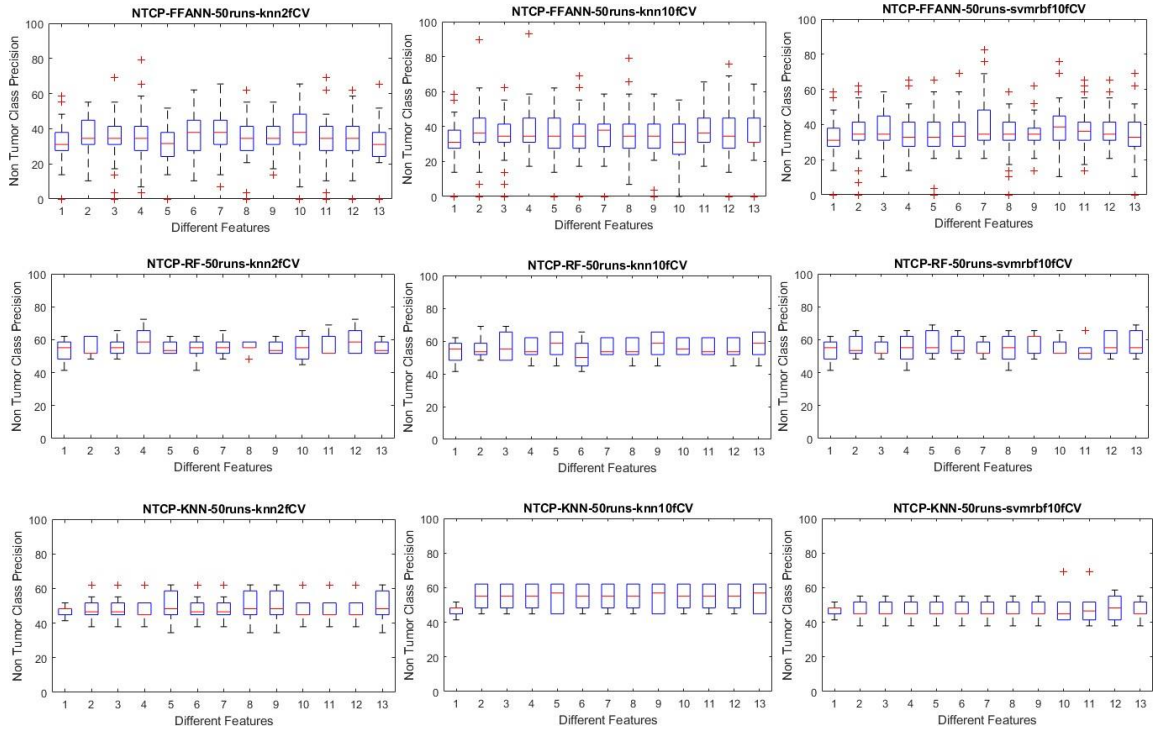
The performance of SVM with RBF and linear kernels, KNN classifiers are constant for the feature subsets that were selected using BGA and BPSO. Increasing the number of selected features from BGA and BPSO are providing better performance for RF, NB and DA classifiers, but where as for FFANN and DT the performance is decreased with increasing the number of selected features.

With the feature subsets selected using BGWO, the performance of SVM with quadratic kernel, FFANN and DT classifiers are similar to the ones obtained using BGA and BPSO features. But the nature of performance using BGWO features on the reset of the classifiers are different from the ones obtained BGA and BPSO features. With SVM classifier using rbf kernel, the performance is improved with increasing the number of features selected by BGWO, but whereas in the same case the performance is decreased using SVM classifier with linear kernel. In both the above cases the performance is constant irrelevant with he number of selected features from BGA and BPSO. Except for SVM with rbf kernel, RF and NB with kernel distribution, the performance of all the other classifiers are decreased with the increase in number of selected features from BGWO.

The box plots showing the distribution of non tumor class precision values for 50 runs with all the classifiers are shown in the figure 5.4. From the box plots it can be observed that more confident results are obtained with the feature selected by BGA, BPSO and BGWO using svmrbf10cv fitness function. Among all the classifiers, the non tumor class precision values obtained using SVM classifier with rbf kernel has produced most confident results.

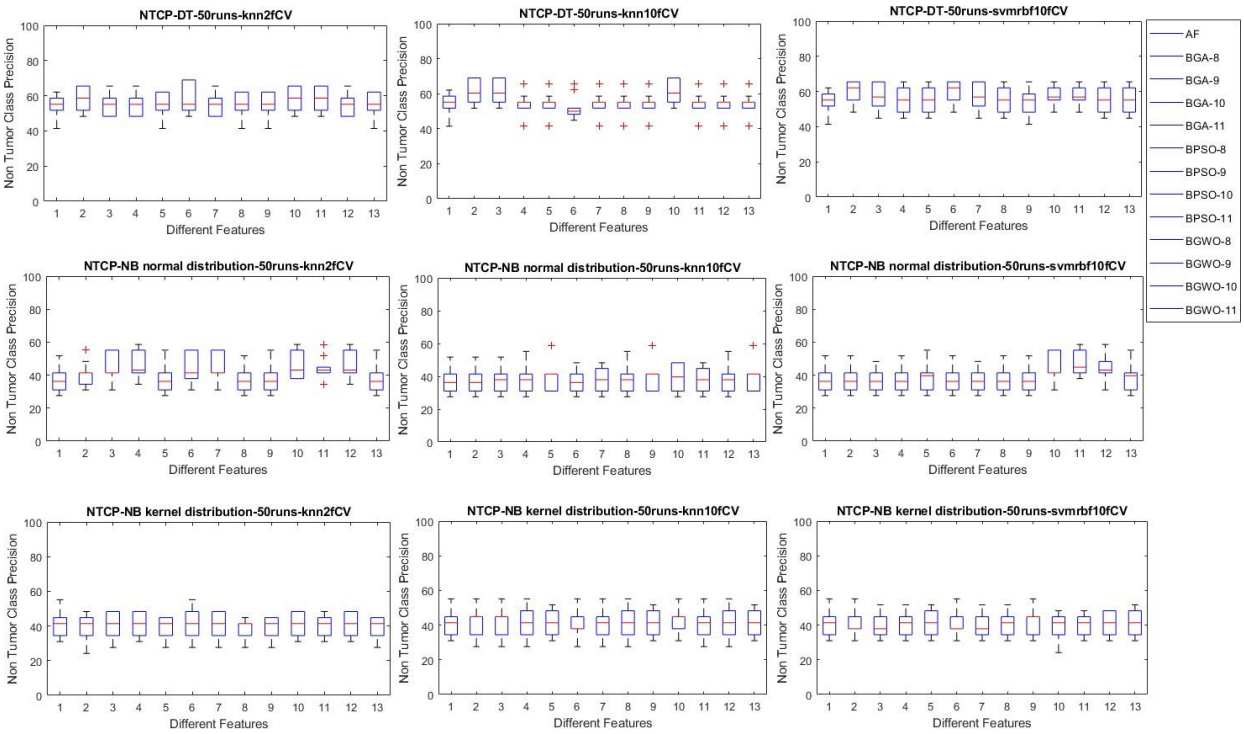


(a) Non Tumor Class Precision of SVM-rbf, SVM-linear, SVM-quadratic classifiers

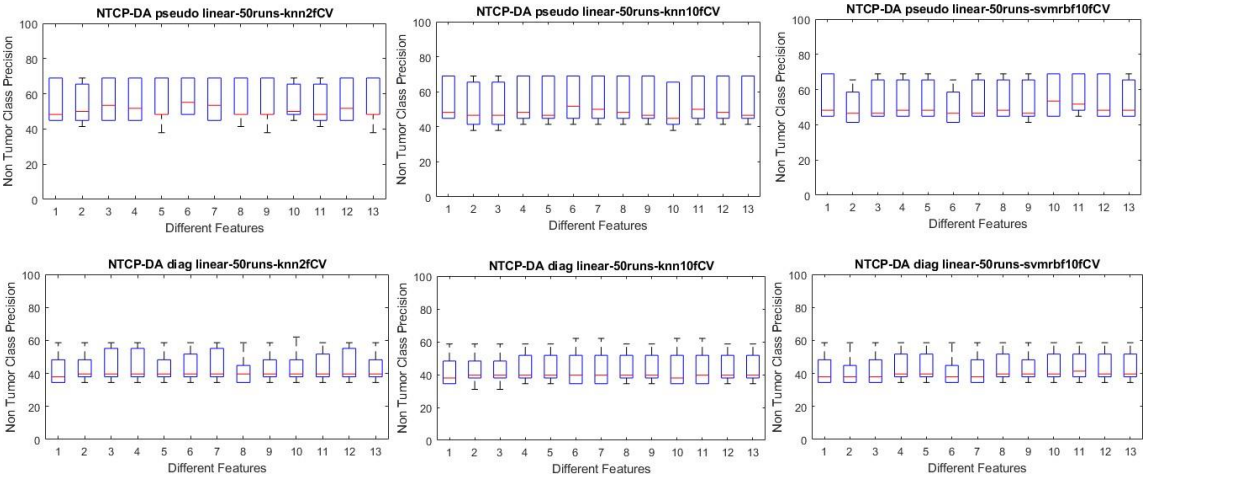


(b) Non Tumor Class Precision of FFANN, RF, KNN classifiers

Figure 5.4: Non Tumor Class Precision of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions



(c) Non Tumor Class Precision of DT, NB-normal, NB-kernel classifiers



(d) Non Tumor Class Precision of DA pseudo linear,DA diag linear classifiers

Figure 5.4: Non Tumor Class Precision of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions

5.3 Tumor Class Precision

Tumor class precision is a measure that specify what proportion of brain MRI images that are predicted as having tumor, does actually have tumor. In this work the brain MRI images that

consists of tumor lesions belong to tumor class category are represented as negative class.

The mean tumor class precision values obtained by eleven classifiers with 50 runs on each feature subset from BGA, BPSO and BGWO using three fitness functions are shown in table 5.9. The tumor class precision is represented as TCP in the table.

From the table 5.9, it can be observed that the performance of classifiers is high in predicting the samples that belong to tumor class than with the non tumor class prediction. And the SVM classifier with linear kernel has provided significantly better precision results on tumor class than with the non tumor class. The SVM classifier with rbf kernel has performed better than any other classifier, and with in itself, the tumor class precision values obtained with the selected features from knn2cv and svmrbf10cv fitness functions are better than the precision obtained using all features.

In order to visualize the mean tumor class precision values, scatter plots are plotted for all the classifiers and are shown in figure 5.5

Table 5.9: Tumor Class Precision of all classifiers with 13 feature subsets from BGA, BPSO, BGWO using 3 different fitness functions

TCP	AF	BGA8	BGA9	BGA10	BGA11	BPSO8	BPSO9	BPSO10	BPSO11	BGWO8	BGWO9	BGWO10	BGWO11
K-Nearest Neighbor													
knn2cv	79.7	78.6	78.6	78.8	78.0	78.6	78.6	79.4	78.0	78.8	78.8	78.8	78.0
knn10cv	79.7	79.2	79.2	79.2	80.3	79.2	79.2	79.2	80.3	79.2	79.2	79.2	80.3
svmrbf10cv	79.7	77.7	77.7	77.7	77.7	77.7	77.7	77.7	77.7	76.4	76.5	76.1	77.7
Discriminant Analysis - Pseudo Linear Discriminant type													
knn2cv	76.9	78.3	76.1	77.2	76.2	75.0	76.1	77.0	76.2	75.7	78.2	77.2	76.2
knn10cv	76.9	78.2	78.2	77.1	76.7	76.7	77.5	77.1	76.7	76.4	77.5	77.1	76.7
svmrbf10cv	76.9	77.8	77.0	77.0	78.4	77.8	77.0	77.0	77.1	78.6	79.6	78.1	78.4
Discriminant Analysis - Diag Linear Discriminant type													
knn2cv	78.3	77.5	77.5	77.5	77.5	77.8	77.5	77.5	77.5	77.8	77.5	77.5	77.5
knn10cv	78.3	77.5	77.5	77.5	77.5	77.9	78.2	77.5	77.5	78.3	78.2	77.5	77.5
svmrbf10cv	78.3	78.6	78.6	77.5	77.5	78.6	78.6	77.5	77.5	77.5	77.5	77.5	77.5
Naive Bayes - Normal Distribution													
knn2cv	86.1	80.4	80.1	75.9	82.9	80.0	80.1	85.7	82.9	79.0	77.3	75.9	82.9
knn10cv	86.1	83.2	82.2	81.8	79.0	81.8	82.5	81.8	79.0	82.9	82.5	81.8	79.0
svmrbf10cv	86.1	83.6	83.2	83.2	79.4	83.6	83.2	83.2	85.4	77.7	71.6	76.6	79.4
Naive Bayes - Kernel Distribution													
knn2cv	77.3	79.9	78.7	78.0	80.8	78.4	78.7	81.6	80.8	79.1	78.8	78.0	80.8
knn10cv	77.3	76.0	76.0	77.3	76.6	77.1	76.5	77.3	76.6	75.4	76.5	77.3	76.6
svmrbf10cv	77.3	75.1	77.3	77.3	77.6	75.1	77.3	77.3	75.4	81.3	80.2	78.0	77.6
Decision Tree													
knn2cv	74.1	73.0	75.5	75.5	74.5	76.9	75.5	74.5	74.5	73.0	73.0	75.5	74.5
knn10cv	74.1	73.3	73.3	73.7	73.7	71.0	73.7	73.7	73.7	73.3	73.7	73.7	73.7
svmrbf10cv	74.1	72.2	74.7	75.0	75.0	72.2	74.7	75.0	75.0	75.8	75.8	75.3	75.0
Support Vector Machine - Linear Kernel													
knn2cv	91.5	86.9	86.9	87.4	89.6	86.9	86.9	89.7	89.6	87.4	87.4	87.4	89.6
knn10cv	91.5	86.0	86.0	85.9	84.7	86.0	85.9	85.9	84.7	86.0	85.9	85.9	84.7
svmrbf10cv	91.5	88.4	88.4	88.4	88.4	88.4	88.4	88.4	88.6	85.8	86.1	85.6	88.4
Support Vector Machine - Radial Basis Function Kernel													
knn2cv	98.4	98.8	98.7	98.5	98.3	98.8	98.7	98.1	98.3	98.5	98.5	98.5	98.3
knn10cv	98.4	98.1	98.1	98.1	98.7	98.1	98.1	98.1	98.7	98.1	98.1	98.1	98.7
svmrbf10cv	98.4	99.2	99.2	99.2	99.2	99.2	99.2	99.2	99.2	99.1	98.2	98.8	99.2
Support Vector Machine - Quadratic Kernel													
knn2cv	68.6	66.8	70.4	58.7	90.8	58.0	70.4	85.8	90.8	61.6	76.8	58.7	90.8
knn10cv	68.6	87.5	88.4	83.5	86.8	87.5	79.9	83.5	86.8	88.6	79.9	83.5	86.8
svmrbf10cv	68.6	56.6	67.1	72.8	78.3	56.6	67.1	72.8	72.9	46.7	45.3	73.4	78.3
Random Forest													
knn2cv	86.2	83.0	82.6	83.7	85.8	82.4	82.6	85.0	85.8	83.9	83.9	83.7	85.8
knn10cv	86.2	83.3	84.6	84.4	84.3	82.2	86.1	84.4	84.3	84.3	86.1	84.4	84.3
svmrbf10cv	86.2	86.7	85.0	85.7	84.4	86.7	85.0	85.7	84.8	85.3	84.0	84.0	84.4
Feed Forward Artificial Neural Network													
knn2cv	83.5	84.9	81.0	83.7	77.8	83.0	81.8	80.7	82.9	81.6	78.1	79.1	78.6
knn10cv	83.5	81.0	82.2	82.0	80.6	84.2	82.0	84.7	81.2	83.7	83.3	80.0	83.2
svmrbf10cv	83.5	82.7	80.5	85.2	81.0	85.7	84.3	83.9	83.0	81.1	83.1	84.8	84.0

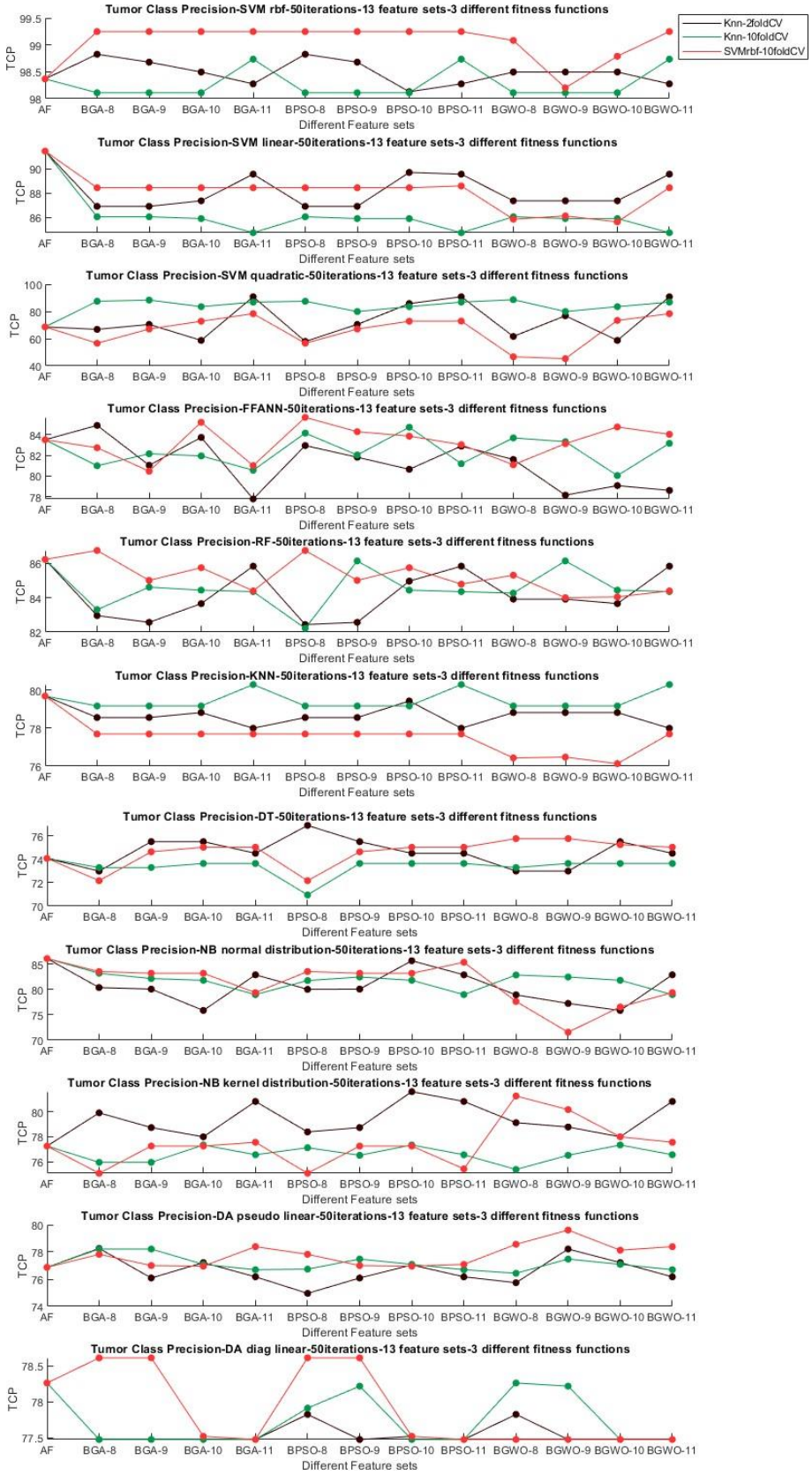


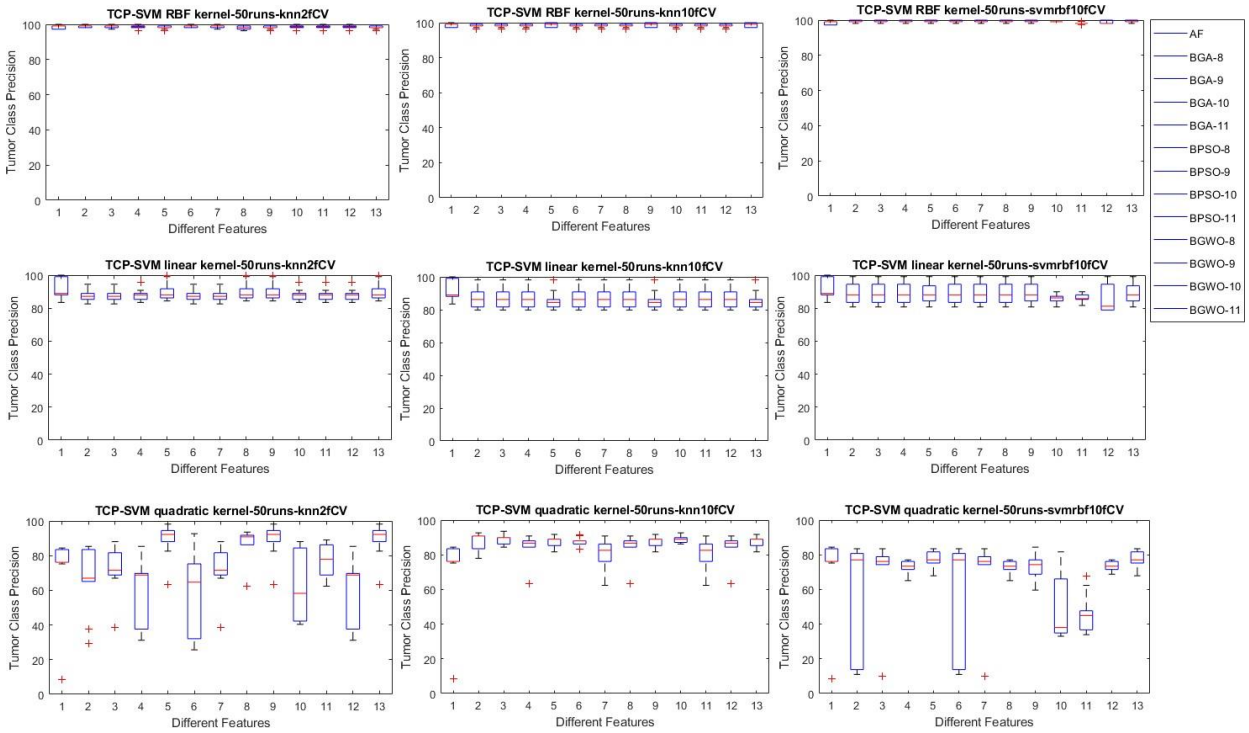
Figure 5.5: Scatter plot with mean tumor class precision of classifiers with 13 feature subsets from BGA,BPSO,BGWO using 3 different fitness functions

From the figure 5.5, it is observed that the performance of SVM with linear and rbf kernels, DT, NB with normal distribution, DA with diag linear discriminant is higher with the selected features from svmrbf10cv fitness function than with knn fitness functions. But with KNN and SVM with quadratic kernel the performance with features selected from knn10cv fitness function are better than with the svmrbf10cv fitness function. For the rest of the classifiers the performance is varying based on the number of selected features.

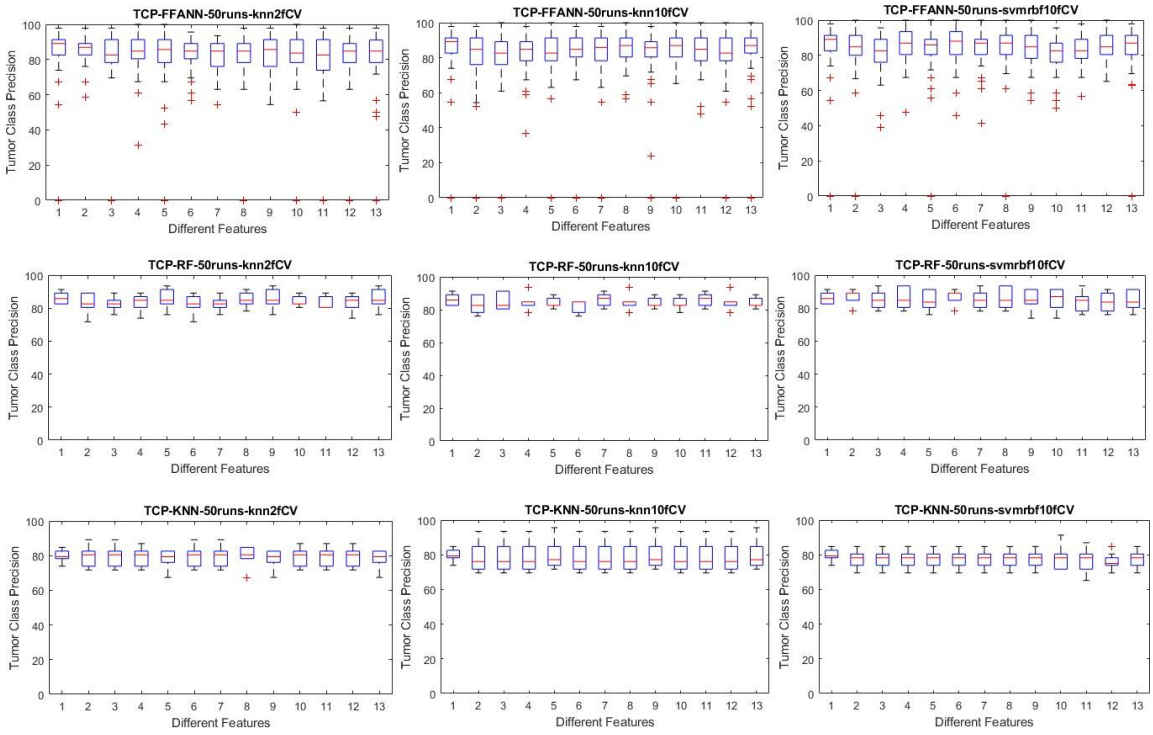
For the NB classifier with normal distribution and DA classifier with diag linear discriminant, the performance with features selected from BGWO with svmrbf10cv fitness function are slightly lower than the ones selected using knn10cv fitness function. For the classifier NB with kernel distribution and DA with pseudo linear discriminant the performance with BGWO features selected with svmrbf10cv fitness function are better than the ones obtained from knn10cv fitness function.

The performance of SVM with all kernels, FFANN, KNN is increased with increasing the number of features selected using BGWO. But whereas, for the classifier NB with normal distribution the performance is decreased with increasing the number of features using BGWO. For the classifiers SVM with rbf and linear kernels and KNN there is not change in precision values irrelevant to the number of selected features using BGA and BPSO, and it is same with the features selected using BGWO for the classifier DA with diag linear discriminant type.

The distribution of tumor class precision values for 50 runs of classifiers are shown as box plots in figure 5.6. From the figure 5.6 it can be seen that the results obtained with features selected form svmrbf10cv fitness function are more confident than the ones selected from other fitness functions. It can be greatly seen from SVM with quadratic kernel and DT, that the confidence of results are increased with the increase in number of features using BGA, BPSO and BGWO.

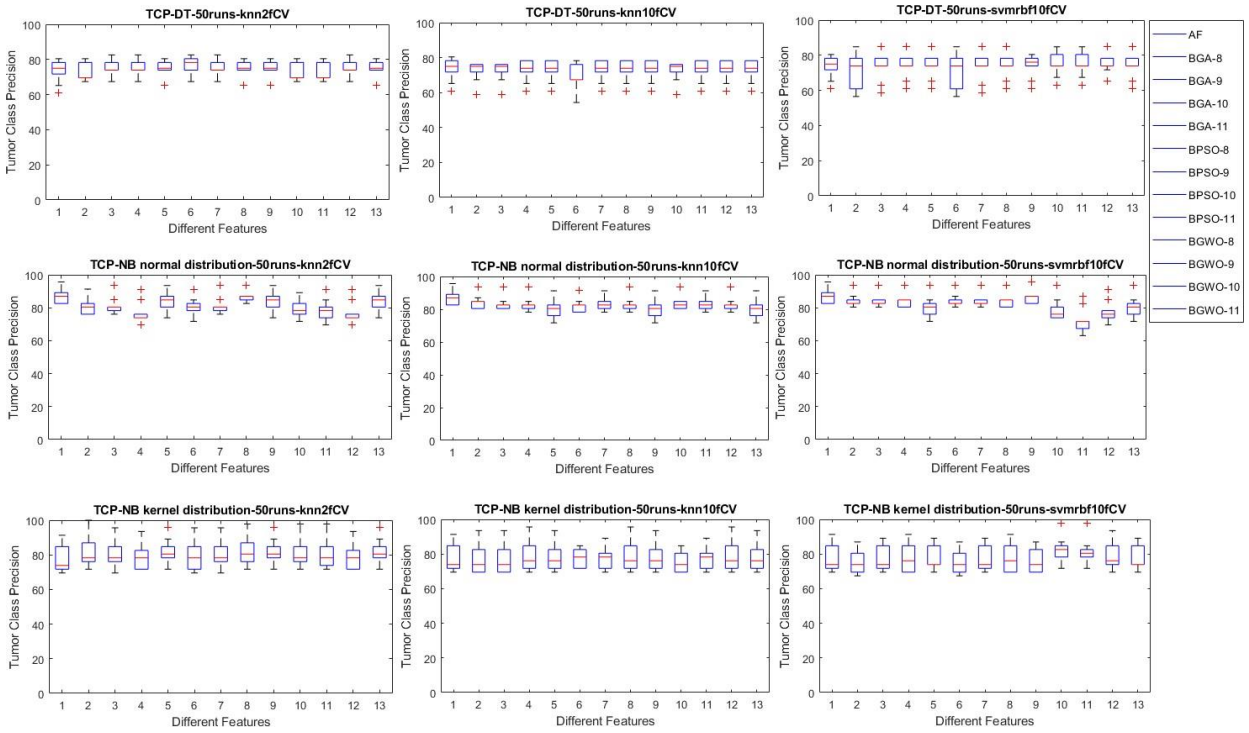


(a) Tumor Class Precision of SVM-rbf, SVM-linear, SVM-quadratic classifiers

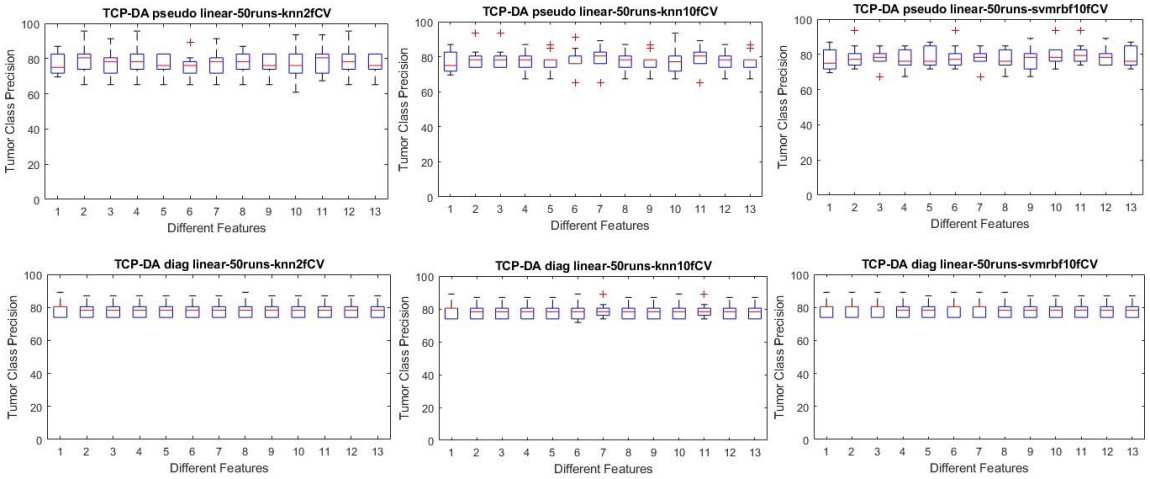


(b) Tumor Class Precision of FFANN, RF, KNN classifiers

Figure 5.6: Tumor Class Precision of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions



(c) Tumor Class Precision of DT, NB-nornal, NB-kernel classifiers



(d) Tumor Class Precision of DA pseudo linear,DA diag linear classifiers

Figure 5.6: Tumor Class Precision of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions

5.4 Specificity

Specificity is a classification measure that provides information about what proportion of images that belongs to tumor class category, are actually predicted by the model as tumorous.

The mean specificity values obtained by eleven classifiers by performing classification for 50 times on each feature subset selected from feature selection are shown in table 5.10. From the table 5.10, it is observed that SVM using rbf kernel has performed better than any other classifier used in this work. Moreover, the performance of all the classifiers using selected feature subsets are either equal of higher than the ones using all features.

To visualize the performance of classifiers on different feature subsets with different fitness functions, scatter plots are plotted using mean specificity values obtained. The figure 5.7 shows the scatter plots of mean specificity values obtained from classification. From the scatter plots it can be observed that, the performance of SVM with all kernels and KNN on the feature subsets from knn10cv fitness function are better than the ones selected from svmrbf10cv fitness function.

The performance of SVM with quadratic kernel and RF are increasing with increase in the number of selected features using BGA, BPSO and BGWO, where as for DT the performance is decreasing with increase in number of selected features. The performance of SVM with rbf and linear kernels, KNN is same irrelevant to the number of features selected from BGA and BPSO.

The performance of the NB and DA classifiers are increased with increase in number of selected features from BGA and BPSO, where as the performance is decreased with increase in number of features from BGWO. Except for SVM with rbf and quadratic kernels, RF and KNN, the performance of BGWO is better with less number of selected features, but even the decrease in performance with BGWO is negligible.

Table 5.10: Specificity of all classifiers with 13 feature subsets from BGA, BPSO, BGWO using 3 different fitness functions

Specificity	AF	BGA8	BGA9	BGA10	BGA11	BPSO8	BPSO9	BPSO10	BPSO11	BGWO8	BGWO9	BGWO10	BGWO11
K-Nearest Neighbor													
knn2cv	70.7	70.8	70.8	70.6	71.0	70.8	70.8	71.4	71.0	70.6	70.6	70.6	71.0
knn10cv	70.7	73.5	73.5	73.5	73.3	73.5	73.5	73.5	73.3	73.5	73.5	73.5	73.3
svmrbf10cv	70.7	69.8	69.8	69.8	69.8	69.8	69.8	69.8	69.8	70.5	70.5	69.9	69.8
Discriminant Analysis - Pseudo Linear Descriminant type													
knn2cv	72.7	72.9	73.5	73.5	72.6	73.6	73.5	72.8	72.6	72.7	72.6	73.5	72.6
knn10cv	72.7	72.2	72.2	72.7	72.4	72.7	73.1	72.7	72.4	71.0	73.1	72.7	72.4
svmrbf10cv	72.7	71.3	72.4	72.7	73.0	71.3	72.4	72.7	72.2	74.1	74.2	73.4	73.0
Discriminant Analysis - Diag Linear Descriminant type													
knn2cv	68.2	68.5	68.9	68.9	68.5	68.8	68.9	68.0	68.5	68.8	68.7	68.9	68.5
knn10cv	68.2	68.2	68.2	68.7	68.7	68.9	68.9	68.7	68.7	68.7	68.9	68.7	68.7
svmrbf10cv	68.2	68.1	68.3	68.7	68.7	68.1	68.3	68.7	68.5	68.7	68.9	68.7	68.7
Naive Bayes - Normal Distribution													
knn2cv	68.6	68.4	69.6	69.1	68.0	69.3	69.6	68.5	68.0	69.9	68.8	69.1	68.0
knn10cv	68.6	67.9	67.8	67.9	68.0	67.3	67.9	67.9	68.0	68.4	67.9	67.9	68.0
svmrbf10cv	68.6	68.0	67.7	67.9	67.4	68.0	67.7	67.9	68.4	69.0	68.2	68.6	67.4
Naive Bayes - Kernel Distribution													
knn2cv	67.8	68.0	67.9	67.8	67.9	68.3	67.9	67.8	67.9	68.2	67.7	67.8	67.9
knn10cv	67.8	67.6	67.6	68.0	67.5	67.9	67.5	68.0	67.5	67.7	67.5	68.0	67.5
svmrbf10cv	67.8	67.1	67.1	67.3	67.6	67.1	67.1	67.3	67.5	68.4	68.1	67.7	67.6
Decision Tree													
knn2cv	71.8	73.6	72.7	72.7	72.4	74.4	72.7	72.4	72.4	73.6	73.6	72.7	72.4
knn10cv	71.8	74.7	74.7	71.7	71.7	69.9	71.7	71.7	71.7	74.7	71.7	71.7	71.7
svmrbf10cv	71.8	74.5	73.6	72.8	72.8	74.5	73.6	72.8	72.3	73.9	73.9	72.8	72.8
Support Vector Machine - Linear Kernel													
knn2cv	68.8	72.0	72.0	72.1	69.5	72.0	72.0	69.4	69.5	72.1	72.1	72.1	69.5
knn10cv	68.8	70.7	70.7	70.7	71.1	70.8	70.7	70.7	71.1	70.7	70.7	70.7	71.1
svmrbf10cv	68.8	69.4	69.4	69.4	69.6	69.4	69.4	69.4	69.4	71.5	71.4	69.3	69.6
Support Vector Machine - Radial Basis Function Kernel													
knn2cv	88.5	87.4	87.4	88.3	87.2	87.4	87.4	87.4	87.2	88.3	88.3	88.3	87.2
knn10cv	88.5	88.8	88.8	88.9	89.3	88.9	88.9	88.9	89.3	88.8	88.9	88.9	89.3
svmrbf10cv	88.5	88.5	88.5	88.5	88.5	88.5	88.5	88.5	88.5	87.0	87.2	87.5	88.5
Support Vector Machine - Quadratic Kernel													
knn2cv	62.6	71.2	72.5	70.0	73.2	64.3	72.5	74.2	73.2	63.4	77.2	70.0	73.2
knn10cv	62.6	72.9	72.4	73.2	74.9	74.5	72.1	73.2	74.9	74.3	72.1	73.2	74.9
svmrbf10cv	62.6	55.7	60.7	64.1	67.6	55.7	60.7	64.1	64.9	56.4	56.3	65.1	67.6
Random Forest													
knn2cv	74.7	75.1	74.7	76.9	75.2	73.8	74.7	75.0	75.2	75.1	75.5	76.9	75.2
knn10cv	74.7	74.5	75.9	74.9	76.3	73.5	75.7	74.9	76.3	75.9	75.7	74.9	76.3
svmrbf10cv	74.7	75.7	74.5	74.9	76.2	75.7	74.5	74.9	76.3	75.0	74.1	75.5	76.2
Feed Forward Artificial Neural Network													
knn2cv	64.5	67.8	65.1	67.1	61.5	67.7	68.0	65.0	67.1	67.0	63.2	63.5	62.8
knn10cv	64.5	66.9	66.3	67.8	64.8	66.9	66.1	68.0	65.7	65.6	68.1	65.3	66.7
svmrbf10cv	64.5	66.6	66.6	67.5	64.7	68.0	68.9	66.4	66.8	68.0	67.9	68.1	66.4

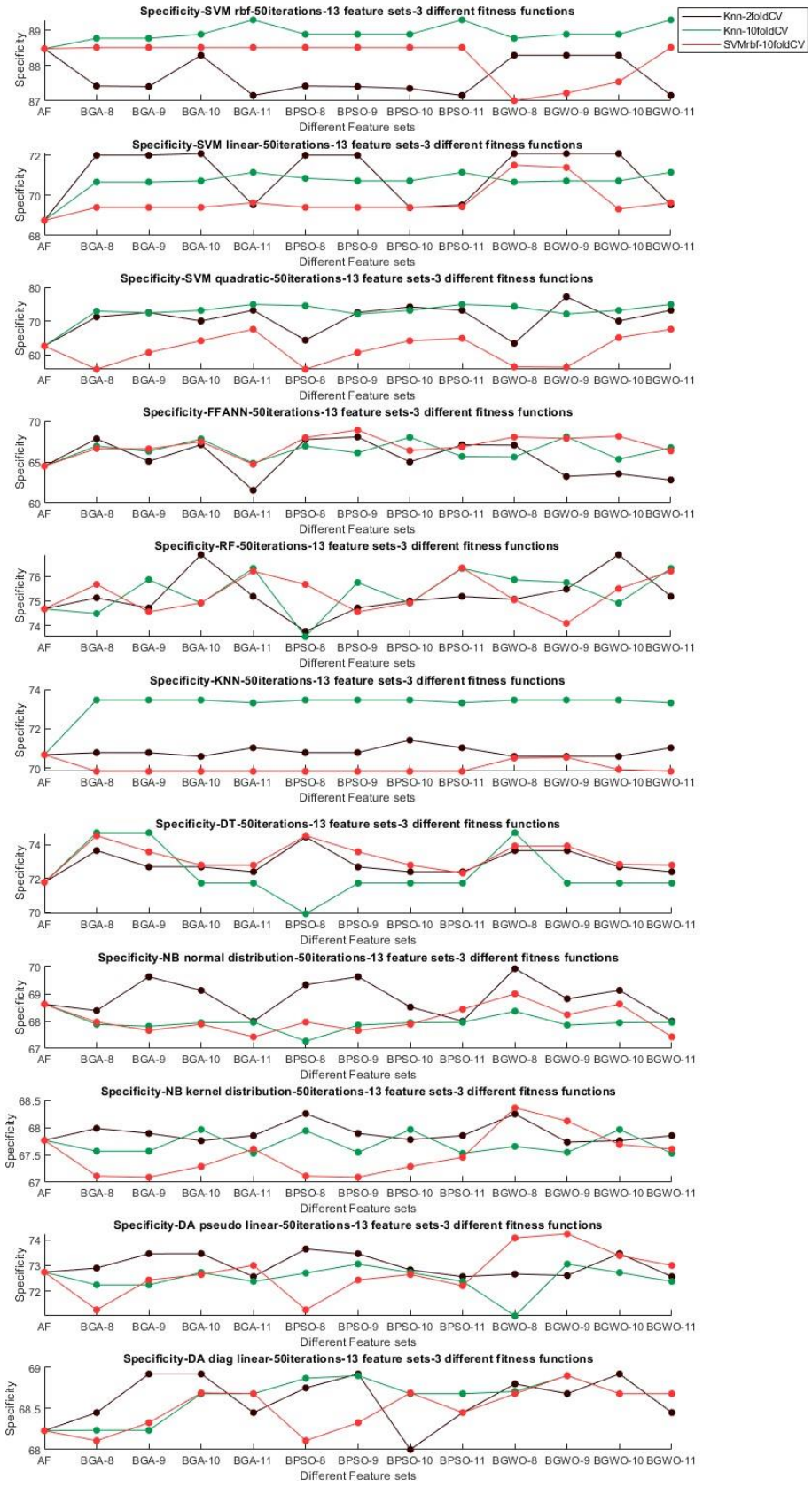
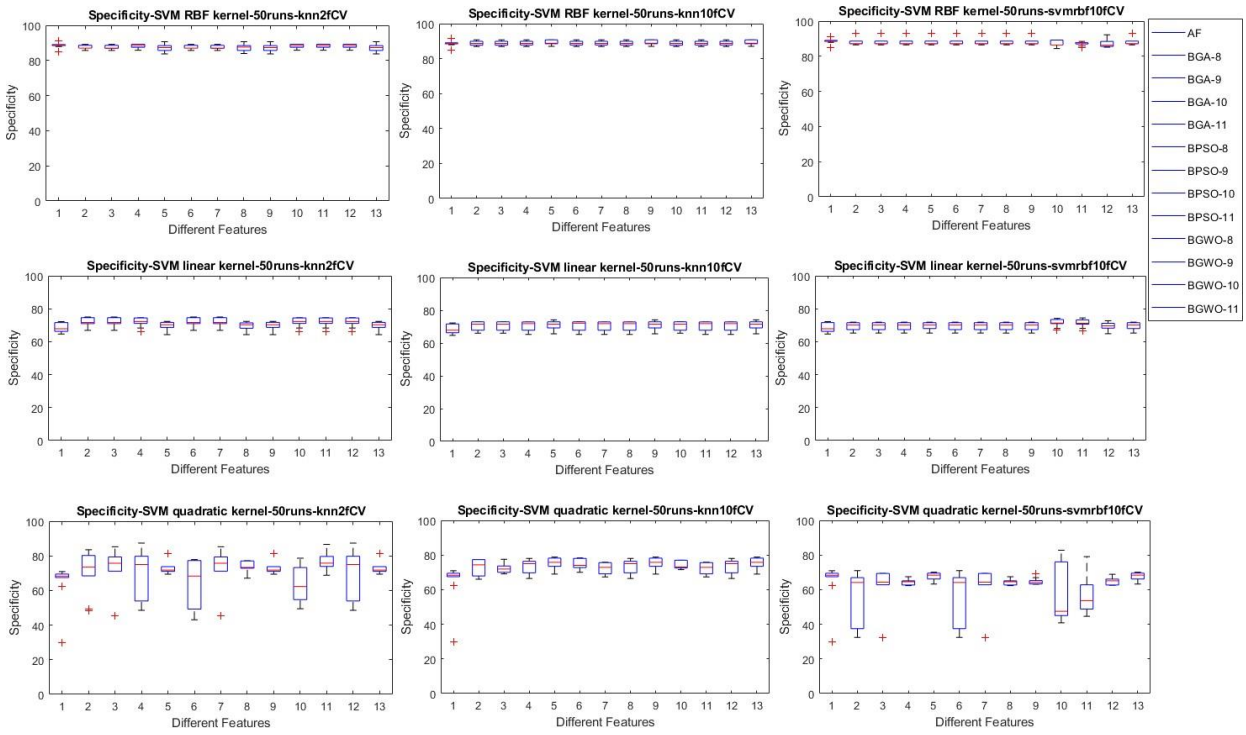
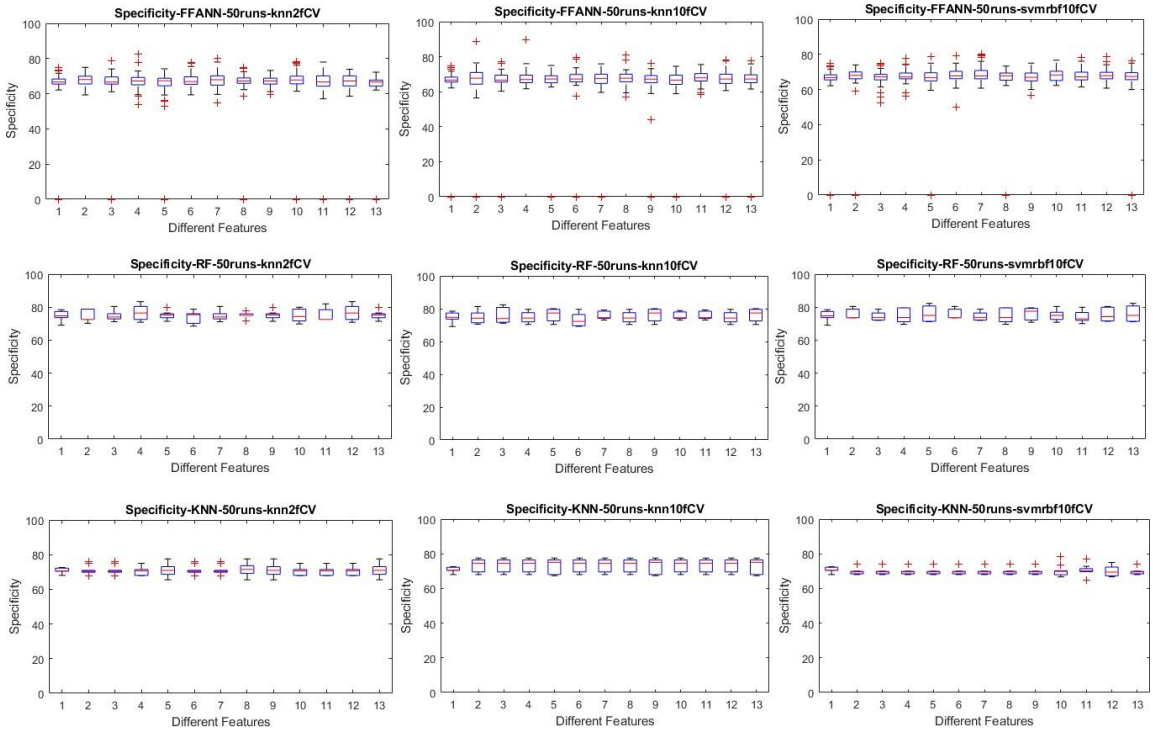


Figure 5.7: Scatter plot with mean specificity of classifiers with 13 feature subsets from BGA,BPSO,BGWO using 3 different fitness functions

The distribution of specificity values for 50 runs of classifiers on different selected feature subsets are shown as box plots in figure 5.8. From the box plots it can be observed that, except for SVM classifier with quadratic kernel, the specificity obtained with all the classifiers with all the selected feature subsets are more confident than the results for tumor class precision. The results are more confident with the feature subsets selected from svmrbf10cv fitness function. Except for the RF classifier, increase in number of features has increased the confidence in results.

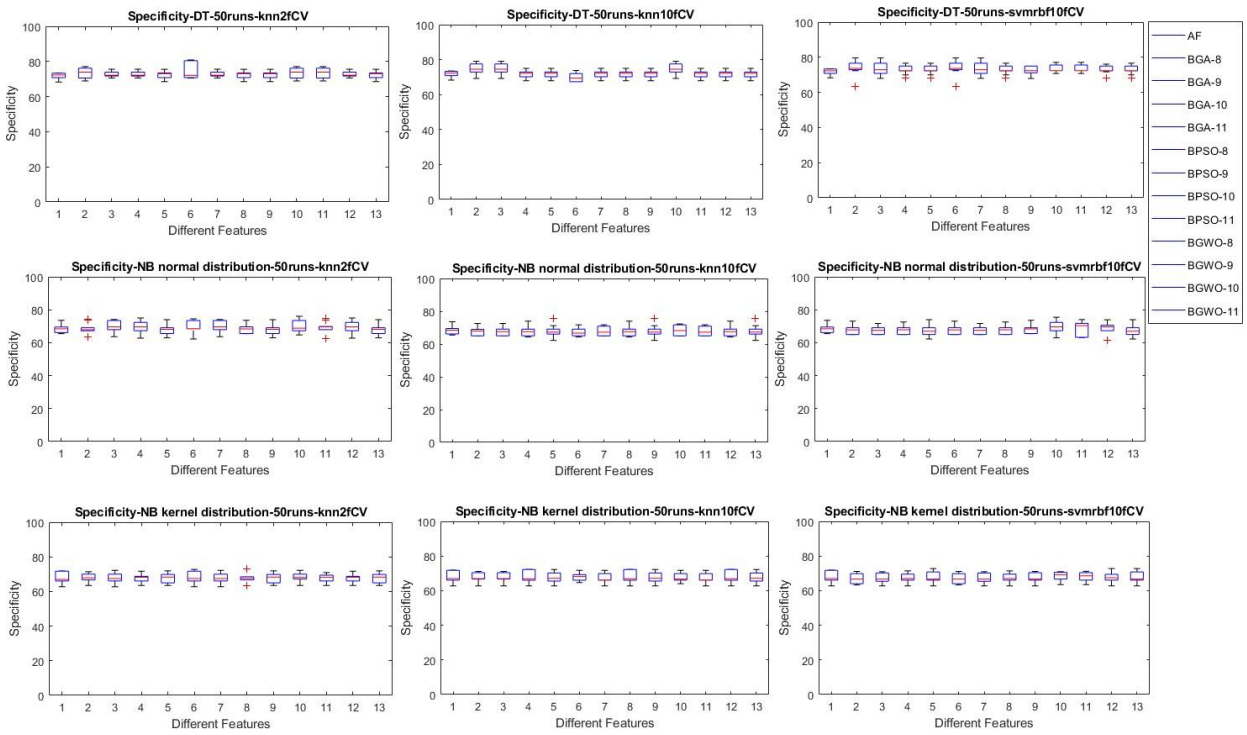


(a) Specificity of SVM-rbf, SVM-linear, SVM-quadratic classifiers

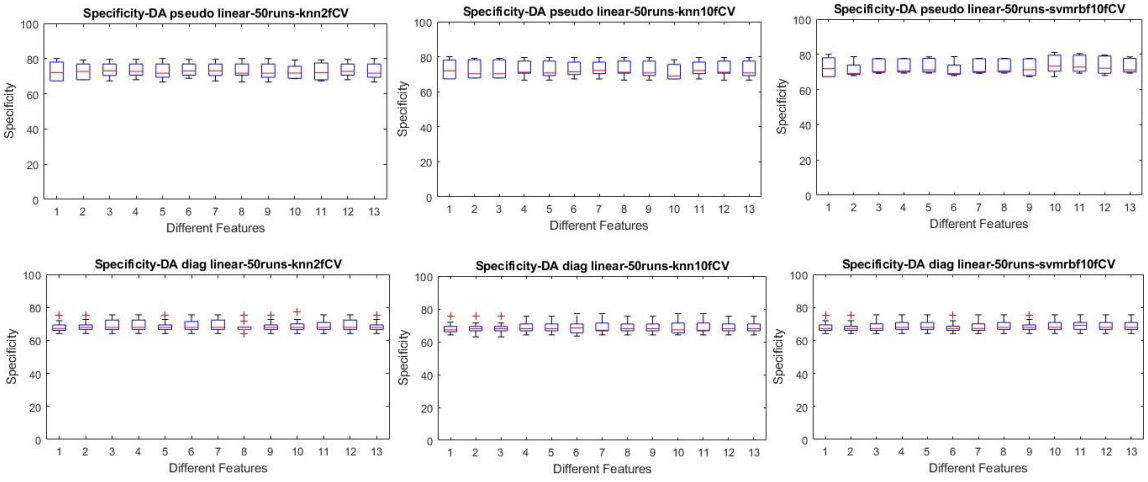


(b) Specificity of FFANN, RF, KNN classifiers

Figure 5.8: Specificity of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions



(c) Specificity of DT, NB-nornal, NB-kernel classifiers



(d) Specificity of DA pseudo linear,DA diag linear classifiers

Figure 5.8: Specificity of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions

5.5 Recall

Recall is the classification measure that provides information on what proportion of images that are belonged to non tumor class category, are actually predicted by the model as non tumorous.

The mean recall values obtained with 50 runs of classifiers on the selected feature subsets from BGA, BPSO and BGWO with three fitness functions are shown in table 5.11. From the table 5.11 it is observed that, the results obtained using the classifier SVM with rbf kernel are better than any other classifiers, and mostly all classifiers have provided better or slightly lesser recall values using selected feature subsets than with the ones using all features.

To visualize the mean recall values, scatter plots are plotted for each classifier applied on different selected feature subsets from BGA, BPSO and BGWO using three fitness functions. The scatter plots are shown in figure 5.9. From the figure 5.9, it is observed that, performance of SVM with rbf and linear kernels,RF and DT with feature subsets selected from svmrbf10cv fitness function are greater than the ones selected from knn10cv fitness function, where as the performance is vice versa with SVM with quadratic kernel and KNN classifiers.

The performance of SVM with rbf and linear kernels and KNN are same irrespective of number of features selected using BGA and BPSO. Except with DT, NB with kernel distribution and DA classifiers, the performance of other classifiers are increased with increase in number of feature selected using BGWO.

Table 5.11: Recall of all classifiers with 13 feature subsets from BGA, BPSO, BGWO using 3 different fitness functions

Recall	AF	BGA8	BGA9	BGA10	BGA11	BPSO8	BPSO9	BPSO10	BPSO11	BGWO8	BGWO9	BGWO10	BGWO11
K-Nearest Neighbor													
knn2cv	59.9	58.9	58.9	59.0	58.4	58.9	58.9	60.3	58.4	59.0	59.0	59.0	58.4
knn10cv	59.9	63.8	63.8	63.8	64.8	63.8	63.8	63.8	64.8	63.8	63.8	63.8	64.8
svmrbf10cv	59.9	57.3	57.3	57.3	57.3	57.3	57.3	57.3	57.3	56.6	57.0	56.0	57.3
Discriminant Analysis - Pseudo Linear Descriminant type													
knn2cv	59.8	61.2	59.8	60.9	58.8	59.1	59.8	59.8	58.8	59.3	60.7	60.9	58.8
knn10cv	59.8	59.6	59.6	59.6	58.8	59.4	60.4	59.6	58.8	57.2	60.4	59.6	58.8
svmrbf10cv	59.8	58.8	59.2	59.7	61.3	58.8	59.2	59.7	59.3	62.1	63.2	60.9	61.3
Discriminant Analysis - Diag Linear Descriminant type													
knn2cv	54.8	54.6	55.1	55.1	54.6	55.1	55.1	53.9	54.6	55.1	54.8	55.1	54.6
knn10cv	54.8	54.1	54.1	54.8	54.8	55.4	55.3	54.8	54.8	55.2	55.3	54.8	54.8
svmrbf10cv	54.8	54.9	55.1	54.9	54.8	54.9	55.1	54.9	54.6	54.8	55.2	54.8	54.8
Naive Bayes - Normal Distribution													
knn2cv	62.8	56.7	58.2	54.8	58.3	57.9	58.2	61.7	58.3	58.0	55.5	54.8	58.3
knn10cv	62.8	58.1	56.8	56.7	54.9	55.7	57.3	56.7	54.9	58.4	57.3	56.7	54.9
svmrbf10cv	62.8	58.5	57.6	58.0	54.3	58.5	57.6	58.0	61.3	55.7	51.6	54.6	54.3
Naive Bayes - Kernel Distribution													
knn2cv	54.6	57.1	55.4	54.5	57.0	55.8	55.4	58.8	57.0	56.4	55.3	54.5	57.0
knn10cv	54.6	53.4	53.4	54.5	53.3	54.2	53.0	54.5	53.3	52.6	53.0	54.5	53.3
svmrbf10cv	54.6	52.0	53.4	53.7	54.2	52.0	53.4	53.7	52.5	58.3	56.7	54.8	54.2
Decision Tree													
knn2cv	56.8	57.8	58.8	58.8	57.7	61.2	58.8	57.7	57.7	57.8	57.8	58.8	57.7
knn10cv	56.8	58.8	58.8	56.3	56.3	52.9	56.3	56.3	56.3	58.8	56.3	56.3	56.3
svmrbf10cv	56.8	59.0	59.2	58.7	58.7	59.0	59.2	58.7	57.9	60.4	60.4	58.8	58.7
Support Vector Machine - Linear Kernel													
knn2cv	76.6	69.2	69.2	69.9	72.5	69.2	69.2	72.5	72.5	69.9	69.9	69.9	72.5
knn10cv	76.6	67.0	67.0	66.7	66.0	67.2	66.7	66.7	66.0	67.0	66.7	66.7	66.0
svmrbf10cv	76.6	70.6	70.6	70.6	70.8	70.6	70.6	70.6	70.8	67.2	67.4	67.6	70.8
Support Vector Machine - Radial Basis Function Kernel													
knn2cv	96.9	97.7	97.4	97.1	96.6	97.7	97.4	96.4	96.6	97.1	97.1	97.1	96.6
knn10cv	96.9	96.5	96.5	96.5	97.6	96.5	96.5	96.5	97.6	96.5	96.5	96.5	97.6
svmrbf10cv	96.9	98.6	98.6	98.6	98.6	98.6	98.6	98.6	98.6	98.1	96.4	97.7	98.6
Support Vector Machine - Quadratic Kernel													
knn2cv	53.7	56.3	57.3	51.0	78.2	48.3	57.3	72.5	78.2	47.3	64.8	51.0	78.2
knn10cv	53.7	70.6	71.6	67.3	72.0	72.5	63.1	67.3	72.0	74.1	63.1	67.3	72.0
svmrbf10cv	53.7	45.6	48.1	45.3	54.5	45.6	48.1	45.3	47.7	34.5	34.8	47.3	54.5
Random Forest													
knn2cv	71.0	68.2	66.9	70.0	71.8	66.1	66.9	70.2	71.8	68.3	68.9	70.0	71.8
knn10cv	71.0	67.9	70.1	69.3	69.9	65.1	72.1	69.3	69.9	69.8	72.1	69.3	69.9
svmrbf10cv	71.0	73.0	70.0	71.1	70.3	73.0	70.0	71.1	71.2	71.2	68.4	69.5	70.3
Feed Forward Artificial Neural Network													
knn2cv	60.7	61.3	58.0	58.6	56.6	58.5	58.6	57.4	59.8	59.4	55.6	55.9	56.1
knn10cv	60.7	59.4	59.5	58.8	59.3	62.6	59.2	61.8	57.9	59.3	61.7	58.0	62.1
svmrbf10cv	60.7	60.8	57.0	63.4	58.2	64.9	63.0	61.7	59.7	58.2	60.2	63.1	61.5

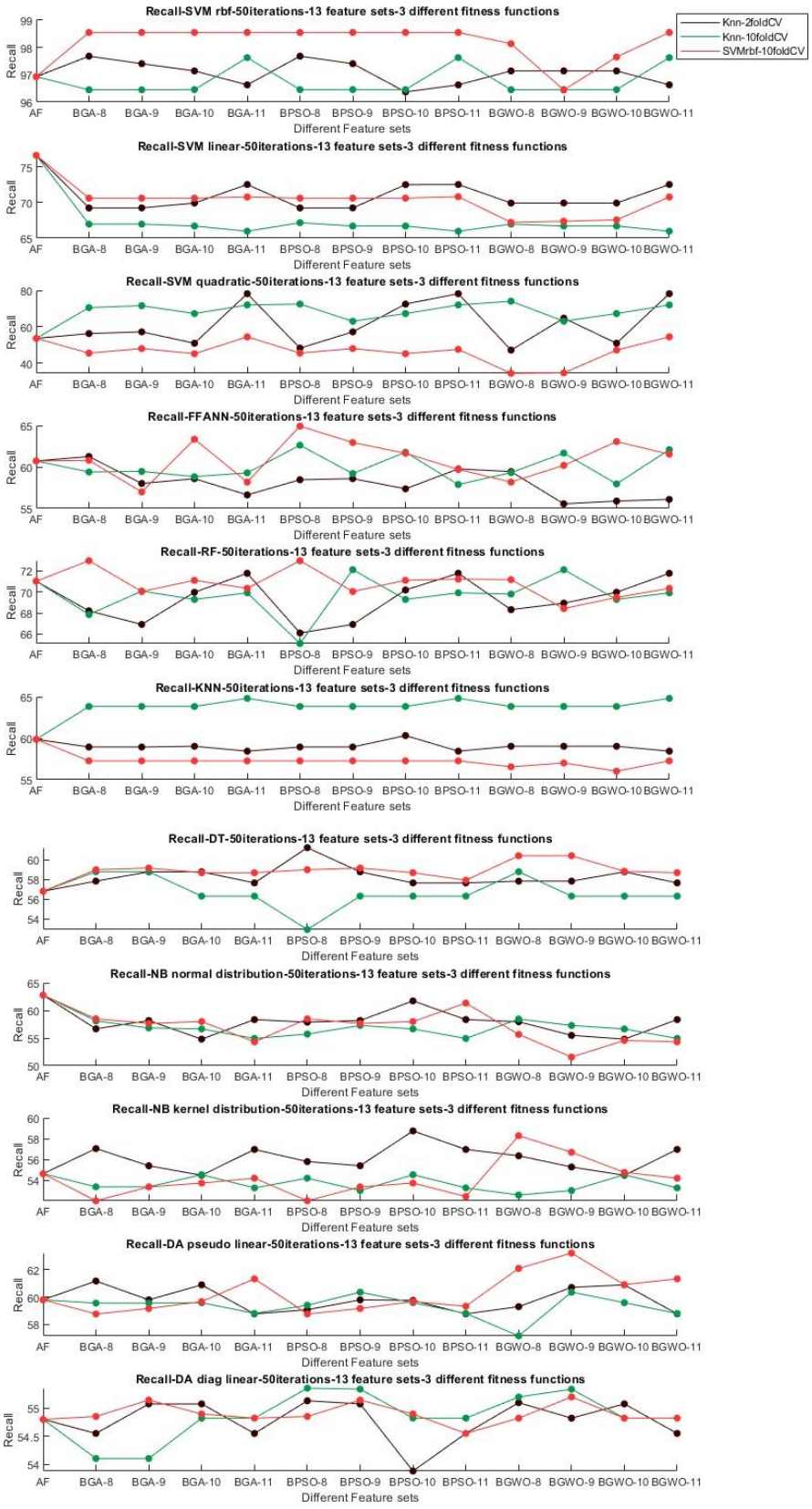
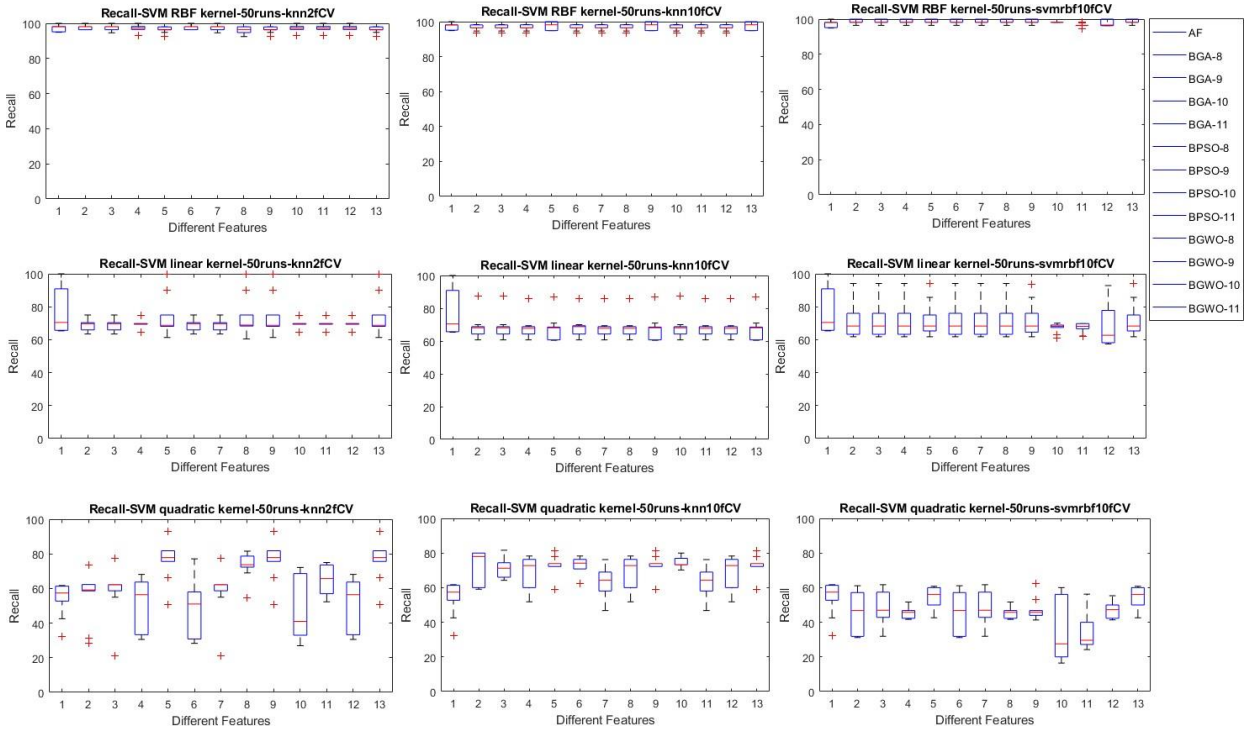


Figure 5.9: Scatter plot with mean recall of classifiers with 13 feature subsets from BGA,BPSO,BGWO using 3 different fitness functions

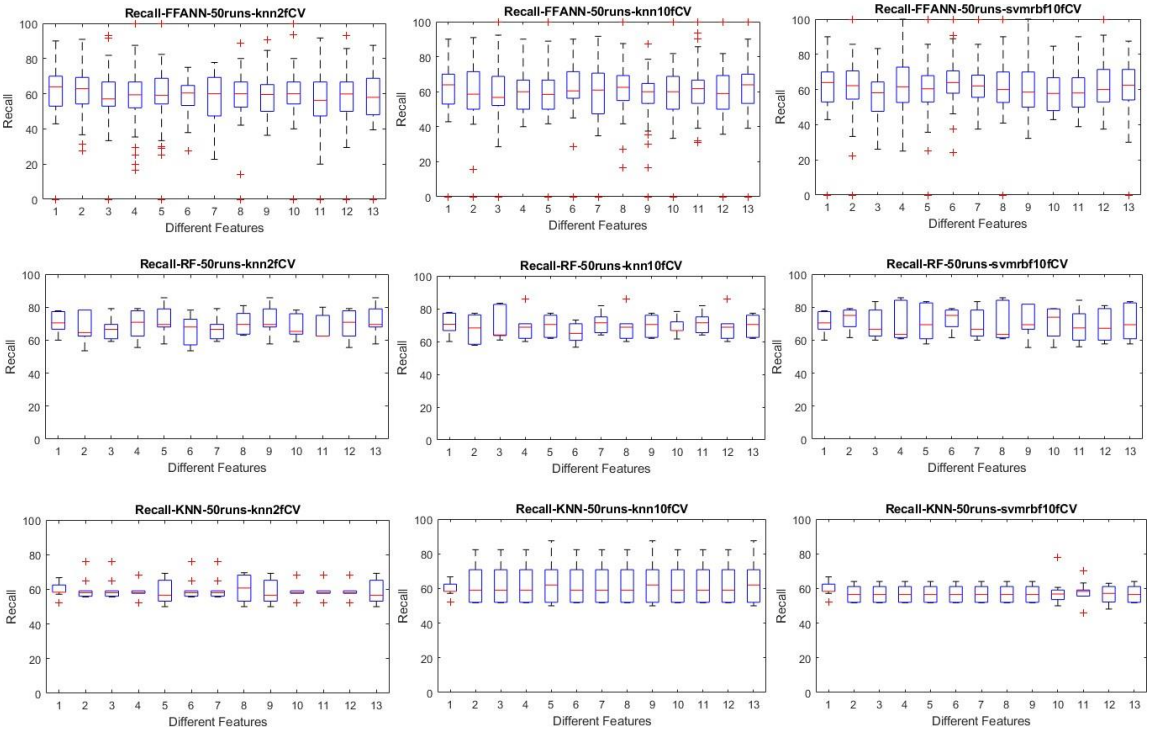
The distribution of recall values obtained by 50 runs of classifiers are shown as box plots in figure 5.10.

From the box plots figure 5.10 it is observed that, for SVM with rbf kernel and KNN the recall values obtained with feature subsets selected from svmrbf10cv fitness function are more confident than the ones obtained using other fitness functions. But for SVM with linear and quadratic kernels the results obtained with features selected from knn10cv fitness function are more confident than the ones obtained using svmrbf10cv fitness function. For the NB and DA classifiers the results are mostly same irrelevant with any chosen fitness function.

For most of the classifiers increase in number of features selected using BGA and BPSO has no effect in confidence, and for most of the classifiers increase in number of features selected using BGWO has increased the confidence.

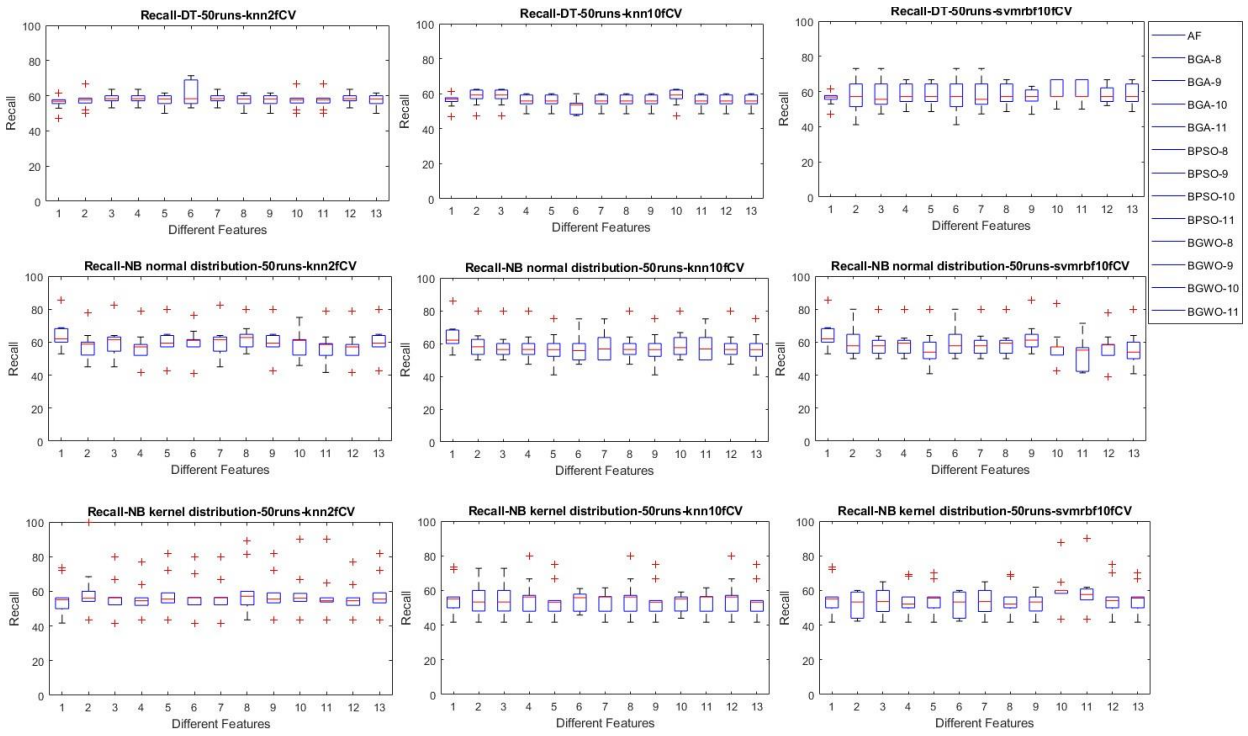


(a) Recall of SVM-rbf, SVM-linear, SVM-quadratic classifiers

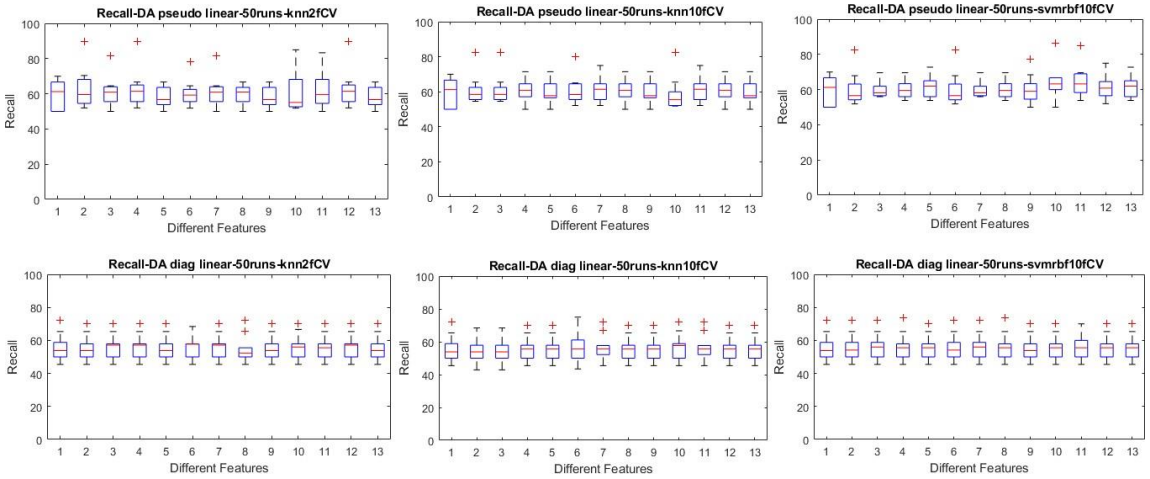


(b) Recall of FFANN, RF, KNN classifiers

Figure 5.10: Recall of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions



(c) Recall of DT, NB-nornal, NB-kernel classifiers



(d) Recall of DA pseudo linear,DA diag linear classifiers

Figure 5.10: Recall of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions

5.6 F-Measure

It is a classification measure which provides a combined information of both non tumor class precision and recall. This metric is often handy in comparing the performance of the model rather than looking at two different metrics.

The mean f-measure values for 50 runs of the classifiers on each feature subset from BGA, BPSO and BGWO with three fitness functions are shown in table 5.12. From the table 5.12 it is observed that, SVM with rbf kernel has provided best results than any other classifier, and the results of the classifier with different feature subsets selected are also better than the ones obtained using all features.

To visualize the mean values of the classifiers, scatter plots are plotted for each classifier with different selected feature subsets with three different fitness functions. The scatter plots are shown in figure 5.11. From the figure 5.11 it is observed that, except for SVM with quadratic kernel and KNN classifiers, the performance of all the classifiers are nearly equal or varying with the number of selected number of features using knn10cv and svmrbf10cv fitness functions. Only for the mentioned classifiers the performance with features selected using knn10cv fitness function are better than the ones selected using svmrbf10cv fitness function. Besides the difference, the trend of the classifiers performance on selected feature subsets are similar to the scatter plots of non tumor class precision as shown in figure 5.3 and recall as shown in figure 5.9.

The distribution of f-measure for 50 runs of classifiers are show as box plots in figure 5.12. From the box plots it is observed that, SVM with rbf kernel, DT, FFANN, KNN and DA produced more confident results with the feature subsets selected with svmrbf10cv fitness function than with others. For SVM with rbf and quadratic kernels, KNN and DT the confidence of the results increased with increase in the number of selected features.

Table 5.12: F-measure of all classifiers with 13 feature subsets from BGA, BPSO, BGWO using 3 different fitness functions

Fmeasure	AF	BGA8	BGA9	BGA10	BGA11	BPSO8	BPSO9	BPSO10	BPSO11	BGWO8	BGWO9	BGWO10	BGWO11
K-Nearest Neighbor													
knn2cv	53.0	52.6	52.6	52.3	53.1	52.6	52.6	53.9	53.1	52.3	52.3	52.3	53.1
knn10cv	53.0	58.4	58.4	58.4	58.1	58.4	58.4	58.4	58.1	58.4	58.4	58.4	58.1
svmrbf10cv	53.0	51.2	51.2	51.2	51.2	51.2	51.2	51.2	51.2	52.0	52.0	51.3	51.2
Discriminant Analysis - Pseudo Linear Descriminant type													
knn2cv	56.1	56.5	57.3	57.3	55.6	57.7	57.3	56.0	55.6	56.3	55.9	57.3	55.6
knn10cv	56.1	54.7	54.7	55.7	55.1	55.7	56.4	55.7	55.1	52.7	56.4	55.7	55.1
svmrbf10cv	56.1	53.6	55.3	55.8	56.5	53.6	55.3	55.8	55.0	58.5	58.9	57.1	56.5
Discriminant Analysis - Diag Linear Descriminant type													
knn2cv	47.2	48.0	48.8	48.8	48.0	48.5	48.8	46.9	48.0	48.5	48.4	48.8	48.0
knn10cv	47.2	47.4	47.4	48.4	48.4	48.6	48.5	48.4	48.4	48.0	48.5	48.4	48.4
svmrbf10cv	47.2	46.9	47.3	48.4	48.4	46.9	47.3	48.4	48.0	48.4	49.0	48.4	48.4
Naive Bayes - Normal Distribution													
knn2cv	46.4	47.3	50.1	49.7	45.5	49.5	50.1	46.2	45.5	51.1	49.1	49.7	45.5
knn10cv	46.4	45.1	45.1	45.4	46.4	44.1	45.4	45.4	46.4	46.5	45.4	45.4	46.4
svmrbf10cv	46.4	45.3	44.7	45.1	45.0	45.3	44.7	45.1	46.1	49.1	49.1	48.7	45.0
Naive Bayes - Kernel Distribution													
knn2cv	46.7	46.6	46.7	46.5	46.1	47.4	46.7	46.2	46.1	47.5	46.3	46.5	46.1
knn10cv	46.7	46.3	46.3	47.0	46.3	47.1	46.1	47.0	46.3	46.6	46.1	47.0	46.3
svmrbf10cv	46.7	45.6	45.2	45.7	46.4	45.6	45.2	45.7	46.2	47.2	46.9	46.5	46.4
Decision Tree													
knn2cv	54.8	57.9	56.4	56.4	55.8	59.4	56.4	55.8	55.8	57.9	57.9	56.4	55.8
knn10cv	54.8	59.4	59.4	54.7	54.7	52.0	54.7	54.7	54.7	59.4	54.7	54.7	54.7
svmrbf10cv	54.8	59.7	58.0	56.6	56.6	59.7	58.0	56.6	55.6	58.6	58.6	56.6	56.6
Support Vector Machine - Linear Kernel													
knn2cv	43.8	55.1	55.1	55.3	47.1	55.1	55.1	46.8	47.1	55.3	55.3	55.3	47.1
knn10cv	43.8	51.5	51.5	51.6	53.1	51.9	51.6	51.6	53.1	51.5	51.6	51.6	53.1
svmrbf10cv	43.8	47.1	47.1	47.1	47.8	47.1	47.1	47.1	47.1	54.1	53.8	47.1	47.8
Support Vector Machine - Radial Basis Function Kernel													
knn2cv	87.4	86.4	86.3	87.3	85.6	86.4	86.3	85.8	85.6	87.3	87.3	87.3	85.6
knn10cv	87.4	87.7	87.7	87.8	88.7	87.8	87.8	87.8	88.7	87.7	87.8	87.8	88.7
svmrbf10cv	87.4	88.0	88.0	88.0	88.0	88.0	88.0	88.0	88.0	86.0	85.8	86.4	88.0
Support Vector Machine - Quadratic Kernel													
knn2cv	48.0	56.8	57.3	55.9	57.3	50.5	57.3	59.2	57.3	46.6	63.3	55.9	57.3
knn10cv	48.0	56.7	56.3	57.3	61.5	60.8	56.2	57.3	61.5	60.6	56.2	57.3	61.5
svmrbf10cv	48.0	42.8	42.8	39.7	46.5	42.8	42.8	39.7	41.9	38.5	39.3	41.9	46.5
Random Forest													
knn2cv	60.8	61.8	60.7	64.6	62.2	59.0	60.7	61.6	62.2	61.2	62.2	64.6	62.2
knn10cv	60.8	60.6	63.0	61.3	63.4	58.1	63.1	61.3	63.4	62.9	63.1	61.3	63.4
svmrbf10cv	60.8	62.9	60.8	61.4	63.6	62.9	60.8	61.4	64.0	61.9	59.8	62.4	63.6
Feed Forward Artificial Neural Network													
knn2cv	40.3	44.5	42.2	41.8	39.1	44.0	45.7	42.1	43.4	44.5	40.4	41.0	39.3
knn10cv	40.3	44.5	43.2	44.1	41.8	43.8	43.2	44.0	42.6	40.7	45.5	42.5	44.0
svmrbf10cv	40.3	43.9	43.0	43.3	41.5	44.5	46.1	42.9	42.5	45.5	44.6	45.3	42.6

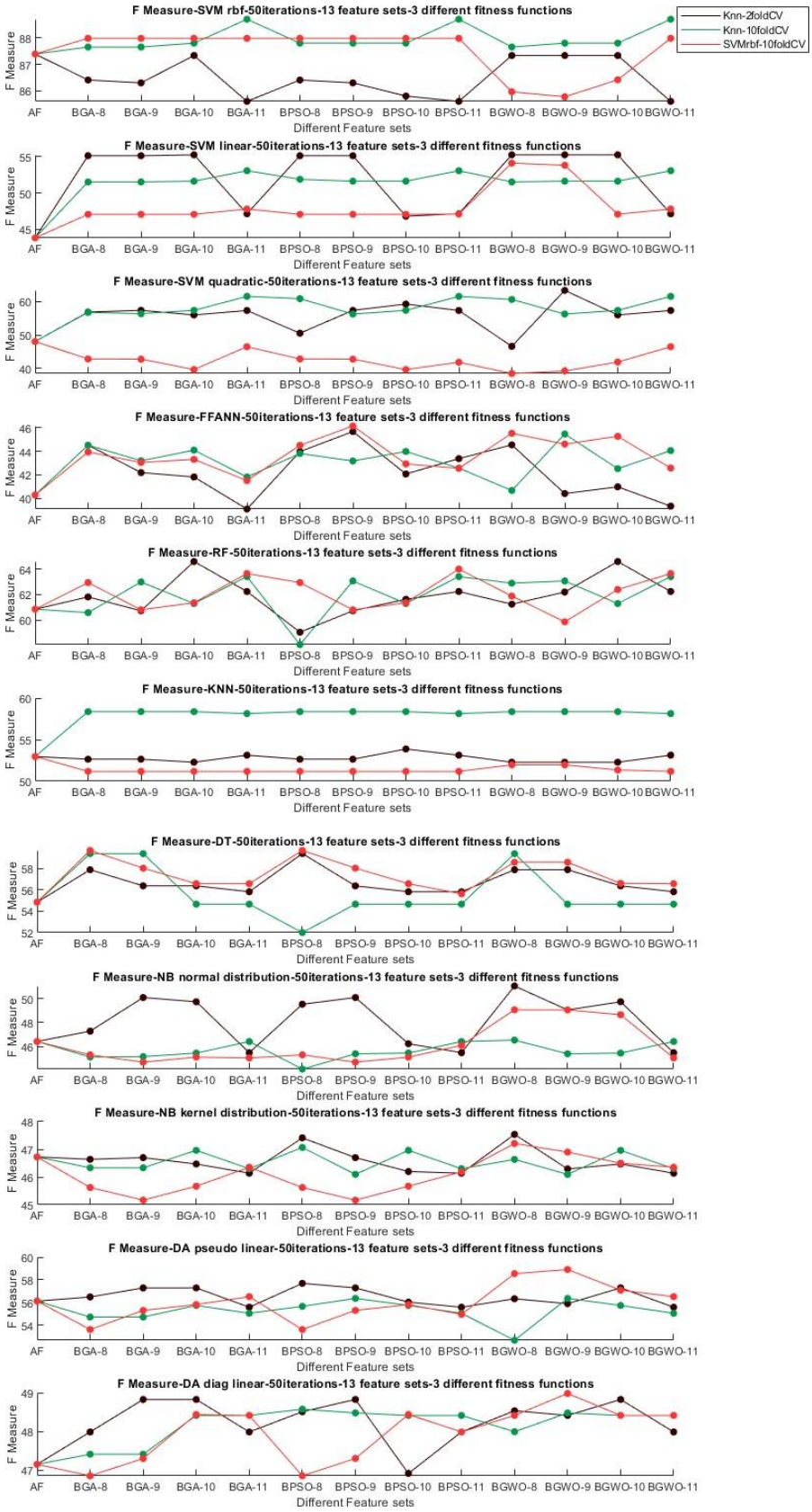
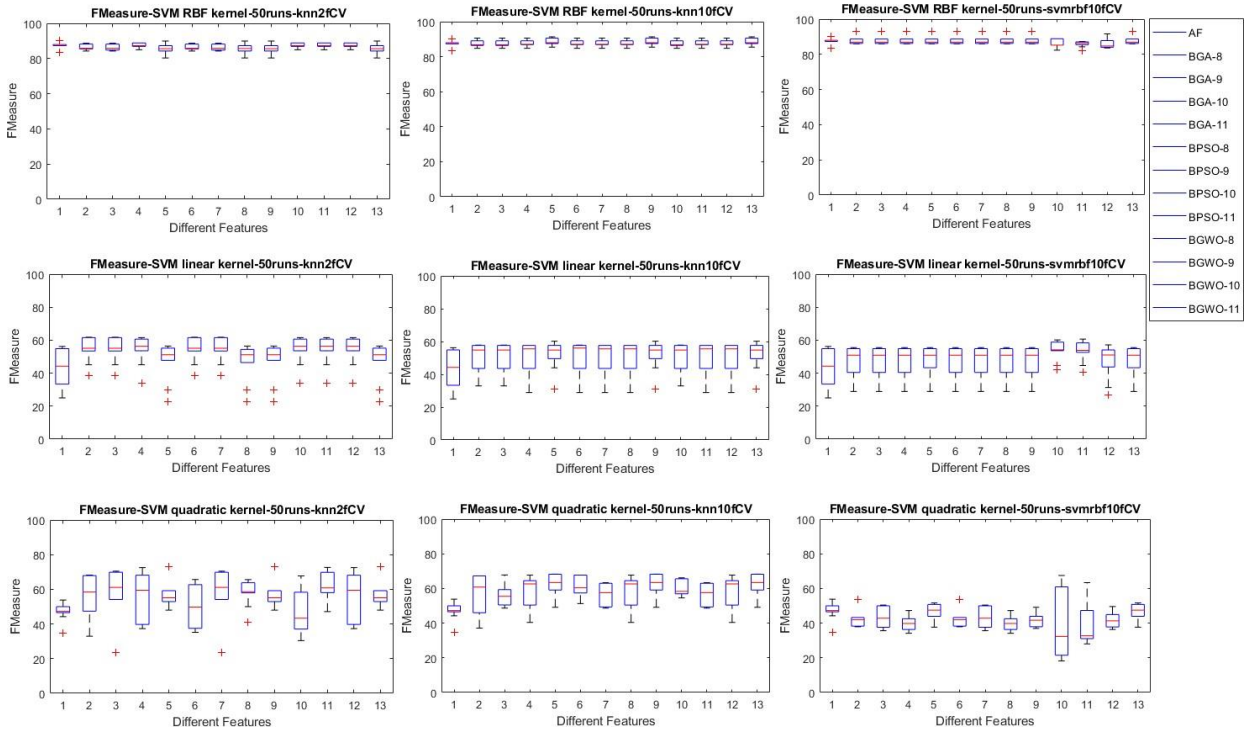
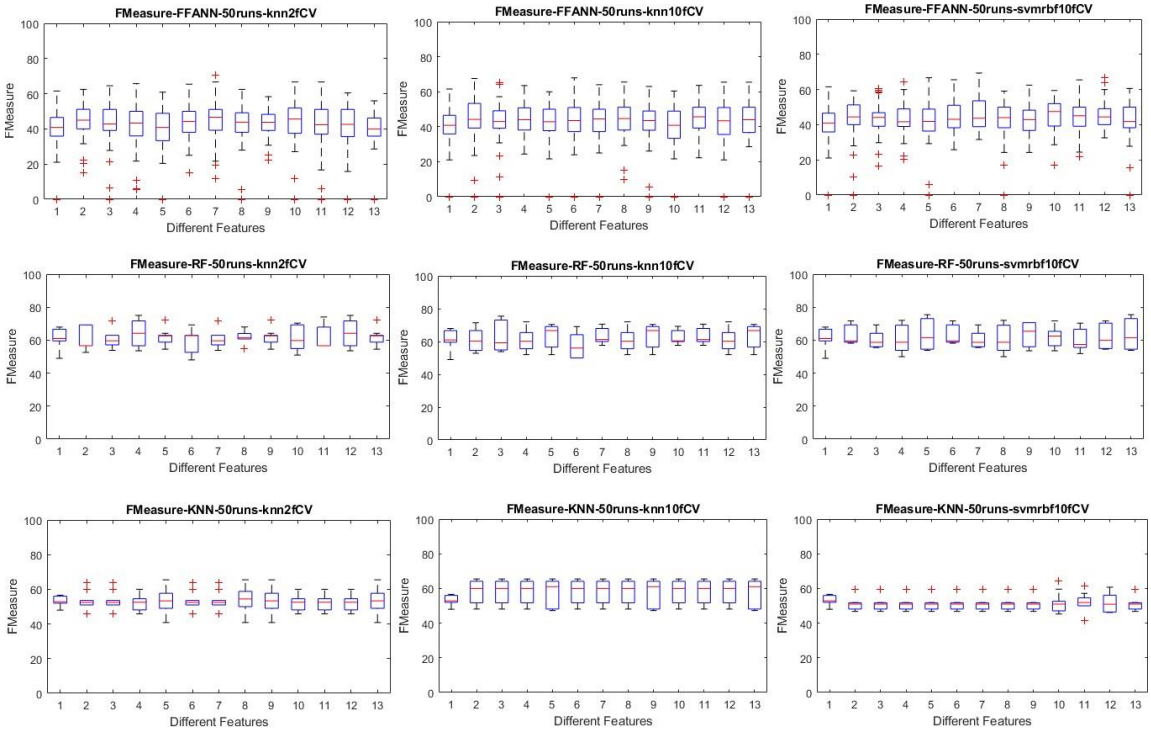


Figure 5.11: Scatter plot with mean f-measure of classifiers with 13 feature subsets from BGA,BPSO,BGWO using 3 different fitness functions

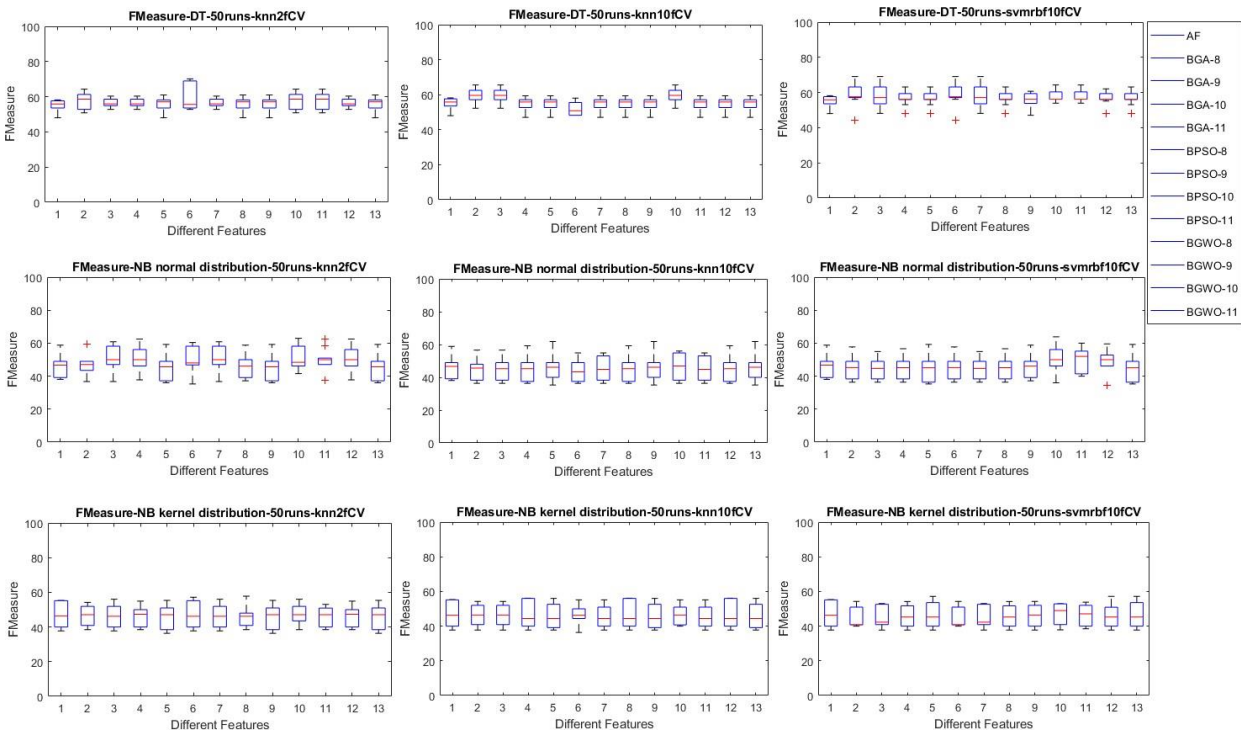


(a) F-Measure of SVM-rbf, SVM-linear, SVM-quadratic classifiers

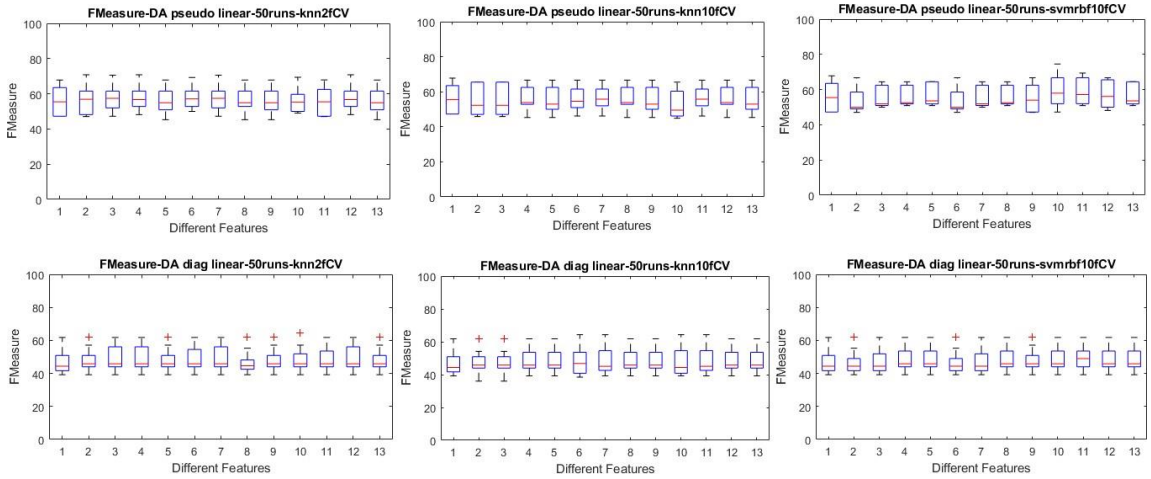


(b) F-Measure of FFANN, RF, KNN classifiers

Figure 5.12: F-Measure of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions



(c) F-Measure of DT, NB-nornal, NB-kernel classifiers



(d) F-Measure of DA pseudo linear,DA diag linear classifiers

Figure 5.12: F-Measure of 11 classifiers with 50 runs using 13 feature subsets from 3 fitness functions

CHAPTER VI: CONCLUSION

Analysis of medical images involves consideration of numerous attributes that constitutes both useful and redundant information. Selecting useful and best information from medical images would result in decreasing complexity of both time and computation of classifiers in categorizing into decease groups.

In this work brain MRI medical images are considered, from which 13 intense and texture features are extracted. Three meta heuristic feature selection algorithms namely Binary Genetic Algorithm, Binary Particle Swarm Optimization and Binary Grey Wolf Optimizer are considered from literature, which were stated to give better performance in feature selection on brain MRI classification. Three fitness functions namely K-nearest neighbor classifier with two variations of cross validation, two and ten fold, and Support Vector Machine classifier using Gaussian Radial Basis Function kernel with ten fold cross validation are considered. Feature subsets were selected based on the frequency of feature present across the 33 runs of the execution of each feature selection algorithm. Bases on that, 8,9,10,11 most repeated feature feature are selected and their performance are evaluated by using them to perform binary classification with 7 different classifiers. Hence, visualization and analysis of the results form this work becomes multidimensional.

The feature selected using svmrbf10cv fitness function have provided much confident results on majority of classifiers in all the evaluation metrics.For the classifiers which work best on non linear data sets, the performance of the results increased as the number of selected features are increased.

Mostly, increasing the number of features selected by Binary Grey Wolf Optimizer are producing better results and performance. This leads to the fact that the algorithm required much complex and higher search spaces in order to find the best optimal solution. Where as the performance exhibited by classifiers with feature subset consisting of 8 best features is similar to the 9, 10 and 11 feature subsets select by Binary Genetic Algorithm and Binary Particle Swarm optimization. The

performance of the feature subsets generated by 10 fold cross validation fitness functions are better when compared to ones generated by 2 fold cross validation fitness function.

Feature subsets that were selected using Support Vector Machine fitness function have produced higher performance than the ones using all features with Support Vector Machine classifier with Gaussian Radial Basis Function kernel. Features selected using non linear fitness function i.e., Support Vector Machine classifier with Gaussian Radial Basis Function kernel are providing better average results, but the confidence of the results when seen from box plots are slightly reduced.

The features subsets selected from K-nearest neighbor fitness functions are performing better than the ones with SVM fitness function. The feature selected by KNN fitness functions are also performing better on linear classifiers.

6.1 Future Work

This work has been dealt with a problem with less number of features, but more interesting patterns can be obtained with larger feature space. This brings the notion of extracting more new features from the images. This work has only considered the intensity and texture features for classifying the images, but they are number of shape features available which can be also used for building a higher dimensional feature set.

The data set that was used in this work only consisted of 253 images and with class imbalance. But with a good availability of brain images this work can be performed. Also the images considered in this work are only from MRI imaging technique, but there are different modalities too for brain scanning, which can be used, and the features obtained would be more generic and robust for the classifiers to train.

This work used some, from the many available meta-heuristic algorithms, which can also be used and compared with results from work. Also only two classifiers KNN and SVM using RBF kernel are used as fitness function in this work. But there is room for many different classifiers which can be used as fitness functions.

The hyper parameters used for the feature selection as well as classification algorithms are the

standard and most used ones from literature. But modification and optimization of which can open room for further research on this area.

Research on Deep Learning is high in recent times. It is a widely used method for image process activities. It also has the capability for performing feature extraction as well as selection, and this open the room for further research using Deep learning networks in this area.

REFERENCES

[1] Ahmed M Anter et al. “A robust swarm intelligence-based feature selection model for neuro-fuzzy recognition of mild cognitive impairment from resting-state fMRI”. In: *Information Sciences* 503 (2019), pp. 670–687.

[2] Setu Basak. *How to perform Roulette wheel and Rank based selection in a genetic algorithm?* 2018. URL: https://medium.com/@setu677/how-to-perform-roulette-wheel-and-rank-based-selection-in-a-genetic-algorithm-d0829a37a189 (visited on 07/04/2018).

[3] Rajesh S. Brid. *Introduction to Decision Trees*. Oct. 2018. URL: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb (visited on 10/26/2018).

[4] Bumghi Choi, Ju-Hong Lee, and Deok-Hwan Kim. “Solving local minima problem with large number of hidden nodes on two-layered feed-forward artificial neural networks”. In: *Neurocomputing* 71.16-18 (2008), pp. 3640–3643.

[5] Pao Hua Chou et al. “Application of back-propagation neural network for e-commerce customerspatterning”. In: *ICIC Express Letters* 3.3 (2009), pp. 775–785.

[6] Thomas Cover and Peter Hart. “Nearest neighbor pattern classification”. In: *IEEE transactions on information theory* 13.1 (1967), pp. 21–27.

[7] El-Sayed Ahmed El-Dahshan, Tamer Hosny, and Abdel-Badeeh M Salem. “Hybrid intelligent techniques for MRI brain images classification”. In: *Digital Signal Processing* 20.2 (2010), pp. 433–441.

[8] Manoranjan Dash and Huan Liu. “Feature selection for classification”. In: *Intelligent data analysis* 1.1-4 (1997), pp. 131–156.

[9] E Roy Davies. *Computer vision: principles, algorithms, applications, learning*. Academic Press, 2017.

[10] Kirthi Devleker. *Understanding Wavelets, Part 1: What Are Wavelets*. URL: https://www.mathworks.com/videos/understanding-wavelets-part-1-what-are-wavelets-121279.html.

[11] Russell Eberhart and James Kennedy. "Particle swarm optimization". In: *Proceedings of the IEEE international conference on neural networks*. Vol. 4. Citeseer. 1995, pp. 1942–1948.

[12] Rohith Gandhi. *Support Vector Machine — Introduction to Machine Learning Algorithms*. June 2018. URL: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 (visited on 06/07/2018).

[13] Rafael C Gonzales and Richard E Woods. *Digital image processing*. 2002.

[14] Shubham Gupta and Kusum Deep. "A novel random walk grey wolf optimizer". In: *Swarm and evolutionary computation* 44 (2019), pp. 101–112.

[15] Isabelle Guyon et al. "Gene selection for cancer classification using support vector machines". In: *Machine learning* 46.1-3 (2002), pp. 389–422.

[16] JS Hallinan. "Data mining for microbiologists". In: *Methods in Microbiology*. Vol. 39. Elsevier, 2012, pp. 27–79.

[17] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. "Textural features for image classification". In: *IEEE Transactions on systems, man, and cybernetics* 6 (1973), pp. 610–621.

[18] Changjun He et al. "Prediction of compressive yield load for metal hollow sphere with crack based on artificial neural network". In: *ICIC Express Letters* 3.4 (2009).

[19] Cost Helper Health. *How much does brain surgery cost?* 2019. URL: https://health.costhelper.com/brain-surgery.html.

[20] Sadegh Bafandeh Imandoust and Mohammad Bolandraftar. “Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background”. In: *International Journal of Engineering Research and Applications* 3.5 (2013), pp. 605–610.

[21] Mohammed J Islam et al. “Investigating the performance of naive-bayes classifiers and k-nearest neighbor classifiers”. In: *2007 International Conference on Convergence Information Technology (ICCIT 2007)*. IEEE. 2007, pp. 1541–1546.

[22] Mehdi Jafari and Reza Shafaghi. “A hybrid approach for automatic tumor detection of brain MRI using support vector machine and genetic algorithm”. In: *Global journal of science, engineering and technology* 3 (2012), pp. 1–8.

[23] Anil K Jain, Robert P. W. Duin, and Jianchang Mao. “Statistical pattern recognition: A review”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.1 (2000), pp. 4–37.

[24] JB Siddharth Jonathan and KN Shruthi. “A Two Tier Neural Inter-Network Based Approach to Medical Diagnosis Using K-Nearest Neighbor Classification for Diagnosis Pruning”. In: *web page available at http://infolab. stanford. edu/˜ jonsid/nimd. pdf* (2011).

[25] Nita Kakhandaki and SB Kulkarni. “A Novel Framework for Detection and Classification of Brain Hemorrhage”. In: ().

[26] Laveen N Kanal and Paruchuri R Krishnaiah. *Classification, Pattern Recognition, and Reduction of Dimensionality*. North-Holland Publishing Company, 1982.

[27] Bahram Karimi, Mohammad Bagher Menhaj, and Iman Saboori. “Multilayer feed forward neural networks for controlling decentralized large-scale non-affine nonlinear systems with guaranteed stability”. In: *Int. J. Innov. Comput., Inf. Control* 6.11 (2010), pp. 4825–4841.

[28] Saurav Kaushik. *Introduction to Feature Selection methods with an example (or how to select the right variables?)* Dec. 2016. URL: https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/ (visited on 12/01/2016).

[29] Ahmed Kharrat et al. “A hybrid approach for automatic classification of brain MRI using genetic algorithm and support vector machine”. In: *Leonardo journal of sciences* 17.1 (2010), pp. 71–82.

[30] Dong Seong Kim and Jong Sou Park. “Network-based intrusion detection with support vector machines”. In: *International Conference on Information Networking*. Springer. 2003, pp. 747–756.

[31] Marcin Kociołek et al. “Discrete wavelet transform-derived features for digital image texture analysis”. In: *International Conference on Signals and Electronic Systems*. 2001, pp. 99–104.

[32] Marcin Kociołek et al. “Discrete wavelet transform-derived features for digital image texture analysis”. In: *International Conference on Signals and Electronic Systems*. 2001, pp. 99–104.

[33] Will Koehrsen. *Chapter 5: Random Forest Classifier*. Dec. 2017. URL: https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d (visited on 12/27/2017).

[34] Arun Kumar, Alaknanda Ashok, and MA Ansari. “Brain Tumor Classification Using Hybrid Model Of PSO And SVM Classifier”. In: *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE. 2018, pp. 1022–1026.

[35] Pat Langley. “Selection of relevant features in machine learning”. In: *Proceedings of the AAAI Fall symposium on relevance*. 1994, pp. 1–5.

[36] Wen Long et al. “Inspired grey wolf optimizer for solving large-scale function optimization problems”. In: *Applied Mathematical Modelling* 60 (2018), pp. 112–126.

[37] Chao Lu, Liang Gao, and Jin Yi. “Grey wolf optimizer with cellular topological structure”. In: *Expert Systems with Applications* 107 (2018), pp. 89–114.

[38] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[39] Stephane G Mallat. “A theory for multiresolution signal decomposition: the wavelet representation”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 7 (1989), pp. 674–693.

[40] Kazuhiro Matsui, Yusuke Suganami, and Yukio Kosugi. “Feature selection by genetic algorithm for MRI segmentation”. In: *Systems and Computers in Japan* 30.7 (1999), pp. 69–78.

[41] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. “Grey wolf optimizer”. In: *Advances in engineering software* 69 (2014), pp. 46–61.

[42] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[43] Peifeng Niu, Songpeng Niu, Lingfang Chang, et al. “The defect of the Grey Wolf optimization algorithm and its verification method”. In: *Knowledge-Based Systems* 171 (2019), pp. 37–43.

[44] Savan Patel. *Chapter 5: Random Forest Classifier*. May 2017. URL: https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1 (visited on 05/18/2017).

[45] Tina R Patil, SS Sherekar, et al. “Performance analysis of Naive Bayes and J48 classification algorithm for data classification”. In: *International journal of computer science and applications* 6.2 (2013), pp. 256–261.

[46] VP Rathi and S Palani. “Brain tumor MRI image classification with feature selection and extraction using linear discriminant analysis”. In: *arXiv preprint arXiv:1208.2128* (2012).

[47] Atiq ur Rehman, Aasia Khanum, and Arslan Shaukat. “Hybrid Feature Selection and Tumor Identification in Brain MRI Using Swarm Intelligence”. In: *2013 11th International Conference on Frontiers of Information Technology*. IEEE. 2013, pp. 49–54.

[48] Javad Sadeghi, Saeid Sadeghi, and Seyed Taghi Akhavan Niaki. “Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: an improved particle swarm optimization algorithm”. In: *Information Sciences* 272 (2014), pp. 126–144.

[49] Yuehjen E Shao and Bo-Sheng Hsu. “Determining the contributors for a multivariate SPC chart signal using artificial neural networks and support vector machine”. In: *International Journal of Innovative Computing, Information and Control* 5.12 (2009), pp. 4899–4906.

[50] Qeethara Kadhim Al-Shayea. “Artificial neural networks in medical diagnosis”. In: *International Journal of Computer Science Issues* 8.2 (2011), pp. 150–154.

[51] V Sheejakumari and B Sankara Gomathi. “MRI brain images healthy and pathological tissues classification with the aid of improved particle swarm optimization and neural network”. In: *Computational and mathematical methods in medicine* 2015 (2015).

[52] Yuhui Shi et al. “Particle swarm optimization: developments, applications and resources”. In: *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*. Vol. 1. IEEE. 2001, pp. 81–86.

[53] Jonathon Shlens. “A tutorial on principal component analysis”. In: *arXiv preprint arXiv:1404.1100* (2014).

[54] Wojciech Siedlecki and Jack Sklansky. “On automatic feature selection”. In: *Handbook of Pattern Recognition and Computer Vision*. World Scientific, 1993, pp. 63–87.

[55] Suchada Tantisatirapong et al. “Magnetic resonance texture analysis: Optimal feature selection in classifying child brain tumors”. In: *XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013*. Springer. 2014, pp. 309–312.

[56] Cancer Research UK. *Recovering from brain tumor surgery*. May 2019. URL: https://www.cancerresearchuk.org/about-cancer/brain-tumours/treatment/surgery/recovering (visited on 08/05/2019).

[57] Alper Unler and Alper Murat. “A discrete particle swarm optimization method for feature selection in binary classification problems”. In: *European Journal of Operational Research* 206.3 (2010), pp. 528–539.

[58] V Vapnik. “Estimation of dependences based on empirical data”. In: *IEEE Transactions on PAMI* 14.9 (1992), pp. 211–222.

[59] Bing Xue, Mengjie Zhang, and Will N Browne. “Particle swarm optimization for feature selection in classification: A multi-objective approach”. In: *IEEE transactions on cybernetics* 43.6 (2012), pp. 1656–1671.

[60] Bahman ZareNezhad and Ali Aminian. “A multi-layer feed forward neural network model for accurate prediction of flue gas sulfuric acid dew points in process industries”. In: *Applied Thermal Engineering* 30.6-7 (2010), pp. 692–696.

[61] Yu-Dong Zhang and Lenan Wu. “An MR brain images classifier via principal component analysis and kernel support vector machine”. In: *Progress In Electromagnetics Research* 130 (2012), pp. 369–388.

[62] Jiayong Zhang and Yanxi Liu. “Cervical cancer detection using SVM based feature screening”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2004, pp. 873–880.

[63] Razia Zia. “Multi-resolution Tranaform Based Feature Extraction Techniques for Differentiating Glioma Grades Using MRI Images”. PhD thesis. National University of Science & Technology, Islamabad, 2018.