CHAPTER 4

FEATURE SELECTION USING GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION

In this chapter, various categories for feature selection have been discussed. Genetic Algorithm (Leardi 1992) and Particle Swarm Optimization (Wang et al. 2007) have been used to select the features which are classified using Minimum Distance Classifier, k-NN and SVM classifier. The results obtained from GA and PSO are used for comparing the results of proposed feature selection algorithm.

4.1 INTRODUCTION

A feature is an individual quantifiable property of the process being observed. Using an identified set of features, any learning algorithm would proceed to classification. Existence of redundant, noisy or irrelevant features might affect the performance of classification. In order to address this issue, feature selection is proposed to choose a subset of features that are relevant to the target idea (Dash & Liu 1997). The objective of feature selection is to select a subset of features from the extracted features which can efficiently describes itself while reducing noise or irrelevant features and still provide good classification results. To remove an irrelevant feature, a feature selection criterion is required which can measure the relevance of each feature with the output class/labels accordingly.
4.1.1 Feature Selection

The underlying purpose of feature selection includes reducing the amount of data needed for learning, improving the system accuracy and increasing the comprehensibility of the learned model (Liu et al. 2005). Figure 4.1 shows the process of a feature selection method (Dash & Liu 1997) which consists of five basic steps. They are

1. Initialization: A feature selection algorithm starts with an initialization procedure based on all the original features.

2. Subset discovery: A discovery procedure to generate candidate subsets. It is a search procedure (Langley 1994), which can start with no features, all features, or a random subset of features. Many search techniques including Evolutionary Computing (EC) techniques would be applied to search for the best subset of features.

3. Subset evaluation: An evaluation function to measure the goodness of the generated feature subsets.

4. Stopping criterion: The algorithm will stop its execution with the given criterion, which commonly relies on the generation procedure or the evaluation function. The former can be a predefined number of features selected or a predetermined maximum number of iterations reached. The later includes whether an optimal feature subset as per the evaluation function is generated or addition or deletion of any feature does not reflect in produced better subset.

5. Results validation: The validity of the selected subset is tested by carrying out tests on unseen data.
4.1.2 Categories of Feature Selection

The feature selection methods were broadly classified (Liu & Yu 2005), into filter, wrapper and embedded methods and the flow diagrams are given in Figure 4.2 to Figure 4.4 (Zhang et al. 2014).

4.1.2.1 Filter approach

Filters perform feature selection independently of the classifier.

Figure 4.2 Flow diagram for filter approach
4.1.2.2 Wrapper approach

In wrapper methods the feature selection criterion is the performance of the classifier i.e. the classifier is wrapped on a search algorithm that eventually will find a subset which gives the highest classification result (Kohavi & John 1997).

![Flow diagram for wrapper approach](image)

**Figure 4.3 Flow diagram for wrapper approach**

4.1.2.3 Embedded approach

Embedded methods include feature selection as a part of the training process without splitting the data into training and testing sets. This approach (Guyon & Elisseeff 2003) is similar to the wrapper approach in which the features are specifically selected for certain learning algorithm. Moreover, in this approach, the features are selected during the learning process.
4.2 GENETIC ALGORITHM

Evolutionary Computing (EC) techniques are population-based heuristic search techniques with a set of genetically motivated operations. These operations are used by a population of candidate solutions to obtain the optimal or near-optimal solution to the problem. In recent times various EC algorithms have been applied to feature selection problems such as GA (Leardi 1992), PSO (Wang et al. 2007), Ant Colony Optimization (ACO) (Cai et al. 2010) and other metaheuristic algorithms, all these algorithms are based on wrapper approach. In the proposed breast cancer classification system, GA and PSO algorithm have been used to select the optimal features and the results are then used for comparing the results of proposed feature selection approach with some conventional techniques.
4.2.1 Introduction

Genetic algorithms (GA) are a component of evolutionary computing, which is a rapid growing area of Artificial Intelligence. In the year 1960s, idea of evolutionary computing was proposed by I. Rechenberg. Evolutionary algorithm (EA) is a subset of EC, a population based metaheuristic optimization algorithm. An EA uses the mechanisms motivated by biological evolution, such as reproduction, mutation, recombination and selection. Figure 4.5 shows the general structure of an evolutionary process (Eiben & Smith 2003). GA is popular type of EA and it seeks the solution to the problem in the form of strings of numbers, by applying operators such as recombination and mutation. GA shares the properties of adaptation through an iterative process that accumulates and strengthens value variation through trial and error (Preux & Talbi 1999). An introduction of GA and the feature selection based on genetic algorithm for breast cancer classification are explained in this chapter with result analysis.

GA is inspired by Darwin's theory about evolution through the survival of the fittest (Espinoza et al. 2003). It is an adaptive heuristic procedure used to find optimal solutions to search problems through the principles of evolutionary biology (Goldberg 2009, Holland 2001).

The characteristics of GA are as follows,

- GA represents an intelligent exploitation of random search used to solve optimization problems
- It exploits the historical information effectively to direct the search into the region of good performance within the search space
- Naturally the competition of among individual will always results in fittest individuals dominating over the weaker ones
Optimization is a process that picks the best solution among the solutions for a problem. Optimization problem are centered around three factors

1. An objective function that needs to be maximized or minimized
2. Set of unknowns or variables that affects the objective function
3. A set of constraints that allow unknowns to take on certain values but exclude others

**Search Space:** In solving problems, some solution will be best among others. The space of all feasible solutions is called search space. Each possible solution can be marked by its value (fitness) for the problem. GA looks for better solution among a number of possible solutions represented by one point in the search space.

GA looks for a solution is then equal to some extreme value in the search space (Maximum or minimum). Using GA, the process of finding solutions generates other points as evolution proceeds.

![Figure 4.5 General structure of an evolutionary process](image-url)
**Representation:** Chromosome representation should be fixed. It means that one should be able to encode all possible solutions to the problem.

**Fitness function:** Fitness function measures the quality of a solution. It should be planned in such a way that better solutions will always have higher fitness value than the other solutions. The fitness function plays an indispensable role in the selection process.

**Initialization of population:** The size of population, usually depends on the problems nature. The initial population is generated randomly, allowing the entire range of possible solutions. Initialization may be

- Randomly generated individuals
- A previously saved population
- A set of solutions provided by a human expert
- A set of solutions provided by another heuristic algorithm

**Selection:** The primary idea of the selection operator is to emphasize the good solutions and eliminate the bad solutions in a population while keeping the population size constant. Selection is based on genetic operator that selects a chromosome from the current generation’s population, and inserted into the next generation’s population. In a selection operator, the possibility of a chromosome getting selected is proportional to its fitness value. It illustrates the concept of survival of the fittest.

Genetic operators are crossover (also called recombination) and mutation. There are different techniques to implement selection in Genetic Algorithms such as tournament selection, roulette wheel selection, proportionate selection, rank selection, steady state selection etc.,.
**Crossover:** The crossover operator is used to create new chromosome (offspring) from the existing chromosomes (parents) available in the mating team after applying selection operator. This operator exchanges the gene information between the solutions in the mating team. Through the random choice, the probability of crossover applied is generally between 0.6 and 1.0. Crossover occurs in evolution according to the user defined crossover probability. The most common method of encoding is binary coded. Chromosomes are strings of ones and zeros and each position in the chromosome represents a particular characteristic of the problem. The most popular crossover selects any two chromosome strings randomly from the mating team and some portion of the strings is exchanged between the strings.

**Mutation:** Mutation is a genetic operator that changes one or more gene values in a chromosome from its initial state. The new gene values being added to the gene pool and by using these new gene values, the GA may be able to arrive a better solution than that was previously generated. The mutation operator changes the value of a chromosome gene randomly with the given probability.

**Accepting:** After producing offsprings they must be inserted into the population. By a reinsertion scheme, individuals should be inserted into the new population and it determines which individuals of the population will be replaced by the offsprings. The selection algorithm determines the reinsertion scheme.

**Termination:** The above process is repeated until a termination condition is reached. There are two common termination criteria. The first allows the GA to have a definite number of generations which may be enough to produce a satisfactory solution. The second requires the convergence of the population. If complete convergence is necessary for termination, the GA will
only stop when all members of the population are same. Figure 4.6 explains the algorithm for Genetic Algorithm.

1. Start: Generate random population of N chromosomes (suitable solutions for the problem).
2. Fitness: Evaluate the fitness \( f(X) \) of each chromosome \( X \) in the population.
3. New Population: Create a new population by iteratively following steps until the new population is complete.
   a. Selection: Select two parent chromosomes from a population according to their fitness value (the best fitness, the bigger chance to be selected).
   b. Crossover: Crossover the parents to form a new offspring (children) using the crossover probability. If no crossover is performed, offspring is an exact copy of parents.
   c. Mutation: mutate new offspring at each locus (position in chromosome) using the mutation probability.
   d. Accepting: Position or place the new offspring in a new population.
4. Replace: Use new generated population for a further execution of algorithm
5. Test: If the end condition is satisfied then stop the process, and return the best solution in current population otherwise go to step 2.

**Figure 4.6 Algorithm for GA**
4.2.2 Feature Selection Based on GA for Breast Cancer Classification

In this work, GA is applied to select useful features from the extracted 123 features. To measure the performance of GA, the various classifiers used are Minimum Distance Classifier, k-NN and SVM for breast cancer classification system.

**Chromosome representation:** Chromosome represents the possible solution for a particular problem. The most often used way of encoding the feature selection from 123 features is a binary string. The random values are generated for feature position. The features are considered when the value in its position is greater than 0.5, otherwise it is ignored. Figure 4.7 shows the chromosome representation as a binary string.

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>-</th>
<th>-</th>
<th>F122</th>
<th>F123</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.67</td>
<td>0.97</td>
<td>0.34</td>
<td>0.18</td>
<td>0.24</td>
<td>0.01</td>
<td>0.83</td>
<td>0.92</td>
<td>0.35</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 4.7 Chromosome representation for feature selection using GA**

**Fitness function:** A fitness function is a specific type of objective function which is used to suggest that how close a given solution is to achieve the target as a particular value. This value reflects how best the optimal the solution is; the higher the number, the better the solution. By maximizing the fitness values in each generation, the genetic string with the global optimum value could be found to be the optimum result. The fitness function of an individual is determined by evaluating the classifier. The performance of breast cancer classification system is measured by its classification accuracy. Classification accuracy plays a major role in the process of breast cancer classification using significant features from 123 features. Here, the
classification accuracy is based on either with k-NN classifier or Minimum Distance Classifier or SVM classifier which is used as the fitness function to GA. The fitness function fitness \( t \) of GA is defined as in equation (4.1).

\[
\text{fitness}(t) = \text{Accuracy}(t)
\]  

(4.1)

Accuracy \( (t) \) is the test accuracy of testing data \( t \) in the classifier which is built with the feature subset selection of training data. The classification accuracy is estimated using equation (4.2).

\[
\text{Accuracy}(t) = \left( \frac{s}{n} \right) \times 100
\]  

(4.2)

\( s \) - Number of samples that are correctly classified in test data either by Minimum Distance Classifier or k-NN Classifier or SVM Classifier.

\( n \) - Total number of samples in test data.

**Initial population** Many individual solutions (chromosomes) are randomly generated in order to generate an initial population. So that each chromosome in the population represents one the possible solution for a specified problem. Therefore, a population represents a number of candidate solutions which allows the entire range of possible solution. In this feature selection method, each chromosome represents an optimal set of features that are chosen to classify the samples correctly.

**Roulette Wheel Selection:** As chromosomes are selected based on their relative fitness, the selection type is independent of representation. The first step is fitness assignment. Each individual in the selection pool receives a reproduction probability. It depends on its own fitness value and all other individuals in the selection pool. This fitness value is used for the actual selection. Murthy & Chowdhury (1996) and Estivill-castro & Murray (1998)
have used proportional selection. The simplest approach for proportional selection is the roulette wheel selection method. This work applies the simplest selection scheme, roulette-wheel selection, also called stochastic sampling with replacement. In this sampling method, each individual from the population occupies a region of the roulette wheel relative to its fitness value. If the individuals S are to be selected, the roulette wheel randomly rotates S times. For each rotation, the individual pointed by the roulette wheel is chosen. As a result, fitter individuals have higher chances of being chosen.

In this breast cancer classification system the chromosomes are elected based on their fitness values. Each chromosome occupies a portion of the wheel, chromosomes with high fitness value occupies larger portion than the less fitness value. Each time the wheel rotates, the individuals with the high fitness value will get elected. So, always the more informative features are carried to the next generation. Each and every run gives the optimal features that are useful for the breast cancer classification process.

**One point crossover:** It occurs with a user defined probability called the crossover probability P_c. (Estivill Castro 2000) uses a single or one point crossover that is context sensitive. Breast cancer classification system uses one point crossover, which is used to generate new children.

It randomly selects one crossover point and then copies everything before this point from the first parent and then everything after the crossover point copied from second parent. Consider the parents with ten features as shown in Figure 4.8(a). A location at which the parents are divided into two parts (crossover point) is randomly selected and shown in Figure 4.8(b). The elements of the parents are then swapped to generate two new children, which are shown in Figure 4.8(c).
<table>
<thead>
<tr>
<th>Feature selected for mutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before Mutation</td>
</tr>
<tr>
<td>After Mutation</td>
</tr>
</tbody>
</table>

**Figure 4.9 Mutation operation of child1 in GA**
**Elitist selection:** Breast cancer classification system uses elitist selection for evaluation. The individuals with the high fitness are copied into the next generation. The evaluation ranks the individuals by accuracy of classifier that corresponds to the quality of the individual solutions. An elitist strategy never replaces the best parent by any children worse than them (Mitchell 2002). Table 4.1 shows the parameters and its value for GA.

The four principal factors of GA are

- Population size: 50
- Crossover rate: 0.65
- Mutation probability $P_m$ as $1/L$, the chromosome length ‘L’ here the feature length 123
- Number of generations: 200

The GA model is selected for the analysis, the methods that are used in GA are

- Roulette wheel selection procedure has been used for the reproduction process
- One-point crossover has been used with a probability of $P_c$
- For every bit of the string, mutation happens with probability $P_m$
- Crossover rate between 0.65 and 1 (Jong 2005)
Table 4.1 Parameters and its value for GA

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of Population</td>
<td>50</td>
</tr>
<tr>
<td>Number of generations</td>
<td>200</td>
</tr>
<tr>
<td>Rate of Selection</td>
<td>0.5</td>
</tr>
<tr>
<td>Rate of Crossover</td>
<td>0.65</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>1/L</td>
</tr>
<tr>
<td>Type of selection</td>
<td>Roulette wheel selection</td>
</tr>
<tr>
<td>Type of crossover</td>
<td>One point crossover</td>
</tr>
<tr>
<td>Type of mutation</td>
<td>Flip bit mutation</td>
</tr>
<tr>
<td>Type of evaluation</td>
<td>Elitist selection</td>
</tr>
</tbody>
</table>

4.2.3 Results of GA

Totally 500 DDSM images (in which 288 are benign and 212 are malignant) have been considered for developing breast cancer classification system. Breast cancer classification using GA for feature selection has been implemented using MATLAB, initially 123 features were extracted for further analysis. Classification system has been evaluated using five-fold cross validation method.

GA with Minimum Distance Classifier selects 69 features and gives 94.23% average classifier accuracy, GA with k-NN classifier selects 62 features and provides 95.21% average classifier accuracy, GA with SVM classifier selects 72 features and gives 93.18% average classifier accuracy. The results are shown in Figure 4.10 and given in Table 4.2. From the above results, it is found that Genetic Algorithm with k-NN classifier selects optimal features with maximum classifier accuracy.
PARTICLE SWARM OPTIMIZATION ALGORITHM

PSO based feature selection methods have been used in classification of breast cancer in mammograms. The main advantage of using PSO is easier to implement, computationally less expensive and can converge quickly (Kennedy & Eberhart 1995).

4.3.1 Introduction

PSO is an evolutionary computation technique proposed by Kennedy & Eberhart (1995), and it has been used for feature selection due to
their global search ability in which, each potential solution is called a bird or particle. PSO is motivated by social behaviors such as bird flocking and fish schooling. In PSO, the algorithm maintains a population of particles, where each particle represents a potential solution to the optimization problem. Each particle is assigned to a randomized velocity. The particles are then flown through the search space (Shi & Eberhart 1998). Search space with velocities which are dynamically adjusted according to their past behaviors. Therefore, the particles have the sympathy to fly towards better search area during the path of the search process.

In PSO, each single solution is called as particle. All particles have fitness values which are estimated by the fitness function, and have velocities which indicate the direction of the particles to fly. The particles fly through the search space by following the particles with the current optimum solutions. PSO is initialized with a group of random particles (solutions) and then it searches the entire search space for optima by updating each generation. Each particle keeps track of the following information in the problem space.

- $x_i$, the current position of the particle
- $v_i$, the current velocity of the particle
- $y_i$, the personal best position of the particle which is the best position that it has achieved so far. This position yields the best fitness value for that particle
- The fitness value of best position is called $p_{best}$ and also stored. There are two approaches to PSO, namely
  - Local best ($l_{best}$) and
  - Global best ($g_{best}$)
For the g\textsubscript{best} model, the best particle is determined from the entire swarm. For the l\textsubscript{best} model, the swarm is divided into overlapping neighborhoods of particles. For each neighborhood, a best particle is determined the g\textsubscript{best}. PSO is a special case of l\textsubscript{best} when the neighborhood is the entire swarm. Another best value that is tracked by the PSO is the overall best value (g\textsubscript{best}) obtained so far by any particle in the population. The location of this overall best value is called y\textsubscript{g}. This location is also tracked by PSO. The PSO changes the velocity of each particle at each time step so that it moves toward its personal best and global best locations. The algorithm for implementing the PSO (Eberhart & Shi 2001) is described in Figure 4.11.

1. Initialize a population of particles with random positions and velocities on a d-dimensional problem space.
2. For each particle, evaluate the desired optimization fitness function of d variables.
3. Compare particle’s fitness evaluation with particle’s personal best value (p\textsubscript{best}). If the current fitness function value is better than p\textsubscript{best}, then set the p\textsubscript{best} value equal to the current value, and the p\textsubscript{best} location equal to the current location in the d dimensional space.
4. Compare fitness evaluation with the population’s overall previous best value. If the current value is better than the global best value (g\textsubscript{best}), then set g\textsubscript{best} to the current particle’s value and set the global best position y\textsubscript{g} to the current particle’s position.
5. Change the velocity and position of the particle according to equations (4.3) and (4.4) respectively.

\begin{align*}
\tilde{v}_i(t + 1) &= w\tilde{v}_i(t) + c_1 r_1(t)(y_i(t) - x_i(t)) + c_2 r_2(t)(y_g(t) - x_i(t)) \\
x_i(t + 1) &= x_i(t) + v_i(t + 1)
\end{align*}

where w- is the inertia weight, c\textsubscript{1} and c\textsubscript{2} are the acceleration constants, r\textsubscript{1}(t) and r\textsubscript{2}(t) are random numbers generated in the range between 0 and 1.
6. Loop to Step 2 until a termination criterion is met. The criterion is usually a sufficiently good fitness or a maximum number of iterations. In this base work, a maximum number of iterations are used as termination criterion.

**Figure 4.11 Algorithm for Particle Swarm Optimization**
**Initial population:** Many individual solutions (particles) are generated randomly to create an initial population. Each particle position in the population denotes one possible solution for a chosen problem. Therefore, a population denotes a number of candidate solutions which allows the entire range of possible solution.

**Velocity calculation:** The change of the particle position can be represented by the concept of velocity. Velocity of each agent can be modified by the above equations (4.3) & (4.4). Inertia weight (w) often is decreased linearly from 0.9 to 0.4 during a run. Appropriate selection of the inertia weight provides a balance between global and local exploration and exploitation, and to find an optimal solution successfully with less number of iterations.

In earlier solutions, the inertia weight was set initially to a constant, but later it is suggested from the results to have a larger value, initially to promote global exploration of search space and then gradually decrease it to get more refined solutions. The following weighting function expressed in equation (4.5) is applied

\[
w = w_{\text{max}} - \frac{w_{\text{max}} - w_{\text{min}}}{n} \times i
\]  

(4.5)

where

- \( w_{\text{max}} \) - Initial inertia weight
- \( w_{\text{min}} \) - Final inertia weight
- \( n \) - Maximum number of iterations
- \( i \) - Current iteration
A large inertia weight facilitates global search, while a small inertia weight facilitates local search. A dynamically changing inertia weight provides PSO a better performance over a fixed value. It can be changed linearly over the course of PSO running or dynamically changed, based on the measurement of the PSO performance. Generally $c_1$ and $c_2$ are set to 2.0 which will make the search around the regions centered at $p_{best}$ and $g_{best}$.

**Personal best and global best positions of particle:** The personal best position of a particle is calculated as follows in Equation (4.6)

$$p_{id}(t + 1) = \begin{cases} p_{id}(t) & \text{if } f(x_{id}(t+1) \leq f(p_{id}(t)) \\ x_{id}(t+1) & \text{if } f(x_{id}(t+1)) \geq f(p_{id}(t)) \end{cases}$$ \hspace{1cm} (4.6)

Here $t$ represents the time period. The particle to be drawn towards the best particle in the swarm is the global best position of each particle. At the initial stage an initial position of the particle is considered as the $p_{best}$. The $g_{best}$ had been identified with maximum fitness function value

**4.3.2 Feature Selection Based on PSO for Breast Cancer Classification**

Feature selection based on PSO algorithm is used for breast cancer classification system is presented in this section. Sensitivity analysis is carried out for different combinations of parameter value, to find the optimal values for the parameters. Table 4.3 shows the parameters and its values for PSO.

**Particle representation:** In the process of feature selection using PSO, each particle position represents a solution. The most used way of encoding the feature selection from the 123 features is a binary string. Here, the random values are generated for feature position. The features are considered when the value in its position is greater than 0.5, otherwise it is
ignored. The particle is represented as a binary string for feature selection which is given in Figure 4.12.

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F122</th>
<th>F123</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0.67</td>
<td>0.97</td>
<td>0.34</td>
<td>0.18</td>
<td>0.24</td>
<td>0.01</td>
<td>0.83</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 4.12 Particle representation for feature selection using PSO**

In PSO based feature selection method, the fitness function of an individual is determined by evaluating the k-NN classifier or Minimum Distance Classifier or SVM classifier. The fitness value of each particle is estimated by using the equation (4.1).

**Initial population:** Initial population size considered for this problem is 50.

**Finding new solutions:** According to the particle’s own individual experience and those of the neighbors, the particle varies the vector position in the vector space at each generation. The new velocity is estimated using equation (4.3) for deterministic inertia weight. The particle’s position is updated based on the equation (4.5).
Table 4.3 Parameters and its value for PSO

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>50</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>200</td>
</tr>
<tr>
<td>c₁</td>
<td>2.2</td>
</tr>
<tr>
<td>c₂</td>
<td>2.2</td>
</tr>
<tr>
<td>w_{\text{max}}</td>
<td>0.9</td>
</tr>
<tr>
<td>w_{\text{min}}</td>
<td>0.4</td>
</tr>
</tbody>
</table>

It is observed that the maximum fitness value is obtained at population size of 50 with the maximum number of iterations as 200, c₁ as 2.2 and c₂ as 2.2, the run gets better optimum result than all other value.

4.3.3 Results of PSO

Totally 500 DDSM images (in which 288 are benign and 212 are malignant) have been considered for developing breast cancer classification system. Breast cancer classification system using PSO algorithm, for feature selection has been implemented using MATLAB. In this system, initially 123 features were extracted from the mass portion. Classification system has been evaluated using five-fold cross validation method. Classifiers used in this work are Minimum Distance Classifier, k-NN classifier and SVM classifier.

PSO with Minimum Distance Classifier selects 64 features and gives 96.25% average classifier accuracy, PSO with k-NN classifier selects 58 features and gives 96.84% average classifier accuracy, PSO with SVM classifier selects 67 features and gives 95.47% average classifier accuracy and are given in Figure 4.13 and Table 4.4. From the above results PSO with k-NN
classifier selects minimum features (58 features) with maximum classifier accuracy (96.84%).

![Performance analysis of PSO with Minimum Distance Classifier, k-NN and SVM classifier](image)

**Figure 4.13** Performance analysis of PSO with Minimum Distance Classifier, k-NN and SVM classifier

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Minimum Distance Classifier</th>
<th>k-NN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO</td>
<td>No of Features Selected from 123</td>
<td>64</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Average Accuracy in %</td>
<td>96.25</td>
<td>96.84</td>
</tr>
</tbody>
</table>

From the above discussions it is found that the PSO with k-NN classifier gives better result than Genetic Algorithm.