

## **CHAPTER 5**

### **HEURISTIC OPTIMIZATION TECHNIQUES FOR FEATURE SELECTION**

#### **5.1 INTRODUCTION**

In the previous chapter the Evolutionary Computing algorithms such as Genetic Algorithm (GA), Particle swarm optimization (PSO) have been applied for feature selection in breast cancer classification system.

From the previous results, it is clear that the maximum numbers of features degrades the classification accuracy as well as increase their computational complexity. Therefore, the consistent feature vectors should be considered to reduce the amount of irrelevant information and thereby the classification accuracy can be increased.

Metaheuristic algorithms are capable of extracting optimal features from a set of features and may generate the best features in practice. The different types of metaheuristic algorithms are Harmony Search (HS) (Geem 2010), Cuckoo Search (CS) algorithm (Yang & Deb 2009), Bat Algorithm (BA) (Yang 2010), Shuffled Frog Leaping algorithm (Eusuff & Lansey 2003), Cat Swarm algorithm (Chu et al. 2006), Monkey Search (Mucherino & Seref 2007), Grey Wolf Optimizer (Mirjalili et al. 2014), Water flow-like algorithm (Yang & Wang 2007), Forest Optimization Algorithm (Ghaemi & Feizi Derakhshi 2014), Lion Optimization Algorithm (LOA) (Maziar Yazdani & Fariborz Jolai 2016).



In the present work totally 123 features have been extracted and the feature vector is used for feature selection. In this work three feature selection approaches have been proposed for breast cancer classification system. The extracted features are given as input to the algorithm and it selects relevant features to produce the highest classification accuracy. The proposed feature selection algorithms are

- Hybrid Harmony Search
- Enhanced cuckoo search
- Improved Lion Optimization Algorithm

These algorithms are based on wrapper approach.

## **5.2 HYBRID APPROACH TOWARDS FEATURE SELECTION**

Hybrid Harmony Search (HHS) is a combination of Harmony Search and Cuckoo Search. HHS initiated to improvise the performance of standard Harmony Search algorithm. The Cuckoo Search (CS) algorithm is inspired by the reproduction strategy of cuckoos. Cuckoo search combines the advantages of both GA and PSO. Since its simple nature, efficiency and optimal random search path, it has been applied to various practical engineering optimization problems. Harmony Search (HS) is a meta-heuristic algorithm, which is conceptualized using the musical process of searching for a perfect state of harmony. Harmony Search is less sensitive to the chosen parameters, which means that not required to fine-tune these parameters to get quality solutions. So the good features of HS and CS are fully exploited and combined into a new hybrid metaheuristic algorithm. Hybrid Harmony search has been proposed to select optimal features for breast cancer classification system. Minimum Distance Classifier, k-NN classifier and SVM classifier have been used for classification of breast cancer. The performance of the

new hybrid algorithm is compared with the Genetic Algorithm, Particle Swarm Optimization algorithm, Cuckoo Search and Harmony Search.

### **5.2.1 Harmony Search**

The Harmony Search (HS) is a meta-heuristic algorithm developed by Geem et al. (2001). HS imitates the process of music improvisation and it was conceptualized using the musical process of searching for a perfect state of harmony. HS attempts to find a solution vector that optimizes a given objective function. During such an iterative search process, each musician (decision variable) chooses a note (variable value) in order to find a best harmony (a potential global optimum) in conjunction with the other musicians.

HS has the following advantages:

1. It has fewer mathematical requirements and does not require initial value settings of the decision variables (Mahdavi et al. 2007)
2. It generates a new solution by considering all of the existing vectors
3. It can explore the search space in a parallel optimization environment, in which each solution (harmony) vector is generated by intelligently exploring and exploiting a search space (Geem 2009)

From the above advantages, HS is very flexible and capable of producing improved solutions.

The HS imitates the musical performance process that occurs when a musician attempts to find a state of harmony. In the music production



process, a musician selects and collects different notes from a whole range of notes (Salam et al. 2016). The musician then plays the notes with musical instruments to search for an ideal state of harmony in the music. This kind of process is similar to an optimum design process that involves the search for an optimum solution for optimization problems where the solution is determined by an objective function. The pitch of each musical instrument determines the aesthetic quality, just as the fitness function or objective function is determined by the set of values assigned to each design variable. Aesthetic sound quality can be iteratively improved the same way a fitness function value can be improved iteration after iteration (Geem et al. 2001).

During the improvisation process of music, a musician has three possible options:

- 1) Play any eminent piece of music exactly from his or her memory (a series of pitches in harmony)
- 2) Play something similar to a known piece (thus adjusting the pitch slightly)
- 3) Compose new or random notes. Geem (2010) outlined the above three choices into quantitative optimization and these three corresponding components become into: usage of harmony memory, pitch adjusting and randomization

Original HS uses five parameters including three core parameters such as

- 1) Size of Harmony Memory Search (HMS)
- 2) The Harmony Memory Considering Rate (HMCR)
- 3) The maximum number of iterations  $K$

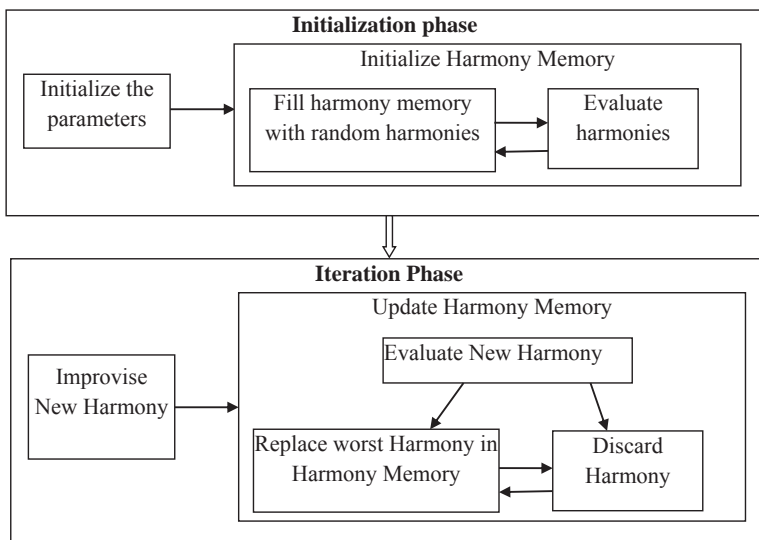
Two optional ones such as

- 1) Pitch Adjustment Rate (PAR) and
- 2) Adjusting bandwidth or Fret Width (FW). The number of musicians  $N$  is equal to the number of variables in the optimization function

Some existing equivalences between the musical terms, optimization terms and feature selection terms are illustrated in Table 5.1. Each musician corresponds to each decision variable, the pitch of each musical instrument corresponds to the value of a decision variable, a new harmony produced by all the musical instruments corresponds to a solution for the optimization problems, and the musical harmony iteratively improving practice after practice corresponds to enhancing the solution vector iteration after iteration.

**Table 5.1 Concept of mapping from HS to FS**

<b>Musical terms</b>	<b>Optimization terms</b>	<b>Feature Selection terms</b>
Improvisation	Generation	Generation
Harmony	Solution Vector	Subset
Harmony memory	Solution storage	Subset storage
Harmony evaluation	Fitness function	Subset evaluation
Musician note	Variable Value	Feature
Pleasing harmony	Optimal solution	Optimal subset
Musician	Decision variable	Feature Selector
Pitch	Value	Feature Value



(Source: Geem et al. 2001)

**Figure 5.1 Initialization and iteration phases of Harmony search**

As shown in Figure 5.1, the HS consists of the following steps and is listed in Figure 5.2.

- Step 1: Initialise the problem and the algorithm parameters.
- Step 2: Initialise the Harmony Memory (HM).
- Step 3: Improvise the new harmony ( $f'$ )
- Step 4: Update the HM.
- Step 5: Check the stopping criterion.
- Step 6: Cadenza.

**Figure 5.2 Algorithm for Harmony Search**

$$\text{HM matrix} = \begin{bmatrix} f_1^1 & f_2^1 & \dots & \dots & \dots & f_N^1 \\ f_1^2 & f_2^2 & \dots & \dots & \dots & f_N^2 \\ f_1^3 & f_2^3 & \dots & \dots & \dots & f_N^3 \\ f_1^4 & f_2^4 & \dots & \dots & \dots & f_N^4 \\ \vdots & \vdots & \dots & \dots & \dots & \vdots \\ f_1^{\text{HMS}} & f_2^{\text{HMS}} & \dots & \dots & \dots & f_N^{\text{HMS}} \end{bmatrix}$$

**Figure 5.3 Harmony Memory as a matrix of solutions**

The above steps are described in detail in the below subsections

### **Step 1: Initialize the problem and the algorithm parameters**

In this step, the HS parameters are initialized. The parameters comprise the following:

**Harmony Memory Size (HMS):** HMS defines the number of harmonies (solutions) that are stored in the HM which is similar to the population size in the GA.

**Harmony Memory Consideration Rate (HMCR):** HMCR is used during the improvisation process to decide whether the value of a decision variable of a new harmony should be chosen from the value of any harmony in the HM or at random from the possible range space. HMCR usually takes a value in the range [0, 1]. The probability of randomly selecting the value of the decision variable from the possible range is given as  $1 - \text{HMCR}$ .

**Pitch Adjusting Rate (PAR):** PAR is also used during the improvisation process to determine whether to modify the values of decision variables that have been selected from the HM to its neighboring value or to make no change. PAR takes a value in the range [0, 1].

**Number of Improvisations (NI):** This parameter represents the number of times that the HS is repeated. It is treated as a stopping criterion.

### Step 2: Initialize the HM

HM is a two-dimensional matrix of solutions that has a size equal to the HMS. Each row in the HM represents solution, as shown in Figure 5.3. The solutions are randomly generated with their respective values of the fitness function  $f(t)$ . Solutions in the HM are arranged in a reverse order based on the values of the fitness function. Each solution is composed of a bit string that has a length equal to the total number of features that encode a candidate feature subset. A bit of 1 implies that the corresponding feature is selected, a bit of 0 implies that corresponding feature is excluded. The fitness function is utilized to calculate the fitness value for each solution in the HM. It is obtained by using the equations (4.2) & (4.3)

### Step 3: Improve the new harmony

Generating a new harmony (which is the core of the HS) is called ‘improvisation’. The HS improvises (generates) a new harmony  $f' = f'_1, f'_2, f'_3, \dots \dots f'_N$  using three rules:

1) Memory consideration 2) Pitch adjustment and 3) Random consideration.

#### 1) Memory consideration

In memory consideration, the value of the first feature  $f'_1$  for the new solution is selected from any existing values that are stored in the HM  $f_1^1, f_1^2, \dots \dots f_1^{HMS}$  with the HMCR probability. The next Feature  $f'_2$  is also chosen from  $f_2^1, f_2^2, \dots \dots f_2^{HMS}$  and the values for the other features are selected in the same manner if they satisfy the probability selection of HMCR between 0 and





1. These cumulative steps ensure that good harmonies are considered as the elements of a new harmony

## 2) Pitch adjustment

Every feature obtained by the HMCR is examined to determine whether it should be tuned (pitch-adjusted) with the probability of the PAR. The adjustment involves the mutation of the feature from 0 to 1 or from 1 to 0.

## Step 4: Update the HM

To update the HM with the improvised solution  $f' = f'_1, f'_2, \dots, f'_N$  the fitness function is calculated for the solution. If the fitness function for the new solution is better than the worst solution in the HM, then the worst solution is excluded and replaced by the new solution. Otherwise, the new solution is ignored.

## Step 5: Check the stopping criterion

The stopping criterion in this algorithm is the maximum number of iterations. Steps 3 and 4 are repeated until the stopping criterion is reached.

## Step 6: Cadenza

A cadenza is a musical passage that occurs in the final step of a movement in a musical work to return the most fantastic harmony played during the improvisation process. In the context of the HS, the cadenza can be referred to as the last step in the HS that occurs at the end of the search process for the best harmony. In this process, the HS returns the best harmony found and stored in the HM based on the fitness function  $f(X)$ . Parameters and its value for HS are shown in Table 5.2.



**Table 5.2 Parameter and its value for Harmony Search**

Parameter	Value
Harmony memory size(HMS)	50
HMCR	0.5
Pitch adjustment rate(PAR)	0.3
Number of improvisations	50

**Step 1: Initialize the problem and the algorithm parameters**

Assume that HMS = 50; HMCR = 0.5, PAR = 0.3, NI= 50.

**Step 2: Initialize the HM**

There are fifty solutions in the HM with the length size equals to the number of features (123 in this example, where each feature is represented by 1 or 0).

**Step 3: Improvise the New Harmony (f)**

Assume that the generated random numbers which are equal to the number of features. The generated random number is compared with the HMCR. If it is greater than HMCR, then a feature from the solution in the HM is selected. Otherwise a random consideration is utilized where a random number either 1 or 0 is chosen. A new solution is generated as follows:

**1) Memory consideration**

First, generate an empty solution with size equals to the number of features. Then, loop through the solution cell one by one and assign a value to each cell based on the HMCR.



The mainly used way of encoding the feature selection is a binary string. Here, the random values are generated for feature position. If the value of the variable is 1, then the feature is selected else feature is not selected. The harmony solution is represented as a binary string; the highlighted text in Figure 5.4 represents the Features which are selected based on the memory consideration.

F1	F2	F3	F4	F5	F6	-	-	F122	F123
0.25	0.75	0.97	0.54	0.18	0.24	-	-	0.92	0.35
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
0	1	1	1	0	0	-	-	1	0

**Figure 5.4 Harmony Solutions by considering HMCR**

Note that, the highlighted text in Figure 5.4 represents the features which are selected based on the memory consideration.

## 2) Pitch adjustment

For every feature that is based on the memory consideration will be tuned (pitch-adjusted) or mutated from 0 to 1 or from 1 to 0 based on the comparison between the random generated number  $[0,1]$  and PAR (selected features that are based on the random consideration) will not be tuned.

### Step 4: Update the HM

The new solution will replace the worst solution in the HM.

### Step 5: Check the stopping criterion

The above steps are repeated until the NI equals to 50.

## Step 6: Cadenza

HSA returns the best found solution.

### 5.2.2 Cuckoo Search

Cuckoo search (CS) is one of the recent Nature-Inspired metaheuristic algorithms developed by Xin-SheYang & Suash Deb (2009). CS is based on the brood parasitism of a few cuckoo species. This algorithm is enhanced by the Levy flights (Rodrigo et al. 2012) rather than by simple isotropic random walks. Cuckoos are attractive birds, because of the beautiful sounds they can make and their aggressive reproduction strategy. A few species such as the Ani and Guira cuckoos lay their eggs in communal nests and they may remove others' eggs to increase the hatching probability of their own eggs. The following three idealized rules for describing standard cuckoo search have been used

- At a time each cuckoo lays one egg and dumps at a random chosen nest
- The best nests with high quality eggs will be passed over to the next generations
- Number of available host nests is fixed and the cuckoo's egg is discovered by the host bird with a probability  $p_a$  belongs to  $(0,1)$ . The host bird can do either get rid of the egg or just abandon the nest and build a completely a new nest

This last assumption can be approximated by a fraction  $p_a$  of the  $n$  host nests that are replaced by new nests (with new random solutions) (Yang & Deb 2009). Cuckoo search is very simple and has extensive search space. It uses the levy flight for global search instead of standard random walk, which makes CS to explore the search space more efficiently.



For a maximization problem, the fitness of a solution can simply be proportional to the value of the objective function. Other forms of fitness can be defined in a similar way to the fitness function in genetic algorithms. The following simple representations have been used, each egg in a nest represents a solution, and cuckoo egg represents a new solution, the main aim is to use the new and better solutions (cuckoos) to replace a not-so-good solution in the nests.

In the proposed work, each nest has only a single egg. Based on the above three rules, the basic steps of the Cuckoo Search (CS) can be summarized in the following pseudo code (Figure 5.5).

```

Begin
    Objective function  $f(x)$ ,  $x = (x_1, x_2 \dots \dots x_d)^T$ 
    Generate initial population of n host nests  $x_i (i=1,2,\dots,n)$ 
    While( $t < \text{MaxGeneration}$ ) or (stop criterion)
        Get a cuckoo randomly by levy flights
        Evaluate its quality/fitness  $F_i$ 
        Choose a nest among n(say,j)randomly
        if( $F_i > F_j$ )
            Replace j by new solution;
        end if
        A fraction ( $P_a$ ) of worst nests are abandoned and new ones are
        built;
        Keep the best solutions (or nests with quality solutions);
        Rank the solutions and find the current best
    end while
    Postprocess the results and visualization
End

```

**Figure 5.5 Pseudo code for Cuckoo Search via levy flights**

When generating new solutions  $x_i(t + 1)$  for, say, a cuckoo Levy flight is performed and given in equation (5.1)

$$x_i(t + 1) = x_i(t) + \alpha \oplus \text{Levy}(\lambda) \quad (5.1)$$

where  $\alpha > 0$  is the step size which can be related to the scales of the problem of interests. In most cases, it can used  $\alpha = 1$ . The above equation is essentially the stochastic equation for random walk. In general, a random walk is a Markov chain whose next status/location only depends on the current location (the first term in the above equation) and the transition probability (the second term). The product  $\oplus$  means entry-wise multiplication, here the random walk via Levy flight is more efficient in exploring the search space as its step length is much longer in the long run.

The Levy flight essentially provides a random walk while the random step length is drawn from a Levy distribution  $\text{Levy} \sim u = t^{-\lambda}$ , ( $1 < \lambda \leq 3$ ) which has an infinite variance with an infinite mean. Some of the new solutions can be generated by Levy walk around the best solution obtained so far, this will speed up the local search. However, a substantial fraction of the new solutions can be generated by far field randomization and whose location can be far enough from the current best solution, this will make sure the system will not be trapped in a local optimum. From a quick look, it seems that there is some similarity between CS and hill-climbing in combination with some large scale randomization. But there are some significant differences. Firstly, CS is a population-based algorithm, in a way similar to GA and PSO, but it uses some sort of elitism and/or selection similar to that used in harmony search. Secondly, the randomization is more efficient as the step length is heavy tailed, and any large step is possible. Thirdly, the number of parameters to be tuned is less than GA and PSO, and thus it is potentially



more generic to adapt to a wider class of optimization problems. In addition, each nest can represent a set of solutions.

**Parameter for Cuckoo Search:** In CS, the parameters  $p_a$ ,  $\alpha$ ,  $\lambda$  and are set as 0.3, 1, 1.5 respectively. It is observed that the maximum fitness value is at population size of 50 with the maximum number of generations as 200. Table 5.3 shows the parameters and its values for CS for feature selection in breast cancer classification.

**Table 5.3 Parameters and its value for CS**

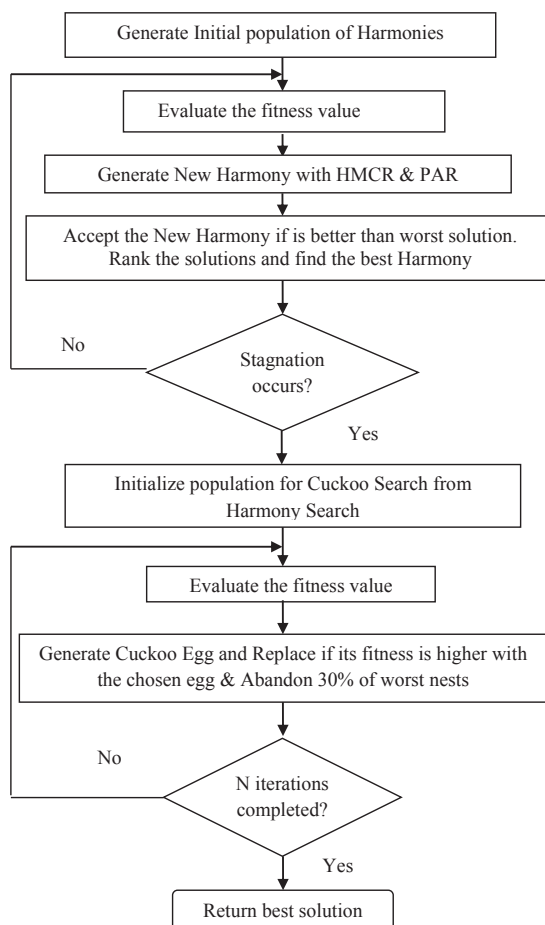
Parameter	Value
Egg size	50
Population size	50
Number of generation	200
$P_a$	0.3
$\alpha$	1
$\lambda$	1.5

### 5.2.3 Hybrid Harmony Search (HHS)

It is an integration of harmony search and cuckoo search. HHS introduced to improve the performance of standard HS algorithm. It is achieved by initializing the harmonies as eggs for cuckoo search from harmony search when no change in output i.e. stagnation occurs. Hybrid approach monitors the output of Harmony Search and if there is no change in output then the algorithm assigns the best harmonies as eggs for cuckoo search algorithm. The flow diagram is shown in Figure 5.6.

The performance of any classification system is calculated by its classification accuracy. Classification accuracy plays a major role in the

process of mammogram image classification using significant features. In the proposed work, Minimum Distance Classifier, k-NN classifier and SVM classifier are as classifier, and the classification accuracy is used as the fitness function for HHS and is estimated using equation (4.1).



**Figure 5.6 Flow diagram for Hybrid Harmony Search**



A distance measure is used to find the closeness between test set and training set in Minimum Distance Classifier or k-NN or SVM Classifier. The various distance measures such as Euclidean distance, City block distance, Minkowski distance, Chi-Square Distance are computed using the Equations (1.4) to (1.7). The accuracy of a classifier is assessed based on 5-fold cross validation techniques.

**Table 5.4 Parameters and its value for HHS**

Parameter	Value
Harmony Memory(HM)	50
Harmony Memory Considering Rate (HMCR)	0.5
Pitch Adjusting Rate (PAR)	0.3
Number of nests	50
$p_a$	0.3
$\alpha$	1
$\lambda$	1.5

#### 5.2.4 Results of HHS

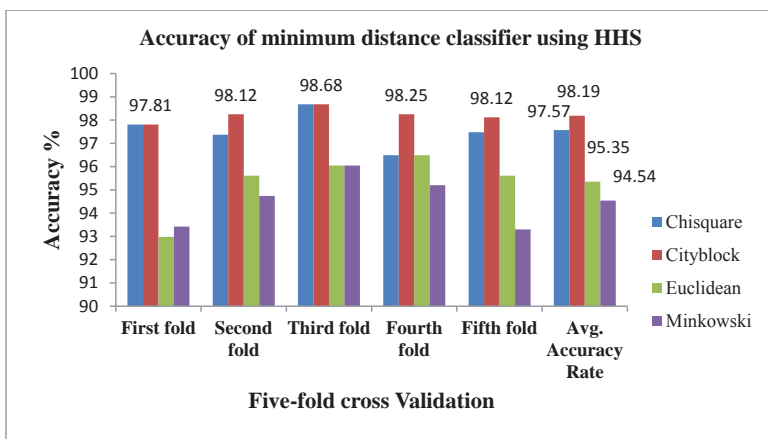
Totally 500 DDSM images (in which 288 are benign and 212 are malignant) have been considered for developing breast cancer classification system. The proposed system has been implemented and the features were also extracted for further analysis using MATLAB. In this work, initially 123 features were extracted.

The proposed feature selection method using Minimum Distance Classifier selects only 36 features and produces 98.19% average classification accuracy with city block distance measure. Other distance measures such as Chi-square, Euclidean and Minkowski using Minimum Distance Classifier

produces 97.57%, 95.35% and 94.54% average classification accuracy respectively. The proposed feature selection method using k-NN classifier selects 34 features and produces 98.34% average classification accuracy with chi-square distance measure. Other distance measures such as city block, Euclidean and Minkowski using k-NN classifier produces 97.63%, 97.45% and 96.96% average classification accuracy respectively. The HHS using SVM classifier selects 39 features and produces 97.18% average classification accuracy. Results of HHS with Minimum Distance Classifier and k-NN classifier based on five-fold cross validation using various distances measures are shown in Figures 5.7 & 5.8 and given in Tables 5.5 & 5.6. The results of HHS with SVM classifier using five-fold cross validation are shown in Figure 5.9. Accuracy of proposed system and number of feature selected are also compared with Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Cuckoo Search (CS) and Harmony Search (HS) and it is shown in Table 5.7.

**Table 5.5 Results of HHS with Minimum Distance Classifier based on five-fold validation**

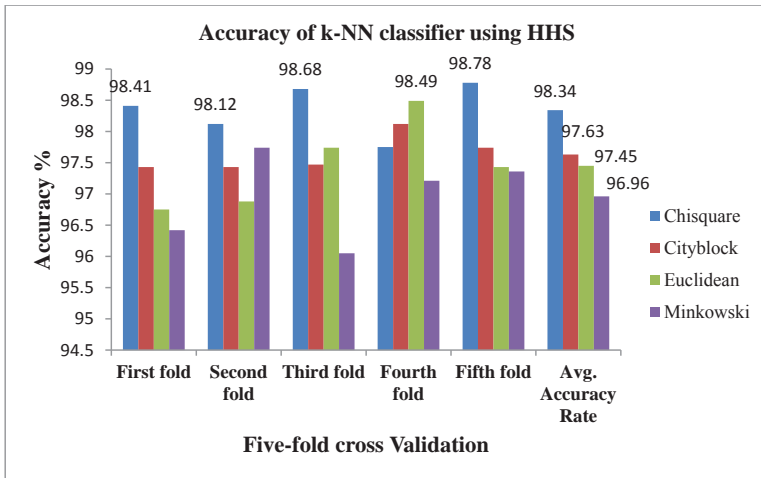
<b>Minimum Distance Classifier Accuracy in %</b>				
	<b>Chi-square</b>	<b>City block</b>	<b>Euclidean</b>	<b>Minkowski</b>
<b>First fold</b>	97.81	97.81	92.98	93.42
<b>Second fold</b>	97.37	98.12	95.61	94.74
<b>Third fold</b>	98.68	98.68	96.05	96.05
<b>Fourth fold</b>	96.49	98.25	96.49	95.2
<b>Fifth fold</b>	97.48	98.12	95.61	93.3
<b>Avg. Accuracy</b>	97.57	<b>98.19</b>	95.35	94.54



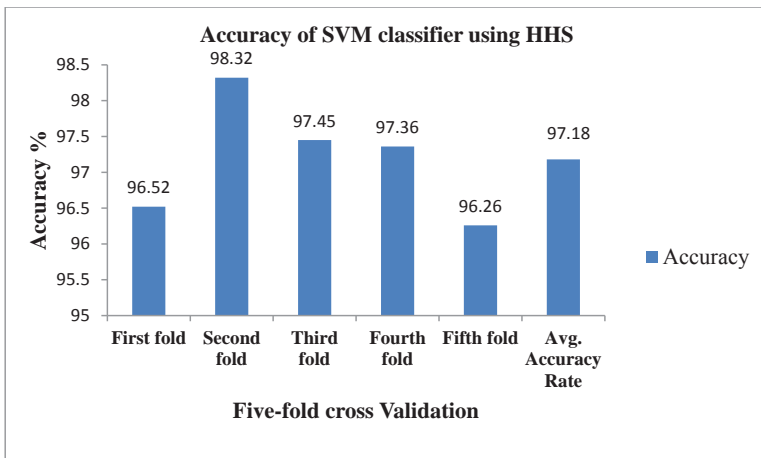
**Figure 5.7 Results of HHS with Minimum Distance Classifier based on five-fold cross validation**

**Table 5.6 Results of HHS with k-NN Classifier based on five-fold cross validation**

k-NN Classifier Accuracy in %				
	Chisquare	Cityblock	Euclidean	Minkowski
<b>First fold</b>	98.41	97.43	96.75	96.42
<b>Second fold</b>	98.12	97.43	96.88	97.74
<b>Third fold</b>	98.68	97.47	97.74	96.05
<b>Fourth fold</b>	97.75	98.12	98.49	97.21
<b>Fifth fold</b>	98.78	97.74	97.43	97.36
<b>Avg. Accuracy</b>	<b>98.34</b>	97.63	97.45	96.96



**Figure 5.8 Results of HHS with k-NN classifier based on five-fold cross validation**



**Figure 5.9 Results of HHS with SVM classifier using five-fold cross validation**

**Table 5.7 Performance analysis of HHS**

Algorithm	Minimum Distance Classifier		k-NN		SVM	
	No of Features Selected from 123	Average Accuracy in %	No of Features Selected from 123	Average Accuracy in %	No of Features Selected from 123	Average Accuracy in %
GA	69	94.23	62	95.21	72	93.18
PSO	64	96.25	58	96.84	67	95.47
CS	53	97.45	49	97.87	55	96.41
HS	45	97.84	36	98.13	42	98.04
HHS	36	98.19	<b>34</b>	<b>98.34</b>	39	97.18

### 5.3 FEATURE SELECTION USING ENHANCED CUCKOO SEARCH

The proposed system has been developed to extract the optimal features for breast cancer classification using Enhanced Cuckoo Search (ECS). In this work Minimum Distance Classifier, k-NN classifier and SVM classifier are used for classification. Overall accuracy of the classifier is used as a fitness value for ECS. The performance of the ECS is compared with the Particle Swarm Optimization algorithm, Cuckoo Search and Harmony Search.

#### 5.3.1 Enhanced Cuckoo Search

Cuckoo search is modified to Enhanced Cuckoo Search with different types of host nest with multiple eggs (Tuba et al. 2011). In general cuckoo selects three types of nests for laying their eggs.

- The common cuckoo selects a group of host nests with egg characteristics similar to their own. (Moksnes & Roskaft 1995)
- Others cuckoo selects a group of host nest with egg characteristics dissimilar to their own

- Some other species of cuckoo lay cryptic egg, which are dark in color when the eggs of their host birds are light. This trick is used to hide the eggs from the host and evolve in cuckoos that parasite host with dark, domed nests

Algorithm steps are listed in Figure 5.10 for Enhanced Cuckoo Search based on egg laying behavior of cuckoo and multiple eggs in the nest.

```

Generate an initial population of n host nests with m eggs;
while (t < MaxGeneration) or (stop criterion)
  for each nest
    Get a cuckoo type randomly (say, i)
    Check the type of the cuckoo
    if cuckoo_type = Common_cuckoo
      Create two eggs using Crossover with the two best eggs in the nest and
      choose the best one among them
    else if cuckoo_type = European_cuckoo
      Create two eggs using Crossover with uniform mutation operator
      with any two eggs in the nest and choose the best one among them.
    else
      Create an egg with random solution (cryptic egg)
    end if
    Evaluate its fitness  $F_i$ 
    Choose an egg with the worst solution in the nest (say, j)
    if ( $F_i > F_j$ )
      Replace j by the new solution i
    end if
    Rank the eggs based on the solution
    Find the best solution (among m eggs) in the nest
    Abandon a fraction ( $p_a$ ) of the eggs in the nest which have worse solutions
    and built new ones using levy flight
    Keep the best solutions
  end for
  Rank the eggs in all nests using fitness value and find the current best
end while

```

**Figure 5.10 Algorithm for Enhanced Cuckoo Search**



**Egg Representation:** In the process of feature selection using ECS, each egg contains the solution it represents. The eggs are represented by 123 features. The mainly used way of encoding the feature selection is a binary string. Here, the random values are generated for feature position. If the value of the variable is 1, then the feature is selected else feature is not selected. The egg is represented as a binary string which is shown in Figure 5.11.

F1	F2	F3	F4	F5	F6	-	-	F122	F123
0.25	0.67	0.97	0.34	0.18	0.24	0.01	0.83	0.92	0.35
↓	↓	↓	↓	↓	↓	↓	↓	↓	
0	1	1	0	0	0	0	1	1	0

**Figure 5.11** Egg representation for feature selection using ECS

**Initial Population:** In this work, each egg represents a possible set of features that are selected and used to classify the samples correctly. Here the initial population size is considered as 50.

**Finding New Solutions and Levy Flight:** ECS based feature selection method makes use of levy flight for finding the new solutions. Some of the new solutions can be generated by a levy walk around the best solution obtained so far, this can speed up the local search. By using Levy flight, the new solution  $x_i(t+1)$  is produced for cuckoo  $i$ , and is calculated using  $x_i(t+1) = x_i(t) + \alpha \oplus \text{Levy}(\lambda)$ , where  $\alpha$  is the step size. The step length follows the Levy distribution  $\text{Levy}(s, \lambda) \sim s^{-\lambda}$ ,  $1 < \lambda \leq 3$ .

**Fitness Function:** The fitness function plays a vital role in the selection process. Here, the classification accuracy of Minimum Distance Classifier or k-NN or SVM classifier is used as the fitness function of ECS. The fitness function  $\text{fitness}(f)$  of ECS is defined as in Equation (4.1).



The classification accuracy of Minimum Distance Classifier or k-NN or SVM is calculated using equation (4.2).

**Parameter Pa:** In enhanced cuckoo search the  $P_a$  value is changed dynamically by using the formula given in Equation (5.2)

$$p_a = p_{a_{\max}} - \frac{p_{a_{\max}} - p_{a_{\min}}}{iter_{\max}} \times iter \quad (5.2)$$

$p_{a_{\max}}$ ,  $p_{a_{\min}}$  is set as 0.5 and 0.3 respectively.

### Crossover and Mutation

- Cuckoo type is common cuckoo then crossover is used to create two eggs in the nest and choose the best one among them
- A cuckoo type is European cuckoo then creates two eggs using crossover with uniform mutation operator and choose the best one among them
- Otherwise eggs (cryptic) are created with random solution

**K-Fold Cross validation** K-fold cross validation procedure has been used to evaluate the efficiency of the proposed system. The average of K results from the folds can be calculated to produce an average accuracy of Minimum Distance Classifier, k-NN and SVM Classifier.

**Similarity Measure:** To find the closeness between training data set and test data set normally some distance measures are used. They are Euclidean distance, City block distance, Minkowski distance and chi-Square Distance, are given in equation (1.4) to equation (1.7)





Table 5.8 shows the parameters and its values for ECS for feature selection in breast cancer classification.

**Table 5.8 Parameters and its value for ECS**

Parameter	Value
Number of nests (Population)	50
Number of eggs	10
No. of generation	200
$p_{a_{\min}}$	0.3
$p_{a_{\max}}$	0.5
$\alpha$	1
$\lambda$	1.5
Mutation Probability	0.01
Crossover rate	0.65

### 5.3.2 Results of ECS

Totally 500 DDSM images (in which 288 are benign and 212 are malignant) have been considered for developing breast cancer classification system. The proposed system has been implemented and the features were also extracted for further analysis using MATLAB.

The proposed feature selection method (ECS) based on Minimum Distance Classifier selects totally 34 features and produces 98.75% average classifier accuracy by using chi-square distance. Other distance measures such as City block, Euclidean and Minkowski using Minimum Distance Classifier produces 97.4%, 97.54% and 96.86% average classification accuracy

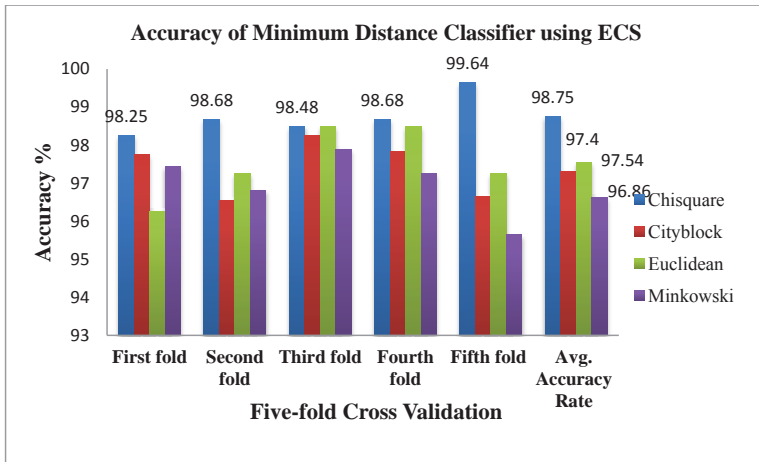


respectively. ECS with k-NN classifier selects totally 29 features and produces 99.13% average classifier accuracy using Euclidean distance.

Other distance measures such as Chi-Square, City block and Minkowski using k-NN classifier produces 97.17%, 98.07% and 97.04% average classification accuracy respectively. ECS with SVM classifier selects totally 29 features and produces 99.22% average classifier accuracy. Results of ECS with Minimum Distance Classifier and k-NN classifier based on five-fold cross validation using various distances measures are shown in Figures 5.12 & 5.13 and given in Table 5.9 & Table 5.10. The results of ECS with SVM classifier using five-fold cross validation are given in Figure 5.14. Optimal features selected from PSO, CS, HS and ECS are shown in Table 5.11.

**Table 5.9 Results of ECS with Minimum Distance Classifier based on five-fold cross validation**

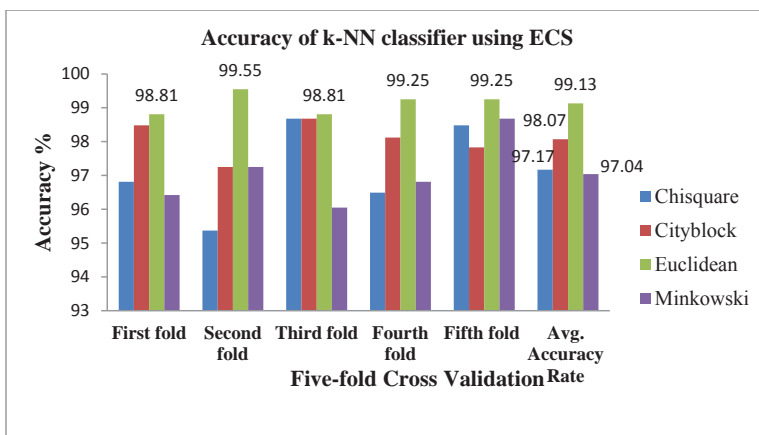
<b>Minimum Distance Classifier Accuracy in %</b>				
	<b>Chisquare</b>	<b>Cityblock</b>	<b>Euclidean</b>	<b>Minkowski</b>
<b>First fold</b>	98.25	97.75	96.25	97.45
<b>Second fold</b>	98.68	96.54	97.25	96.81
<b>Third fold</b>	98.48	98.25	98.48	97.89
<b>Fourth fold</b>	98.68	97.82	98.48	96.54
<b>Fifth fold</b>	99.64	96.64	97.25	95.64
<b>Avg. Accuracy</b>	<b>98.75</b>	97.4	97.54	96.86



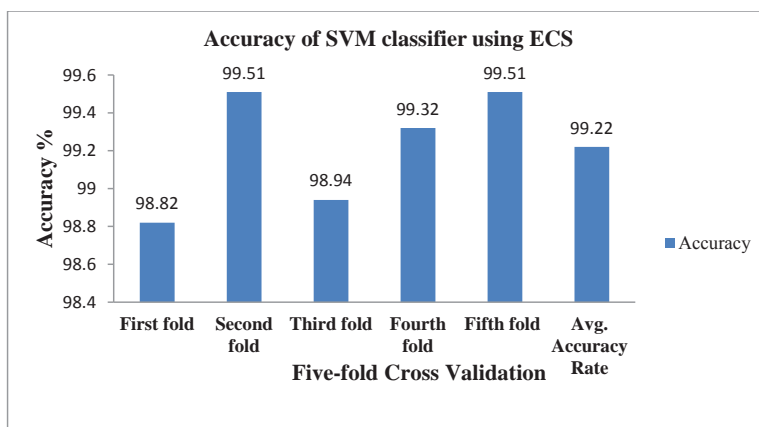
**Figure 5.12 Results of ECS with Minimum Distance Classifier based on five-fold cross validation**

**Table 5.10 Results of ECS with k-NN classifier based on five-fold cross validation**

k-NN classifier Accuracy in %				
	Chisquare	Cityblock	Euclidean	Minkowski
First fold	96.81	98.48	98.81	96.42
Second fold	95.37	97.25	99.55	97.25
Third fold	98.68	98.68	98.81	96.05
Fourth fold	96.49	98.12	99.25	96.81
Fifth fold	98.48	97.83	99.25	98.68
Avg. Accuracy	97.17	98.07	<b>99.13</b>	97.04



**Figure 5.13 Results of ECS with k-NN classifier based on five-fold cross validation**



**Figure 5.14 Results of ECS with SVM classifier based on five-fold cross validation**

**Table 5.11 Performance analysis of ECS**

Algorithm	Minimum Distance Classifier		k-NN		SVM	
	No of Features Selected from 123 features	Average accuracy in %	No of Features Selected from 123 features	Average accuracy in %	No of Features Selected from 123 features	Average accuracy in %
PSO	64	96.25	58	96.84	67	95.47
CS	53	97.45	49	97.87	55	96.41
HS	45	97.84	36	98.13	42	98.04
ECS	34	98.75	29	99.13	<b>29</b>	<b>99.22</b>

## 5.4 OPTIMAL FEATURE SELECTION USING IMPROVED LION OPTIMIZATION ALGORITHM

Improved Lion Optimization Algorithm (ILOA) is used to select optimal features for breast cancer classification. In this work Minimum Distance Classifier, K-Nearest Neighbor Classifier and Support Vector machine have been used for classification. Accuracy of the classifier is used as a fitness value for ILOA. The performance of the ILOA is compared with the Cuckoo Search, Harmony Search and Lion Optimization Algorithm (LOA).

### 5.4.1 Lion Optimization Algorithm

It is based on life style of lions, its behavior and social organization (Maziar Yazdani & Fariborz Jolai 2016). Two types of social organization of lion are residents and nomads. Residents lives in a groups called pride (Schaller 1972), which includes five females and one or more adult males. Nomads live singularly or pairs. Residents may become nomads and nomads become resident, in such a way that lions may switch their life styles.



Some Female Lions in pride work together and go for group hunting bring success, male lions and some female lion stay in their territory and waiting for other lions to return. A female lion may mate with multiple lions (Hrdy 2006) and young males are excluded from their birth pride when they become mature.

Male and female nomad lion moves randomly to seek for better place. If male nomad lion invade the male pride lion then male pride lion is driven out of the pride by male nomad lion. Some pride female lions migrate from one pride to another and become nomad, some nomad female lions migrates to prides.

#### 5.4.2 Improved Lion Optimization Algorithm

This algorithm is introduced to improve the performance LOA. The following steps are modified in lion optimization algorithm.

- Some female prides in each pride go for hunting and provide prey for their pride. The female prides are selected based on Roulette Wheel Selection
- The stopping condition is based on maximum number of iteration. But, If best fitness value stagnates for designated number of iterations then merge the entire population of lions can be merged and split the prides and nomads based on fitness value
- Percentage of females from pride is selected based on Roulette Wheel Selection. Selected females in pride mate with one or several resident male

**Initialization:** The first step is to arbitrarily generate the M population over the solution space. In this algorithm, every single solution is

called Lion. Here, the population size is 50. 20% of solutions randomly chosen as nomad lions, the rest of the populations divided into 2 prides and lions in each pride are 20. In each pride 70% are considered as females and rest as males. In nomad lions 30% are considered as females and rest as males.

In breast cancer classification problem, a Lion represents a solution that is set of features that are chosen to classify the samples. Let  $n$  ( $=123$ ) denote the number of features associated with  $k$  ( $=50$ ) objects (Lions). If the feature  $j$  is selected for an object  $i$ , then 1 is assigned for feature  $j$ . Otherwise 0 is assigned ( $j \leq n$ ,  $i \leq k$ ). Table 5.12 shows the Lion representation for feature selection using ILOA.

Lions	Features											
	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>	F <sub>6</sub>	F <sub>7</sub>	F <sub>8</sub>	...	F <sub>n-2</sub>	F <sub>n-1</sub>	F <sub>n</sub>
L <sub>1</sub>	1	0	0	0	1	1	0	0	...	0	1	0
L <sub>2</sub>	0	1	0	1	1	0	0	0	...	1	0	1
L <sub>3</sub>	0	1	0	1	1	0	0	1	...	1	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
L <sub>k</sub>	1	0	0	1	0	0	1	1	...	0	1	1

**Table 5.12 Lion representation for feature selection using ILOA**

**Hunting :** Some female prides in each pride go for hunting and provide prey for their pride. The female lions are selected based on Roulette wheel selection. Sometimes prey will escape from hunter and new position of prey is obtained using equation (5.3).

$$\text{prey}' = \text{prey} + \text{ran}(0,1) \times \text{PerI} \times (\text{prey} - \text{Hunter}) \quad (5.3)$$

where prey is current position of prey, hunter is new position hunter who attaches to prey and PerI is the percentage of improvement in fitness of hunter.

The following equations (5.4) and (5.5) are used to encircle prey by mentioned hunter groups.

$$\text{Hunter}' = \begin{cases} \text{rand}((2 \times \text{prey} - \text{Hunter}), \text{prey}), (2 \times \text{prey} - \text{Hunter}) < \text{prey} \\ \text{rand}(\text{prey}, (2 \times \text{prey} - \text{Hunter})), (2 \times \text{prey} - \text{Hunter}) > \text{prey} \end{cases} \quad (5.4)$$

$$\text{Hunter}' = \begin{cases} \text{rand}(\text{Hunter}, \text{prey}), \text{Hunter} < \text{prey} \\ \text{rand}(\text{prey}, \text{Hunter}), \text{Hunter} > \text{prey} \end{cases} \quad (5.5)$$

Hunter' is new position of hunter. In above equations, rand() generates a random number between upper and lower bounds.

**Moving Towards Safe Place :** Some females in each pride go for hunting. Remaining females go toward one of the safe areas of territory. Best solutions obtained so far are saved and used as valuable information to improve solutions. Therefore, the new position for a female lion may be obtained using the following Equation (5.6)

$$\text{NewPosition FemaleLion}' = \text{CurrentPosition FemaleLion} + 2D \times \text{rand}(0,1)\{R_1\} + U(-1,1) \times (\tan \theta) \times D \times \{R_2\} \quad (5.6)$$

where  $\{R_1\}, \{R_2\} = 0, \|R_2\| = 1$ , D is distance between current position and new position of female lion,  $R_1$  is start point and direction towards the selected position,  $R_2$  is perpendicular to  $R_1$ .

**Roaming:** Roaming is a strong local search and assists the algorithm to search around of a solution to improve it. Each male pride lion





roams in that pride's territory. To imitate this behavior of resident male pride, 20% of pride territory are selected randomly and are visited by that lion. While roaming, if resident male visits a new position which is better than its current best position then update the best visited solution. Both male and female nomad lion moving randomly in search space and are given in equation (5.7).

$$\text{Lion}_{ij} = \begin{cases} \text{Lion}_{ij} & \text{if } \text{rand}_j > \text{pr}_i \\ \text{rand}_j & \text{Otherwise} \end{cases} \quad (5.7)$$

where  $\text{Lion}_{ij}$  is current of  $i^{\text{th}}$  nomad lion in  $j^{\text{th}}$  dimension,  $\text{rand}_j$  is a uniform random number within (0,1),  $\text{pr}_i$  is probability calculated for each nomad lion independently and given in equation (5.8).

$$\text{pr}_i = 0.1 + \min \left( 0.5, \frac{\text{Nomad}_i - \text{Best}_{\text{nomad}}}{\text{Best}_{\text{nomad}}} \right) \quad (5.8)$$

$i = 1, 2, \dots$  no. of nomad lions, where  $\text{Nomad}_i$  and  $\text{Best}_{\text{nomad}}$  are cost of current position of  $i^{\text{th}}$  lion in nomad and the best cost of the nomad lion

**Mating** : In every pride, 30% of female lions mate with one or several resident males to produce offspring and these males are selected randomly. In nomad lions, a nomad female only mates with one of the males which are selected randomly. 30% of female pride lions are selected based on Roulette Wheel Selection. The new cubs are produced after selecting the female lion and male(s) for mating using the following equations (5.9) and (5.10).

$$\text{offspring}_1 = \beta \times \text{Female Lion}_j + \sum_{i=1}^{\text{noR}} \frac{(1-\beta)}{S_i} \times \text{Male Lion}_j^i \times S_i \quad (5.9)$$

$$\text{offspring}_2 = (1 - \beta) \times \text{Female Lion}_j + \sum_{i=1}^{\text{noR}} \frac{\beta}{S_i} \times \text{Male Lion}_j^i \times S_i \quad (5.10)$$

where  $j$  is dimension,  $S_i$  equals 1 if male  $i$  is selected for mating, Otherwise it equals 0. 30% is the number of resident males in a pride,  $\beta$  is a randomly

generated number. A mutation is applied on each gene of one of the produced offspring with probability of 1%, noR is the number of resident males in a pride.

**Defense:** In a pride, when male lions mature, they fight with other males in their pride. Beaten males driven out their pride and become a nomad. If a nomad male lion is fight with a pride by fighting its males, the beaten male lion is driven out of the pride and becoming a nomad.

**Migration:** Size of migrated female in each pride is equal to 40%. Selected females migrate from prides and become nomad. Next, the new nomad female lion and old nomad female lion are sorted according to the accuracy (fitness), best among the lions are selected to fill the prides empty place of migrated female.

**Lions' Population Equilibrium :** Number of live lions will be controlled at the end of each iteration with respect to the maximum permitted number of each gender in nomads. Nomad lions with the least fitness value will be removed.

**Convergence :** Commonly considered stopping condition is based maximum number of iteration. Here the maximum number of iteration chosen is 200. If best fitness value stagnates for designated number of iterations then merge the entire population of lions and split the prides and nomads based on fitness value. Always the more informative features are carried to the next generation based on accuracy of the classifier.

**Fitness Function:** In this work, the features are selected based on their fitness values. The fitness function plays a major role in the feature selection process. Here, the classification accuracy of Minimum Distance Classifier or k-NN classifier or SVM classifier is used as the fitness function of ILOA. The fitness function fitness(f) of ILOA is calculated using equation (4.1). The classification accuracy is estimated using Equation (4.2).



**K-Fold Cross validation:** K-fold cross validation procedure has been used to evaluate the efficiency of the proposed system. In this procedure, the 123 features are randomly divided into K disjoint parts of approximately the equal size. The average of K results from the folds can be calculated to produce an average accuracy of minimum Distance Classifier, k-Nearest Neighbor Classifier and SVM Classifier.

**Similarity Measure:** To find the closeness between training data set and test data set some distance measures are used. Distance measures are Euclidean distance, City block distance, Minkowski distance and Chi-square distance are calculated using equation (1.4) to equation (1.7).

Table 5.13 shows the parameters and its values for ILOA for feature selection in breast cancer classification and Figure 5.15 explains the algorithm for ILOA.

**Table 5.13 Parameters and its value for ILOA**

Parameter	Value
Population size	50
Nomad Lions	20%
Number of Prides	2
Lions in each Pride	20
Female Prides	70%
Male Prides	30%
Female Nomads	30%
Male Nomads	70%
Number of Iterations	200
Type of selection for Moving	Tournament Selection
Type of selection for Mating	Roulette Wheel Selection
Mating probability	0.3
Roaming percent	0.2
Immigrate rate	0.4
Mutation probability	0.01

```

Begin
Define the objective function  $f(x)$ ,  $x = \{x_1, x_2, x_3, x_4 \dots x_n\}$ 
Set the following parameters
Population of Lions, Number of Iterations
Number of Prides, Female Prides, Male Prides, Female Nomads, Male Nomads
Immigration Rate, Mutation Probability, Numbers of lions for mating, moving
Generate  $N_{pop}$  of lions randomly
Initialize prides and nomad lions in the population
Randomly select %N of initial population as nomad lions. Partition remained lions into P
prides randomly and form each pride's territory.
In each pride %S of entire population are known as females and the rest as males.
In each pride %(1-S) of entire population are known as females and the rest as males.
While termination criterion is not satisfied
For each pride do
    Select some female lions using Roulette Wheel Selection and go hunting
    Each of remained female lion in pride go towards one of the best selected position
    from territory
    For each resident male; %R of territory are randomly selected and checked
    Select %Ma of females from pride using Roulette Wheel Selection
    Selected females in pride mate with one or several resident male.
    Do mutation with Mutation Probability Mu. New cubs become mature
    Merge new off springs with prides
    Weakest male drive out from pride and become nomad
End For
Select some female lions using Roulette Wheel Selection and go hunting
For Nomad do
    Nomad lion (Both male and female) moving randomly in search space
    Select %Ma of females from nomad using Roulette Wheel Selection
    Selected nomad females mate with one of the best nomad male
    Do mutation with Mutation Probability Mu. New cubs become mature
    Merge new off springs with prides
    Selected prides randomly attacked by nomad male
End For
For each pride do
    Some female from pride with Immigration Rate I become nomad lions
End For

```



```

Do
    Based on their fitness vale each gender of the nomad lions is sorted. After that,
    the best females among them are selected and distributed to prides filling empty
    places of migrated females.
    With respect to maximum permitted number of each gender, nomad lions with
    least fitness value will be removed.
End
Find the overall best fitness value from the entire population of lions
If best fitness value stagnates for designated number of iterations
    Merge the entire population of lions and Split the prides and nomads based on
    fitness value
End If
End While
End

```

**Figure 5.15 Algorithm for ILOA**

### 5.4.3 Results of ILOA

Totally 500 DDSM images (in which 288 are benign and 212 are malignant) have been considered for developing breast cancer classification system. The proposed system has been implemented using MATLAB and the features were also extracted for further analysis. In this work, initially 123 features were extracted.

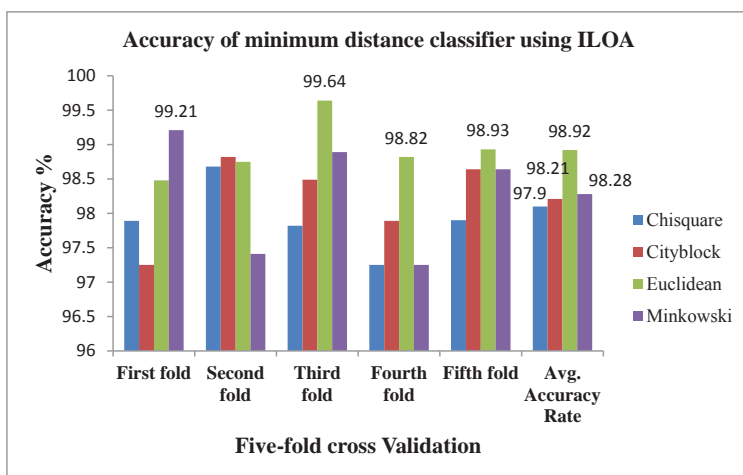
The proposed ILOA with Minimum Distance Classifier selects 30 features and produced 98.92% average classifier accuracy using Euclidean distance measure. Other distance measures such as Chi-Square, City block and Minkowski using Minimum distance classifier produces 97.9%, 98.21% and 98.28% average classification accuracy respectively. ILOA with k-NN classifier selects 28 features and produces 99.31% average classifier accuracy using Euclidean distance measure. Other distance measures such as Chi-Square, City block and Minkowski using k-NN classifier produces 97.56%, 98.65% and 98.69% average classification accuracy respectively. ILOA with SVM

classifier selects 32 features and produces 98.62% average classifier. Results of ILOA with Minimum Distance Classifier and k-NN classifier based on five-fold cross validation using various distances measures are shown in Figure 5.16 & 5.17 and given in Table 5.14 & Table 5.15. The results of ILOA with SVM classifier using five-fold cross validation are given in Figure 5.18. Average accuracy and optimal features selected from Genetic Algorithm (GA), Cuckoo Search (CS), Harmony Search (HS) and Lion Optimization Algorithm (LOA) and Improved Lion Optimization Algorithm (ILOA) are given in Table 5.16. Performance analysis based average accuracy is shown in Figure 5.19.



**Table 5.14 Results of ILOA with Minimum Distance Classifier based on five-fold cross validation**

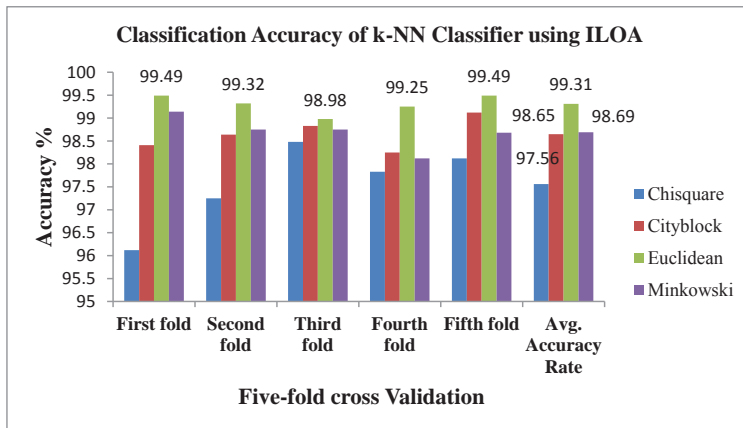
Minimum Distance Classifier Accuracy in %				
	Chi-square	City block	Euclidean	Minkowski
<b>First fold</b>	97.89	97.25	98.48	99.21
<b>Second fold</b>	98.68	98.82	98.75	97.41
<b>Third fold</b>	97.82	98.49	99.64	98.89
<b>Fourth fold</b>	97.25	97.89	98.82	97.25
<b>Fifth fold</b>	97.90	98.64	98.93	98.64
<b>Avg. Accuracy</b>	97.90	98.21	<b>98.92</b>	98.28



**Figure 5.16 Results of ILOA with Minimum Distance Classifier based on five-fold cross validation**

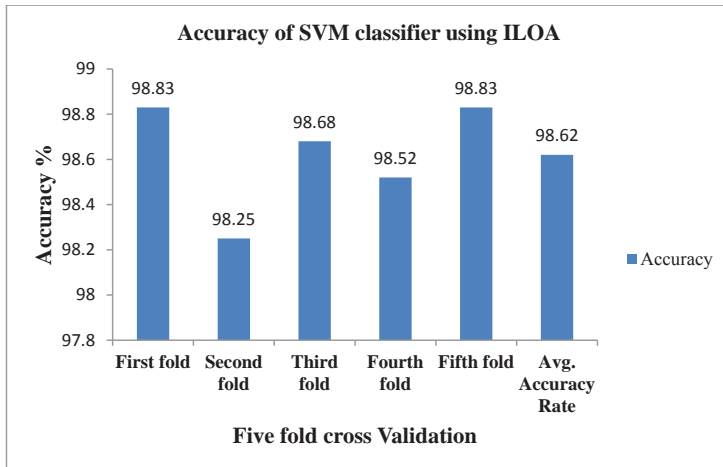
**Table 5.15 Results of ILOA with k-NN classifier based on five-fold cross validation**

k-NN Classifier Accuracy in %				
	Chisquare	Cityblock	Euclidean	Minkowski
<b>First fold</b>	96.12	98.41	99.49	99.14
<b>Second fold</b>	97.25	98.64	99.32	98.75
<b>Third fold</b>	98.48	98.83	98.98	98.75
<b>Fourth fold</b>	97.83	98.25	99.25	98.12
<b>Fifth fold</b>	98.12	99.12	99.49	98.68
<b>Avg. Accuracy</b>	97.56	98.65	<b>99.31</b>	98.69

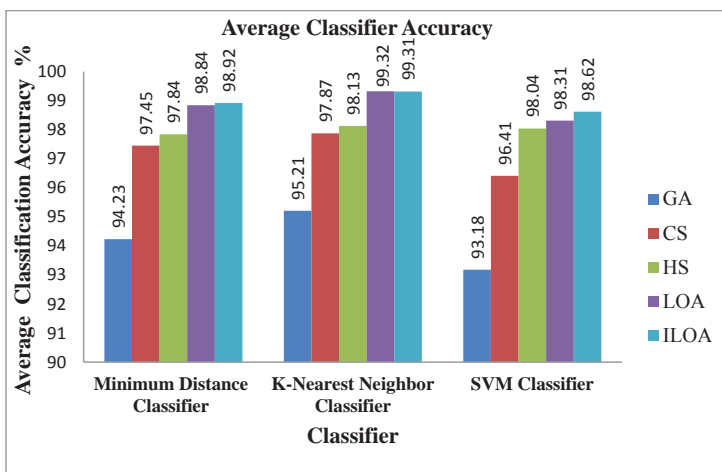


**Figure 5.17 Results of ILOA with k-NN classifier based on five-fold cross validation**





**Figure 5.18 Results of ILOA with SVM classifier based on five-fold cross validation**



**Figure 5.19 Performance analysis based on Average accuracy**

**Table 5.16 Performance analysis of ILOA**

Algorithm	Minimum Distance Classifier		k-NN		SVM	
	No of Features Selected from 123 features	Average accuracy in %	No of Features Selected from 123 features	Average accuracy in %	No of Features Selected from 123 features	Average accuracy in %
GA	69	94.23	62	95.21	72	93.18
CS	53	97.45	49	97.87	55	96.41
HS	45	97.84	36	98.13	42	98.04
LOA	32	98.84	31	99.32	37	98.31
ILOA	30	98.92	<b>28</b>	<b>99.31</b>	32	98.62