Optimal Cancer Classification of Microarray Data Using Different Optimization Techniques

By

Payal Patel

A thesis submitted in partial fulfillment of

the requirements for the degree of

Master of Science (MSc) in Computational Sciences

The Faculty of Graduate Studies

Laurentian University

Sudbury, Ontario, Canada

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne
Faculty of Graduate Studies/Faculté des études supérieures

Title of Thesis Titre de la thèse	Optimal Cancer Classification of Microarray Data Using Different Optimization Techniques
Name of Candidate Nom du candidat	Patel, Payal
Degree Diplôme	Master of Science
Department/Program Département/Programme	Computational Sciences
Date of Defence Date de la soutenance	September 17, 2019

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Kalpdrum Passi
(Supervisor/Directeur de thèse)

Dr. Ratvinder Grewal
(Committee member/Membre du comité)

Dr. Mazen Saleh
(Committee member/Membre du comité)

Dr. Gulshan Wadhwa
(External Examiner/Examinateur externe)

Approved for the Faculty of Graduate Studies
Approuvé pour la Faculté des études supérieures
Dr. David Lesbarrères
Monsieur David Lesbarrères
Dean, Faculty of Graduate Studies
Doyen, Faculté des études supérieures

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

Abstract

Cancer being one of the most vital diseases in the medical history needs adequate focuses on its cause, symptom and detection. Various algorithms and software have been designed so far to predict the disease at cellular level. The most crucial data for sorting the cancerous tissue is the classification of such tissues based on the gene expression data. Gene expression data consists of high amount of genetic data as compared to the number of data samples. Thus, sample size and dimensions are a major challenge for researchers. In this work, four different types of cancers are analyzed viz., breast cancer, lung cancer, leukemia and colon cancer. The analysis is done using various nature-inspired algorithms like Grasshopper Optimization (GOA), Interval Value Based Particle Swarm Optimization (IVPSO) and Particle Swarm Optimization (PSO). To study the accuracy of the data, five different classifiers are used – Random Forest, K-Nearest Neighborhood (KNN), Neural Network and Support Vector Machine (SVM). The comprehensive data analysis is done with the combination of these five classifiers over various datasets of each of the selected cancer type. After deep analyzing different combinations GOA outperformed for almost each dataset. The research work addresses the issue of dimensionality reduction and efforts towards improving accuracy.

Acknowledgments

On a remarkable note I am heartily thankful to my supervisor, **Dr. Kalpdrum Passi**, whose encouragement, supervision and support from the preliminary to the concluding level enabled me to develop an understanding of the subject. At the end, I offer my regards and blessings to all of those who supported us in any respect during the completion of the research and to our school for providing a resources and materials.

I humbly extend my thanks to all concerned person who give their time and attention for completion of this research.

Finally, I extend my gratitude to my family for their valuable help.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to Microarray

Microarray technology is one of the significant techniques that allow the diagnosis of various diseases, such as cancers by using gene expression data [4]. Genes are encoding regions that build basic forming block inside the cell and show the way to proteins, which are attaining a variety of functions.

In recent years, scientists have tried to anatomize a large amount of microarray data for obtaining relevant information that can be used in cancer classification. To achieve this goal, one can use decision tree (Quinlan, 1996) [28], Support Vector Machine (SVM) [6], K-Nearest Neighbor (KNN) [11], Neural Networks, statistical methods such as multinomial linear regression [8] as well as various sophisticated machine learning and pattern recognition methods [9].

The microarray dataset is structured in the form of an expression value matrix where each row represents a gene and each column represents distinct samples. Figure1 shows the microarray matrix g in which x, y is a numeric value representing the gene expression level of gene x in sample y. However, there exist some challenges for classification of gene expression data. The difficult challenge is the small number of samples in comparison with the massive number of features “genes” [19]. For such asymmetric dimension space, it is hard to use traditional classifiers directly. One way to address this situation is to use feature selection methods, which lower not only the redundancy but also the dimensionality of gene expression data [21].

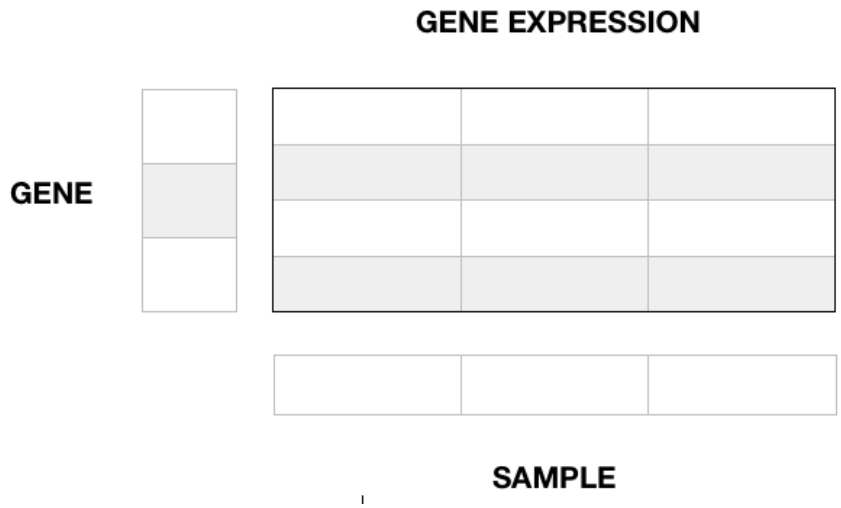

Figure 1: Gene expression microarray

1.2 Gene expression and tissue sample

Gene expression is the process, in which gene's data is used in the synthesis process of the functional gene product. This gene product is divided into two parts. One is proteins and the second is non-protein coding genes like transfer RNA (tRNA) or small nuclear RNA (snRNA). All known life like eukaryotes (including multicellular organisms), prokaryotes (bacteria), and viruses use the gene expression procedure to produce macromolecular machinery for life.

To identify, medical professionals recommend doing biopsy, which is the process of examining the extraction of the sample cells or tissues. This whole process is done by the pathologist using the tool called "microscope". Basically, biopsies are performed for the cancerous and inflammatory conditions. Biopsies can be done for the bone, bone marrow, breast, gastrointestinal tract, lung, liver, prostate, nervous system, urogenital system.

1.3 Classification Techniques

In this study, we use different classification techniques to predict cancer. These include:

- Support Vector Machine (SVM)
- K - nearest neighbor (KNN)
- Naive Bayes (NB)
- Neural Network (NN)
- Random Forest (RF)

Performances of these classifiers are compared using the following parameters:

- Accuracy

  Accuracy (ACC) is calculated as the number of all true predictions (TP + TN) divided by the total number of data sets (P + N) i.e., the summation of all true predictions (TP +TN) and all true false predictions (FN + FP). The highest accuracy is 1.0, while the worst is 0.0. It can also be calculated by 1-ERR.

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}} \quad (1)$$

- Sensitivity

  Sensitivity (SN) is calculated as the number of true positive predictions (TP) divided by the total number of positives (P). The highest sensitivity is 1.0, while the worst is 0.0.

$$\mathrm{SN} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = \frac{\mathrm{TP}}{\mathrm{P}} \quad (2)$$

- Specificity

  Specificity is calculated as the number of true negative predictions (TN) divided by the total number of negatives (N). It is also referred to as the true negative rate (TNR). The highest specificity is 1.0, while the worst is 0.0.

$$\mathrm{SP} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} = \frac{\mathrm{TN}}{\mathrm{N}} \quad (3)$$

- Area Under the Curve (AUC)

AUC curve is a classification problem performance measurement at different threshold settings. ROC is a curve of probability and AUC is a degree or separability measure. AUC is used in the analysis of classification to determine which of the models used best predicts the classes. ROC curves are an example of its application.

- PPV (Positive Predictive Value or Precision)

  PPV is calculated as the number of true positive predictions (TP) divided by the total number of positive predictions (TP + FP). The best precision or PPV value is 1.0, whereas the worst is 0.0.

$$\mathrm{PREC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4}$$

- NPV (Negative Predictive Value)

  NPV is calculated as the number of true negative predictions (TN) divided by the total number of negative predictions (TN + FN). The best NPV value is 1.0 and the worst is 0.0.

$$\mathrm{NPV} = \frac{\mathrm{TN}}{(\mathrm{FN}+\mathrm{TN})} \tag{5}$$

- FNR (False Negative Ratio)

  The summation of true positive and false negative (TP + FN). It can also be calculated as 1 – Sensitivity.

- FPR (False Positive Ratio)

  The summation of true negative and false positive (TN + FP). The best false positive rate is 0.0 while the worst is 1.0. It can also be calculated as 1 – Specificity.

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} = 1 - \mathrm{SP} \tag{6}$$

- Recall: is calculated as the number of true positive predictions (TP) divided by the total number of positives (P) i.e., the summation of true positive (TP) and false negative (FN). The highest recall value is 1.0, while the worst is 0.0.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (7)$$

Depending on these performance parameters, the best classifier is selected.

1.4 Optimization Techniques

In our study, we use optimization techniques to optimize the data to achieve higher accuracy in classification. Optimization techniques used are:

- Grasshopper Optimization Algorithm (GOA)
- Particle Swarm Optimization (PSO)
- Interval Valued Particle Swarm Optimization (IVPSO)

Grasshopper optimization technique is a population-based optimization algorithm inspired by grasshoppers. While Interval Valued Particle Swarm Optimization is a kind of swarm optimization technique. Both have been discussed thoroughly in the chapter 4**.**

1.5 Objectives of the thesis

In this thesis we analyze four types of cancer data and their classification. Datasets of different types of cancer were taken from UCI repository which have collections of thousands of microarray datasets. In this research, we implemented three optimization algorithms namely Grasshopper Optimization (GOA), Particle Swarm Optimization (PSO) and Interval Valued Particle Swarm Optimization (IVPSO) in conjunction with different classifiers such as Support Vector Machine (SVM), Naive Bayes (NB), Random Forest, K-Nearest-Neighbor (KNN), and Neural Network and also 10-Fold cross validation is being used further in each classification method with different training-testing ratios to predict better accuracy for each dataset. The main objective is to find the best optimization technique among GOA, PSO and IVPSO in conjunction with the five classifiers to achieve the best accuracy and AUC values for the four cancer microarray datasets.

The selection of algorithm depends behavior of classification and optimization techniques. Among stochastic optimization approaches, nature-inspired, population-based algorithms are the most popular. Such techniques mimic natural problems-solving methods, often those used by creatures. Machine Learning (ML) is coming into its own, with a growing recognition that ML can play a key role in a wide range of critical applications, such as data mining, natural language processing, image recognition, and expert systems.

Time and memory are also an important entity to determine the performance of any classification algorithm. Time complexity is a feature that describes how much time an algorithm takes to the algorithm in terms of the quantity of input. "Time" can mean the number of memory accesses performed, the number of integers compared, the number of times that some internal loop is performed, or some other natural unit related to the amount of real time that the algorithm takes.

Space complexity is a feature that describes the amount of memory (space) that an algorithm requires to the algorithm in terms of the quantity of input. We often talk about the extra memory needed to store the input itself, not counting the memory needed. Sometimes spatial complexity is ignored because the space used is minimal and evident, but sometimes it becomes a problem as significant as time.

The thesis is organized as follows:

We present the literature review in Chapter 2. In Chapter 3, different datasets and classification methods are presented. Optimization algorithms will be discussed in Chapter 4. Chapter 5 will show tools and techniques. In Chapter 6 we discuss the results. Chapter 7 concludes and discussed future work.

Chapter 2

Literature Review

With the invention of the microarray technique, scientists and researchers have immense opportunity to evaluate the expression levels of thousands of genes concurrently in a single experiment. In Ghorai et al. [1], the Nonparallel Plane Proximal Classifier (NPPC) was proposed for cancer classification in a Computer Aided Diagnosis (CAD) framework to ensure high classification accuracy and to minimize the computation time. But Valvular heart disorders were one of the most challenging classification problems. Sengur et al. [2] used three powerful and popular ensemble learning representative called bagging, boosting, and random subspaces to early detect Valvular heart disorders. However, the classification time was minimized using methods, but the rate at which the accuracy was said to be attained remained unaddressed. In Costa et al. [3], three Generalized Mixture (GM) functions were applied via dynamic weights to improve the classification accuracy of the classification system. Though the service handles single label classification, multi-label classification problem was not addressed.

Huda et al. [7] proposed a case study for brain tumor diagnosis using global optimization-based hybrid wrapper filter feature selection with ensemble classification methods.[4]. It increases the classification accuracy, but the classification time was not minimized. In an average, 40% of the world's population is affected by cancer. A Proportion SVM was used by Hussein et al. [5] for efficient categorization of Lung Nodules, which results in the improved diagnosing accuracy. The proportion of SVM failed to minimize the error rate in disease categorization. Using Radial Basis Function Neural Network with Affine Transforms, which in turn achieved high classification accuracy and low mean square error. But the feature extraction performance was not improved. A review of feature selection and parallel classification systems was carried out by Jain et al. [7] to

enhance the classification accuracy for disease prediction, but classification time was not minimized. A Critical assessment of ANN was carried out in Dander et al. [8] which increases the efficacy and specificity of the diagnostic techniques, but it fails to minimize the computational complexity. Tumor tissue based on pathological evaluation is one of the most pivotal for early diagnosis in cancer patients. However, the automated image analysis methods have the potential to improve the accuracy of disease diagnosis and to minimize human errors. Khosravia et al. [9] proposed different computational methods using convolution neural networks (CNN), where a stand-alone pipeline was constructed expertly to classify several histopathology images across different types of cancer. But it fails to minimize the computation cost while classifying the various types of cancer.

Sharma et al. [10] proposed a two-stage hybrid ensemble classification technique to increase the prediction accuracy of chronic kidney disease with the ML technique. It improves the disease diagnosis, but the multistage classification was not performed with minimum time. Early diagnoses of lung cancers and differentiation between the tumor types and non-tumor types have been required to improve the patient survival rate. In Hosseinzadeh et al. [11], a diagnostic system with structural and physicochemical attributes of proteins via feature extraction, feature selection, and prediction models was designed. Then, the ML models were applied to both original and newly created database *to predict the lung cancer type of tumors, which results in improved accuracy. However, the model reduces the processing time, but the false positive rate was not minimized. Evaluation of ML algorithm for lung cancer diagnosis was carried out by Podolsky et al. [12]. It accurately predicts cancer vulnerability as well as minimizes the false positive rate. But the classification time was not exploited, which can be helpful for early lung cancer detection.

Rabbani et al proposed a narrative review based on radiometric features to help diagnose lung cancer in an early stage [13], where the ML algorithms were combined with artificial intelligence approches. A systematic review of mortalities and survival rate of lung cancer with evolutionary algorithms was conducted by Dubey et al. [15] to identify a better method for early lung cancer diagnosis and to

achieve higher accuracy rate with deep learning techniques. It does not minimize the error rate. Liu et al. [16] proposed a Multi-view Convolutional Neural Networks (MV-CNN) for efficient lung nodule classification, to improve the accuracy, and the classification time. Here, accurate detection was not performed with the features. Baz et al. [17] explored some critical challenges and methodologies with the CAD system for lung cancer. It increases the detection and diagnosis of lung nodules, but the proper feature selection was not performed to minimize the detection time.

Deep feature fusion and hand-crafted features for lung nodule classification were developed by Wang et al. [18]. But classification performance was not accurate. CAD was introduced for enhancing the performance of nodule candidate classification by Chen et al. [19]. However, classification time was not minimized. To effectively classify the lung nodules, in-depth features were extracted in CT images with higher accuracy by Kumar et al. [20]. But the error rate remained unaddressed. Image-based feature selection method was developed for classifying the lung cancer images with higher accuracy by Baranidharan et al. [21]. In this method, the novel fusion-based selection was used to select the features for classification. During the feature selection, the redundant features were not removed, thus introducing an error in the classification process. Data analysis of population statistics and data mining techniques were used in [22] to determining the cancer morbidity and mortality data in a regional cancer registry. However, the false positive rate was not minimized. Various aspects of large-scale knowledge mining were covered in [23] for medical and diseases examination. A new image-based features selection method was planned in [24] to categorize the lung computed tomography images with higher accuracy.

Existing studies primarily adopt artificial neural networks (ANNs), decision tree analysis (DTA), Naïve Bayes (NB), Support Vector Machines (SVM), and so on. Due to the neural network has the advantage of capturing the correlations between attributes. Therefore, it has been widely utilized for breast cancer diagnosis (Lundin et al., 1999 [25]; Ravdin & Clark, 1992 [26]; Yao, 1999 [28]). Liu,

Wang, and Zhang (2009) [9] designed a decision tree prediction model for breast cancer survivability and adopted an under-sampling method to balance the training data, and the results have shown that when the ratio is equal to 15%, the AUC of the model is 0.7484. On top of the decision tree algorithm, Quinlan (1996) [27] introduced MDL-inspired penalty and designed an improved C4.5 decision tree algorithm for breast cancer prediction and attained the prediction accuracy of 94.74%. However, the performance of a single learning classification algorithm can't reflect the Interactive factors of the breast cancer survival and recurrence rate (Wang, Zheng, Yoon, & Ko, 2018) [31], Therefore, to overcome the drawbacks brought by single algorithm, various hybrid algorithms have been proposed. Akay (2009) [33] presents F-score method for feature selection and SVM for breast cancer prediction. On top of that, another hybrid algorithm presented by Chen et al. (2011) [35] designed a hybrid classifier with a rough set for feature selection and SVM for classification.

Zheng, Yoon, and Lam (2014) [36] proposed K-means and SVM hybrid algorithm for breast cancer diagnosis, K-means method for breast tumor feature extraction and SVM for classification. In another study, Onan (2015) [37] designed a hybrid intelligent classification model for breast cancer diagnosis, which consists of fuzzy-rough approach for instance selection, consistency-based for feature selection and fuzzy-rough nearest neighbor algorithm for breast tumor classification.

Also, Sheikhpour, Sarram, and Sheikhpour (2016) [39] proposed PSO and nonparametric kernel density estimation (KDE) based classifier to diagnose breast cancer. To summarize, their results showed that the proposed hybrid models had achieved high classification accuracy with fewer feature variables. Nevertheless, to the best of our knowledge, in medical diagnosis, comparing the cost of misclassifying a cancerous patient as a noncancerous to misclassifying a noncancerous patient as harmful, the consequences vary significantly. Therefore, this study constructs a hybrid intelligent classification model which has a competitive performance compared to other existing methods. The main advantage of our proposed classification model is that it can achieve not only the minimum misclassification cost but also obtain the maximum classification accuracy with fewer input feature.

Chapter 3

Datasets and Methods

3.1 Dataset Selection

3.1.1. Leukemia dataset

The leukemia dataset was taken from leukemia patient samples revealed by Golub et. al., (1999) [64]. This dataset frequently fills in as a benchmark for microarray examination techniques. It contains quality articulations relating to intense lymphoblast leukemia (ALL) and intense myeloid leukemia (AML) tests from bone marrow and fringe blood. The dataset comprised of 72 tests in which 49 tests of ALL and 23 tests of AML. Each sample is estimated more than 7,129 genes. A framework with 7130 genes (7129 genes demonstrates the quality articulations while the last gene reports the relating test's class mark), and from all 72 samples tested and categorized as two types.

1. Acute lymphoblast leukemia sample (ALL).
2. Acute myeloid leukemia sample (AML).

3.1.2 Lung Cancer Dataset

Dataset was taken from UCI repository [67]. Hong and Young used this dataset to describe optimal classification even in large number of attributes compared to the number of samples.

Class label is represented by attribute 1. In the dataset there are 56 attributes and number of samples is 32, and all data have integer attribute type.

3.1.3. Colon Cancer Dataset

Colon cancer microarray data was downloaded from UCI repository [68]. In the dataset the file ‘names’ contain the description of the 2000 genes. In file tissues there is an identity of 62 tissues. The

numbers compare the patients, where a positive sign attributes to an ordinary tissue, and a negative sign indicated a tumor tissue.

The Colon Cancer Dataset was first in use in 1999 having 62 samples, including 20 normal samples and 40 tumor samples, collected from patients of colon-cancer.

The matrix contains the expression of the 2000 genes with highest minimal intensity across the 62 tissues. The genes are placed in order of descending minimal intensity. Each entry in I2000 is a gene intensity derived from the ~20 feature pairs that correspond to the gene on the chip

The file ‘names’ contains the EST number and description of each of the 2000 genes, in an order that corresponds to the order in I2000. Note that some ESTs are repeated which means that they are tiled a number of times on the chip, with different choices of feature sequences.

3.1.4 Breast cancer dataset

We have collected this data from UCI repository for the Breast Cancer dataset [69].

Significant Information:

Samples arrive periodically as Dr. Wolberg reports his clinical cases. Integer is taken attribute character type. In this dataset there are 699 instances; 2 were removed from the dataset. It also contains 11 attributes which are described below.

1. Sample code number: id number
2. Clump thickness: 1-10
3. Uniformity of cell size: 1-10
4. Uniformity of cell shape: 1-10
5. Single epithelial cell size: 1-10
6. Bare Nuclei: 1-10
7. Bland Chromatin: 1-10
8. Normal Nucleoli: 1-10

9. Mitoses: 1-10
10. Class: (2 for benign, 4 for maligant)

3.2 Classification methods

Classification algorithms are supervised methods which means that the data is already labelled and they perform prediction of the classes by assigning a categorical label to the current class. The classification performance is measured in several parameters like Accuracy, Sensitivity, Specificity, Area Under the Curve (AUC), PPV (Positive Precision Value), NPV (Negative Precision Value), FNR (False Negative Ratio), FPR (False Positive Ratio). Graphical assessment ways area under the curve (AUC), precision and recall curves offer different interpretations of the classification performance.

Training and testing data

Dataset is divided into training and testing to train the model for better prediction of new data. Training data includes both expected output and input data. In training, algorithm learns how to apply different machine learning technologies and produce complex results, so that it can give more precise decision. It uses in various fields like natural language processing, sentiment analysis, etc.

On other hand, testing data is used to examine how well your algorithm was trained and evaluate model features [69].

3.3 Classification Methods

3.3.1 Support Vector Machine

The goal of support vector machine is to search hyperplane in an N-dimensional space that uniquely classifies the data points. There are many possible hyperplanes that could be possible to separate two classes. Our main objective is to find optimal hyperplane that has maximum margin.

Maximizing the margin distance offers some strengthening in order to be able to classify future information points with greater confidence [69].

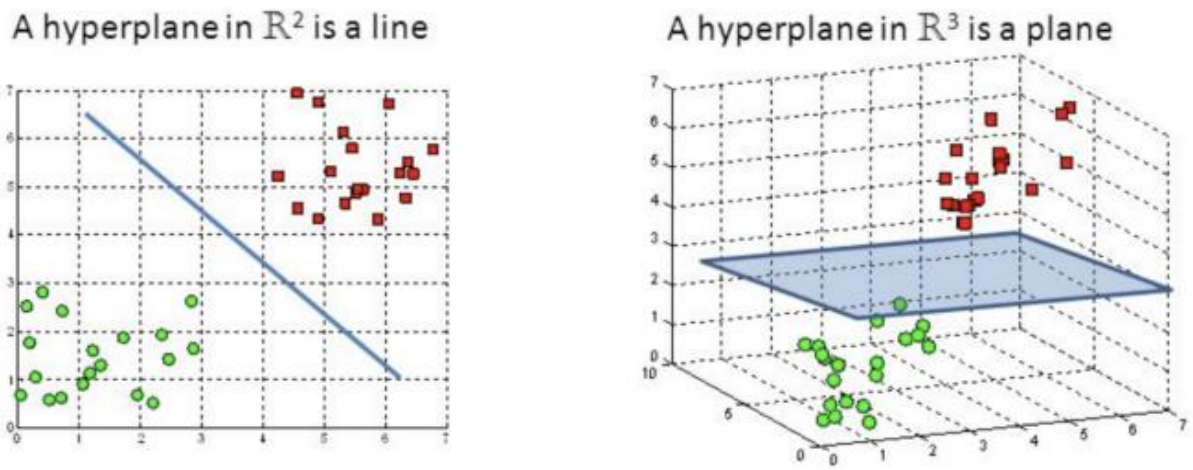

Figure 2: A hyperplane in different dimensional space [69]

In this algorithm, each data item is plotted as a point in n-dimensional space (where n is the number of characteristics) with the value of each function being the value of a specific coordinate. It uses a nonlinear mapping to transform the training data into a higher dimension. In this new dinsion, a linear optimal separating hyperplane separates the data points into two classes. If the number of features of the input is 2, the hyperplane is only a line (Figure 2). If the amount of feature entered is 3, then a two-dimensional plane becomes the hyperplane. When the amount of feature exceeds 3, it becomes hard to imagine.

Support vectors are information points nearer to the hyperplane and affect the hyperplane's position and orientation. We maximize the classifier's margin by using these support vectors. The removal of the support vectors will change the hyperplane's position. These are the points that help us to construct our SVM.

Advantages of SVM classifier

- SVMs are effective when the number of features is quite large.
- It works effectively even if the number of features is greater than the number of samples.

- Non-Linear data can also be classified using customized hyperplanes built by using kernel trick.
- It is a robust model to solve prediction problems since it maximizes margin.

Disadvantages of SVM classifier

- Support Vector Machine's greatest limitation is the kernel decision. The kernel's incorrect selection can lead to a rise in error percentage.
- With a larger number of samples, poor performance begins [69].
- SVMs have excellent efficiency in generalization, but in the test stage they can be highly slow.
- SVM have high algorithmic complexity and extensive memory requirement due to the use of quadratic programming.

3.3.2 Random Forest

Random forest algorithm is a supervised algorithm for classification. As the name suggests, with a amount of trees, this algorithm produces the forest. The more trees in the forest the more robust the forest appears. Similarly, the greater number of trees provides more precision in the random forest classification.

Following steps shows the pseudo code of random forest [70].

1) Randomly select “k” features from total “m” features.

   Where k << m.

2) Among the “k” features, calculate the node “d” using the best split point.
3) Split the node into sibling nodes using the best split.
4) Repeat 1 to 3 steps until “l” number of nodes has been reached.
5) Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.

Algorithm starts with randomly selecting k number of features out of m features. In the next step we are using randomly selected k feature and find the root note by selecting best split approach [70]. We will be calculating sibling nodes using same approach. Until first 3 stages, with a root node we form a tree and having leaf node as a target. We repeat above 4 stages to create n random tree and this tree forms the random forest.

Advantages of random forest

- It overcomes the problem of overfitting by combining results of different decision trees.
- If there is a large number of data missing in dataset, algorithm tries to get better accuracy.
- There is no need of scaling in this algorithm, it is flexible and maintains high accuracy.
- Instead of single decision tree it works well with large number of datasets.

Disadvantages of random forest

- High complexity and more computational resources are main disadvantages of the algorithm.
- Prediction process is very time-consuming compared to other algorithms.
- Construction of random forest is much harder than making decision tree.

3.3.3 Neural Network

The Artificial Neural Network (ANN) derives from human brain activity. The construction of the neural network involves three different layers with feed forward architecture. This is the most popular network architecture in use today. The input layer of this network is a set of input units, which accept the elements of input feature vectors [84].

Widely used for image processing and speech processing, neural network technique has gained enough attention in the field of data mining. The Neural Networks train the models by tuning the weights after several iterations and also tuning the biases so as to minimize the loss function of data. A neural framework, as a rule, incorporates a gigantic number of processors working in parallel and engineered in levels [58]. The chief dimension gets the unrefined information - undifferentiated from

optic nerves in human visual planning. Every dynamic dimension gets the yield from the dimension going before it, rather than from the rough information - likewise neurons further from the optic nerve get signals from those closer to it. The last dimension makes the yield of the system.

Advantages of Neural Network

- Neural networks are flexible and can be used for regression as well as classification. Any information that can be numerical value can be used in the model as a mathematical model with approximation function.
- Once the neural network is trained, predictions are quite fast.
- It works well with more data points.
- Nonlinear information with a big amount of inputs are good for modeling neural network. For example, images. It works by splitting the problem of classification into a layered network of simple components.

Disadvantages of neural network

- It is very time consuming and expensive for traditional processors.
- Neural networks depend on training dataset, which leads to the problem of overfitting and generalization.
- In neural network we don't know about how independent variable affect the dependent variable. It is like a black box.

3.3.4 K Nearest-Neighbor Algorithm

KNN is a learning algorithm that is non-parametric and lazy. Non- parametric [85] implies that the fundamental distribution of information is not assumed. In other words, from the dataset, the model structure is determined. This will be very helpful where most mathematical theoretical assumption are not followed by the actual world dataset.

Lazy algorithm implies that there is no need for model generation training information points.

K is the number of closest neighbors in KNN. The number of neighbors is the key factor that decides. K is usually an unusual number when the number of classes is 2. In Figure 3, if K=1, the algorithm is recognized as the nearest neighbour in algorithm.

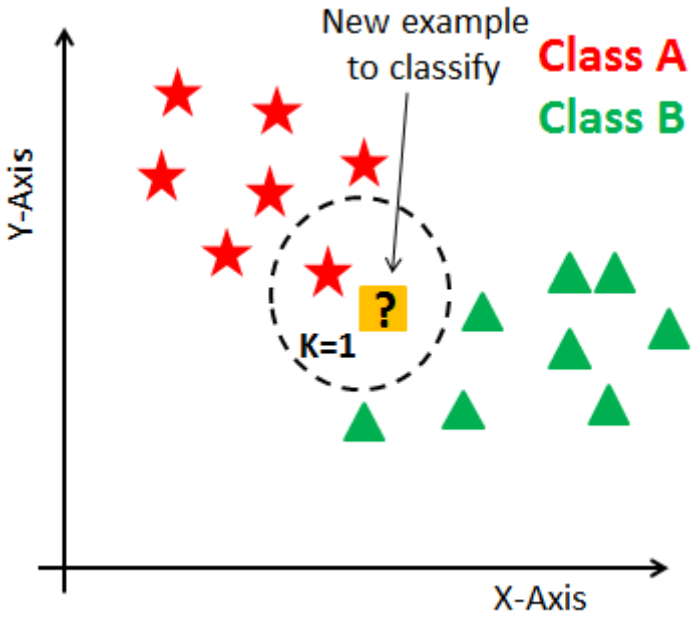

Figure 3: Determine K-nearest point (Avinash Navlani, August 2018) [85]

Suppose P1 is the point that must be predicted for the label. First, discover the k closest to P1 and then classify their k neighbors by majority vote. Each object vote is taken as the prediction for their class and the class with the most votes defines the label of the class.

Advantages of KNN

- **No Training Period**: KNN is called Lazy Learner (Instance based learning). It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g. SVM, Linear Regression etc.

- **Easy to implement:** There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc).
- **Easy to add new data:** since the KNN algorithm requires no training before making predictions, new data can be added seamlessly which will not impact the accuracy of the algorithm.

Disadvantages of KNN:

- **Doesn't work well with high dimensions**: It becomes difficult for algorithm to measure the distance in each dimension.
- **Doesn't function well with a big dataset**: The cost of calculating distance between the recent point and each existing point is huge which reduces the performance of the algorithm.
- **Need feature scaling:** Before implementing KNN algorithms to any dataset, we need to perf orm feature standardization and normalization. KNN produces wrong prediction without first normalizing data.
- **Sensitive to noisy data:** we need to add missing values and remove outliers.

3.3.5 Naive Bayes

Naïve Bayes is a method of statistical classification based on the theorem of Bayes. It is one of the easiest learning algorithms that is monitored [86]. The Naïve Bayes classifier is the algorithm that is quick, precise and reliable. On big datasets Naïve Bayes classifiers have high precision and velocity. Naïve Bayes classifier assumes that independent, i.e. there is no dependence of relations between the attributes. This hypothesis reduces computation cost which is why it is deemed naïve. This hypothesis is called independence conditional class.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \quad (8)$$

In equation (8), P(h) is the prior probability of hypothesis h being true, P(D) is the probability of data. P(h|D) is the posterior probability of hypothesis h given the data D. P(D|h) is the probability of data d if hypothesis h was true. Calculate the probability of class labels in prior. Find likelihood of each class with each attribute. Put this value in the formula of Bayes and calculate the posterior probability. See which class is more likely given that the input belongs to the greater class of probability.

Advantages of Naïve Bayes

- Naïv**e** Bayes is easy to understand and interpret, it also useful to predict the class for test dataset.
- It is effective for binary as well as multiclass prediction.
- If input is numeric variable, then classifier performs well.

Disadvantages of Naïve Bayes

- The model will assign a 0 (zero) probability and will not be able to predict if the categorical variable has a category (in test information set) that was not present in the training data set. This sort of error is often referred to as "zero frequency."
- Naïve Bayes classifier suffers from multicollinearity.
- This classifier is feasible for few categories of variable.

Chapter 4

Optimization Techniques

4.1. Introduction to optimization techniques

Optimization plays essential role in scientific research, management, and industry because numerous real-world problems can be primarily modeled as optimization tasks. “Traditional” mathematical programming methods (e.g., gradient-based methods) are no longer completely effective in solving complex optimization problems characterized by multi-modality, discontinuity, and noise. Different kinds of population-based optimization algorithms (POAs) have been emerging as promising alternatives in response to these challenges.

In POAs, multiple individuals search the solution space cooperatively and globally with operators and mechanisms like mutation, crossover, selection, information sharing, and learning. Besides, randomness is usually embedded into one or more operators so that POAs possess the capability to escape local optimal points and better explore the search space. Compared with other optimization algorithms, the essential characteristics of POAs could be three-fold. First, it searches the solution space through multiple points (solutions or individuals) simultaneously. Second, they have mechanisms for information sharing and interactive learning between individuals with different search behaviors. Third, POAs are stochastic as randomness is usually incorporated into the search behaviors, including mutation, crossover, selection, and others. The workflow of a POA is shown in Figure 4.

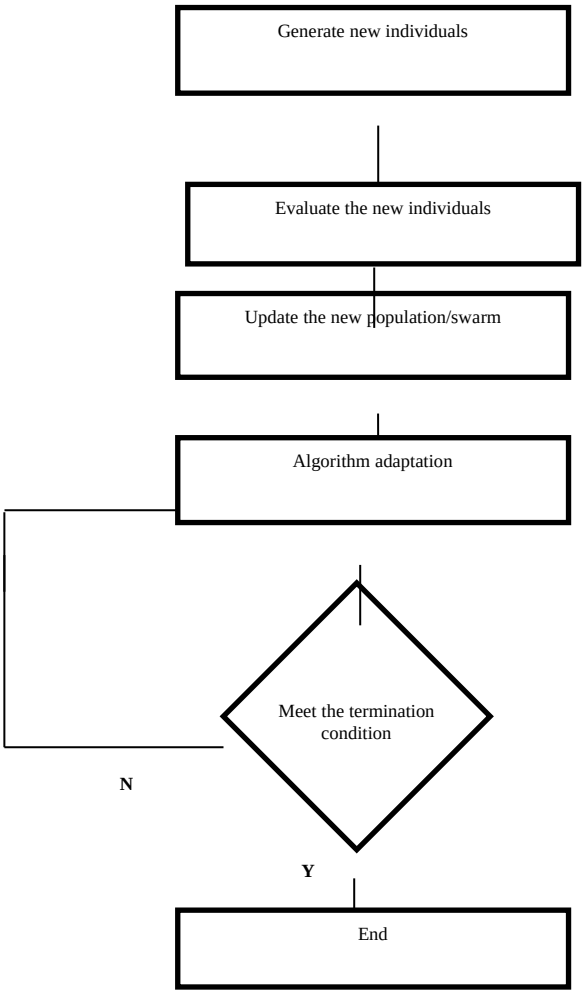

Figure 4: The workflow of a POA

4.2 Types of optimization techniques

POAs could be roughly categorized into evolutionary algorithms (EAs) and swarm intelligence Algorithms (SIAs). Classical and popular EAs include Genetic Algorithm (GA) [35], Evolution Strategy (ES) [35], Evolution Programming (EP) [36] and Differential Evolution (DE) [37]. Besides, popular SIAs include Ant Colony Optimization (ACO) [38], Particle Swarm Optimization (PSO)[39],

Artificial Colony Bee (ABC) [40], Artificial Immune Systems (AIS) [41], Across Neighborhood Search (ANS) and several others. These POAs are well-known for their ability to solve optimization problems with multiple objectives and constraints that may not always be continuous and differentiable and may be characterized by chaotic disturbances, randomness, and complex non-linear dynamics. POAs being stochastic "Generate-and-Test" algorithms only need the objective and constraint function values and do not require information regarding their characteristics. Thus, POAs are particularly suitable for Black-box optimization.

Numerous methods and strategies have been proposed to enhance the precision, time efficiency, and robustness of POAs. The ensemble strategy is one of the most promising approaches and has resulted in many efficient and versatile POA variants for unconstrained single objective numerical optimization, constrained optimization, multi-objective optimization, inching, etc. Motivations of using ensemble strategies in the design of POAs can be summarized as follows:

First, the No Free Lunch (NFL) [89] Theorem states that theoretically there exists no algorithm which is superior to the other algorithms in solving all possible optimization problems. In practice, NFL theorem indicates that it is impossible to design an algorithm (e.g., population-based optimization algorithm) that is more effective than all other algorithms in solving different optimization problems with different characteristics. However, researchers are always devoted to developing versatile POAs that are suited to many kinds of optimization problems. Tailor an existing algorithm to meet their requirements. Furthermore, to select an efficient one from thousands of candidate algorithms is time-consuming. This sounds a frustrating contradiction. Is it possible that there exists a Particle and useful algorithm that can deal with a set of optimization problems of different characteristics? Our answer is yes. This is because although the NFL tells that there is no algorithm being efficient for all possible optimization problems, in practice, the problems considered are always a subset of the all. Hence, it is quite possible to develop a POA that is efficient for this specific problem subset. How to design such a versatile POA is precisely a vital concern of the algorithm researchers. Ensemble strategy

provides a useful tool and paradigm to implement versatile POAs. In an ensemble POA, there are multiple search operators (e.g., mutation), parameters, constraint handling techniques or neighborhood structures, which generally have different characteristics and capabilities (e.g., exploration and exploitation) and therefore are suited to different types of optimization problems. As a result, a sophisticatedly designed ensemble POA with useful and distinguished ensemble components is potentially able to deal with different kinds of optimization problems. Note that, in this study, we call these constituent search operators, parameters, constraint handling techniques or neighborhood structures in one ensemble as components of the ensemble.

Second, to improve the probability of finding the optimal solution for a hard optimization problem complex landscapes, it is better to use a different search (sampling) approaches. The same search approach may make the POA follow similar trajectories and be trapped in one of the local optima. Therefore, ensemble of multiple strategies could make POA particularly efficient for complicated optimization problems (e.g. composition functions of CEC 2014 single objective optimization benchmark) [43]. In addition, fixed strategies may not be most appropriate for the entire search process. Therefore, during the optimization process the search strategy needs to be updated to suit the search process as the search landscape changes during the population evolution towards the optimal global solution. Ensemble of multiple strategies with proper adaptation mechanism could enable a POA to have a higher probability to select the most appropriate strategy during the optimization process [44].

Moreover, the ensemble could also make search strategies of different capabilities support each other thus significantly strengthen the performance of a POA. For example, in an ensemble, exploration search strategies could find more unvisited promising areas which could be further refined with exploiting search strategies [45].

Third, given an optimization problem, the quality of the solution to the problem and convergence speed of an algorithm to the optimal solution highly depends on the parameter and search strategy

configuration of the optimization algorithm. Configuring the parameters and search strategies of a POA refers to finding the best combination of operators, parameter values and search strategies before or during the optimization process to maximize the performance of the algorithm on the given problem. Algorithmic configuration can be performed before or during the optimization process. When the algorithmic configuration is conducted before the optimization process; it is usually referred to as tuning the algorithm. On the other hand, when the algorithmic configuration is conducted during the optimization process, it is referred to as adapted operator, parameter, and strategy control. It should be noted that traditional algorithmic configuration is to search for a fixed combination of strategies, operators and parameter values. However, traditional algorithmic configuration methods based on trial and error are usually time-consuming inefficient and incapable of changing during evolution.

Adapted control approaches dynamically adjust the operators, parameters, and strategies of POAs according to the optimization states and have attracted much attention recently [46]. The ensemble of a set of promising candidate parameters values and strategies can properly realize the adapted control of parameters, operators and strategies, thereby alleviating the burden of configuring and selecting parameters and strategies for POAs.

The ensemble concept for a POA can be defined as a combination of different strategies, operators, parameter values and methods (with a single or parallel population) referred to as an ensemble can provide better results on a set of optimization problems compared to a single set of strategies, operators, parameter values, and methods. It should be noted that in machine learning, researchers use the concept of ensemble learning where multiple diverse models are combined to form an ensemble. For instance, the neural network ensemble is a successful learning paradigm where a collection of a finite number of neural networks trained for the same task are combined to achieve a better performance on the same task and has successful applications in diverse areas [47]. The basic idea of ensemble learning is based on the intuition that a group of ‘unstable and diverse’ learning

algorithms can be combined to obtain an overall better learning algorithm. The extension of the ensemble concept to stochastic population-based optimization aims to construct a "stable" optimization algorithm by an appropriate combination of "unstable and diverse" stochastic optimization algorithms.

It should be noted that there are some other concepts related to the ensemble, including multi-strategy [48], multi-method [49], algorithm portfolios [50], hyper-heuristic [51], memetic algorithm [52] and hybrid algorithm [53]. In this survey, we take the concepts of multi-strategy and multi-method under the umbrella of the ensemble. This is because multi-strategy can be viewed as a special case of the low-level ensemble of algorithmic components, while multi-method can be viewed as special cases of high-level ensemble of different EA variants. In contrast, the term of the memetic algorithm (MA) was proposed by Moscato [54] as being a paradigm for integrating EAs with one or more refinement methods [55]. In population-based EAs (e.g., DE, PSO, and ABC) are taken as global search techniques while refinement methods (e.g., Nelder-Mead simplex search method, Hill Climber with Sidestep and trust-region derivative-free methods) play the role of local search [56]. In response to the challenge that the performance of an algorithm may vary significantly from problem to problem, algorithm portfolio attempts to find a less risky way to distribute the time among multiple different algorithms [57]. Hyper-heuristic is an approach which, when given a specific issue occasion or a class of examples, and a few low-level heuristics (or their segments), consequently creates a sufficient blend of the gave parts to successfully illuminate the given problem(s) [58]. Burke et al. classified hyper-heuristic into heuristic choice and heuristic are [59]. It can be perceived that hyper-heuristic aims to develop new search algorithms based on a pool of automatically low-level heuristics. The hybrid algorithm is a broader concept that may cover the other interrelated ones. Nevertheless, there is still a difference between ensemble algorithm and the hybrid algorithm. For example, a differential evolution algorithm with an ensemble of multiple mutation strategies is an ensemble algorithm while it would not be classified as a hybrid algorithm. On the other hand, a particle swarm optimizer

combined with a crossover operator is a traditional hybrid algorithm instead of an ensemble algorithm while simultaneous usage of multiple crossover strategies will make it a hybrid ensemble method.

Ensemble POAs have attracted much attention and resulted in encouraging achievements during the last decade. Different ensemble strategies have been proposed for a low-level ensemble of algorithmic components including multiple search strategies, parameter values, etc. as well as high-level ensemble of multiple EA variants. Ensemble strategies have been incorporated into differential evolution (DE) [59], particle swarm optimization (PSO) [60], artificial bee colony (ABC) [61], biogeography-based optimization (BBO) [62] and so on. Also, ensemble strategies are widely applied to different optimization areas, such as bound constrained single-objective optimization [63], constrained optimization [64], multi-objective optimization [65], dynamic optimization 192 [66], multi-modal optimization [67].

4.2.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) was created in 1995 by authors Kennedy and Eberhart [83], affected by the conduct of social creatures in communities like bird and fish schooling or ant colonies. This technology emulates the connection of communication between creatures. The PSO algorithm is population-based algorithm oriented on the simulation of bird's social behavior within flock. In PSO, individuals are "flown" through hyperdimensional search space, referred to as particles. Changes in individuals' particle position within search space based on psychological tendency to evaluate individuals' success compare to another particle.

A PSO algorithm maintains a particle swarm where each particle is a potential solution. A swarm is comparable to a population, while a particle is comparable to an individual. The particles are simply “flown” through a multidimensional search space, where each particle’s location is changes according to their own knowledge and neighbor knowledge.

In optimization and in integration with other current methods, particle swarm optimization has been implemented to various fields. This technique plays out the search for the optimum arrangement by methods for particles, whose directions are balanced by a stochastic and deterministic way. Each particle is affected by its ' pBest ' position and the ' gBest ' position of the swarm but continues to search for optimum solution. A particle i is denoted by its vector of location, xi and matrix of velocity, each iteration, each particle changes its place depending on the current velocity. The social contribution to particle velocity, with regard to the velocity equation, is proportional the distance between a particle and the best position achieved by the particle neighborhood. Vnew = updated velocity

At every iteration (equation 9), each particle shifts its place depending on the current velocity.

$$v_i(t+1) = v_i(t) + c_1 r_1[y_i(t) - x_i(t)] + c_2 r_2[g_i(t) - x_i(t+1)] \quad (9)$$

where the i is represented as index of particle, $v_i(t)$ is the velocity of particle i at time t, $x_i(t)$ is the position of particle at time t; c_1 (social parameter) and c_2 (cognitive parameter) are coefficients between 0 and 1.

In equation 10, $x_i(t+1)$ denote the position of particle i in the next time step t in search space by adding a velocity, $v_i(t+1)$, to the present position, the particle location is changed.

$$x_i(t+1) = x_i t + v_i(t+1) \quad (10)$$

The personal best position associated with particle y_i, is the best place that has been visited by the particle since the first iteration. In view of minimizing problems, the personal best position is calculated as the next iteration, t + 1 (equation 11, 12).

$$y_i(t+1) = y_i(t) \qquad \text{If } f(x_i(t+1)) \geq f(y_i(t)) \quad (11)$$

$$y_i(t+1) = x_i(t+1) \qquad \text{If } f(x_i(t+1)) < f(y_i(t)) \quad (12)$$

In figure 5 shows the graphical representation of pBest to calculate the next iteration.

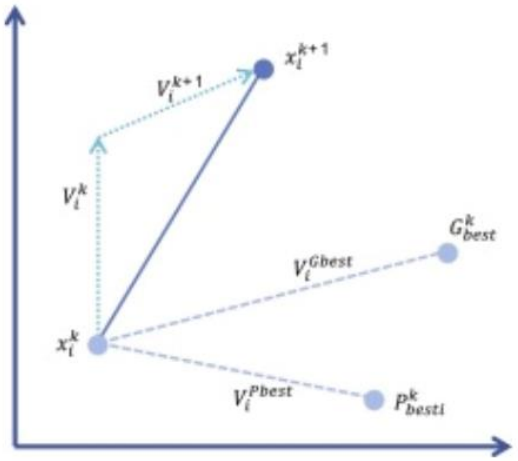

Figure 5: graphical presentation of finding the pBest

Algorithm 1 shows the pseudo code for PSO. I have taken this pseudo code from [107].

```
For each particle
    Initialize particle
End

Do
    For each particle
        Calculate fitness value
        If the fitness value is better that the best fitness value(pbest) in history
        Set current value as the new pBest
End
Choose the particle with the best fitness value of all the particles as the gBest
For each particle

    Calculate particle velocity according equation (9)
    Update particle position according equation (10)
end
```

4.2.2 Interval Value Based Particle Swarm Optimization (IVPSO)

The particle swarm optimization is a computational method which optimizes a problem by continuously trying to enhance a candidate solution with regard to a given measure of quality. In every iteration process, each candidate solution is calculated by the objective function being optimized, deciding the fitness of that solution. Every particle preserves its position, composed of the candidate solution and its evaluated fitness, and its velocity.

Algorithm IVPSO

1. Initialize N number of swarm particles with

   x_i position and v_i velocity of each particle. Let pBest be the best position

   of particle i and gbest as the best known position in the swarm.
2. Particle's initial position is x_i.
3. Take i=1, 2,…..,N for each particle
4. For every particle calculate fitness value.
5. If fitness value is better than the best fitness value (pBest)
6. Position the new pBest as current value.
7. Until a criterion of termination has been met.
8. Among all the particle select the best fitness value as gBest
9. For each particle calculate particle velocity,

   $v_i(t+1) = w*v_i(t)+ c_1 r_1 [(t)-x_i(t)] + c_2 r_2 [g(t)-x_i(t)]$

   where the i is represented as index of particle, $v_i(t)$ is the velocity of particle i at time t, at time t position of particle is $x_i(t)$, x, c_1, c_2 are coefficients.
10. Update particle position $x_i(t+1) = x_i(t)+v_i(t+1)$.

Until some stopping conditions are fulfilled.

4.2.3 Grasshopper Optimization algorithm (GOA)

Grasshopper Optimization Algorithm is a recent nature-inspired algorithm that mimics the swarming behavior of grasshoppers in nature (Saremi, Mirjalili & Lewis, 2017) [91]. The original version of GOA was designed for solving continuous optimization problems (Saremi et al., 2017) [90]. However, many optimization problems have discrete decision variables and search space. In this paper, new binary versions of GOA are proposed and designed to solve the FS problem. In the first two approaches, two transfer functions that belong to two families (i.e., S-shaped and V-shaped) are used to convert the continuous solutions in GOA to binary ones.

Characteristic of GOA:

Grasshoppers are insects that are considered as plant-eaters. Grasshoppers are usually seen individually in nature; however, when they join in one swarm, which may consist of millions of grasshoppers, they become a severe pest; due to their damage to crops, pasture and grain (Simpson, McCaffery & Haegele, 1999) [87]. The life cycle of a grasshopper consists of three stages; egg, nymph, and adult. The grasshopper can be found in the swarm in nymph or adulthood life phases (Rogers, Matheson, Despland, Dodgson, Burrows & Simpson, 2003) [89]. The swarm in the nymph phase is characterized by slow movement with small steps by the grasshoppers. By contrast, the swarm in the adult phase is characterized by sudden and long-distance moves. Seeking food sources is a critical characteristic of the swarming grasshoppers.

Exploration and exploitation are the two main phases in the nature-inspired algorithms that aim to improve the convergence speed and local optima avoidance in the algorithm when searching for a target. In the exploitation process, the search agents tend to move locally in the search space. In contrast, they are encouraged to move abruptly during the exploration process. Grasshoppers perform

these two processes as well as naturally seeking the target (food source). Equation (6) presents the simulation of the swarming behavior of grasshoppers (Saremi et al., 2017) [91].

$X_i = r1 * S_i + r2 * G_i + r3 * A_i$ (13)

where X_i defines the position of the i^{th} grasshopper, S_i is the social interaction defined in Equation (13). A_i shows the wind advection Equation (14), and r1, r2, and r3 are random numbers in the interval [0,1].

$$s_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} s\left(d_{ij}\right)\hat{d}_{ij} \quad (14)$$

where d_{ij} is the Euclidean distance between the i^{th} and the j^{th} grasshopper that is calculated as $d_{ij} = |x_j - x_i|$, d_{ij} is a unit vector from the i^{th} grasshopper to the j^{th} grasshopper, s is defined as the strength of social forces calculated as in Equation (15).

$$s(r) = fe^{\frac{-r}{l}} - e^{r} \quad (15)$$

Where f indicates the intensity of attraction, and l is the attractive length scale. More details about the impact of these two parameters (i.e., attraction and repulsion) on the social behaviors of artificial grasshoppers can be found in the original GOA paper (Saremi et al., 2017) [91]. The authors presented extensive experiments to study the behavior of the grasshoppers with different values of l and f, and they found that the repulsion occurs between any two grasshoppers if the distance between them is in the interval [81]. G_i is the gravity force on the i^{th} grasshopper in Equation (16).

$$G_i = -g\hat{e}_g \quad (16)$$

Where g is the gravitational constant and $\hat{e}_g$ shows a unity vector towards the center of the earth.

$$A_i = u\hat{e}_w \quad (17)$$

Where u is a constant drift and $\hat{e}_w$ is a unit vector in the direction of the wind. Nymph grasshoppers have no wings, so their movements are highly correlated with wind direction. Equation (18) presents how to determine the new position of the ith grasshopper based on its current position, the position of all other grasshoppers, and the position of the target (food source).

$$X_i^d = c_1\left(\sum_{j=1,\, j\neq i}^{N} c_2 \frac{ub_d - lb_d}{2} s\left(|x_j^d - x_i^d|\right)\frac{x_j - x_i}{d_{ij}}\right) + \widehat{T}_d \tag{18}$$

Where u_{bd} is the upper bound in the Dth dimension, l_{bd} is the lower bound in the Dth dimension, $s(r) = fe^{\frac{-r}{l}} - e^r$; $\hat{T}_d$ is the value of the Dth dimension in the target (best solution found so far), and c is a decreasing coefficient to shrink the comfort zone, attraction region and repulsion region. Note that S is like the S component in Equation (15). However, the gravity component G has not been considered, and the wind direction A has been assumed towards the target $\hat{T}_d$. In Equation. (18), the adaptive parameter c has been used twice to simulate the deceleration of grasshoppers approaching the source of food and eventually consuming it. The outer c_1 is used to reduce the search coverage toward the target grasshopper as the iteration count increases, while the inner c_2 has been used to reduce the effect of the attraction and repulsion forces between grasshoppers proportionally to the number of iterations.

The mathematical formulations of GOA are capable of efficiently exploiting and exploring the search space. However, it still needs a mechanism for tuning the exploration and exploitation degree during the search process. In nature, the nymph grasshoppers have no wings, so they locally search for foods by moving in the surrounding area. However, adult grasshoppers fly freely in the air and explore a much larger scale region. In population-based optimization techniques, however, exploring the search space comes first; due to the need for finding promising areas. Later, the exploitation process is applied to locally intensify the search for finding better solutions that lead to the global optimum. For controlling the degree of exploratory and exploitative behaviors in GOA, the parameter c should be

decreased proportionally to the number of executed iterations. This mechanism increases the degree of exploitation as the iteration count increases. It reduces the comfort zone proportionally to the number of iterations as well. Parameter c is calculated as in equation (19).

$$c = cmax - l\ \frac{cmax - cmin}{\mathrm{L}} \tag{19}$$

Where cMax is the maximum value of parameter c, cMin is the minimum value of parameter c, l is the current iteration number, and L is the maximum number of iterations.

The pseudo code given in Algorithm 1 is taken from [91] and used in this research. In algorithm 1, at the beginning of the search process, GOA creates a set of random initial solutions (grasshoppers) and calculates the fitness for each of them. The grasshoppers update their positions based on Equation (18). In each iteration, the position of the best grasshopper (target) that is obtained so far is updated. Furthermore, in each iteration, the parameter c is calculated using Equation. (19) and the distances between grasshoppers are normalized between 1 and 4. Updating the position of grasshoppers is performed iteratively until the stopping criterion is satisfied. Finally, the best grasshopper (target) is returned which represents the best approximation for the global optimum. I have taken pseudo code from [91].

Algorithm 2: Pseudo code of the GOA algorithm. Initialize cmax, cmin and *Max*_Iterations Initialize a population of solutions X_i (i = 1, 2, . . . , n) Evaluate each solution in the population Set **T** as the best solution

while *t* <*Max*_Iterations **do**

Update *c* using Equation. 8 **for** each solution **do**

Update the location of the current solution using Equation. 6

Bring the current grasshopper back if it goes outside the boundaries

Update **T** if there is a better solution in population t=*t*+1.

Return T

Chapter 5

Tools and Techniques

5.1 Tool

We have used Microsoft excel for the storage of the dataset. For the analysis and implementation of algorithms, we have used MATLAB 2016.

Microsoft Excel

Microsoft Excel is a spreadsheet for Windows, macOS, Android and iOS developed by Microsoft. It functions calculation, graphing instruments, pivot tables, and a Visual Basic for Applications micro programming language. For these systems, particularly since release 5 in 1993, it has been a very commonly implemented spreadsheet and has substituted Lotus 1-2-3 as the sector norm for spreadsheets. Excel is component of the computer package of Microsoft Office.

Microsoft Excel has the fundamental characteristics of all spreadsheets to arrange information manipulations such as arithmetic operations using a matrix of cells organized in ordered lines and letter-named columns. To meet statistical, manufacturing and economic requirements, it has a battery of provided features. It can also show information as row graphs, histograms and charts with a very restricted 3D graphical display. It enables information sectioning to display its dependencies for distinct views on distinct variables (using pivot tables and situation manager). It has a programming element, Visual Basic for Applications, which enables the customer to use a broad range of numerical methods.

MATLAB

MATLAB is a high-performance computing framework. It integrates computation, visualization, and programming in a user-friendly setting where recognizable mathematical notation expresses issues and answers. Typical applications include:

- Math and computing production
- Algorithm modeling,
- Modeling and prototyping
- Data assessment,
- Exploring and visualization Scientific and manufacturing design
- Design of applications including graphical user interface construction.

MATLAB is an intuitive framework whose fundamental information component is a cluster that does not require dimensioning. This enables you to tackle numerous specialized processing issues, particularly those with grid and vector definitions, in a small amount of the time it would take to compose a program in a scalar noninteractive language, for example, C or Fortran.

The name MATLAB represents lattice lab. MATLAB was initially composed to give simple access to framework programming created by the LINPACK and EISPACK ventures, which together speak to the cutting edge in programming for network calculation.

MATLAB has advanced over a time of years with contribution from numerous clients. In college conditions, it is the standard instructional device for early on and propelled courses in arithmetic, designing, and science. In industry, MATLAB is the instrument of decision for high-profitability research, advancement, and examination.

MATLAB highlights a group of use explicit arrangements called tool compartments. Important to most clients of MATLAB, tool stash enable you to learn and apply innovation. Tool kits are exhaustive accumulations of MATLAB

capacities (M-documents) that stretch out the MATLAB condition to take care of specific classes of issues. Zones in which tool stash are accessible incorporate sign handling, control frameworks, neural systems, fluffy rationale, wavelets, recreation, and numerous others.

In each part, the results differed.

5.2 10- fold cross validation

Cross-validation is a method for frequent holdout to improve. Cross-validation is a comprehensive method to do continuous holdout that effectively enhances it by decreasing the estimation variance. We take a training set and a classifier is created. Then we're looking to assess that classifier's efficiency, and there's a certain level of variance in that assessment because it's all underneath the statistics. We want to maintain the difference as small as feasible in the assessment. Cross-validation is a method to reduce the variance, and it is further reduced by a cross-validation version called "stratified cross-validation."

We split it only once with cross-validation, but we split dataset into 10 parts. Then we bring 9 of the parts and use them to train, and we use the last item to test. Then we bring another 9 parts with the same separation and use them for practice and experimentation with the hold-out item. We do the whole process 10 occasions, each moment we use a distinct section to test. In other cases, we split the dataset into 10 parts, and then we keep each piece in turn for monitoring, training on the remainder, monitoring, and averaging the 10 outcomes. That would be a "cross-validation of 10 times."

Divide the dataset into 10 components, keep each portion in turn, and evaluate the outcomes. Therefore, each data point in the dataset is used for testing once and for training nine times. That's a cross-validation of 10 times. By standard, Weka does stratify cross-validation. Weka invokes the learning algorithm 11 occasions with 10-fold cross-validation, once for each cross-validation unit, and then a third moment for the whole dataset.

5.3 Hybrid Approach

This thesis proposes an approach for prediction in microarray cancer data. The proposed methodology includes optimization algorithm with different classifiers. Interval value-based particle swarm optimization (IVPSO) uses different classifiers to evaluate precise prediction on various datasets. In proposed methodology firstly IVPSO combine the results of local search problem and extracted information from data is encoded in different classifiers with optimal parameters by attempting each combination of parameter for better accuracy.

In GOA approach evaluating best fitness function for each grasshopper and update the new position for it with respect to global optimal and particle's best position in given search problem. Obtained results of fitness function used in different classifier with appropriate parameters for better prediction on each dataset.

Third proposed algorithm is PSO in which particles affected by best position in swarm of particle whose velocity and weight of each iteration, in each iteration particle shifts depending on the current velocity. Different classifiers with best combination of parameters used for computing better accuracy in classification prediction. Accuracy of all experimental result computed by averaging resultant accuracies from all 10-folds. The results mentioned in discussion section shows the effectiveness of hybrid model.

5.4 Ratio Comparison

Comparison of ratio is the ratio taken using cross validation.

In our research, we have used ratios as follows:

1. **90:10** where 90% is training set and 10% is testing
2. **80:20** where 80% is training set and 20% is testing
3. **70:30.** where 70% is training set and 30% is testing
4. **60:40.** where 60% is training set and 40% is testing.

Chapter 6

Results and Discussion

6.1 Process and experiment

Various experiments are done on the available datasets to attain high classification accuracy. The performance was evaluated based on five different classifiers: Random Forest, Neural Network, KNN and SVM. The training-testing ratios are 90:10, 80:20, 70:030 and 60:40.

6.2 Results on Leukemia Dataset

The proposed model is tested on the dataset of Leukemia. The dataset is directly downloaded from UPI repository. For detailed analysis, different training-testing ratios are considered over the entire range and based on algorithms applied over them, results are concluded. Table 1 shows the parameters like accuracy, sensitivity, specificity, precision, recall, AUC, positive predictive value, negative predictive value, false negative rate and false positive rate. The comparison ratio is derived from the cross validation as per the Leukemia results for the training test ratios. We use IVPSO, GOA and PSO algorithms with the five classifiers to compare with the different parameters for the ratio comparison.

Table 1: Leukemia results for training-testing ratio 90:10

	ACCURACY	Sensitivity	Specificity	ROC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (90-10)**									
KNN	59.72%	0.5	0.6	0.485	0.55556	0.54545	0.5	0.44444	0.52632	64.5
Naïve Bayes	6923%	0.7	0.2	0.3	0.46667	0.4	0.3	0.53333	0.56	31
SVM	62.5%	0.8	0.4	0.43	0.57143	0.66667	0.2	0.42857	0.66667	39
Random forest	51.38%	0.6	0.7	0.605	0.66667	0.63636	0.4	0.33333	0.63158	59.5
NN	57.95%	0.68434	0.47475	0.53502	0.56576	0.60064	0.31566	0.43424	0.61943	-1.0045
GOA	**Training-Testing (90-10)**									
KNN	63.88%	0.5	0.5	0.435	0.5	0.5	0.5	0.5	0.5	63

Naïve Bayes	76.92%	0.5	0.8	0.59	0.71429	0.61538	0.5	0.28571	0.58824	77
SVM	62.5%	0.8	0.4	0.435	0.57143	0.66667	0.2	0.42857	0.66667	31.5
Random forest	58.33%	0.6	0.8	0.68	0.75	0.66667	0.4	0.25	0.66667	55.5
NN	50.88%	0.54419	0.47348	0.47711	0.50825	0.50951	0.45581	0.49175	0.52561	-1.0122
PSO	**Training-Testing (90-10)**									
KNN	81.56%	0.5	0.5	0.453	0.5564	0.54463	0.5	0.4445	0.5677	64
Naïve Bayes	89.48%	0.6	0.6	0.43	0.48556	0.5966	0.4	0.3455	0.5744	65
SVM	87.06%	0.7	0.3	0.446	0.56443	0.6677	0.2	0.43556	0.6655	35
Random forest	56.73%	0.6	0.6	0.639	0.64337	0.6767	0.4	0.3555	0.63445	57
NN	55.34%	0.65447	0.46553	0.52667	0.52334	0.57446	0.35466	0.43356	0.57446	-1.0443

In Table 2, For the training-testing ratio 80:20, highest accuracy is 97.38% in PSO optimization technique with SVM classifier. Highest PPV is in Naïve Bayes classifier is 0.69231. SVM and Random Forest shows highest sensitivity which is 0.7 in GOA and IVPSO technique respectively. Naïve Bayes algorithm have lowest False Negative results compare to other classifier.

Table 2: Leukemia results for Training-Testing ratio 80:20

	ACCURACY	Sensitivity	Specificity	ROC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (80-20)**									
KNN	55.55%	0.6	0.6	0.65	0.6	0.6	0.4	0.4	0.6	51
Naïve Bayes	61.53%	0.9	0.6	0.735	0.69231	0.85714	0.1	0.30769	0.78261	31.5
SVM	61.11%	0.3	0.6	0.26	0.42857	0.46154	0.7	0.57143	0.35294	62.5
Random forest	66.66%	0.6	0.7	0.665	0.66667	0.63636	0.4	0.33333	0.63158	51.5
NN	53.72%	0.61237	0.46212	0.48894	0.53238	0.54383	0.38763	0.46762	0.56958	1.039
GOA	**Training-Testing (80-20)**									
KNN	63.88%	0.6	0.5	0.465	0.54545	0.55556	0.4	0.45455	0.57143	49.5
Naïve Bayes	76.92%	0.6	0.6	0.48	0.6	0.6	0.4	0.4	0.6	46
SVM	63.88%	0.6	0.7	0.56	0.66667	0.63636	0.4	0.33333	0.63158	64
Random forest	52.77%	0.7	0.6	0.65	0.63636	0.66667	0.3	0.36364	0.66667	53.5
NN	55.80%	0.61995	0.49621	0.51889	0.55169	0.56628	0.38005	0.44831	0.58383	0.947
PSO	**Training-Testing (80-20)**									
KNN	97.22%	0.6	0.5	0.533	0.57	0.4557	0.4	0.4077	0.58663	49
Naïve Bayes	81.01%	0.9	0.5	0.64877	0.65	0.75446	0.3	0.3577	0.75449	45
SVM	97.38%	0.4	0.5	0.4566	0.45366	0.54336	0.5	0.4355	0.45779	63.5
Random forest	60.45%	0.6	0.6	0.65	0.65788	0.65443	0.4	0.35667	0.63441	52
NN	55.43%	0.61445	0.47563	0.49667	0.54223	0.55467	0.38445	0.45339	0.56448	0.975

In Table 3, For the leukemia results SVM classifier outperformed among all classifier in each optimization techniques. In GOA, SVM classifier gives highest accuracy is 97.593%, it also gives highest sensitivity which is 0.9. False negative ration is very low in SVM with value 0.1 which shows that classifier is more accurate. For Random Forest in GOA technique False positive ratio is also lower with 0.125 and sensitivity is higher which is 0.9.

Table 3: Leukemia results for training-testing ratio 70:30

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (70-30)**									
KNN	55.55%	0.55556	0.5	0.4	0.365	0.45455	0.44444	0.5	0.54545	47.619
Naïve Bayes	76.92%	0.8	0.4	0.45	0.57143	0.66667	0.2	0.42857	0.66667	34.5
SVM	97.50%	0.7	0.5	0.545	0.58333	0.625	0.3	0.41667	0.63636	41.5
Random forest	55.55%	0.5	0.8	0.62	0.71429	0.61538	0.5	0.28571	0.58824	78.5
NN	54.29%	0.58586	0.5	0.49857	0.53953	0.54696	0.41414	0.46047	0.56174	0.94213
GOA	**Training-Testing (70-30)**									
KNN	63.88%	0.4	0.5	0.36	0.44444	0.45455	0.6	0.55556	0.42105	56.5
Naïve Bayes	69.23%	0.5	0.7	0.54	0.625	0.58333	0.5	0.375	0.55556	63
SVM	97.53%	0.9	0.5	0.63	0.64286	0.83333	0.1	0.35714	0.75	32.5
Random forest	51.38%	0.7	0.9	0.76	0.875	0.75	0.3	0.125	0.77778	51
NN	56.12%	0.62247	0.5	0.50817	0.55456	0.56978	0.37753	0.44544	0.58656	0.94243
PSO	**Training-Testing (70-30)**									
KNN	56.74%	0.5447	0.5	0.39	0.4456	0.45454	0.5543	0.5443	0.45663	54
Naïve Bayes	72.49%	0.6	0.7	0.42	0.58664	0.6554	0.4	0.39554	0.57884	40
SVM	96.34%	0.8	0.5	0.6	0.63556	0.7554	0.3	0.39554	0.65774	35
Random forest	53.75%	0.6	0.85	0.75	0.85445	0.70554	0.4	0.25334	0.64557	75
NN	55.39%	0.58664	0.5	0.50445	0.54663	0.54337	0.3557	0.45566	0.57884	0.94556

In Table 4, For the training-testing ratio 60:40, GOA gives more accurate result for Naïve Bayes classifier. While IVPSO gives more accurate result in the classifier of SVM which is 99.28%. By comparing all the classifier, IVPSO gives more precise output than others. Lowest FNR is 0.2 in Naïve Bayes classifier with IVPSO which shows more accurate prediction of results in classifier. In GOA, Random Forest gives "1" sensitivity and highest AUC which is 0.7.

Table 4: Leukemia Results for training-testing ratio 60:40

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (60-40)**									
KNN	58.33%	0.8	0.3	0.345	0.53333	0.6	0.2	0.46667	0.64	34.5
Naïve Bayes	46.15%	0.5	0.6	0.495	0.55556	0.54545	0.5	0.44444	0.52632	65
SVM	99.28%	0.5	0.7	0.505	0.625	0.58333	0.5	0.375	0.55556	64.5
Random forest	54.16%	0.6	0.6	0.505	0.6	0.6	0.4	0.4	0.6	55.5
NN	54.86%	0.59596	0.50126	0.50285	0.54441	0.5537	0.40404	0.45559	0.56902	-0.93684
GOA	**Training-Testing (60-40)**									
KNN	56.94%	0.6	0.4	0.38	0.5	0.5	0.4	0.5	0.54545	42.5
Naïve Bayes	69.23%	0.4	0.9	0.53	0.8	0.6	0.6	0.2	0.53333	72.5
SVM	98.00%	0.6	0.6	0.48	0.6	0.6	0.4	0.4	0.6	61
Random forest	58.33%	1	0.6	0.7	0.71429	1	0	0.28571	0.83333	46.5
NN	47.79%	0.48232	0.47348	0.42523	0.4781	0.47771	0.51768	0.5219	0.4802	1.0123
PSO	**Training-Testing (60-40)**									
KNN	57.67%	0.65	0.4	0.35	0.52	0.6	0.3	0.4666	0.59664	37.3
Naïve Bayes	53.86%	0.47	0.8	0.52	0.675	0.5664	0.5	0.35554	0.5223	70.6
SVM	99.17%	0.68	0.65	0.49	0.6	0.5433	0.4	0.455	0.57885	65
Random forest	56.49%	0.7	0.5	0.6	0.6547	0.65	0.3	0.4332	0.7544	48.3
NN	52.37%	0.54734	0.49753	0.47	0.5233	0.8	0.4335	0.47665	0.5344	-1.0123

6.3 Results on Lung Cancer Dataset

For three different algorithms, the lung cancer data is evaluated. The dataset is directly taken from UPI repository. Analysis is done based on parameters like accuracy, sensitivity, specificity, precision, recall, area under the AUC curve, positive predictive value, negative predictive value, false negative rate and false positive rate. In Table 5 the IVPSO has given more accurate data than the PSO and GOA in the classifier KNN which is 0.75. For the Naive Bayes classifier, IVPSO is again better than other techniques. While GOA has given more accurate data in the SVM, random forest and NN classifiers which is 0.99577, 0.6875 and 0.61976 respectively. By comparing all these results, we can say GOA is more accurate and precise than other classifier.

Table 5: Lung cancer for 90:10 training-testing ratio

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (90-10)**									
KNN	75%	0.6	0.6	0.605	0.6	0.6	0.4	0.4	0.6	60.5
Naïve Bayes	83.33%	0.6	0.8	0.6	0.75	0.66667	0.4	0.25	0.66667	49
SVM	99.16%	1	0.6	0.71	0.71429	1	0	0.28571	0.83333	38.5
Random Forest	53.12%	0.7	0.4	0.34	0.53846	0.57143	0.3	0.46154	0.6087	51
NN	59.88%	0.475	0.71264	0.49571	0.60317	0.59615	0.525	0.39683	0.53147	2.0229
GOA	**Training-Testing (90-10)**									
KNN	60.71%	0.4	0.4	0.205	0.4	0.4	0.6	0.6	0.4	51.5
Naïve Bayes	33.33%	0.7	0.6	0.6	0.63636	0.66667	0.3	0.36364	0.66667	55.5
SVM	99.57%	0.6	0.6	0.485	0.6	0.6	0.4	0.4	0.6	53
Random Forest	68.75%	0.7	0.4	0.315	0.53846	0.57143	0.3	0.46154	0.6087	42.5
NN	61.97%	0.51875	0.71264	0.55159	0.62406	0.61692	0.48125	0.37594	0.56655	2
PSO	**Training-Testing (90-10)**									
KNN	72.33%	0.5	0.6	0.4065	0.5	0.6	0.5	0.4	0.45	57.5
Naïve Bayes	55.67%	0.6	0.7	0.6	0.7348	0.6677	0.4	0.34968	0.6677	53
SVM	99.01%	0.8	0.6	0.5664	0.5342	0.7	0.3	0.354	0.7459	45
Random Forest	55.39%	0.7	0.4	0.3255	0.58343	0.57231	0.3	0.4596	0.6056	47
NN	60.77%	0.4865	0.7123	0.53446	0.61452	0.58463	0.5134	0.37465	0.5456	1.9945

In Table 6, For the lung cancer, 80:20 training-testing ratio, highest accuracy is 93.49% in SVM classifier in GOA. In Table 6 Random Forest shows fewer number of false predictive values compared to other classifiers. Highest sensitivity in Naïve Bayes classifier is 0.8 in IVPSO technique. In Table 6, Naïve Bayes with GOA technique gives “0” False positive result with sensitivity of "1" and Positive Predictive value “1”.

Table 6: Lung cancer data for 80:20 training-testing ratio

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (80-20)**									
KNN	64.28%	0.4	0.7	0.435	0.57143	0.53846	0.6	0.42857	0.47059	72
Naïve Bayes	66.66%	0.8	0.5	0.535	0.61538	0.71429	0.2	0.38462	0.69565	40
SVM	93.42%	0.6	0.7	0.57	0.66667	0.63636	0.4	0.33333	0.63158	61.5
Random Forest	53.12%	0.7	0.7	0.7	0.7	0.7	0.3	0.3	0.7	44.5

NN	60.03%	0.47813	0.71264	0.51172	0.60474	0.59759	0.52187	0.39526	0.53403	2.0057
GOA	**Training-Testing (80-20)**									
KNN	75%	0.7	0.3	0.29	0.5	0.5	0.3	0.5	0.58333	32.5
Naïve Bayes	50%	0.4	1	0.54	1	0.625	0.6	0	0.57143	67.5
SVM	93.49%	0.3	0.7	0.4	0.5	0.5	0.7	0.5	0.375	68
Random Forest	71.87%	0.5	0.7	0.595	0.625	0.58333	0.5	0.375	0.55556	49.5
NN	59.43%	0.46563	0.71264	0.51874	0.59839	0.59189	0.53438	0.40161	0.52373	2.0019
PSO	**Training-Testing (80-20)**									
KNN	73.44%	0.5	0.4	0.3564	0.56342	0.52447	0.4	0.47665	0.52338	56
Naïve Bayes	64.55%	0.6	0.7	0.548	0.65473	0.65887	0.3	0.32447	0.59643	47
SVM	93.44%	0.5	0.7	0.475	0.66773	0.57446	0.4	0.48665	0.56998	63.5
Random Forest	64.55%	0.5	0.7	0.64	0.67854	0.67445	0.3	0.3546	0.63887	47.5
NN	59.66%	0.46576	0.6894	0.5146	0.59664	0.59645	0.5	0.38554	0.52885	2.106

In Table 7, GOA gives highest accuracy among all optimization techniques with SVM classifier is 96.54%. Highest sensitivity in Naïve Bayes classifier is 0.9 in GOA optimization technique. False Negative ratio is 0.1 and sensitivity is also higher in Naïve Bayes which is 0.9 with GOA optimization technique. AUC of Random Forest is 0.74 with a greater number of positive predictive results which is 0.7 with IVPSO algorithm.

Table 7: Lung cancer result for training-testing ratio of 70:30

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (70-30)**									
KNN	50%	0.4	0.7	0.455	0.57143	0.53846	0.6	0.42857	0.47059	70
Naïve Bayes	66.66%	0.1	0.9	0.23	0.5	0.5	0.9	0.5	0.16667	66.5
SVM	93.71%	0.7	0.6	0.575	0.63636	0.66667	0.3	0.36364	0.66667	38.5
Random Forest	62.5%	0.7	0.7	0.74	0.7	0.7	0.3	0.3	0.7	59
NN	58.53%	0.44688	0.71264	0.49204	0.58848	0.58353	0.55312	0.41152	0.50799	2.0117
GOA	**Training-Testing (70-30)**									
KNN	60.71%	0.6	0.5	0.53	0.54545	0.55556	0.4	0.45455	0.57143	43
Naïve Bayes	66.66%	0.9	0.5	0.73	0.64286	0.83333	0.1	0.35714	0.75	45.5
SVM	96.54%	0.5	0.7	0.455	0.625	0.58333	0.5	0.375	0.55556	62
Random Forest	68.75%	0.7	0.3	0.385	0.5	0.5	0.3	0.5	0.58333	46
NN	59.88%	0.79688	0.41667	0.51312	0.55677	0.69048	0.20313	0.44323	0.65553	1.0004
PSO	**Training-Testing (70-30)**									

KNN	57.44%	0.5	0.6	0.51	0.56987	0.54886	0.6	0.43665	0.53887	56
Naïve Bayes	68.85%	0.6	0.7	0.65	0.62886	0.74456	0.7	0.487	0.63887	62
SVM	95.33%	0.6	0.6	0.468	0.62997	0.64887	0.5	0.3655	0.53665	57.5
Random Forest	64.76%	0.5	0.4	0.586	0.68	0.64	0.4	0.35	0.68554	42
NN	58.77%	0.6455	0.65887	0.4933	0.57699	0.62887	0.49665	0.42776	0.63755	1.86467

In Table 8, For the lung cancer training-testing ratio 60:40, GOA is more accurate for SVM classifier. In GOA highest AUC is 0.64 also FPR is lower with 0.28571 in Naïve Bayes classifier. Thus, Naïve Bayes is good with GOA technique but not in accuracy. The highest accuracy is achieved with SVM classifier (96.77%) as it gives highest sensitivity value compared to other classifiers.

Table 8: Lung cancer data for training-testing ratio of 60:40

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (60-40)**									
KNN	57.14%	0.4	0.9	0.585	0.8	0.6	0.6	0.2	0.53333	78.5
Naïve Bayes	83.33%	0.7	0.4	0.485	0.53846	0.57143	0.3	0.46154	0.6087	46.5
SVM	94.69%	0.4	0.8	0.49	0.66667	0.57143	0.6	0.33333	0.5	61.5
Random Forest	71.87%	0.4	0.7	0.47	0.57143	0.53846	0.6	0.42857	0.47059	66
NN	58.98%	0.77812	0.41667	0.49265	0.55088	0.6713	0.22187	0.44912	0.64508	1.0037
GOA	**Training-Testing (60-40)**									
KNN	57.14%	0.6	0.7	0.61	0.66667	0.63636	0.4	0.33333	0.63158	55.5
Naïve Bayes	66.66%	0.5	0.8	0.64	0.71429	0.61538	0.5	0.28571	0.58824	71.5
SVM	96.77%	0.8	0.4	0.535	0.57143	0.66667	0.2	0.42857	0.66667	31.5
Random Forest	68.75%	0.5	0.7	0.555	0.625	0.58333	0.5	0.375	0.55556	76
NN	53.59%	0.66563	0.41667	0.45844	0.51202	0.5754	0.33437	0.48798	0.5788	1.0103
PSO	**Training-Testing (60-40)**									
KNN	58.26%	0.4	0.7	0.596	0.7844	0.6544	0.6	0.31776	0.5733	62.5
Naïve Bayes	75.54%	0.6	0.6	0.6437	0.6344	0.59133	0.5	0.34887	0.58832	53.4
SVM	95.34%	0.6	0.6	0.5287	0.6387	0.57443	0.3	0.35887	0.62334	35
Random Forest	69.55%	0.4	0.6	0.5744	0.59665	0.55997	0.5	0.3955	0.52958	61
NN	54.88%	0.66538	0.45332	0.4956	0.50774	0.6533	0.32554	0.47665	0.62443	1.1335

6.4 Results on Colon Cancer Dataset

The colon cancer datasets that are extracted from UCI repository are evaluated for three different algorithms viz., IVPSO, PSO and GAO. The classification is done using distinct classifiers and for different training-testing ratios. Highest accuracy 100% is obtained for various classifiers, using various algorithms. The detailed analysis is given in the Table 9, Table 10, Table 11, Table 12 for different training-testing ratios (90-10, 80-20, 70-30, 60-40 respectively).

Table 9: Colon Cancer data for 90-10 ratio

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (90-10)**									
KNN	100%	0.8	0.6	0.69	0.66667	0.75	0.2	0.33333	0.72727	55
Naïve Bayes	100%	0.5	0.4	0.335	0.45455	0.44444	0.5	0.54545	0.47619	55
SVM	100%	0.7	0.4	0.48	0.53846	0.57143	0.3	0.46154	0.6087	47
Random Forest	100%	0.9	0.4	0.495	0.6	0.8	0.1	0.4	0.72	38.5
NN	51.17%	0.49413	0.52933	0.50546	0.51216	0.51133	0.50587	0.48784	0.50299	3405.9
GOA	**Training-Testing (90-10)**									
KNN	100%	0.7	0.3	0.385	0.5	0.5	0.3	0.5	0.58333	31
Naïve Bayes	100%	0.8	0.8	0.9	0.8	0.8	0.2	0.2	0.8	58.5
SVM	95.16%	0.6	0.5	0.515	0.54545	0.55556	0.4	0.45455	0.57143	55.5
Random Forest	100%	0.5	0.6	0.51	0.55556	0.54545	0.5	0.44444	0.52632	61
NN	48.82%	0.44428	0.53226	0.46853	0.48714	0.48922	0.55572	0.51286	0.46472	3415.8
PSO	**Training-Testing (90-10)**									
KNN	96.77%	0.7	0.5	0.574	0.56437	0.58674	0.4	0.45345	0.54876	46
Naïve Bayes	93.55%	0.6	0.5	0.753	0.6873	0.6759	0.3	0.33334	0.74539	57.4
SVM	100%	0.7	0.3	0.503	0.54743	0.57498	0.3	0.34267	0.41648	49
Random Forest	100%	0.7	0.6	0.5	0.5864	0.57421	0.2	0.39756	0.52976	47
NN	49.49%	0.46545	0.53715	0.49724	0.49636	0.48891	0.50553	0.53472	0.49673	3409.5

Table 10 shows that, we have got accuracy 100% for each optimization techniques for each ratio. Also, we can compare the maximum values for AUC is 0.73 in GOA classifier with KNN classifier. In colon cancer dataset itself has better attributes so that every algorithm gives fewer number of False Positives. Also, sensitivity is near to 1 for every classifier.

Table 10: Colon cancer data for 80:20 ratio

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (80-20)**									
KNN	100%	0.8	0.5	0.575	0.61538	0.71429	0.2	0.38462	0.69565	34
Naïve Bayes	100%	0.7	0.6	0.56	0.63636	0.66667	0.3	0.36364	0.66667	46
SVM	100%	0.7	0.3	0.38	0.5	0.5	0.3	0.5	0.58333	39.5
Random Forest	100%	0.7	0.4	0.485	0.53846	0.57143	0.3	0.46154	0.6087	49.5
NN	56.30%	0.51906	0.60704	0.56323	0.56913	0.55795	0.48094	0.43087	0.54294	3806.7
GOA	**Training-Testing (80-20)**									
KNN	100%	0.6	0.9	0.73	0.85714	0.69231	0.4	0.14286	0.70588	54.5
Naïve Bayes	100%	0.8	0.4	0.53	0.57143	0.66667	0.2	0.42857	0.66667	36.5
SVM	98.38%	0.9	0.3	0.45	0.5625	0.75	0.1	0.4375	0.69231	42.5
Random Forest	100%	0.6	0.6	0.6	0.6	0.6	0.4	0.4	0.6	52
NN	51.46%	0.41202	0.6173	0.4925	0.51845	0.51217	0.58798	0.48155	0.45915	3871.8
PSO	**Training-Testing (80-20)**									
KNN	100%	0.7	0.4	0.64583	0.61295	0.68567	0.3	0.28464	0.68956	50
Naïve Bayes	100%	0.7	0.5	0.5582	0.56398	0.66253	0.2	0.40375	0.66534	35
SVM	100%	0.75	0.3	0.39	0.49367	0.69	0.3	0.49365	0.55725	40.5
Random Forest	98.45%	0.7	0.35	0.37	0.49563	0.57	0.2	0.42474	0.60365	47
NN	53.12%	0.48523	0.6167	0.52348	0.49264	0.52385	0.49145	0.45865	0.50735	3804.5

In Colon cancer dataset (Table 11), all the algorithms (IVPSO, GOA, PSO) give the highest accuracy which is 100% for training-testing ratio 70:30. Which is the best value considered as per the parameters. Further, the maximum values of AUC in the algorithms IVPSO, GOA and PSO are 0.645, 0.59 and 0.6 respectively. Negative predictive value is "1" while false negative value is "0" that shows the preciseness of random forest algorithm in GOA optimization technique.

Table 11: Colon cancer data for 70:30 ratio

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (70-30)**									
KNN	100%	0.8	0.6	0.62	0.66667	0.75	0.2	0.33333	0.72727	55
Naïve Bayes	100%	0.7	0.8	0.645	0.77778	0.72727	0.3	0.22222	0.73684	71.5
SVM	95.16%	0.6	0.4	0.4	0.5	0.5	0.4	0.5	0.54545	49
Random Forest	100%	0.7	0.3	0.35	0.5	0.5	0.3	0.5	0.58333	52.5

NN	52.85%	0.54839	0.5088	0.52824	0.5275	0.52977	0.45161	0.4725	0.53774	3304.5
GOA	**Training-Testing (70-30)**									
KNN	100%	0.8	0.6	0.59	0.66667	0.75	0.2	0.33333	0.72727	48.5
Naïve Bayes	100%	0.5	0.4	0.38	0.45455	0.44444	0.5	0.54545	0.47619	57.5
SVM	93.54%	0.5	0.6	0.355	0.55556	0.54545	0.5	0.44444	0.52632	56.5
Random Forest	100%	1	0.3	0.44	0.58824	1	0	0.41176	0.74074	28
NN	51.83%	0.42962	0.60704	0.49777	0.52228	0.51557	0.57038	0.47772	0.47144	3807.9
PSO	**Training-Testing (70-30)**									
KNN	100%	0.8	0.6	0.6	0.66665	0.7	0.2	0.3	0.7272	53
Naïve Bayes	100%	0.6	0.7	0.45	0.5556	0.5555	0.4	0.44555	0.5674	59
SVM	94.35%	0.6	0.5	0.385	0.5	0.5344	0.4	0.4456	0.5345	54.4
Random Forest	100%	0.8	0.3	0.4	0.57	0.7	0.3894	0.3999	0.60456	39.8
NN	53.34%	0.50456	0.5023	0.51365	0.5252	0.52312	0.513452	0.4793	0.5143	3517.6

In Table 12, GOA gives highest AUC value which is 0.835 for colon cancer data. In IVPSO with KNN classifier gives lowest false positive results is 0.125 which shows more preciseness of algorithm.

Table 12: Colon cancer data for 60:40 ratio

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing (60-40)**									
KNN	100%	0.7	0.9	0.71	0.875	0.75	0.3	0.125	0.77778	64.5
Naïve Bayes	100%	0.6	0.6	0.515	0.6	0.6	0.4	0.4	0.6	39.5
SVM	98.38%	0.8	0.5	0.61	0.61538	0.71429	0.2	0.38462	0.69565	42
Random Forest	100%	0.8	0.3	0.35	0.53333	0.6	0.2	0.46667	0.64	42
NN	51.24%	0.5132	0.51173	0.49681	0.51245	0.51248	0.4868	0.48755	0.51282	3331.1
GOA	**Training-Testing (60-40)**									
KNN	100%	0.9	0.4	0.495	0.6	0.8	0.1	0.4	0.72	29.5
Naïve Bayes	100%	0.7	0.3	0.395	0.5	0.5	0.3	0.5	0.58333	35.5
SVM	91.93%	0.8	0.8	0.835	0.8	0.8	0.2	0.2	0.8	54
Random Forest	100%	0.5	0.7	0.51	0.625	0.58333	0.5	0.375	0.55556	64
NN	56.74%	0.58358	0.55132	0.5722	0.56534	0.5697	0.41642	0.43466	0.57431	3520.5
PSO	**Training-Testing (60-40)**									
KNN	100%	0.7	0.7	0.69	0.756	0.75	0.2	0.2546	0.7655	56
Naïve Bayes	100%	0.6	0.4	0.437	0.5	0.5	0.3	0.4	0.6	37
SVM	93.42%	0.7	0.5	0.756	0.7476	0.7534	0.674	0.3548	0.77	47

Random Forest	100%	0.6	0.4	0.47	0.5677	0.55	0.597	0.45634	0.5567	48
NN	53.84%	0.53456	0.51243	0.5246	0.5755	0.5466	0.4766	0.4735	0.5645	3345

6.5 Results on Breast Cancer Dataset

For breast cancer, datasets that are directly collected from UPI repository are analyzed using GAO, IVPSO and PSO algorithms. The comprehensive results sheet is displayed in the Table 13, Table 14, Table 15, Table 16 for different training-testing ratios (90-10, 80-20, 70-30, 60-40 respectively). The table also shows the data with highest accuracy.

Table 13: Breast cancer result for 90:10 ratio

	ACCURACY	**Sensitivity**	**Specificity**	**AUC**	**PPV**	**NPV**	**FNR**	**FPR**	**Precision**	**Recall**
IVPSO	**training-testing(90-10)**									
KNN	95.60%	0.5	0.5	0.43	0.5	0.5	0.5	0.5	0.5	54.5
Naïve Bayes	97.72%	0.5	0.5	0.285	0.5	0.5	0.5	0.5	0.5	58
SVM	93.61%	0.7	0.5	0.52	0.58333	0.625	0.3	0.41667	0.63636	39
NN	83.42%	0.75236	0.9089	0.70817	0.88268	0.80115	0.24764	0.11732	0.81233	180.3
Random Forest	93.84%	0.7	0.6	0.485	0.63636	0.66667	0.3	0.36364	0.66667	60.5
GOA	**Training-Testing(90-10)**									
KNN	95.9%	0.6	0.7	0.51	0.66667	0.63636	0.4	0.33333	0.63158	54.5
Naïve Bayes	96.21%	0.6	0.6	0.53	0.6	0.6	0.4	0.4	0.6	51.5
SVM	94.62%	0.7	0.8	0.62	0.77778	0.72727	0.3	0.22222	0.73684	55
NN	71.70%	0.50644	0.9679	0.46431	0.8351	0.66903	0.49356	0.1649	0.63051	205.82
Random Forest	93.70%	0.9	0.6	0.64	0.69231	0.85714	0.1	0.30769	0.78261	51.5
PSO	**Training-Testing(90-10)**									
KNN	90.32%	0.53	0.6	0.48	0.6	0.5578	0.42	0.3365	0.5	52.3
Naïve Bayes	83.88%	0.7876	0.5335	0.8012	0.6671	0.4174	0.6529	0.2661	0.812	78.6
SVM	91.94%	0.7	0.7	0.554	0.6573	0.657	0.4	0.3356	0.6545	56
NN	84.53%	0.6073	0.8654	0.6549	0.8629	0.7345	0.3957	0.1386	0.7239	304.67
andom Forest	93.79%	0.8	0.7	0.5673	0.6847	0.7856	0.261	0.3045	0.7012	55.7

For Breast cancer results the highest accuracy is 99.37%, which is highest among all the optimization techniques. While value of AUC is highest in IVPSO with NN classifier which is 0.80252. False positive ratio is lowest in random forest compared to other classifiers with each optimization technique which is 0.10336 with IVPSO, 0.1463 with GOA, and 0.1386 with PSO. Highest sensitivity value is observed in Neural Network (0.86753) with IVPSO algorithm.

Table 14: Breast cancer result for 80:20 ratio

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing(80-20)**									
KNN	95.31%	0.4	0.8	0.55	0.66667	0.57143	0.6	0.33333	0.5	76.5
Naïve Bayes	94.69%	0.8	0.5	0.565	0.61538	0.71429	0.2	0.38462	0.69565	46.5
SVM	96.92%	0.7	0.6	0.685	0.63636	0.66667	0.3	0.36364	0.66667	54.5
NN	88.91%	0.86753	0.9089	0.80252	0.89664	0.88278	0.13247	0.10336	0.88184	904.16
Random Forest	94.42%	0.7	0.8	0.775	0.77778	0.72727	0.3	0.22222	0.73684	45.5
GOA	**Training-Testing(80-20)**									
KNN	95.9%	0.6	0.3	0.325	0.46154	0.42857	0.4	0.53846	0.52174	44.5
Naïve Bayes	95.45%	0.7	0.3	0.445	0.5	0.5	0.3	0.5	0.58333	33
SVM	99.37%	0.5	0.4	0.335	0.45455	0.44444	0.5	0.54545	0.47619	51.5
NN	75.38%	0.58355	0.9089	0.53248	0.8537	0.70551	0.41645	0.1463	0.69324	10.482
Random Forest	92. 99%	0.5	0.5	0.34	0.5	0.5	0.5	0.5	0.5	36
PSO	**Training-Testing(80-20)**									
KNN	90.32%	0.5	0.6	0.458	0.6	0.8578	0.52	0.4265	0.55	56.3
Naïve Bayes	83.78%	0.776	0.5335	0.6012	0.6371	0.4574	0.629	0.3261	0.702	76.6
SVM	95.94%	0.7	0.7	0.554	0.5713	0.587	0.34	0.4356	0.5045	53
NN	84.53%	0.6373	0.8654	0.6749	0.8429	0.7895	0.3557	0.1386	0.729	454.67
Random Forest	93.95%	0.8	0.7	0.573	0.6147	0.65486	0.41	0.45	0.612	40.7

In Table 15, the highest accuracy is 96.97% for GOA with Naïve Bayes classifier in 70-30 ratio and AUC is 0.78904 in GOA with NN classifier. NN works well with this kind of dataset because sensitivity of NN with each optimization technique is higher compare to other techniques.

Table 15: Breast cancer data for 70:30 ratio

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing(70-30)**									
KNN	95.02%	0.6	0.8	0.675	0.75	0.66667	0.4	0.25	0.66667	73
Naïve Bayes	96.21%	0.5	0.6	0.475	0.55556	0.54545	0.5	0.44444	0.52632	71
SVM	93.35%	0.6	0.5	0.415	0.54545	0.55556	0.4	0.45455	0.57143	57.5
NN	69.51%	0.46052	0.9089	0.42252	0.82159	0.64905	0.53948	0.17841	0.59021	10.371
Random Forest	94.56%	0.3	0.6	0.315	0.42857	0.46154	0.7	0.57143	0.35294	60
GOA	**Training-Testing(70-30)**									
KNN	95.9%	0.5	0.8	0.53	0.71429	0.61538	0.5	0.28571	0.58824	74
Naïve Bayes	96.97%	0.4	0.6	0.445	0.5	0.5	0.6	0.5	0.44444	62.5
SVM	93.89%	0.7	0.7	0.69	0.7	0.7	0.3	0.3	0.7	39
NN	88.73%	0.86366	0.9089	0.78904	0.89623	0.87978	0.13634	0.10377	0.87964	220.43
Random Forest	93.84%	0.6	0.6	0.495	0.6	0.6	0.4	0.4	0.6	57.5
PSO	**Training-Testing (70-30)**									
KNN	91.22%	0.4	0.6	0.48	0.723	0.6528	0.45	0.265	0.607	72.3
Naïve Bayes	87.58%	0.6	0.4535	0.412	0.5671	0.5534	0.612	0.476361	0.546	66
SVM	96.24%	0.5	0.7	0.557	0.6723	0.647	0.43	0.356	0.5453	54.3
NN	72.53%	0.7321	0.8654	0.5349	0.8429	0.7345	0.3457	0.1386	0.7139	154.6
Random Forest	92.95%	0.53	0.7	0.473	0.5147	0.5476	0.58	0.435	0.542	45.3

In Table 16, For the Breast cancer data for 60:40 ratio, PSO with Naïve Bayes classifier gives the highest accuracy of 97.58% for breast cancer data with training-testing ratio 60:40. For AUC parameter value is 0.79025 for the NN classifier in IVPSO technique. We can see that Neural network is giving higher positive predictions (PPV) compared to other algorithms which is 0.89473, 0.74965, 0.7339 with IVPSO, GOA, PSO optimization techniques, respectively. AUC is higher in Neural network which is 0.79025 compared to other optimization techniques.

Table 16: Breast cancer data for 60:40 ratio

	ACCURACY	Sensitivity	Specificity	AUC	PPV	NPV	FNR	FPR	Precision	Recall
IVPSO	**Training-Testing(60-40)**									
KNN	95.9%	0.8	0.5	0.505	0.61538	0.71429	0.2	0.38462	0.69565	48.5
Naïve Bayes	96.07%	0.7	0.2	0.275	0.46667	0.4	0.3	0.53333	0.56	39.5
SVM	95%	0.9	0.6	0.705	0.69231	0.85714	0.1	0.30769	0.78261	48
NN	88.07%	0.84993	0.9089	0.79025	0.89473	0.86925	0.15007	0.10527	0.87175	132.65
Random Forest	93.27%	0.8	0.6	0.65	0.66667	0.75	0.2	0.33333	0.72727	46
GOA	**Training-Testing(60-40)**									
KNN	94.87%	0.2	0.7	0.22	0.4	0.46667	0.8	0.6	0.26667	73.5
Naïve Bayes	96.21%	0.9	0.5	0.64	0.64286	0.83333	0.1	0.35714	0.75	34.5
SVM	94.11%	0.6	0.7	0.58	0.66667	0.63636	0.4	0.33333	0.63158	60
NN	85.82%	0.80272	0.9089	0.74965	0.88922	0.83491	0.19728	0.11078	0.84376	161.62
Random Forest	94.13%	0.5	0.8	0.595	0.71429	0.61538	0.5	0.28571	0.58824	53
PSO	**Training-Testing(60-40)**									
KNN	94.22%	0.5	0.6667	0.386	0.513	0.5728	0.4	0.5	0.547	67
Naïve Bayes	97.58%	0.64	0.403	0.412	0.551	0.5834	0.2	0.4861	0.632	34
SVM	95.14%	0.72	0.6678	0.557	0.6701	0.607	0.3	0.3678	0.7353	57
NN	82.13%	0.8329	0.954	0.7339	0.8019	0.8452	0.2157	0.1756	0.8539	158.4
Random Forest	92.05%	0.67`	0.8	0.671	0.6657	0.764	0.38	0.2351	0.6402	47.9

6.6 Discussion

6.6.1 Leukemia Cancer

For individual algorithms, different classifiers give different results. Such results for IVPSO, GOA and PSO are taken for Leukemia cancer data in Table 17. The analysis is done based on accuracy and AUC. According to the results obtained, the highest AUC for 80:20 ratio is obtained for IVPSO algorithm using Naïve Bayes classifier. Whereas 99.28% is the accuracy for 60:40 training-testing ratio obtained using IVPSO algorithm and SVM classifier. Blue color is used for accuracy and green color is used to indicate AUC in the table.

Similarly, for GOA algorithm, Neural Network gives the highest AUC value 0.76 for training-testing ratio 70:30. For PSO optimization algorithm, Neural Network gives highest AUC value 0.75 for training-testing ratio 70:30. The highest accuracy of 99.17% is achieved with SVM for training-testing ratio 60:40.

Table 17: Specific Results of Accuracy and AUC in Leukemia Cancer data

	ACCURACY	AUC	ACCURACY	AUC	ACCURACY	AUC	ACCURACY	AUC
IVPSO	**Training-Testing (90-10)**		**Training-Testing (80-20)**		**Training-Testing (70-30)**		**Training-Testing (60-40)**	
KNN	59.72%	0.485	55.55%	0.65	55.55%	0.4	58.33%	0.345
Naïve Bayes	69.23%	0.3	61.53%	0.735	76.92%	0.45	46.15%	0.495
SVM	62.5%	0.43	61.11%	0.26	97.50%	0.545	99.28%	0.505
NN	51.38%	0.605	66.667	0.665	55.55%	0.62	54.16%	0.505
Random Forest	57.95%	0.53502	53.72%	0.48894	54.29%	0.49857	54.86%	0.50285
GOA	**Training-Testing (90-10)**		**Training-Testing (80-20)**		**Training-Testing (70-30)**		**Training-Testing (60-40)**	
KNN	63.88%	0.435	63.88%	0.465	63.88%	0.36	56.94%	0.38
Naïve Bayes	76.92%	0.59	76.92%	0.48	69.23%	0.54	69.23%	0.53
SVM	62.5%	0.435	63.88%	0.56	97.53%	0.63	99.80%	0.48
NN	58.33%	0.68	52.77%	0.65	51.38%	0.76	58.33%	0.7
Random Forest	50.88%	0.47711	5580%	0.51889	56.12%	0.50817	47.79%	0.42523
PSO	**Training-Testing (90-10)**		**Training-Testing (80-20)**		**Training-Testing (70-30)**		**Training-Testing (60-40)**	
KNN	81.56%	0.453	97.22%	0.533	56.74%	0.39	57.67%	0.35
Naïve Bayes	89.48%	0.43	81.01%	0.64877	72.49%	0.42	53.86%	0.52
SVM	87.06%	0.446	97.38%	0.4566	96.34%	0.6	99.17%	0.49
NN	56.73%	0.639	60.45%	0.65	53.75%	0.75	56.49%	0.6
Random Forest	55.34%	0.52667	55.43%	0.49667	55.39%	0.50445	52.37%	0.47

Table 18 shows the highest AUC value 0.76, which is for NN classifier and 60:40 ratio and highest accuracy is 99.80%, for SVM classifier in Leukemia cancer data. For Naïve Bayes, 0.735 is the highest AUC which is obtained for 60:40 training-testing ratio and highest accuracy obtained using SVM classifier is 99.28% for respective training-testing ratio.

The main reason behind the highest accuracy with SVM classifier is that it works well with large number of gene expression compare to number of samples as leukemia cancer data having 7129 genes with 72 samples.

Table 18: Highest Accuracy and AUC of Leukemia Cancer Data

Highest Accuracy and AUC from different Training-Testing ratio in each optimization technique ratio						
	Ratio	**Classifier**	**Accuracy**	**Ratio**	**Classifier**	**AUC**
IVPSO	80:20	SVM	99.28%	60:40	Naïve Bayes	0.735
GOA	60:40	SVM	99.80%	60:40	NN	0.76
PSO	70:30	SVM	99.17%	60:40	NN	0.75

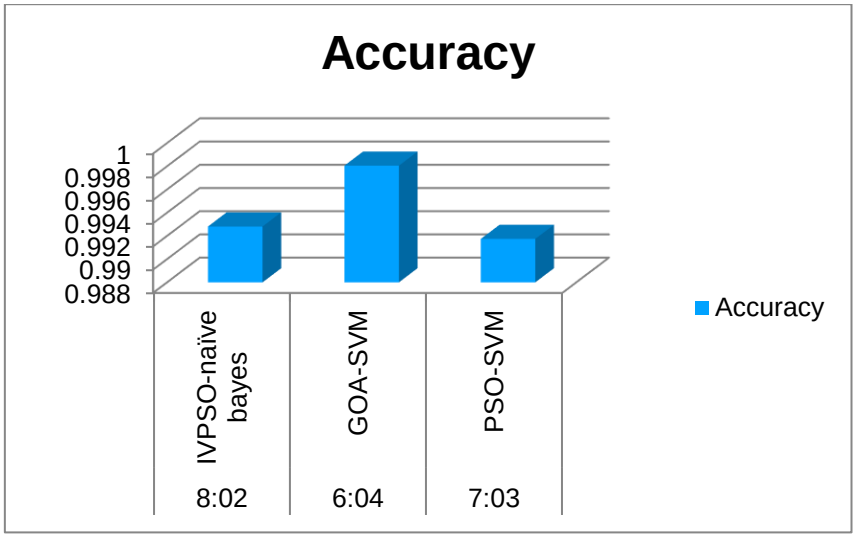

Figure 6(a): Accuracy of Leukemia Cancer Data

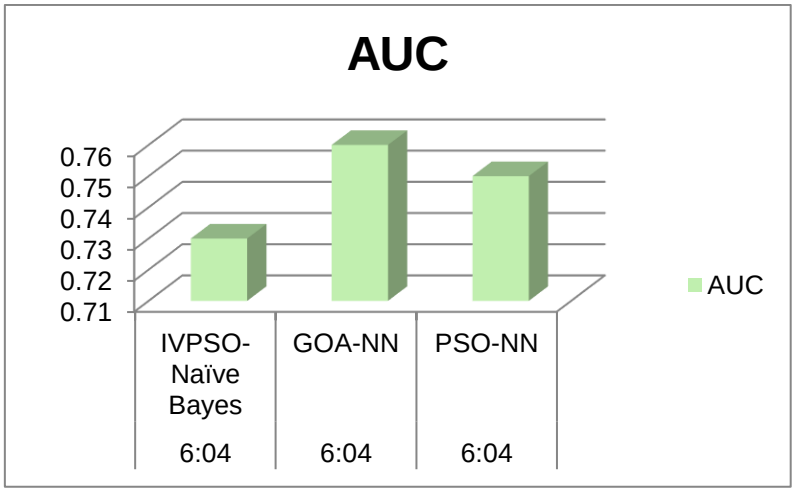

Figure 6(b): AUC of Leukemia Cancer Data

6.6.2 Lung Cancer

For the data of lung cancer, using IVPSO algorithm, highest AUC is 0.74, obtained using NN algorithm at 70:30 ratio. Similarly, for IVPSO, highest accuracy is 99.16%. It is obtained in 90:10 ratio, using SVM classifier.

Table 19 shows the highest accuracy of 99.57% for GOA algorithm, which is recorded for 90:10 training-testing ratio and SVM classifier. Highest AUC in this case is 0.73 which is obtained for 70:30 ratio using Naïve Bayes algorithm. Using PSO algorithm, the highest accuracy is noted for 90:10 ratio using SVM classifier and the highest AUC is noted for 70:30 ratio using Naïve Bayes classifier. Light green color indicates the highest accuracy in each optimization algorithm for each classifier among each training-testing ratio. The middle green color shows the highest accuracy among all classifiers. The dark green color represents the highest accuracy among all the algorithms.

Table 19: Specific Results for Accuracy and AUC in Lung Cancer data

	ACCURACY	AUC	ACCURACY	AUC	ACCURACY	AUC	ACCURACY	AUC
IVPSO	Training-Testing (90-10)		Training-Testing (80-20)		Training-Testing (70-30)		Training-Testing (60-40)	
KNN	75%	0.605	64.28%	0.435	5%	0.455	57.14%	0.585
Naïve Bayes	83.33%	0.6	66.66%	0.535	66.66%	0.23	83.33%	0.485
SVM	99.16%	0.71	93.42%	0.57	93.71%	0.575	94.69%	0.49
NN	53.12%	0.34	53.12%	0.7	62.5%	0.74	71.87%	0.47
Random Forest	59.88%	0.49571	60.03%	0.51172	58.53%	0.49204	58.98%	0.49265
GOA	Training-Testing (90-10)		Training-Testing (80-20)		Training-Testing (70-30)		Training-Testing (60-40)	
KNN	60.71%	0.205	75%	0.29	60.71%	0.53	57.14%	0.61
Naïve Bayes	33.33%	0.6	5%	0.54	66.66%	0.73	66.66%	0.64
SVM	99.57%	0.485	93.49%	0.4	96.54%	0.455	96.77%	0.535
NN	68.75%	0.315	71.87%	0.595	68.75%	0.385	68.75%	0.555
Random Forest	61.97%	0.55159	59.43%	0.51874	59.88%	0.51312	53.59%	0.45844
PSO	Training-Testing (90-10)		Training-Testing (80-20)		Training-Testing (70-30)		Training-Testing (60-40)	
KNN	72.33%	0.4065	73.44%	0.3564	57.44%	0.51	58.26%	0.596
Naïve Bayes	55.67%	0.6	64.55%	0.548	68.85%	0.65	75.54%	0.6437
SVM	99.01%	0.5664	93.44%	0.475	95.33%	0.468	95.34%	0.5287
NN	55.39%	0.3255	64.55%	0.64	64.76%	0.586	69.55%	0.5744
Random Forest	60.77%	0.53446	59.66%	0.5146	58.77%	0.4933	54.88%	0.4956

In case of Lung cancer, the highest recorded accuracy is 0.99577 at the ratio 90:10 and SVM classifier. Further, the highest AUC is 0.74 recorded at 90:10 ratio and NN classifier. Observing the data individually for each optimization algorithm at 90:10 training-testing ratio, accuracy obtained is 99.16%, 99.57% and 99.01% obtained respectively for IVPSO, GOA and PSO algorithms with SVM classifier. Similarly, highest AUC for 90:10 ratio is 0.74, 0.73 and 0.65 ratio for IVPSO, GOA and PSO algorithms respectively.

For highest accuracy in GOA algorithm with SVM classifier at 90:10 ratio is because of well co-ordination of GOA with classifier which are not constrain bounded. SVM classifier is quite faster with large amount of training dataset. Here, data is divided in 90:10 as 90% data is used to train the model.

Table 20: Highest Accuracy and AUC of Lung Cancer Data

Highest Accuracy and AUC from different Training-Testing ratio in each optimization technique						
	Ratio	**Classifier**	**Accuracy**	**Ratio**	**Classifier**	**AUC**
IVPSO	9:01	SVM	99.16%	7:03	NN	0.74
GOA	9:01	SVM	99.57%	7:03	Naïve Bayes	0.73
PSO	9:01	SVM	99.01%	7:03	Naïve Bayes	0.65

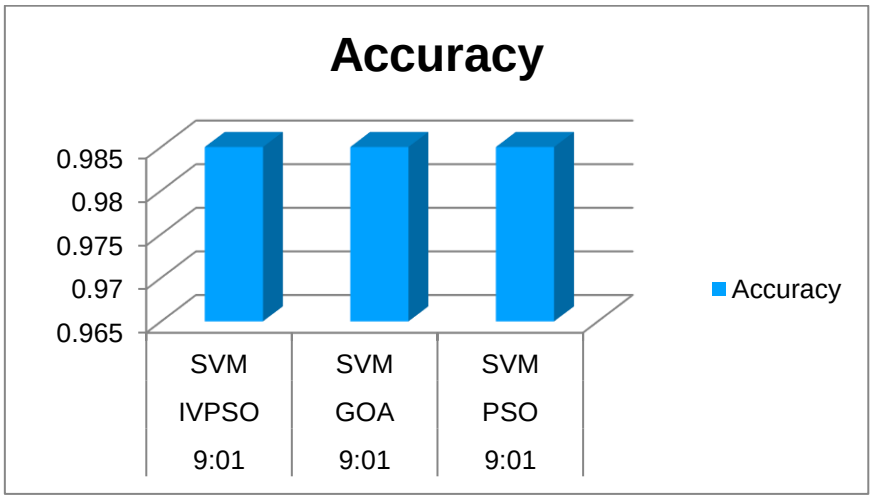

Figure 7(a): Accuracy of Lung Cancer Data

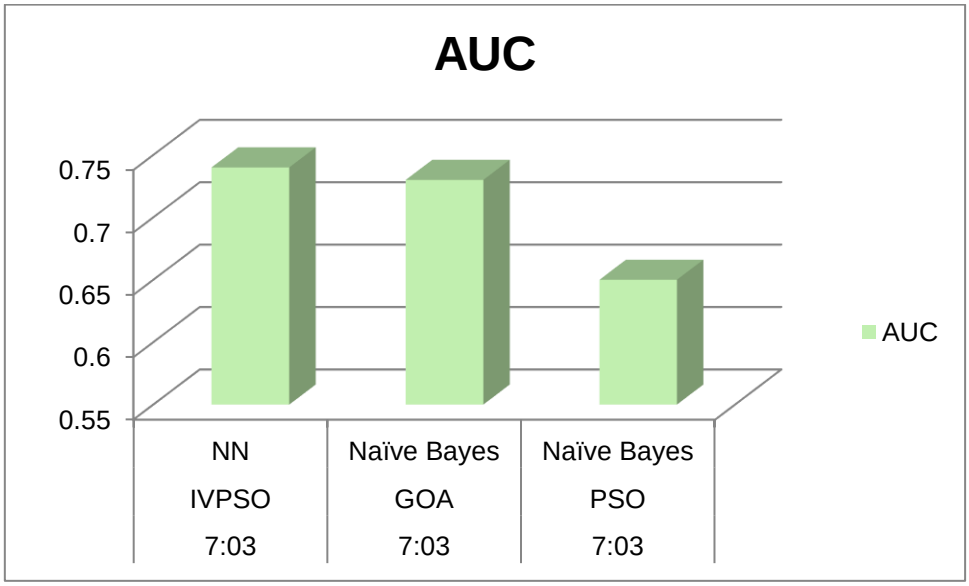

Figure 7(b): AUC of Lung Cancer Data

6.6.3 Colon Cancer

For colon cancer data (Table 21), highest accuracy recorded is 100%. It is obtained quite frequently for different algorithms and classifiers. The highest AUC for IVPSO algorithm is 0.71, which is obtained at 60:40 training-testing ratio using KNN classifier. For GOA, highest AUC recorded is 0.9, which is obtained using Naïve Bayes classifier and 90:10 training-testing ratio. For PSO, highest AUC obtained is for 60:40 ratio using SVM classifier is 0.756.

Table 21: Specific Results of Accuracy and AUC for Colon Cancer data

	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
IVPSO	**Training-Testing (90-10)**		**Training-Testing (80-20)**		**Training-Testing (70-30)**		**Training-Testing (60-40)**	
KNN	100%	0.69	100%	0.575	100%	0.62	100%	0.71
Naïve Bayes	100%	0.335	100%	0.56	100%	0.645	100%	0.515
SVM	100%	0.48	100%	0.38	95.16%	0.4	98.38%	0.61
Random Forest	100%	0.495	100%	0.485	100%	0.35	100%	0.35
NN	51.17%	0.50546	56.30%	0.56323	52.85%	0.52824	51.24%	0.49681
GOA	**Training-Testing (90-10)**		**Training-Testing (80-20)**		**Training-Testing (70-30)**		**Training-Testing (60-40)**	

KNN	100%	0.385	100%	0.73	100%	0.59	100%	0.495
Naïve Bayes	100%	0.9	100%	0.53	100%	0.38	100%	0.395
SVM	95.16%	0.515	98.38%	0.45	93.54%	0.355	91.93%	0.835
Random Forest	100%	0.51	100%	0.6	100%	0.44	100%	0.51
NN	48.82%	0.46853	51.46%	0.4925	51.83%	0.49777	56.74%	0.57216
PSO	**Training-Testing (90-10)**		**Training-Testing (80-20)**		**Training-Testing (70-30)**		**Training-Testing (60-40)**	
KNN	96.77%	0.574	100%	0.64583	100%	0.6	100%	0.69
Naïve Bayes	93.55%	0.753	100%	0.5582	100%	0.45	100%	0.437
SVM	100%	0.503	100%	0.39	94.35%	0.385	93.42%	0.756
Random Forest	100%	0.5	98.45%	0.37	100%	0.4	100%	0.47
NN	49.49%	0.49724	53.12%	0.52348	53.34%	0.51365	53.84%	0.5246

For colon cancer dataset (Table 22), best accuracy of 100% was achieved with all three optimization techniques and all the classifiers except Neural Networks with different training-testing ratios. GOA gives 100% accuracy with all classifiers except SVM and Neural Networks. Best AUC of 90% was achieved with GOA and Naïve Bayes with a training-testing ratio of 90:10. Neural Networks does not perform well in colon cancer dataset due to the data being linearly separable

Table 22: Highest Accuracy and AUC From Different Training- Testing Ratios of Colon Cancer Data

Highest Accuracy = 1 from different Training-Testing ratio in each optimization technique				
	KNN	**Naïve Bayes**	**SVM**	**Random Forest**
IVPSO	90:10,80:20,70:30,60:40	90:10,80:20,70:30,60:40	90:10,80:20	90:10,80:20,70:30,60:40
GOA	90:10,80:20,70:30,60:40	90:10,80:20,70:30,60:40		90:10,80:20,70:30,60:40
PSO	80:20,70:30,60:40	80:20,70:30,60:40	90:10,80:20	90:10,70:30,60:40
Highest AUC from different Training-Testing ratio in each optimization technique				
	Ratio	**Classifier**	**AUC**	
IVPSO	6:04	KNN	0.71	
GOA	9:01	Naïve Bayes	0.835	
PSO	6:04	SVM	0.756	

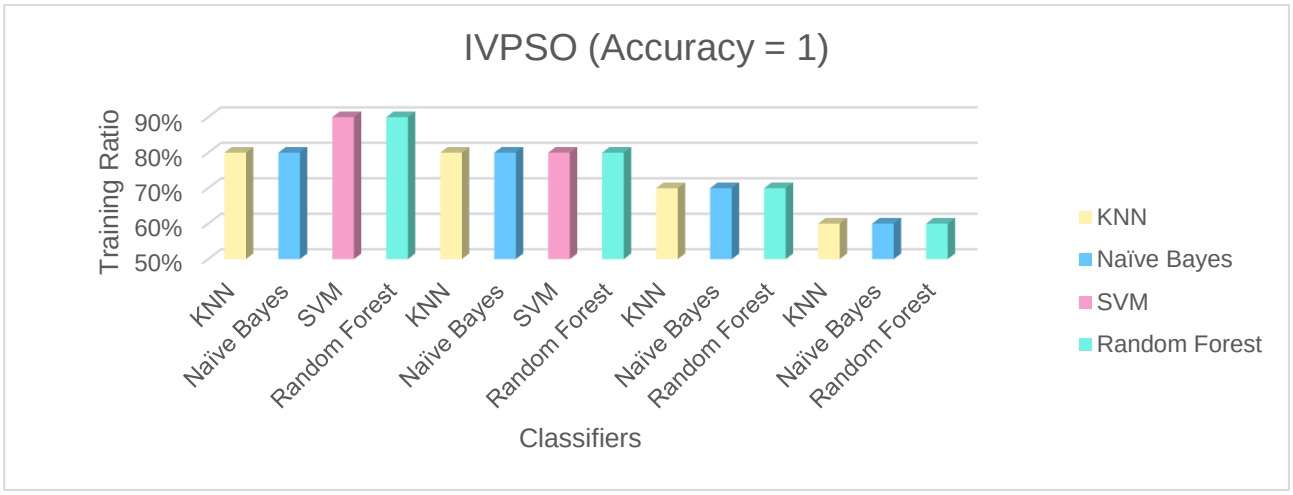

Figure 8(a): Accuracy=1 for IVPSO Technique of Colon Cancer Data

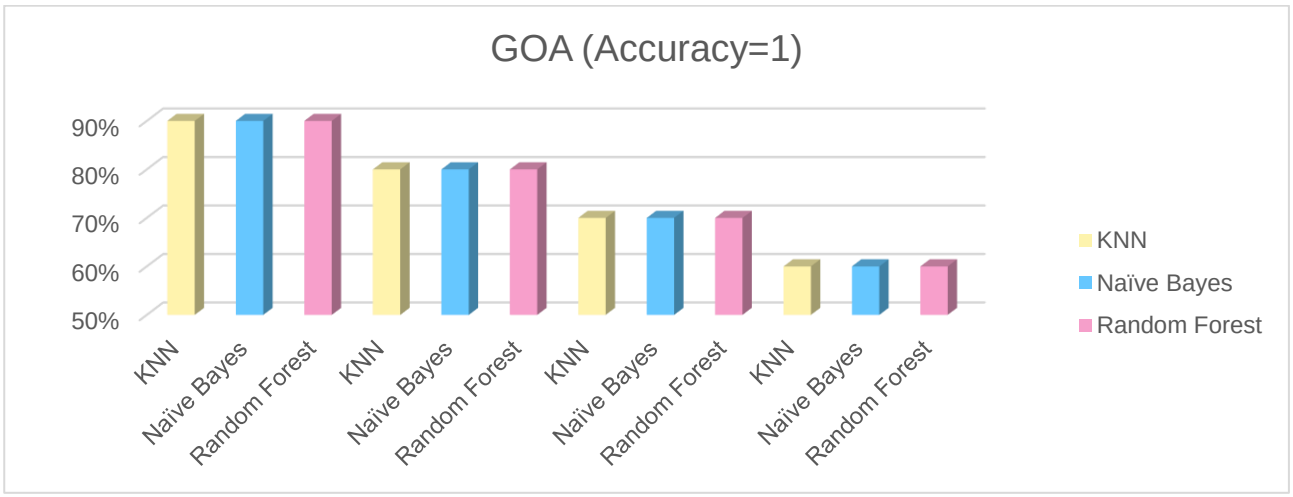

Figure 8(b): Accuracy=1 for GOA Technique of Colon Cancer Data

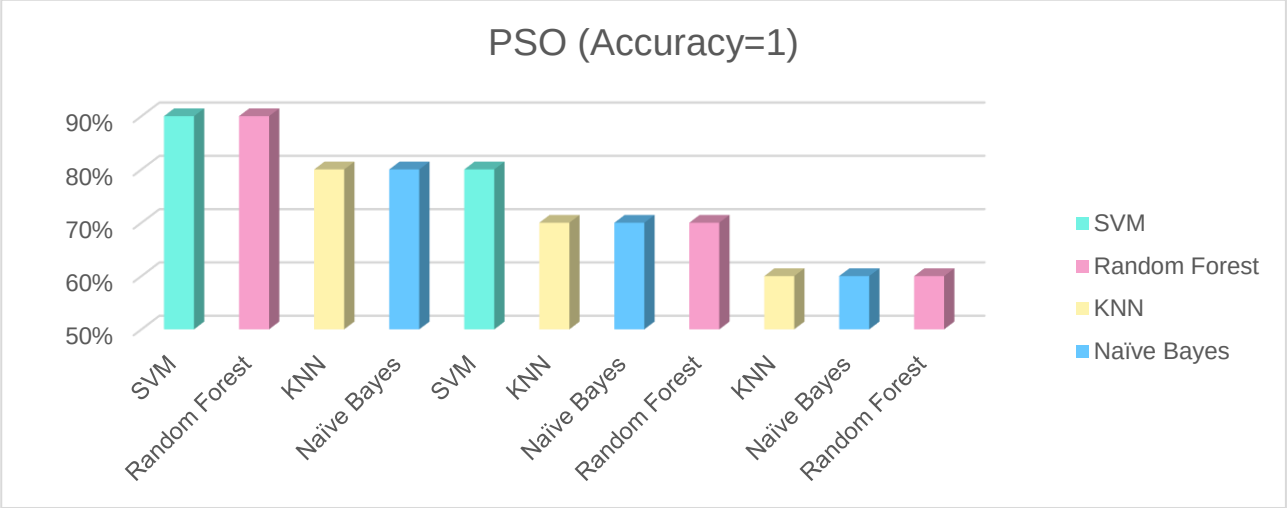

Figure 8(c): Accuracy=1 for PSO Technique of Colon Cancer Data

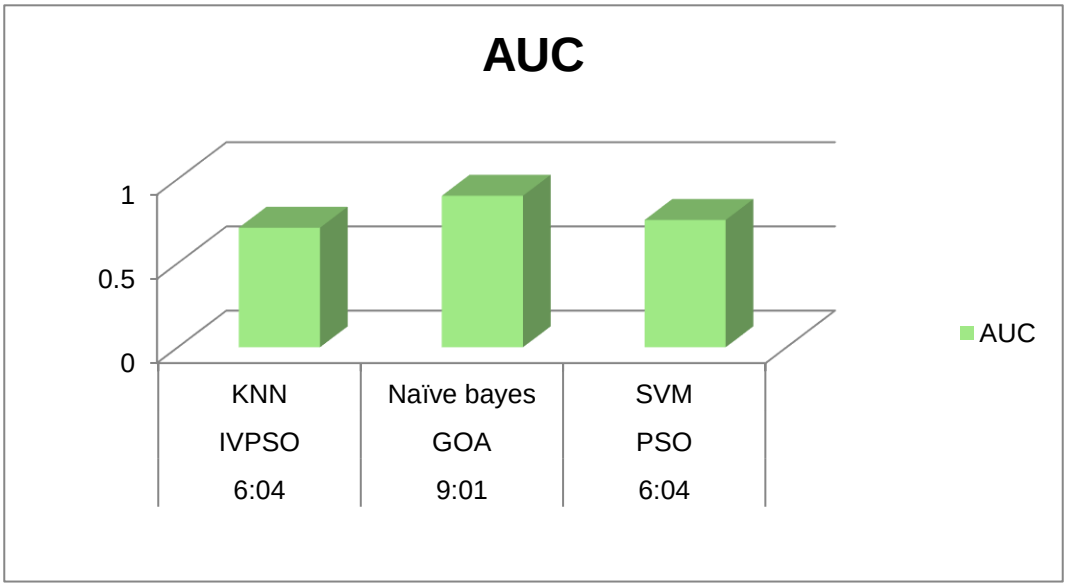

Figure 8(d): AUC of Colon Cancer Data

6.6.4 Breast Cancer

For IVPSO, maximum accuracy is 97.72% which is obtained at 90:10 ratio using Naïve Bayes classifier. For the same optimization algorithm, highest AUC (0.80252) is obtained at 80:20 ratio using NN classifier. For GOA, highest accuracy is 0.99375 which is obtained at 80:20 ratio using SVM. Similarly, highest AUC (0.78904) is obtained at 70:30 training-testing ratio using NN

classifier. For PSO algorithm (Table 23), the maximum accuracy of 97.58% is obtained for training-testing ratio 60:40 with Naïve Bayes classifier. Likewise, maximum recorded AUC (0.8012) is obtained at 90:10 training-testing ratio using Naïve Bayes classifier.

Table 23: Specific Results of Accuracy and AUC for Breast Cancer

	ACCURACY	AUC	ACCURACY	AUC	ACCURACY	AUC	ACCURACY	AUC
IVPSO	Training-Testing (90-10)		Training-Testing (80-20)		Training-Testing (70-30)		Training-Testing (60-40)	
KNN	95.60%	0.43	95.31%	0.55	95.02%	0.675	95.9%	0.505
Naïve Bayes	97.72%	0.285	94.69%	0.565	96.21%	0.475	96.07%	0.275
SVM	93.61%	0.52	96.92%	0.685	93.35%	0.415	95%	0.705
NN	83.42%	0.70817	88.91%	0.80252	69.51%	0.42252	88.07%	0.79025
Random Forest	93.84%	0.485	94.42%	0.775	94.56%	0.315	93.27%	0.65
GOA	Training-Testing (90-10)		Training-Testing (80-20)		Training-Testing (70-30)		Training-Testing (60-40)	
KNN	95.9%	0.51	95.9%	0.325	95.9%	0.53	94.87%	0.22
Naïve Bayes	96.21%	0.53	95.45%	0.445	96.97%	0.445	96.21%	0.64
SVM	94.62%	0.62	99.37%	0.335	93.89%	0.69	94.11%	0.58
NN	71.70%	0.46431	75.38%	0.53248	88.73%	0.78904	85.82%	0.74965
Random Forest	93.70%	0.64	92.99%	0.34	93.84%	0.495	94.13%	0.595
PSO	Training-Testing (90-10)		Training-Testing (80-20)		Training-Testing (70-30)		Training-Testing (60-40)	
KNN	90.32%	0.48	90.32%	0.458	91.22%	0.48	94.22%	0.386
Naïve Bayes	83.88%	0.8012	83.78%	0.6012	87.58%	0.412	97.58%	0.412
SVM	91.94%	0.554	95.94%	0.554	62.4%	0.557	95.14%	0.557
NN	84.53%	0.6549	84.53%	0.6749	72.53%	0.5349	82.13%	0.7339
Random Forest	93.79%	0.5673	93.95%	0.573	92.95%	0.473	92.05%	0.671

For breast cancer data analysis, highest accuracy is 99.37%, which is highlighted with dark green color in 80:20 ratio with SVM classifier. Highest AUC is 0.80252, which is highlighted with dark blue color in 80:20 ratio with NN classifier. Similarly, when results are analyzed for each optimization technique ratio, the output statistics are as mentioned in Table 24.

Table 24: Highest Accuracy and AUC of Breast Cancer Data

	Breast Cancer					
	Highest Accuracy and AUC from different Training-Testing ratio in each optimization technique ratio					
	Ratio	Classifier	Accuracy	Ratio	Classifier	AUC
IVPSO	90:10	Naïve Bayes	97.73%	80:20	NN	0.80252
GOA	80:20	SVM	99.37%	70:30	NN	0.78904
PSO	60:40	Naïve Bayes	97.58%	90:10	Naïve Bayes	0.8012

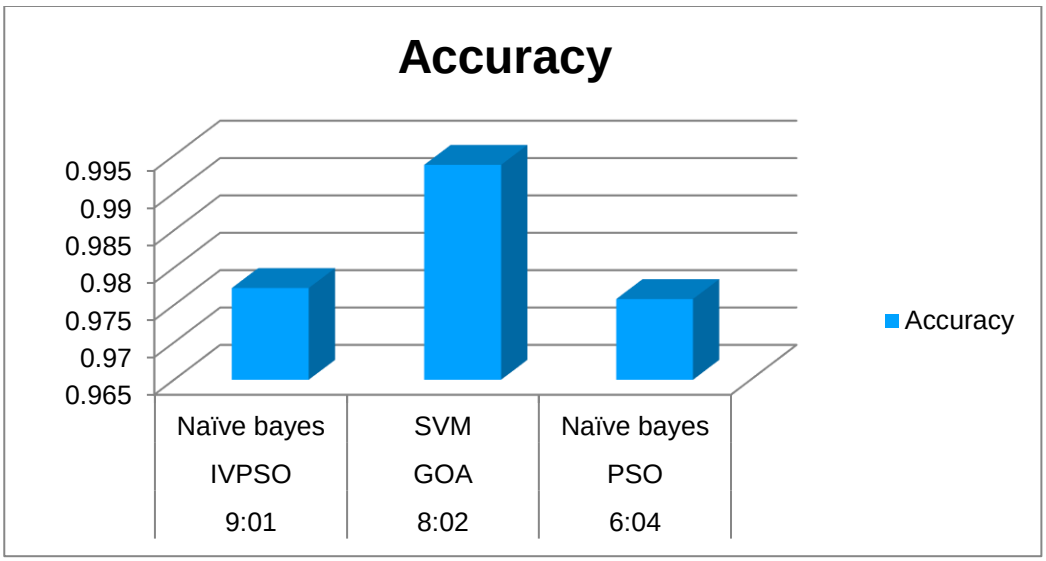

Figure 9(a): Accuracy of Breast Cancer Data

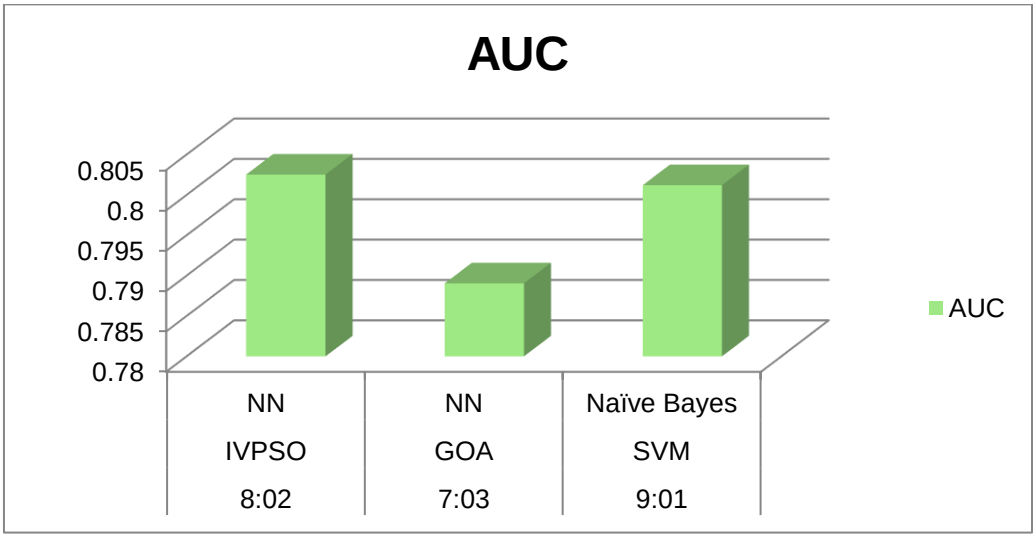

Figure 9(b): AUC of Breast Cancer Data

6.6.5 Comprehensive Study

Figure 10 indicates an analytical pie chart that represents the best accuracy obtained for the data of each of the four cancer datasets. The highest accuracy for breast cancer is obtained at 80:20 training-testing ratio using GOA and SVM classifier. For Lung cancer, the value is obtained using GOA and SVM at 90:10 ratio. For Leukemia, the maximum accuracy is obtained for 60:40 ratio using GOA optimization algorithm and SVM. However, for colon cancer, the highest value is obtained multiple times.

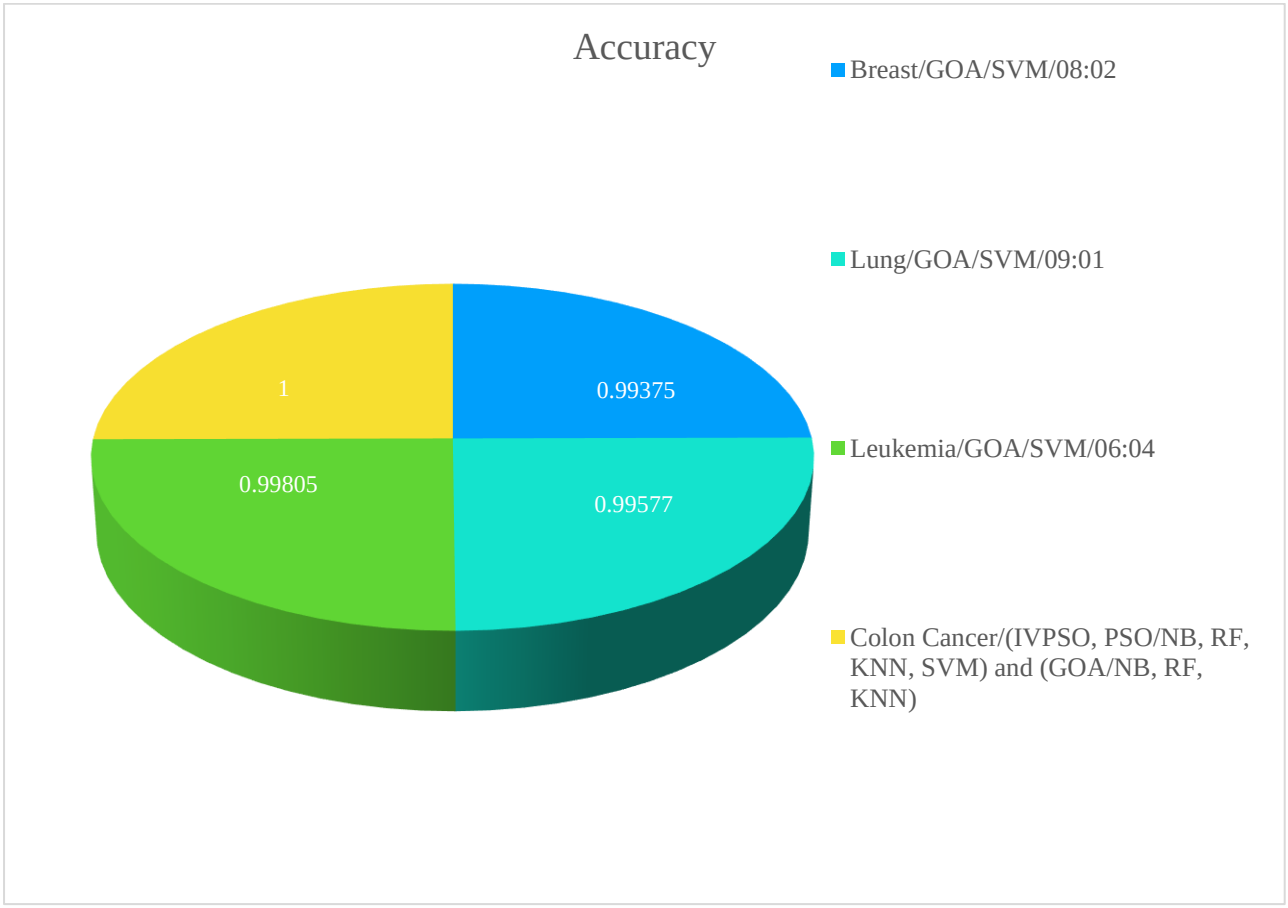

Figure 10: Highest Accuracy of all dataset

Chapter 7

Conclusions and Future Work

7.1 Conclusion

We have analyzed the datasets of four different types of cancer and compared the accuracy and Area Under the ROC Curve (AUC) of three optimization techniques with five classifiers. In three datasets, namely, breast, lung and leukemia, Grasshopper Optimization (GOA) gave the best accuracy with SVM and Neural Networks classifiers.

- For the breast cancer dataset, best accuracy of 99.4% was achieved with GOA and SVM with training-testing ratio of 80:20. The best AUC of 80.25% was achieved with IVPSO and Neural Networks with a training-testing ratio of 80:20.

- For lung cancer dataset, best accuracy of 99.58% was achieved with GOA and SVM with a training-testing ratio of 90:10 whereas, the best AUC of 74% was achieved with IVPSO and Neural Networks with a training-testing ratio of 70:30.

- For leukemia dataset, best accuracy of 99.80% was achieved with GOA and SVM with a training-testing ratio of 60:40. The best AUC of 76% was achieved with GOA and Neural Networks with a training-testing ratio of 70:30.

- For colon cancer dataset, best accuracy of 100% was achieved with all three optimization techniques and all the classifiers except Neural Networks with different training-testing ratios.

- GOA gives 100% accuracy with all classifiers except SVM and Neural Networks

- Best AUC of 90% was achieved with GOA and Naïve Bayes with a training-testing ratio of 90:10
- Neural Networks does not perform well in colon cancer dataset due to the data being linearly separable

7.2 Future Work

- Other optimization techniques can be explored with different classifiers to achieve better accuracy in prediction of cancers.
- Hybrid optimization techniques can be explored with different classifiers to improve the prediction accuracy of cancers.

References

[1] Barkai N., Notterman D., Gish K., Ybarra S., Mack D., Levine A.J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. P AUC. Natl. Acad. Sci. U. S. A. 1999;96:6745–6750. [PMC free article] [PubMed] [Google Scholar]

[2] Baxevanis, Ouellette B.F.F. 2nd ed. John Wiley & Sons; 2001. Bioinformatics: "A Particle Guide to the Analysis of Genes and Proteins" [Google Scholar]

[3] Ben-Dor A., Bruhm L., Friedman Tissue classification with gene expression profiles. Comput. Biol. 2000:559–584. [PubMed] [Google Scholar]

[4] Qi Y., Yang X. Interval-valued analysis for discriminative gene selection and tissue sample classification using microarray data. Genomics. 2013;101:38–48. [PubMed] [Google Scholar]

[5] Statnikov A., Aliferis C., Tsamardinos I., Hardin D., Levy S. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics. 2005;21(5):631–643. [PubMed] [Google Scholar]

[6] Cortes C., Vapnik V. Support vector networks. Mach. Learn. 1995;20(3):273–297. [Google Scholar]

[7] Salem D.A., AbulSeoud R.A.A.A., Ali H.A. A new gene selection technique based on hybrid methods for cancer classification using microarrays. Int. J. Biosci. Biochem. Bioinforma. November 2011;1(4)[Google Scholar]

[8] Golub T.R., Slonim D.K., Tamayo Classification of cancer: class discovery and class prediction by gene expression monitoring. Science. 1999;286:315–333. [PubMed] [Google Scholar]

[9] Hong H., Li J., Wang H., Daggard G. Combined gene selection methods for microarray data analysis knowledge-based intelligent information and engineering systems. Lect. Notes Computer Sci. 2006;4251:976–983.[Google Scholar]

[10] Kadota K, Yeh YC, Sima CS, et al. The cribriform pattern identifies a subset of acinar predominant tumors with poor prognosis in patients with stage I lung adenocarcinoma: a conceptual proposal to classify cribriform predominant tumors as a distinct histologic subtype. Mod Pathol 2014;27(5):690–700.

[11] Comprehensive vertical sample-based KNN/LSVM classification for gene expression, August 2004Journal of Biomedical Informatics 37(4):240-8,
DOI: 10.1016/j.jbi.2004.07.003.

[12] Tsutsumida H, Nomoto M, Goto M, et al. A micropapillary pattern is predictive of a poor prognosis in lung adenocarcinoma, and reduced surfactant apoprotein A expression in the micropapillary pattern is an excellent indicator of a poor prog- nosis. Mod Pathol 2007;20(6):638–47.

[13] Nitadori J, Bograd AJ, Kadota K, et al. Impact of micropapillary histologic sub- type in selecting limited resection vs lobectomy for lung adenocarcinoma of 2cm or smaller. J Natl Cancer Inst 2013;105(16):1212–20.

[14] Nonaka D. A study of DNp63 expression in lung non-small cell carcinomas. Am J Surg Pathol 2012;36(6):895–9.

[15] Travis WD. Pathology of lung cancer. Clin Chest Med 2011;32(4):669–92.

[16] Travis WD, Brambilla E, Noguchi M, et al. Diagnosis of lung adenocarcinoma in resected specimens: implications of the 2011 International Association for the Study of Lung Cancer/American Thoracic Society/European Respiratory Society

[17] Kadota K, Yeh YC, Sima CS, et al. The cribriform pattern identifies a subset of acinar predominant tumors with poor prognosis in patients with stage I lung adenocarcinoma: a

conceptual proposal to classify cribriform predominant tumors as a distinct histologic subtype. Mod Pathol 2014;27(5):690–700.

[18] Miyoshi T, Satoh Y, Okumura S, et al. Early-stage lung adenocarcinomas with a micropapillary pattern, a distinct pathologic marker for a significantly poor prog- nosis. Am J Surg Pathol 2003;27(1):101–9.

[19] Tsutsumida H, Nomoto M, Goto M, et al. A micropapillary pattern is predictive of a poor prognosis in lung adenocarcinoma, and reduced surfactant apoprotein A expression in the micropapillary pattern is an excellent indicator of a poor prog- nosis. Mod Pathol 2007;20(6):638–47.

[20] Nitadori J, Bograd AJ, Kadota K, et al. Impact of micropapillary histologic sub- type in selecting limited resection vs lobectomy for lung adenocarcinoma of 2cm or smaller. J Natl Cancer Inst 2013;105(16):1212–20.

[21] Nonaka D. A study of DNp63 expression in lung non-small cell carcinomas. Am J Surg Pathol 2012;36(6):895–9.

[22] Travis WD. Pathology of lung cancer. Clin Chest Med 2011;32(4):669–92.

[23] Travis WD, Brambilla E, Noguchi M, et al. Diagnosis of lung adenocarcinoma in resected specimens: implications of the 2011 International Association for the Study of Lung Cancer/American Thoracic Society/European Respiratory Society

[24] Brambilla C, Laffaire J, Lantuejoul S, et al. Lung squamous cell carcinomas with basaloid histology represent a specific molecular entity. Clin Cancer Res 2014; 20(22):5777–86.

[25] Lundin S, Mang H, Smithies M, Stenqvist O, Frostell C.Intensive Care Med. Inhalation of nitric oxide in acute lung injury: results of a European multicentre study. The European Study Group of Inhaled Nitric Oxide. 1999 Sep;25(9):911-9.

[26] Ravdin, P.M. & Clark, G.M. Breast Cancer Res Tr (1992) 22: 285. https://doi.org/10.1007/BF01840841Gold KA, Wistuba II, Kim ES. New strategies in squamous cell carcinoma of the lung: identification of tumor drivers to personalize therapy. Clin Cancer Res 2012;18(11):3002–7.

[27] Quinlan, J. R. (1990). Learning logical definitions from relations. Machine Learning, 5:239-266. C4.5 improved: discrete and continuous attributes, missing attribute values, attributes with differing costs, pruning trees (replacing irrelevant branches with leaf nodes).

[28] Xin Yao, Ensemble of Classifiers based on Multi-Objective Genetic Sampling for Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering PP (99):1-1 12 February 2019.

[29] Pardo J, Martinez-Penuela AM, Sola JJ, et al. Large cell carcinoma of the lung: an endangered species? Appl Immunohistochem Mol Morphol 2009;17(5): 383–92.

[30] Barbareschi Li, Yanying & Che, Jinxing & Yang, Youlong, 2018. ubsampled support vector regression ensemble for short term electric load forecasting.

[31] Sholl LM. Large-cell carcinoma of the lung: a diagnostic category redefined by immunohistochemistry and genomics. Curr Opin Pulm Med 2014;20(4):324–31.

[32] Hwang DH, Szeto DP, Perry AS, et al. Pulmonary large cell carcinoma lacking squamous differentiation is clinicopathologically indistinguishable from solid-subtype adenocarcinoma. Arch Pathol Lab Med 2014;138(5):626–35.

[33] Rekhtman N, Tafe LJ, Chaft JE, et al. Distinct profile of driver mutations and clinical features in immunomarker-defined subsets of pulmonary large-cell carci-noma. Mod Pathol 2013;26(4):511–22.

[34] Chen, I.-C., Hill, J.K., Ohlemüller, R., Roy, D.B. & Thomas, C.D. (2011) Rapid Range Shifts of Species Associated with High Levels of Climate Warming. Science, 333, 1024-1026.

[35] (Zheng, Yoon, and Lam 2014) Expert Systems with Applications 41(4):1476–1482. DOI: 10.1016/j.eswa.2013.08.044

[36] Maeda H, Matsumura A, Kawabata T, et al. Adenosquamous carcinoma of the lung: surgical results as compared with squamous cell and adenocarcinoma cases. Eur J Cardiothorac Surg 2012;41(2):357–61.

[37] Mordant P, Grand B, Cazes A, et al. Adenosquamous carcinoma of the lung: surgical management, pathologic characteristics, and prognostic implications. Ann Thorac Surg 2013;95(4):1189–95.

[38] Martin LW, Correa AM, Ordonez NG, et al. Sarcomatoid carcinoma of the lung: a predictor of poor prognosis. Ann Thorac Surg 2007;84(3):973–80.

[39] Travis WD. Sarcomatoid neoplasms of the lung and pleura. Arch Pathol Lab Med 2010;134(11):1645–58.

[40] Gustafsson BI, Kidd M, Chan A, et al. Bronchopulmonary neuroendocrine tumors. Cancer 2008;113(1):5–21.

[41] Litzky LA. Pulmonary neuroendocrine tumors. Surg Pathol 2010;3:27–59.

[42] Moran CA, Suster S, Coppola D, et al. Neuroendocrine carcinomas of the lung: a critical analysis. Am J Clin Pathol 2009;131:206–21.

[43] Klimstra DS, Modlin IR, Coppola D, et al. The pathologic classification of neuroendocrine tumors: a review of nomenclature, grading, and staging systems. Pancreas 2010;39(6):707–12.

[44] Rindi G, Klersy C, Inzani F, et al. Grading the neuroendocrine tumors of the lung: an evidence-based proposal. Endocr Relat Cancer 2014;21(1):1–16.

[45] Lin O, Olgac S, Green I, et al. Immunohistochemical staining of cytologic smears with MIB-1 helps distinguish low-grade from high-grade neuroendocrine neoplasms. Am J Clin Pathol 2003;120(2):209–16.

[46] Pelosi G, Rodriguez J, Viale G, et al. Typical and atypical pulmonary carcinoid tumor overdiagnosed as small-cell carcinoma on biopsy specimens: a major pitfall in the management of lung cancer patients. Am J Surg Pathol 2005; 29(2):179–87.

[47] Dishop MK, Kuruvilla S. Primary and metastatic lung tumors in the pediatric population: a review and 25-year experience at a large children's hospital. Arch Pathol Lab Med 2008;132(7):1079–103.

[48] Yu DC, Grabowski MJ, Kozakewich HP, et al. Primary lung tumors in children and adolescents: a 90-year experience. J Pediatr Surg 2010;45(6):1090–5.

[49] Filosso PL, Guerrera F, Evangelista A, et al. Prognostic model of survival for typical bronchial carcinoid tumours: analysis of 1109 patients on behalf of the European Society of Thoracic Surgeons (ESTS) Neuroendocrine Tumours Working Group. Eur J Cardiothorac Surg 2015;48(3):441–7.

[50] Fink G, Krelbaum T, Yellin A, et al. Pulmonary carcinoid: presentation, diagnosis, and outcome in 142 cases in Israel and review of 640 cases from the literature. Chest 2001;119(6):1647–51.

[51] Hobe AUCk TR, Knutson CO, Polk HC Jr. Clinical aspects of invasive carcinoid tumors. South Med J 1975;68(1):33–7.

[52] Sachithanandan N, Harle RA, Burgess JR. Bronchopulmonary carcinoid in multiple endocrine neoplasia type 1. Cancer 2005;103(3):509–15.

[53] Filosso PL, Rena O, Guerrera F, et al. Clinical management of atypical carcinoid and large-cell neuroendocrine carcinoma: a multicentre study on behalf of the European Society of (Wolpert, APRIL 1997)Cardiothorac Surg 2015;48(1):55–64.

[54] National Cancer Institute. Surveillance, Epidemiology, and End Results (SEER) Cancer Statistics Review 1975-2010. Available at: http://seer.concer.gov/csr/ 1975_2010/. Accessed January 15, 2016.

[55] Mike W. Shields, Matthew C. Casey, A theoretical framework for multiple neural network systems, ScienceDirect, March 2008.

[56] Nakamura N, Miyagi E, Murata S, et al. Expression of thyroid transcription factor- 1 in normal and neoplastic lung tissues. Mod Pathol 2002;15(10):1058–67.

[57] Zamecnik J, Kodet R. Value of thyroid transcription factor-1 and surfactant apo-protein A in the differential diagnosis of pulmonary carcinomas: a study of 109 cases. Virchows Arch 2002;440(4):353–61.

[58] Travis WD, Linnoila RI, Tsokos MG, et al. Neuroendocrine tumors of the lung with proposed criteria for large-cell neuroendocrine carcinoma. An ultrastructural, immunohistochemical, and flow cytometric study of 35 cases. Am J Surg Pathol 1991;15(6):529–53.

[59] Battafarano RJ, Fernandez FG, Ritter J, et al. Large cell neuroendocrine carcinoma: an aggressive form of non-small cell lung cancer. J Thorac Cardiovasc Surg 2005;130(1):166–72.

[60] Mukhopadhyay S. Utility of small biopsies for diagnosis of lung nodules: doing more with less. Mod Pathol 2012;25:S43–57.

[61] Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML Downing JR, Caligiuri MA, Bloomfield CD, Lander ES. *Science*, 1999 Oct 15; 286(5439):531-537.

[62] Thunnissen E, Noguchi M, Aisner S, et al. Reproducibility of histopathological diagnosis in poorly differentiated NSCLC: an international multiobserver study. J Thorac Oncol 2014;9(9):1354–62.

[63] Loo PS, Thomas SC, Nicolson MC, et al. Subtyping of undifferentiated non-small cell carcinomas in bronchial biopsy specimens. J Thorac Oncol 2010;5(4): 442–7.

[64] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[65] 'Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays'.U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine,Proc. Natl. Acad. Sci. USA, Vol. 96, Issue 12, 6745-6750, June 8,1999.

[66] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[67] Tarang Shah, about Train, Validation and test sets in Machine learning,(https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7) dec 6, 2017.

[68] Rohith Gandhin, Support Vector Machine, Introduction to Machine Learning Algorithm, towards science direct, https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47, jun 7, 2017.

[69] Rahul saxena, how decision tree algorithm works (https://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/), data science, machine learning, dataaspirants. January 30, 2017.

[70] Li Y, Pan Y, Wang R, et al. ALK-rearranged lung cancer in Chinese: a compre-hensive assessment of clinicopathology, IHC, FISH and RT-PCR. PLoS One 2013;8(7):e69016.

[71] Gruber K, Kohlhaufl M, Friedel G, et al. A novel, highly sensitive ALK antibody 1A4 facilitates effective screening for ALK rearrangements in lung adenocarcinomas by standard immunohistochemistry. J Thorac Oncol 2015;10(4):713–6.

[72] Sholl LM, Sun H, Butaney M, et al. ROS1 immunohistochemistry for detection of ROS1-rearranged lung adenocarcinomas. Am J Surg Pathol 2013;37(9):1441–9.

[73] Lin F, Liu H. Immunohistochemistry in undifferentiated neoplasm/tumor of uncertain origin. Arch Pathol Lab Med 2014;138(12):1583–610.

[74] Zhang K, Deng H, Cagle PT. Utility of immunohistochemistry in the diagnosis of pleuropulmonary and mediastinal cancers: a review and update. Arch Pathol Lab Med 2014;138(12):1611–28.

[75] Thunnissen E, van der Oord K, den Bakker M. Prognostic and predictive biomarkers in lung cancer. A review. Virchows Arch 2014;464(3):347–58.

[76] Verma M. The role of epigenomics in the study of cancer biomarkers and in the development of diagnostic tools. Adv Exp Med Biol 2015;867:59–80.

[77] Lindeman NI, Cagle PT, Beasley MB, et al. Molecular testing guideline for selection of lung cancer patients for EGFR and ALK tyrosine kinase inhibitors. Arch Pathol Lab Med 2013;137:828–60.

[78] Koboldt DC, Steinberg KM, Larson DE, et al. The next-generation sequencing revolution and its impact on genomics. Cell 2013;155(1):27–38.

[79] Nie K, Jia Y, Zhang X. Cell-free circulating tumor DNA in plasma/serum of non- small cell lung cancer. Tumour Biol 2015;36(1):7–19.

[80] Linardou H, Dahabreh IJ, Bafaloukos D, et al. Somatic EGFR mutations and efficiacy of tyrosine kinase inhibitors in NSCLC. Nat Rev Clin Oncol 2009;6(6): 352–66.

[81] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, Proc, colon cancer dataset. Natl. Acad. Sci. USA, Vol. 96, Issue 12, 6745-6750, June 8, 1999.

[82] Pines G, Ko¨stler WJ, Yarden Y. Oncogenic mutant forms of EGFR: lessons in signal transduction and targets for cancer therapy. FEBS Lett 2010;584(12): 2699–706.

[83] International Journal of Emerging Engineering Research and Technology, Volume 3, Issue 7, July 2015, PP 172 -178 ISSN 2349-4395 (Print) & ISSN 2349-4409.

[84] Avinash Navlani, KNN classification using scikit-learn (https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn). August 2nd, 2018.

[85] Avinash Navlani, Naïve Bayes classification (https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn). December 4th, 2018.

[86] Paez JG, Janne PA, Lee JC, et al. EGFR mutations in lung cancer: correlation with clinical response to gefitinib therapy. Science 2004;304(5676):1497–500.

[87] Lynch TJ, Bell DW, Sordella R, et al. Activating mutations in the epidermal growth factor receptor underlying responsiveness of non-small-cell lung cancer to gefitinib. N Engl J Med 2004;350(21):2129–39.

[88] David H. Wolpert and Willian G. Macready, No Free Lunch Theorems for Optimization. IEEE transactions on evolutionary computation. Vol. 1, NO. 1, APRIL 1997.

[89] Shepherd FA, Rodrigues Pereira J, Ciuleanu T, et al. Erlotinib in previously treated non-small-cell lung cancer. N Engl J Med 2005;353(2):123–32.

[90] (PDF) Application of Grasshopper Optimization Algorithm for Constrained and Unconstrained Test Functions. Available from: https://www.researchgate.net/publication/322597187_Application_of_Grasshopper_Optimization_Algorithm_for_Constrained_and_Unconstrained_Test_Functions [accessed Aug 07 2019].

[91] Maione P, Sacco PC, Sgambato A, et al. Overcoming resistance to targeted therapies in NSCLC: current approaches and clinical application. Ther Adv Med Oncol 2015;7(5):263–73.

[92] Toyokawa G, Seto T. Anaplastic lymphoma kinase rearrangement in lung cancer: its biological and clinical significance. Respir Investig 2014;52(6):330–8.

[93] Gainor JF, Shaw AT. Novel targets in non-small cell lung cancer: ROS1 and RET fusions. Oncologist 2013;18(7):865–75.

[94] Yi ES, Chung JH, Kulig K, et al. Detection of anaplastic lymphoma kinase (ALK) gene rearrangement in non-small cell lung cancer and related issues in ALK inhibitor therapy: a literature review. Mol Diagn Ther 2012;16(3):143–50.

[95] Rothschild SI. Targeted therapies in non-small cell lung cancer-beyond EGFR and ALK. Cancers (Basel) 2015;7(2):930–49.

[96] Masters GA, Temin S, Azzoli CG, et al. Systemic therapy for stage IV non-small-cell lung cancer: American Society of Clinical Oncology clinical practice guideline update. J Clin Oncol 2015;33(30):3488–515.

[97] Finocchiaro G, Toschi L, Gianoncelli L, et al. Prognostic and predictive value of MET deregulation in non-small cell lung cancer. Ann Transl Med 2015;3(6):83.

[98] Stewart EL, Tan SZ, Liu G, et al. Known and putative mechanisms of resistance to EGFR targeted therapies in NSCLC patients with EGFR mutations—a review. Transl Lung Cancer Res 2015;4(1):67–81.

[99] Stella GM, Scabini R, Inghilleri S, et al. EGFR and KRAS mutational profiling in fresh non-small cell lung cancer (NSCLC) cells. J Cancer Res Clin Oncol 2013; 139(8):1327–35.

[100] Stinchcombe TE. Novel agents in development for advanced non-small cell lung cancer. Ther Adv Med Oncol 2014;6(5):240–53.

[101] Khoo C, Rogers TM, Fellowes A, et al. Molecular methods for somatic mutation testing in lung adenocarcinoma: EGFR and beyond. Transl Lung Cancer Res 2015;4(2):126–41.

[102] Tucker T, Marra M, Friedman JM. Massively parallel sequencing: the next big thing in genetic medicine. Am J Hum Genet 2009;85(2):142–54.

[103] Popper HH, Ryska A, Timar J, et al. Molecular testing in lung cancer in the era of precision medicine. Transl Lung Cancer Res 2014;3(5):291–300.

[104] Deeb KK, Hohman CM, Risch NF, et al. Routine clinical mutation profiling of non- small cell lung cancer using next-generation sequencing. Arch Pathol Lab Med 2015;139(7):913–21.

[105] Coco S, Truini A, Vanni I, et al. Next generation sequencing in non-small cell lung cancer: new avenues toward the personalized medicine. Curr Drug Targets 2015;16(1):47–59.

[106] http://www.swarmintelligence.org/tutorials.php