Performance Evaluation of PSO Based Classifier for Classification of Multidimensional Data with Variation of PSO Parameters in Knowledge Discovery Database

Sarita Mahapatra[1], Alok Kumar Jagadev[1] and Bighnaraj Naik[1]

[1]*Dept of Information Technology,*
Institute of Technical Education and Research,
Siksha 'O' Anusandhan University, Bhubanewar, Orissa, India

saritamahapatra4@gmail.com, a_jagadev@yahoo.co.in,
bighnaraj_naik@yahoo.co.in

Abstract

In this paper we have proposed a modified PSO based classification model for multidimensional real dataset and we have studied the results of experiment by implementing particle swarm optimization in classification. Evaluation of the performance of PSO based classifier has been made by considering several variation of parameters of the standard particle swarm optimizer. Here we have used a multidimensional real cancer dataset for classification using PSO to study the behavior of PSO parameters and to observe the accuracy of classification of PSO based classifier in different iterations. Extensive simulation has been carried out using UCI data, on which classification is done using our proposed algorithm. We have also explored the possible influence of variants of PSO on accuracy of classification. The obtained results indicate that particle swarm optimization is an effective technique for classification and can be used successfully on more demanding problem domain.

Keywords: *classification; Particle swarm optimization; Euclidean distance*

1. Introduction

Particle swarm optimization (PSO)[1] is a stochastic based search algorithm widely used to find the optimum solution introduced by Kennedy and Eberthart[1] in 1995. PSO is a effective optimization technique to search for global optimized solution[9] but time of convergence[3] is uncertain. Like other population based optimization[2][6] methods the particle swarm optimization starts with randomly initialized population[6] for individuals.PSO works on the social behavior[1] of particle. It finds the global best solution by adjusting each individual's positions[12] with respect to global best position of particle of the entire population. Each individuals is adjusting by altering the velocity[12] according to its own experience and by observing the experience of the particles in search space. According to the used fitness function, local best (lbest) and global best(gbest) will be calculated. The positions and velocities of the particles initially in search space denoted by V and X. Then the new velocities and positions of the particles for next iterations [5] can be evaluated by using the equations 1 and 2.

$$V_{id}\ (t+1) = V_{id}\ (t) + C_1{*}Rand\ (\ )\ (lb_{id} - X_{id}) + C_2\ {*}Rand\ (\ )\ (gb_{id} - X_{id}) \quad \textbf{(1)}$$

$$Xid\ (t+1) = Xid\ (t) + Vid\ (t+1) \quad \textbf{(2)}$$

Where C1 and C2 are the constants [8, 10] and Rand () is random function which generate random numbers in between 0 and 1. In above equation 'i' is the number of instance , 'd' is the dimensions of instances and 't' is the iteration number. 'gb' is the particle in the neighborhood with the best fitness and 'lb' is the position for a particle's best fitness yet encountered. Equation (3) is responsible for social influence the particles in the search space and equation (4) is known as cognition model[12] of particles.

$$V_{id} = V_{id} + C_1{*}Rand\ (\ ){*}(gb_{id} - X_{id}) \quad \textbf{(3)}$$

$$V_{id} = V_{id} + C_2{*}Rand\ (\ ){*}(lb_{id} - X_{id}) \quad \textbf{(4)}$$

Knowledge discovery is a concept of the field of computer science that describes the process of automatically searching large volumes of data for patterns that can be considered knowledge about the data. It is often described as deriving knowledge from the input data. The most well-known branch of data mining is known as KDD(Knowledge Discovery in Databases). Just as many other forms of knowledge discovery it creates abstractions of the input data. The knowledge obtained through the process may become additional data that can be used for further usage and discovery. Classification[7] is a popular knowledge discovery technique to find previously unknown, valid patterns and relationships in large data set. Classification used to predict group membership for data instances. Classification can be done in two phase first one is training phase and second is testing phase. In the training phase, we develop a classifier and in testing phase we develops a model.

Instance based learning is the most widely and simple classification algorithm which requires less computation time during the training phase. Instances can be considered as points within an *n*-dimensional instance space [13] where each of the *n*-dimensions corresponds to one of the *n*-features that are used to describe an instance. Closeness (belongingness) of the instance can be measured using relative distance. We minimize the distance between two similar classified instances, while maximizing the distance between instances of different classes. To measure the relative distance between instances we are using Euclidean distance by using equation(5).

$$\text{Euclidean Distance : } D(X, Y) = \left(\sum_{i=1}^{N} (x_i - y_i)^2 \right)^{½} \quad \textbf{(5)}$$

Where X and Y are two instances of the dataset and i is the no of attributes. x and y are two attribute values of X and Y instances.

2. Training Phase from Real Data

We have taken Breast cancer dataset (a multi dimensional dataset) for classification. Breast cancer dataset is having 699 instances(Table-2). Description of dataset is given in table-1.Each instance is having 10 attribute. First 9 attribute represent symptoms of a disease and last attribute represent class of disease. Breast cancer dataset have 2 class of disease indicated by class-1 and classs-2. Value 1 and 2 in 10^{th} attribute of any instances represents class-1 and class-2 disease respectively(given the description in Table-2). Training phase consists of several procedures (figure-1).

Table-1: Breast Cancer Dataset Description

BREAST CANCER DATASET FROM UCI REPOSITORY
Attribute Information:
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. Marginal Adhesion: 1 - 10 6. Single Epithelial Cell Size: 1 - 10 7. Bare Nuclei: 1 - 10 8. Bland Chromatin: 1 - 10 9. Normal Nucleoli: 1 - 10 10. Mitoses: 1 - 10 11. Class: (2 for benign, 4 for malignant)
Attribute Information: (class attribute has been moved to last column)
Attribute Domain -- --- 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. Marginal Adhesion 1 - 10 6. Single Epithelial Cell Size 1 - 10 7. Bare Nuclei 1 - 10 8. Bland Chromatin 1 - 10 9. Normal Nucleoli 1 - 10 10. Mitoses 1 - 10 11. Class: (2 for benign, 4 for malignant)
Class distribution:
Benign: 458 (65.5%) **Malignant:** 241 (34.5%)
Attributes 2 through 10 have been used to represent instances. Each instance has one of 2 possible classes: benign or malignant.

Table-2: Breast Cancer Dataset

1 1 2 1 2 1 2 1 1 1	4 1 1 1 2 1 1 1 1 1	3 1 1 1 1 1 2 1 1 1	3 1 1 1 2 1 1 1 1 1
5 7 7 1 5 8 3 4 1 1	1 1 1 1 2 1 1 1 1 1	1 1 1 1 2 1 1 1 1 1	2 1 1 1 2 1 1 1 1 1
5 3 1 2 2 1 2 1 1 1	1 2 2 1 2 1 2 1 1 1	1 1 2 2 2 1 3 1 1 1	1 1 1 1 2 1 2 1 1 1
3 1 1 1 2 1 2 1 1 1	3 1 1 1 3 1 2 1 1 1	5 3 3 2 3 1 3 1 1 1	1 1 1 1 2 1 2 1 1 1
3 1 1 1 2 1 2 1 1 1	5 1 2 1 2 1 1 1 1 1	1 1 1 2 2 1 2 1 1 1	4 1 1 1 2 1 1 1 1 1
1 1 1 1 2 1 3 1 1 1	1 1 1 1 2 1 1 1 1 1	2 1 1 1 2 1 1 1 1 1	2 2 2 1 1 1 7 1 1 1
1 1 1 1 3 2 2 1 1 1	4 1 1 1 2 1 2 1 1 1	2 1 1 1 2 1 3 1 1 1	1 1 3 1 2 1 2 1 1 1
1 1 1 1 1 1 1 1 1 1	5 1 3 1 2 1 2 1 1 1	1 1 2 1 3 1 1 1 1 1	1 1 1 1 2 1 2 1 1 1
4 1 1 1 2 1 2 1 1 1	5 1 1 1 2 2 2 1 1 1	1 1 1 1 2 1 3 1 1 1	1 2 1 3 2 1 1 2 1 1
5 1 3 3 2 2 2 3 1 1	3 1 1 1 2 1 2 1 1 1	3 1 3 1 2 1 2 1 1 1	1 1 1 1 4 3 1 1 1 1
5 1 1 1 2 1 2 1 1 1	2 1 1 1 2 1 1 1 1 1	5 1 1 1 2 1 2 1 1 1	1 1 1 1 2 5 1 1 1 1
3 1 1 1 2 1 1 1 1 1	3 1 1 1 2 1 3 1 1 1	1 1 2 1 2 2 4 2 1 1	1 1 1 1 1 1 1 3 1 1
1 1 4 1 2 1 2 1 1 1	7 1 2 3 2 1 2 1 1 1	5 1 1 1 2 1 2 2 1 1	3 1 2 1 2 1 2 1 1 1
1 1 1 1 2 1 1 1 1 1	3 1 1 1 3 2 1 1 1 1	2 1 1 1 2 1 3 1 1 1	1 1 1 1 2 1 2 1 1 1
4 1 1 3 2 1 1 1 1 1	1 1 1 1 2 1 1 1 1 1	3 2 2 3 2 3 3 1 1 1	1 2 3 1 2 1 3 1 1 1
1 1 1 1 2 1 2 1 1 1	1 1 1 1 3 1 1 1 1 1	8 4 4 5 4 7 7 8 2 1	1 1 1 1 2 1 3 1 1 1
3 1 1 2 2 1 1 1 1 1	2 1 1 1 2 1 3 1 1 1	5 2 2 4 2 4 1 1 1 1	4 1 2 1 2 1 3 1 1 1
5 1 1 1 2 1 1 1 1 1	1 1 1 1 2 1 3 1 1 1	3 1 1 1 1 1 2 1 1 1	4 1 1 1 2 1 3 1 1 1
3 1 1 1 2 1 1 1 1 1	1 1 1 1 2 10 3 1 1 1	1 1 1 1 2 1 2 1 1 1	3 1 1 1 2 3 3 1 1 1
2 1 1 1 2 5 1 1 1 1	4 1 3 1 2 1 2 1 1 1	4 4 2 1 2 5 2 1 2 1	2 1 1 1 3 1 2 1 1 1
2 1 1 1 2 1 2 1 1 1	2 1 1 1 2 1 2 1 1 1	5 1 1 3 2 1 1 1 1 1	5 1 1 1 1 1 1 1 1 1
3 1 1 1 2 1 1 1 1 1	1 1 1 1 2 1 1 1 1 1	1 1 1 1 2 1 1 1 1 1	1 1 1 1 1 1 2 1 1 1
2 1 1 1 2 1 1 1 1 1	4 1 1 1 2 1 2 1 1 1	3 2 2 2 2 1 3 2 1 1	4 1 1 1 2 1 1 1 1 1
2 1 1 1 2 1 2 1 1 1	5 1 1 1 2 1 2 1 1 1	6 1 1 1 2 1 3 1 1 1	4 2 1 1 2 2 3 1 1 1
1 1 1 1 2 2 2 1 1 1	3 1 1 1 2 1 2 1 1 1	4 2 2 1 2 1 2 1 1 1	1 1 1 1 2 1 2 3 1 1
1 1 1 1 2 1 2 1 1 1	3 1 1 1 2 1 3 1 1 1	5 1 1 1 2 1 2 2 1 1	1 2 3 1 2 1 1 1 1 1
5 1 1 6 3 1 2 1 1 1	4 1 1 1 1 1 2 1 1 1	1 1 1 1 2 1 1 1 1 1	1 1 3 1 2 1 1 1 1 1
5 1 1 1 3 2 2 2 1 1	1 2 2 1 2 1 1 1 1 1	1 1 1 1 2 5 1 1 1 1	4 1 1 3 2 1 3 1 1 1
1 1 1 1 2 2 1 1 1 1	4 1 1 1 2 1 2 1 1 1	5 1 1 1 2 1 2 2 1 1	3 1 1 1 2 5 5 1 1 1
4 4 4 2 2 3 2 1 1 1	2 1 1 1 2 1 3 1 1 1	3 1 1 1 2 1 2 1 1 1	5 2 2 2 2 2 3 2 2 1
4 1 2 1 2 1 1 1 1 1	1 1 1 1 2 1 3 1 1 1	4 2 1 1 2 1 1 1 1 1	4 1 1 1 2 1 3 2 1 1
4 1 1 1 2 1 1 1 1 1	4 1 1 1 2 1 1 2 1 1	5 1 1 3 2 1 1 1 1 1	1 1 1 1 2 1 3 1 1 1
5 1 1 1 2 1 3 1 1 1	1 1 1 1 2 1 2 1 1 1	5 1 1 6 3 1 1 1 1 1	1 1 1 1 2 1 3 1 1 1
4 2 4 3 2 2 2 1 1 1	3 1 1 1 2 2 3 1 1 1	1 1 1 1 2 1 2 1 1 1	1 1 1 1 5 1 3 1 1 1
1 1 1 2 2 1 3 1 1 1	5 1 2 1 2 1 1 1 1 1	1 2 2 1 2 1 1 1 1 1	5 1 1 1 2 1 2 1 1 1
5 1 1 2 2 2 3 1 1 1	5 1 1 1 2 1 3 1 1 1	1 1 1 1 1 1 2 1 1 1	1 1 1 1 2 1 3 1 1 1
3 1 2 1 2 1 2 1 1 1	6 3 3 5 3 10 3 5 3 1	1 1 1 1 2 1 2 1 1 1	6 1 1 1 2 1 2 1 1 1
5 1 1 1 2 1 1 1 1 1	4 1 2 1 2 1 2 1 1 1	3 1 1 1 2 4 1 1 1 1	3 1 1 1 2 1 2 1 1 1
1 1 1 1 2 1 3 1 1 1	2 1 1 2 2 1 3 1 1 1	3 1 4 1 2 1 3 1 1 1	1 1 1 1 1 1 3 1 1 1
1 1 1 1 1 1 2 1 1 1	2 1 1 1 2 1 3 1 1 1	1 1 1 1 2 1 3 2 1 1	4 1 1 1 3 1 2 2 1 1
1 1 1 1 1 1 1 1 1 1	5 1 1 1 2 1 2 1 1 1	4 1 1 1 2 1 3 1 1 1	2 3 1 1 5 1 1 1 1 1
5 3 3 1 2 1 2 1 1 1	5 1 1 1 2 1 3 1 1 1	3 4 5 3 7 3 4 6 1 1	3 1 1 1 2 1 1 1 1 1
1 1 1 1 2 1 2 1 1 1	4 1 1 2 2 1 1 1 1 1	4 1 1 1 1 1 2 1 1 1	3 1 1 3 8 1 5 8 1 1
1 1 3 1 2 1 2 1 1 1	3 2 2 1 2 1 2 3 1 1	4 3 3 1 2 1 3 3 1 1	4 1 1 1 2 1 2 1 1 1
2 1 1 1 1 1 3 1 1 1	5 1 1 1 2 1 2 1 1 1	6 6 6 9 6 2 7 8 1 1	1 1 1 1 2 1 1 1 1 1
3 1 1 1 2 1 2 1 1 1	1 3 1 1 2 1 2 2 1 1	4 1 1 1 2 1 3 1 1 1	2 1 3 2 2 1 2 1 1 1
1 1 1 1 2 1 2 1 1 1	5 1 1 1 2 1 3 1 1 1	1 1 1 1 2 1 3 1 1 1	1 1 1 1 2 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1	3 1 2 2 2 1 1 1 1 1	2 1 1 1 2 1 2 1 1 1	1 1 1 2 1 1 1 1 1 1
3 2 2 1 4 3 2 1 1 1	3 1 1 1 2 1 3 1 1 1	5 1 1 1 1 1 3 1 1 1	6 3 3 3 3 2 6 1 1 1
4 1 1 1 2 1 2 1 1 1	1 1 1 1 2 1 1 1 1 1	5 1 1 1 2 1 1 1 1 1	3 1 2 1 2 1 2 1 1 1
5 1 1 1 2 1 3 1 1 1	4 1 1 1 2 1 3 1 1 1	3 1 1 3 2 1 2 1 1 1	1 1 1 1 2 1 3 1 1 1
5 1 1 1 2 1 3 1 2 1	3 1 1 4 3 1 2 2 1 1	1 1 1 1 2 1 3 1 1 1	1 1 3 1 1 1 2 1 1 1
3 1 1 1 2 1 1 1 1 1	1 1 1 1 2 1 3 1 1 1	1 1 1 1 2 1 3 1 1 1	2 1 1 2 2 1 3 1 1 1
4 1 1 2 2 1 2 1 1 1	3 3 2 2 3 1 1 2 3 1	5 1 1 1 2 1 3 1 1 1	1 1 1 1 1 1 3 1 1 1
1 1 1 1 2 1 1 1 1 1	8 2 1 1 5 1 1 1 1 1	3 1 1 1 2 1 2 1 1 1	3 3 2 1 3 1 3 6 1 1
3 1 1 1 1 1 2 1 1 1	5 3 4 3 4 5 4 7 1 1	2 1 1 1 2 1 3 1 1 1	1 1 1 1 2 1 1 1 1 1
1 1 1 1 2 1 2 1 1 1	3 1 1 1 2 1 1 1 1 1	1 1 1 1 2 1 1 1 1 1	1 1 1 1 2 1 3 1 1 1
3 1 1 1 2 1 2 1 1 1	5 1 1 1 2 1 1 1 1 1	3 1 1 1 2 1 2 1 1 1	1 1 3 2 2 1 3 1 1 1
2 1 1 1 1 1 2 1 1 1	3 1 1 1 2 1 2 1 1 1	1 1 1 1 10 1 1 1 1 1	1 1 3 1 2 1 2 1 1 1
6 1 1 1 2 1 3 1 1 1	3 1 1 1 2 1 2 1 1 1	1 1 1 1 1 1 2 1 1 1	1 1 1 1 2 3 3 1 1 1
2 1 2 1 2 1 3 1 1 1	4 1 1 1 2 1 3 6 1 1	8 3 3 1 2 2 3 2 1 1	4 1 1 1 2 1 2 1 1 1
1 1 1 3 2 1 1 1 1 1	4 1 1 1 2 1 3 2 1 1	1 1 1 1 2 1 2 1 1 1	1 1 1 1 2 1 1 1 1 1
3 1 1 3 1 1 3 1 1 1	2 3 2 2 2 2 3 1 1 1	2 1 1 1 2 1 1 1 1 1	3 1 1 2 3 4 1 1 1 1
6 9 7 5 5 8 4 2 1 1	4 1 1 1 2 2 3 2 1 1	1 1 1 1 2 1 1 1 1 1	3 1 1 1 2 2 3 1 1 1
5 1 1 1 2 1 2 1 1 1	4 2 1 1 2 1 2 1 1 1	3 1 4 1 2 1 1 1 1 1	2 1 1 1 2 1 3 1 1 1
5 1 2 10 4 5 2 1 1 1	4 1 1 1 2 3 1 1 1 1	3 1 1 1 2 1 3 1 1 1	5 1 1 1 2 1 2 1 1 1

1 1 3 1 2 1 1 1 1 1	3 1 1 3 2 1 1 1 1 1	1 1 1 1 2 1 2 1 1 1	1 1 1 1 2 1 1 1 8 1
5 1 1 1 2 1 1 1 1 1	4 1 1 1 2 1 3 1 1 1	1 1 1 1 1 1 3 1 1 1	4 1 1 1 2 1 1 1 1 1
1 1 1 1 2 1 1 1 1 1	5 2 2 2 3 1 1 3 1 1	6 1 1 3 2 1 1 1 1 1	2 1 1 2 2 1 1 1 1 1
3 1 2 1 2 1 3 1 1 1	3 1 1 1 2 1 1 1 1 1	1 1 1 1 1 1 3 1 1 1	3 1 1 1 2 1 2 1 1 1
1 1 1 1 2 1 1 1 1 1	5 3 1 1 2 1 1 1 1 1	2 1 1 1 2 1 1 1 1 1	3 1 1 1 1 1 1 1 1 1
1 2 3 1 2 1 2 1 1 1	3 1 1 1 2 1 1 1 1 1	4 1 1 1 2 1 1 1 1 1	5 1 1 1 2 1 2 1 1 1
3 1 1 1 2 1 2 1 2 1	1 1 1 1 2 1 3 1 1 1	5 3 4 1 4 1 3 1 1 1	5 1 2 1 2 1 3 1 1 1
3 1 1 1 1 1 2 1 1 1	5 1 2 1 2 1 3 1 1 1	5 3 6 1 2 1 1 1 1 1	3 1 1 1 2 1 2 1 1 1
5 3 2 1 3 1 1 1 1 1	5 3 2 4 2 1 1 1 1 1	5 1 3 1 2 1 1 1 1 1	1 2 1 3 2 1 2 1 1 1
1 1 1 3 2 1 1 1 1 1	6 8 8 1 3 4 3 7 1 1	1 1 3 1 2 1 1 1 1 1	1 1 1 1 1 1 2 1 1 1
6 1 1 3 2 1 1 1 1 1	1 1 1 1 2 1 3 1 1 1	5 2 2 2 2 1 1 1 2 1	1 1 1 1 2 1 3 1 1 1
1 1 1 1 2 1 1 1 1 1	2 1 1 1 2 1 1 1 1 1	1 1 1 1 2 1 2 1 1 1	4 6 5 6 7 2 4 9 1 1

2 1 1 1 2 1 1 1 5 1 2 1 1 1 2 1 2 1 1 1 1 1 1 3 1 3 1 1 1 1 3 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 4 1 2 1 2 1 2 1 1 1 3 1 1 1 2 1 3 1 1 1 3 1 1 1 2 1 1 1 1 1 3 2 1 2 2 1 3 1 1 1 5 1 1 3 2 1 1 1 1 1 5 4 5 1 8 1 3 6 1 1 5 2 2 2 2 1 2 2 1 1 4 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 4 1 1 2 2 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1 5 1 2 1 2 1 1 1 1 1 3 2 2 2 2 1 4 2 1 1 1 1 1 1 2 1 1 1 1 1 5 1 1 1 2 1 3 2 1 1 3 1 3 1 3 4 1 1 1 1 2 1 1 1 2 1 1 1 1 1 4 1 1 3 2 1 3 1 1 1 8 4 6 3 3 1 4 3 1 1 1 1 1 1 2 1 3 1 1 1 3 1 1 1 2 1 2 1 1 1 3 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 2 1 1 1 1 1 5 4 4 5 7 10 3 2 1 1 1 1 2 1 2 1 2 1 1 1 5 2 4 1 1 1 1 1 1 1 1 1 1 1 2 1 3 1 1 1 1 1 1 1 2 1 2 1 2 1 4 1 2 1 2 1 2 1 1 1 3 3 2 6 3 3 3 5 1 1 3 1 1 1 2 5 1 1 1 1 2 1 1 1 2 1 2 1 1 1 2 3 1 1 3 1 1 1 1 1 1 1 1 1 2 1 3 1 1 1 1 1 1 1 2 1 3 1 1 1 5 1 1 2 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 4 3 2 1 3 1 2 1 1 1 5 1 3 1 2 1 3 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 2 1 2 2 1 1 5 1 1 1 2 1 2 1 1 1 3 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 5 1 1 1 2 1 1 1 1 1	1 1 1 3 2 3 1 1 1 1 4 3 1 1 2 1 4 8 1 1 3 3 2 1 2 3 3 1 1 1 4 2 2 1 2 1 2 1 1 1 5 2 2 2 1 1 2 1 1 1 4 1 1 1 2 1 3 1 1 1 1 3 3 2 2 1 7 2 1 1 1 1 1 1 2 1 1 1 1 1 4 1 4 1 2 1 1 1 1 1 5 2 1 1 2 1 3 1 1 1 4 1 1 1 2 1 2 1 1 1 5 1 1 3 4 1 3 2 1 1 4 1 1 1 2 1 1 1 1 1 3 1 1 1 2 1 3 1 1 1 6 1 1 1 2 1 3 1 1 1 3 1 1 1 2 1 3 1 1 1 3 1 1 1 2 1 2 1 1 1 3 1 1 1 2 1 3 1 1 1 4 4 4 4 6 5 7 3 1 1 3 1 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 5 1 1 4 2 1 3 1 1 1 4 1 3 3 2 1 1 1 1 1 3 1 1 1 2 1 2 2 1 1 5 1 4 1 2 1 3 2 1 1 4 1 1 3 1 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 4 1 1 1 3 1 1 1 1 1 2 1 1 1 2 1 3 1 1 1 6 1 3 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 1 1 1 1 1 1 1 1 1 3 1 1 1 2 1 1 1 1 1 3 1 1 1 2 1 3 2 1 1 1 3 1 2 2 2 5 3 2 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2 3 1 1 2 1 2 1 1 1 3 1 1 1 2 2 7 1 1 1 3 1 1 1 2 1 3 1 1 1 3 2 1 1 2 2 3 1 1 1 6 2 1 1 1 1 7 1 1 1 3 1 1 1 2 1 3 1 1 1 5 1 1 1 2 1 1 1 1 1 5 1 2 1 2 1 3 1 1 1 1 1 1 1 1 1 3 1 1 1 2 1 1 1 3 1 2 1 1 1 1 1 1 1 2 1 3 1 1 1 6 1 3 1 2 1 3 1 1 1 1 1 1 1 2 1 1 1 1 1 4 1 1 1 2 1 3 1 1 1	10 5 5 6 8 8 7 1 1 2 7 5 10 10 10 10 4 10 3 2 10 10 10 10 3 10 10 6 1 2 7 2 4 1 6 10 5 4 3 2 8 10 10 1 3 6 3 9 1 2 8 10 5 3 8 4 4 10 3 2 6 10 2 8 10 2 7 8 10 2 10 8 10 10 6 1 3 1 10 2 7 8 7 2 4 8 3 8 2 2 8 8 9 6 6 3 10 10 1 2 5 10 10 9 6 10 7 10 5 2 5 10 10 10 10 10 10 1 1 2 7 3 4 4 3 3 3 2 7 2 10 3 4 5 3 10 4 1 1 2 10 10 10 10 7 10 7 10 4 2 8 7 6 4 4 10 5 1 1 2 10 8 8 4 10 10 8 1 1 2 8 7 8 2 4 2 5 10 1 2 7 8 8 7 3 10 7 2 3 2 10 10 10 10 5 10 10 10 7 2 10 8 10 1 3 10 5 1 1 2 5 3 3 4 2 4 3 4 1 2 7 4 5 10 2 10 3 8 2 2 10 6 6 3 4 5 3 6 1 2 8 6 5 4 3 10 6 1 1 2 8 3 8 3 4 9 8 9 8 2 10 3 6 2 3 5 4 10 2 2 8 8 9 4 5 10 7 8 1 2 4 8 6 3 4 10 7 1 1 2 5 10 10 10 5 2 8 5 1 2 7 6 10 5 3 10 9 10 2 2 7 5 6 10 5 10 7 9 4 2 8 10 10 10 5 10 8 10 6 2 7 5 6 3 3 8 7 4 1 2 5 10 10 10 6 10 6 5 2 2 5 7 9 8 6 10 8 10 1 2 7 6 6 3 2 10 7 1 1 2 9 10 10 1 10 8 3 3 1 2 6 10 7 7 6 4 8 10 2 2 6 6 7 10 3 10 8 10 2 2 10 6 6 2 4 10 9 7 1 2 5 3 2 8 5 10 8 1 2 2 8 3 4 9 3 10 3 3 1 2 4 8 7 10 4 10 7 5 1 2 8 6 7 3 3 10 3 4 2 2	2 5 7 6 4 10 7 6 1 2 10 8 4 4 4 10 3 10 4 2 5 8 7 7 10 10 5 7 1 2 3 4 4 10 5 1 3 3 1 2 9 9 10 3 6 10 7 10 6 2 6 10 10 10 8 10 7 10 7 2 7 4 7 4 3 7 7 6 1 2 8 8 7 4 10 10 7 8 7 2 7 8 7 6 4 3 8 8 4 2 10 10 10 8 6 1 8 9 1 2 9 10 10 10 10 10 10 10 1 2 6 10 10 10 8 10 10 10 7 2 5 3 3 1 3 3 3 3 3 2 9 6 9 2 10 6 2 9 10 2 5 4 6 10 2 10 4 1 1 2 3 3 6 4 5 8 4 4 1 2 8 10 10 8 6 9 3 10 10 2 4 2 3 5 3 8 7 6 1 2 5 10 10 3 7 3 8 10 2 2 4 6 6 5 7 6 7 7 3 2 5 4 6 7 9 7 8 10 1 2 4 10 8 5 4 1 10 1 1 2 10 2 2 1 2 6 1 1 2 2 9 7 7 5 5 10 7 8 3 2 5 8 4 10 5 8 9 10 1 2 10 10 10 3 10 10 9 10 1 2 7 2 4 1 3 4 3 3 1 2 6 3 4 1 5 2 3 9 1 2 5 7 10 10 5 10 10 10 1 2 5 3 5 5 3 3 4 10 1 2 6 1 3 1 4 5 5 10 1 2 8 4 10 5 4 4 7 10 1 2 10 10 10 7 10 10 8 2 1 2 10 10 10 4 8 1 8 10 1 2 9 5 5 4 4 5 4 3 3 2 7 9 4 10 10 3 5 3 3 2 10 10 8 6 4 5 8 10 1 2 10 5 7 3 3 7 3 3 8 2 7 6 4 8 10 10 9 5 3 2 10 10 7 8 7 1 10 10 3 2 5 5 5 2 5 10 4 3 1 2 8 7 8 5 5 10 9 10 1 2 8 10 10 8 5 10 7 8 1 2 5 2 3 4 2 7 3 6 1 2 4 7 8 3 4 10 9 1 1 2

1 1 1 1 2 1 2 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 2 1 3 1 1 7 1 3 2 1 1 1 1 2 1 1 1 3 1 1 1 2 1 2 1 1 1 5 1 1 3 2 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 2 1 2 1 1 1 5 1 2 1 2 1 2 1 1 1 5 1 1 2 2 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 6 2 3 1 2 1 1 1 1 1 5 1 1 1 2 2 3 3 1 1 5 1 3 1 2 1 2 1 1 1 3 2 1 1 2 1 2 2 1 1 2 1 1 1 2 1 2 1 1 1 3 1 1 1 2 1 3 1 1 1 1 1 1 1 2 5 5 1 1 1 5 4 3 1 2 2 2 3 1 1 3 1 1 1 3 1 2 1 1 1 4 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 5 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 5 1 3 1 2 1 2 1 1 1 4 1 1 1 2 1 3 1 1 1 3 2 2 3 2 1 1 1 1 1 1 1 1 1 2 4 1 1 1 1	4 1 1 1 2 1 2 1 1 1 3 1 1 1 2 1 2 3 1 1 5 1 1 1 2 1 1 1 1 1 3 1 1 1 2 1 1 1 1 1 5 1 1 1 2 1 3 1 1 1 4 1 1 1 2 1 2 1 1 1 3 1 1 1 2 1 2 1 1 1 2 1 1 2 3 1 2 1 1 1 5 2 1 1 2 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 2 1 1 1 2 1 3 1 1 1 5 1 1 1 2 1 3 1 1 1 3 3 1 1 2 1 1 1 1 1 4 1 1 1 2 1 3 1 1 1 4 1 1 1 2 3 2 1 1 1 2 3 4 4 2 5 2 5 1 2 3 3 5 2 3 10 7 1 1 2 10 5 6 10 6 10 7 7 10 2 10 3 5 1 10 5 3 10 2 2 6 10 10 2 8 10 7 3 3 2 8 7 4 4 5 3 5 10 1 2 8 10 4 4 8 10 8 2 1 2 10 4 6 4 5 10 7 1 1 2 10 6 5 8 5 10 8 6 1 2 10 8 8 2 8 10 4 8 10 2 10 10 10 3 10 8 8 1 1 2 8 10 10 7 10 10 7 3 8 2 8 10 10 10 8 10 10 7 3 2 8 10 8 8 4 8 7 7 1 2	5 6 6 2 4 10 3 6 1 2 10 10 9 3 7 5 3 5 1 2 10 1 1 1 2 10 5 4 1 2 5 3 5 1 8 10 5 3 1 2 5 5 5 6 3 10 3 1 1 2 4 5 5 10 4 10 7 5 8 2 8 10 10 10 6 10 10 10 10 2 10 9 7 3 4 2 7 7 1 2 5 3 4 1 8 10 4 9 1 2 4 8 6 4 3 4 10 6 1 2 10 10 8 10 6 5 10 3 1 2 9 1 2 6 4 10 7 7 2 2 8 6 4 10 10 1 3 5 1 2 10 3 3 1 2 10 7 6 1 2 6 10 10 10 10 10 8 10 10 2 9 10 10 1 10 8 3 3 1 2 5 10 10 10 10 2 10 10 10 2 10 10 10 2 10 10 5 3 3 2 5 7 10 6 5 10 7 5 1 2 3 10 7 8 5 8 7 4 1 2 6 3 2 1 3 4 4 1 1 2 8 7 8 7 5 5 5 10 2 2 7 6 3 2 5 10 7 4 6 2 10 5 8 10 3 10 5 1 3 2 8 5 5 5 2 10 4 3 1 2 8 10 10 8 7 10 9 7 1 2 8 2 3 1 6 3 7 1 1 2 10 5 5 3 6 7 7 10 1 2 6 6 6 5 4 10 7 6 2 2 10 3 5 4 3 7 3 5 3 2 10 4 4 10 2 10 5 3 3 2 1 4 3 10 4 10 5 6 1 2 5 6 5 6 10 1 3 1 1 2 10 8 7 4 3 10 7 9 1 2 10 10 10 10 6 10 8 1 5 2	10 5 7 4 4 10 8 9 1 2 10 6 4 3 10 10 9 10 1 2 10 10 10 8 6 8 7 10 1 2 10 4 3 10 4 10 10 1 1 2 8 7 8 5 10 10 7 2 1 2 4 5 5 8 6 10 10 7 1 2 6 10 10 10 4 10 7 10 1 2 10 4 7 2 2 8 6 1 1 2 3 6 6 6 5 10 6 8 3 2 6 5 5 8 4 10 3 4 1 2 8 8 8 1 2 1 6 10 1 2 10 4 3 10 3 10 7 1 2 2 7 5 6 10 4 10 5 3 1 2 5 8 9 4 3 10 7 1 1 2 9 8 8 9 6 3 4 1 1 2 10 4 4 6 2 10 2 3 1 2 5 3 3 3 6 10 3 1 1 2 6 8 7 5 6 8 8 9 2 2 8 4 4 1 6 10 2 5 2 2 7 5 3 7 4 10 7 5 5 2 5 7 4 1 6 1 7 10 3 2 5 10 10 6 10 10 10 6 5 2 5 8 8 10 5 10 8 10 3 2 8 2 4 1 5 1 5 4 4 2 1 6 8 10 8 10 5 7 1 2 3 5 7 8 8 9 7 10 7 2 9 5 8 1 2 3 2 1 5 2 8 9 9 5 3 5 7 7 1 2 5 6 7 8 8 10 3 10 3 2 10 4 3 1 3 3 6 5 2 2 8 3 5 4 5 10 1 6 2 2 5 5 7 8 6 10 7 4 1 2 5 4 6 6 4 10 4 3 1 2 1 5 8 6 5 8 7 10 1 2 3 10 3 10 6 10 5 1 4 2

10	4	5	4	3	5	7	3	1	2
4	8	8	5	4	5	10	4	1	2
6	10	10	2	8	10	7	3	3	2
7	8	3	7	4	5	7	8	2	2
8	10	10	10	6	10	10	10	1	2
9	5	5	2	2	2	5	1	1	2
5	6	6	8	6	10	4	10	4	2
7	3	2	10	5	10	5	4	4	2
5	8	8	8	5	10	7	8	1	2
5	10	6	1	10	4	4	10	10	2
9	4	5	10	6	10	4	8	1	2
8	4	4	1	2	9	3	3	1	2
10	3	3	10	2	10	7	3	3	2
3	10	8	7	6	9	9	3	8	2
10	4	6	1	2	10	5	3	1	2
8	7	5	10	7	9	5	5	4	2
9	8	8	5	6	2	4	10	4	2
10	9	8	7	6	4	7	10	3	2
10	4	3	2	3	10	5	3	2	2
10	5	5	6	3	10	7	9	2	2
10	10	10	6	8	4	8	5	1	2
4	10	4	7	3	10	9	10	1	2
7	4	6	4	6	1	4	3	1	2
5	3	3	3	2	3	4	4	1	2
2	7	10	10	7	10	4	9	4	2
10	8	8	2	3	4	8	7	8	2
6	8	7	8	6	8	8	9	1	2
3	6	4	10	3	3	3	4	1	2
10	7	7	6	4	10	4	1	2	2
5	5	5	8	10	8	7	3	7	2
5	10	10	8	5	5	7	10	1	2
8	10	10	10	7	5	4	8	7	2
5	2	3	1	6	10	5	1	1	2
10	10	10	10	10	1	8	8	8	2
3	4	5	2	6	8	4	1	1	2
9	10	10	10	10	5	10	10	10	2
6	5	4	4	3	9	7	8	3	2
10	4	4	10	6	10	5	5	1	2
10	6	4	1	3	4	3	2	3	2
10	4	2	1	3	2	4	3	10	2
8	6	4	3	5	9	3	1	1	2
5	10	10	10	4	10	5	6	3	2
10	10	6	3	3	10	4	3	2	2
10	7	7	3	8	5	7	4	3	2
5	4	4	9	2	10	5	6	1	2
10	5	10	3	5	8	7	8	3	2
5	10	8	10	8	10	3	6	3	2
4	1	1	3	1	5	2	1	1	2
3	7	7	4	4	9	4	8	1	2
10	10	10	7	9	10	7	10	10	2
10	7	7	4	5	10	5	7	2	2
10	10	10	8	2	10	4	1	1	2
2	5	3	3	6	7	7	5	1	2
6	10	5	5	4	10	6	10	1	2
5	10	10	3	8	1	5	10	3	2
10	4	5	5	5	10	4	1	1	2
10	10	10	1	6	1	2	8	1	2
3	3	5	2	3	10	7	1	1	2
8	5	6	2	3	10	6	6	1	2
10	10	10	10	10	10	4	10	10	2
5	4	6	8	4	1	8	10	1	2
10	6	3	6	4	10	7	8	4	2
5	10	10	5	4	5	4	4	1	2
7	4	4	3	4	10	6	9	1	2
8	4	5	1	2	10	7	3	1	2
8	4	7	1	3	10	3	9	2	2
8	10	3	2	6	4	3	10	1	2

Algo PSOBASEDCLASSIFIER(Breast_Cancer_Dataset)

1. Create CLASS1 dataset from given Breast_Cancer_Dataset
2. Create CLASS2 dataset from given Breast_Cancer_Dataset
3. Generate initial velocity V randomly
4. Create a dataset POSITION by taking 50% data from class1 dataset and 50% data from class2 dataset
5. Create a population dataset SWARM by taking 50% data from class1 dataset and 50% data from class2 dataset
6. FV = CALCULATE_FITNESS(POSITION,SWARM);
7. [gbest_cls1,gbest_cls2]= CALCULATE_GBEST(FV,POSITION);
8. repeat step 8 to 13 until stop cretiria
9. next_velocity=VELOCITY_NEXT(POSITION,gbest_cls1,gbest_cls2,lbest,V);
10. next_position = POSITION_NEXT(POSITION,next_velocity);
11. POSITION = LBEST_NEXT(POSITION,next_position);
12. V = next_velocity;
13. check for stop cretiria
14. end step 7
15. exit

CLASS1 and CLASS2 dataset contain class-1 and class-2 data respectively. Initial velocity V generated randomly. Test dataset from CLASS1 and CLASS2 has been taken as initial position called POSITION. All instances of the dataset POSITION will be treated as a particle in 9 dimensional space. SWARM is a dataset generated from breast_cancer_dataset, having both class of data (figure-1).

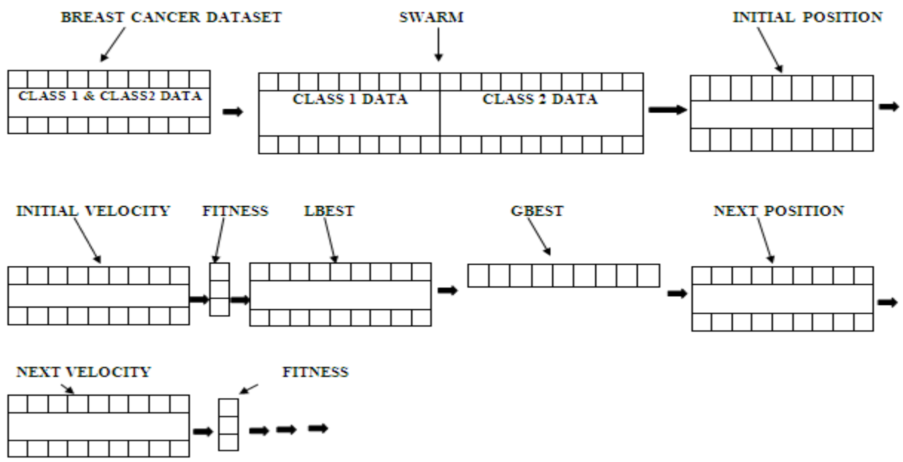


Figure-1: Generation of next position and next velocity of particles from its initial position and velocity in subsequence iterations.

Initial fitness of each particle will be store in a array FV. Fitness value of each particle will be calculated by using procedure CALCULATE_FITNESS() (figure-1). Euclidean distance will be computed from particle of POSITION to class-1 and class2 particle of SWARM (figure-2). Distance from particle of POSITON to class-1 particle of SWARM is ED1 and Distance from particle of POSITON to class-2 particle of SWARM is ED2. If it is a class-1 particle from SWARM and ED1 is minimum then we check class level of that particle of SWARM. If class level of that particle is 1 then it will be consider be a hit for class-1 else it will be miss for class1. If it is a class-2 particle from SWARM and ED2 is minimum then we check class level of that particle of SWARM. If class level of that particle is 2, then it will be consider be a hit for class-2 else it will be miss for class2. Based on number of hit for class1, number of miss for class1, number of hit for class2 and number of miss for class2 accuracy vector will be generated for each particle and will be stored in CLASS1_ACCURACY and CLASS2_ACCURACY. CLASS1_ACCURACY stores the percentage of belongingness of each particle toward class-1 data and CLASS2_ACCURACY stores percentage of belongingness of each particle toward class-2 dataset.

Algo CALCULATION_FITTNESS(POSITION,SWARM)

1. Repeat for each particle for POSITION
2. Repeat for each particle for SWARM
3. calculate euclidean distance (ED1) from Particle for POSITION to class1 particle of SWARM
4. calculate euclidean distance (ED2) from particle for POSITION to class2 particle of SWARM
5. if (ED1<ED2)
6. if (class level of particle is 1)

7. count a hit for class 1
8. else
9. count miss for class 1
10. if (ED2<ED1)
11. if (class level of particle is 2)
12. count a hit for class 2
13. else
14. count a miss for class 2
15. end of step-2
16. store accuracy of particle for class-1 in CLASS1_ACCURACY vector
17. store accuracy of particle for class-2 in CLASS2_ACCURACY vector
18. end of step-1
19. create a vector FV, which is the collection of accuracy for class1 particle from CLASS1_ACCURACY vector and collection of accuracy for class2 particle from CLASS2_ACCURACY vector
20. return (FV)

$$\text{CLASS1\_ACCURACY} = (nh1 / n)*100 \quad \textbf{(6)}$$

$$\text{CLASS2\_ACCURACY} = (nh2/ n)*100 \quad \textbf{(7)}$$

Here nh1 and nh2 is number of hit for class1 and class2 respectively. **n** is total number of instances (particles) of dataset.

Fitness vector FV will be created (figure-1). Local best particle will be chosen for each particle and initial local best vector (lbest) will be initial position POSITION. Particle having best fitness among class1 and class2 will be chosen as global best particle for class1 and class2 respectively known as gbest_cls1 and gbest_cls2. New velocities of class1 and class2 particles will be generated by using the equation-8 and equation-9 respectively.

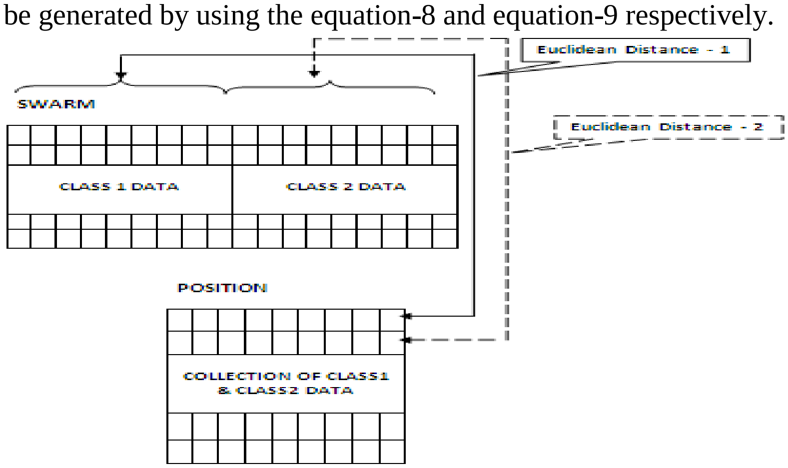


Figure-2: Calculation of closeness of each particle from training dataset POSITION to initial population SWARM (Collection of class1 and class2 dataset).

next_velocity(particle) =
$V(particle)+C_1*rand()*(lbest(particle)-POSITION(particle))+C_2*rand()*(gbest\_cls1-POSITION(particle))$ **(8)**

next_velocity(particle) =
V(particle)+C_1*rand()*(lbest(particle)-POSITION(particle))+C_2*rand()*(gbest_cls2-POSITION(particle)) **(9)**

Next_position(particle)=POSITION(particle)+next_velocity(particle))) **(10)**

New position will be generated by using equation-10.Position of all particles will be updated by comparing fitness values of particles of current position from POSITION dataset with fitness values of newly generated next position of particles from next_position dataset. Position of a particle having low fitness in current position in POSITION will be updated its position with the position having high fitness of next_position. These new particles positions in POSITION will be used in next iteration.

Algo CALCULATE_GBEST(FV,POSITION)

1. choose a particle whose fitness value is maximum among class-1 accuracies
 From FV,as global best for class-1 particles and set it to gbest_cls1
2. choose a particle whose fitness value is maximum among class-2 accuracies
 From FV,as global best for class-2 particles and set it to gbest_cls2
3. return (gbest_cls1 and gbest_cls2)

Procedure CALCULATE_GBEST() calculate global best particle of class1 and class2. gbest_cls1 and gbest_cls2 are global best particle of class1 and class2 respectively. Based on these global best particle next velocity will be generated. Procedure VELOCITY_NEXT() used to calculate next velocities of all particle.

Algo VELOCITY_NEXT(POSITION,gbest_cls1, gbest_cls2,lbest,V)

1.Repeat for each chosen class-1 particle of POSITION
2. next_velocity(particle) =
V(particle)+C_1*rand()*(lbest(particle)-POSITION(particle))+C_2*rand()*(gbest_cls1-POSITION(particle));
3. end of step-1
4. Repeat for each choosen class-2 particle of POSITION
5. next_velocity(particle) =
V(particle)+C_1*rand()*(lbest(particle)-POSITION(particle))+C_2*rand()*(gbest_cls2-POSITION(particle));
6. end of step-2
7. return (next_velocity)

Algo POSITION_NEXT(POSITION,next_velocity)
1. for each particle
2. next_position(particle) = POSITION(particle)+next_velocity(particle);
3. end of step-1
4. maintain class level indicator is next_position
5. return (next_positon)

Algo LBEST_NEXT(POSITION,SWARM)
1. FV1=CALCULATION_FITTNESS(POSITION,SWARM);
2. FV2=CALCULATION_FITTNESS(next_position,SWARM);
3. repeat step 4 to7 for each particle
4. if (FV1(particle of POSITION)<FV2(particle of next_position))

5. POSITION(particle) = next_position(particle);
6. else
7. keep current position of the particle unchanged
8. end of step-3
9. return (POSITION);

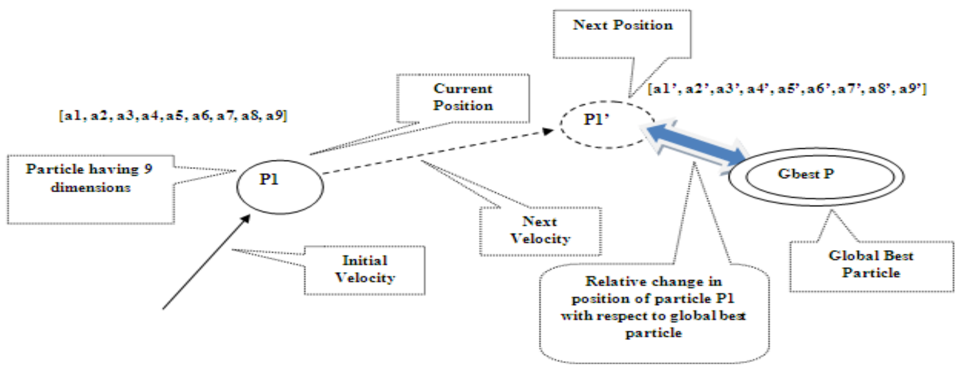


Figure-3: Change of position of particles with new velocity with respect to global best particle.

Algo GBEST_NEXT(POSITION,SWARM)
1. FV = CALCULATION_FITTNESS(POSITION,SWARM);
2. [gbest_cls1,gbest_cls2] = CALCULATE_GBEST(FV,POSITION);
3. return (gbest_cls1, gbest_cls2);

Next positions of all particles will be generated from current positions and next velocity. POSITION_NEXT () procedure is used to calculate next positions of all particles from current position POSITION and next velocity next_velocity. Again local best will be calculated by comparing fitness of current position POSITION and next position next_position. Based on these POSITION will be updated (figure-3). These processes will be repeated until stopping criteria.

3. Result Analysis and Experimentation

Training dataset consist of 200 instances (particles), 100 from class1 and 100 from class2. This dataset is used as training dataset. At training phase, we have done classification without using PSO, and observed that 95% and 68% of class1 and class2 instances has been well classified. 5% and 34% of class1 and class2 instances has been misclassified. Figure-5 shows output of classification without using PSO and figure-4 shows confusion matrix. We have set 80% of belongingness (accuracy) to decide a class of a particle (instance). That means 80% times of total comparison, if we get hit for that particular instance, then it will be counted as well classified, else miss classified.

	Class1	Class2
Class1	95	5
Class2	32	68

Figure-4: Confusion matrix contains information about actual and predicted classification of training dataset without using PSO.

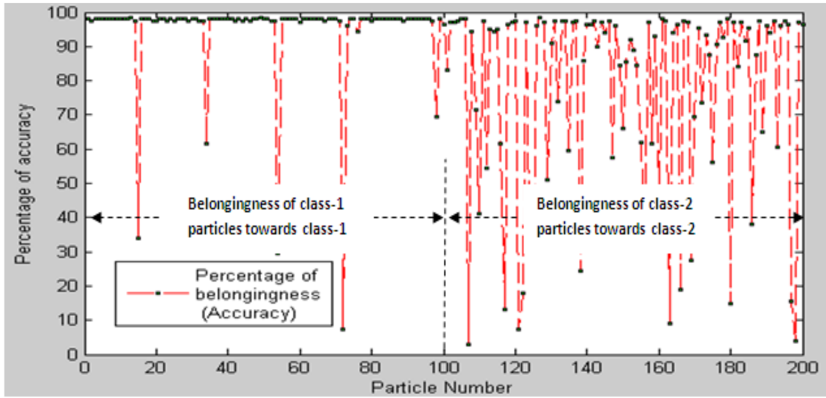


Figure-5: The percentage of accuracy of classification without using PSO for training dataset.

	Class1	Class2
Class1	100	0
Class2	5	95

Figure-6: Confusion Matrix contains information about actual and predicted classification of training dataset using PSO.

Table-3 shows initial fitness (Accuracy) of each particle (instances) which is the output of classification without using PSO. Here 5 and 32 numbers of instances (particles) of class1 and class2 have been misclassified and 95 and 68 numbers of instances has been classified properly. Here accuracy greater than 80 has been set for classification. Table-4 shows final accuracy of each instance after classification using PSO. We have observed that 100 and 95 number of instances of class1 and class2 respectively has been well classified. So 100% and 95% of accuracy of classification of class1 and class2 data is noted. 5% and 27% of enhancement on accuracy of classification have been observed using PSO based classification. Table-4 shows final accuracy of instances.figure-6 shows confusion matrix after classification using PSO based classification and figure-7 shows accuracy of all particles after using modified PSO classifier. Table-5 shows increased accuracy of missed classified instances of class1 and class2 after PSO based classification.

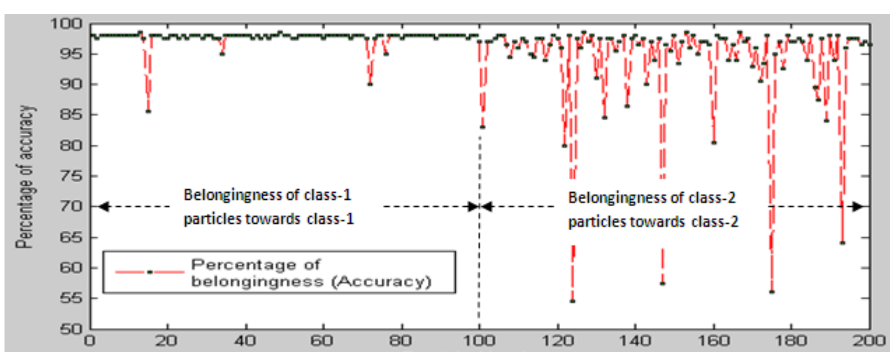


Figure-7: The percentage of accuracy of classification of training dataset using PSO.

Table-3: Initial belongingness (accuracy) of each instance (particles) of training dataset.

98	97.5	98	98	98	98	98	98	98	98
98	98	98.5	97.5	**34**	98	98	98	97.5	97.5
98	98	97.5	98	97.5	98	98	98	97.5	98
98	97.5	97.5	**61.5**	98	98	98	98	98	98
98	97.5	98	97.5	98	97.5	98	98	98.5	98
98	97.5	97.5	**29.5**	98	98	98	98	98	97
98	98	98	98	98	98	97.5	98	98	98
97.5	**7.5**	96	98	98	94.5	98	98	98	97.5
98	98	98	98	98	98	98	98	97.5	98
98	98	98	98	98	98	97.5	**69.5**	98	96.5
83	97	97	97.5	98	98	**3**	94.5	**71.5**	**41**
97.5	**54.5**	95	94.5	95	**61.5**	**13**	96.5	97	97.5
7.5	**18**	97	**39.5**	**31**	96	98.5	97	51	91
97.5	**74**	97.5	97.5	**59.5**	97	97.5	**24.5**	86	96.5
96.5	97.5	90	97	94	97.5	**57.5**	96	84.5	**66**
85.5	92	89	84.5	**62**	**31.5**	97	**61.5**	93	**47**
98	97.5	**9**	94	96.5	**19**	97.5	97	**27.5**	**69.5**
95.5	**73.5**	93.5	87.5	**56**	90.5	97	92.5	98	**15**
97	84	97	91.5	95.5	**38**	87.5	97.5	**65**	96
94	97.5	**60.5**	96	97.5	96.5	15.5	**4**	97	96.5

Table-4: Final belongingness (accuracy) of each instance (particles) of training dataset using PSO

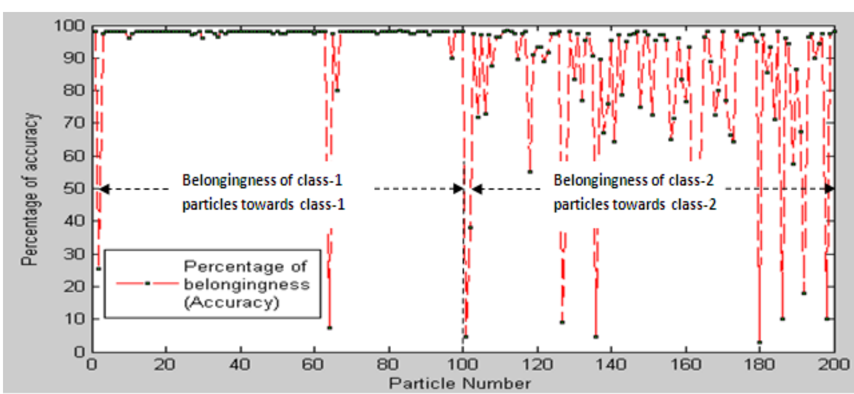
98	97.5	98	98	98	98	98	98	98	98
98	98	98.5	97.5	**85.5**	98	98	98	97.5	97.5
98	98	97.5	98	97.5	98	98	98	97.5	98
98	97.5	97.5	**95**	98	98	98	98	98	98
98	97.5	98	97.5	98	97.5	98	98	98.5	98
98	97.5	97.5	**98**	98	98	98	98	98	97.5
98	98	98	98	98	98	97.5	98	98	98
97.5	**90**	97.5	98	98	95	98	98	98	97.5
98	98	98	98	98	98	98	98	97.5	98
98	98	98	98	98	98	97.5	**98**	98	97
83	97	97	97.5	98	98	**96.5**	94.5	**97**	**96**
97.5	**97**	95	94.5	97.5	**97.5**	**94**	96.5	98	97.5
96	80	98	**54.5**	**97.5**	96	98.5	97	**98**	91
97.5	**84.5**	97.5	97.5	**95.5**	97	98	**86.5**	97.5	98
96.5	97.5	90	97	94	97.5	**57.5**	96.5	95.5	**98**
93.5	97	98.5	96	**98**	**95**	97	**97**	96.5	**80.5**
98	97.5	**97.5**	94	96.5	**94**	98.5	97	**97.5**	93
96	**90.5**	93.5	98	**56**	95	97	92.5	98	**97**
97	97.5	97	94	98	**89.5**	87.5	97.5	**84**	98
94	98	**64**	96	97.5	97.5	97.5	96.5	97	96.5

Like this we have taken a training dataset and we have done classification using our PSO based classification model. As a result, we got 97% and 67% of class1 and class2 data are well classified using non-PSO based classification. Figure-8 describes this statistic and figure-9 is the confusion matrix of non-PSO based classification. Figure-10 shows output of classification of testing dataset using our PSO based classification model. Using PSO based classification we have noted 100% and 92% of accuracy on classification of class1 and class2 data respectively using PSO based classification. Figure-10 shows these statistics and figure-11 shows confusion matrix for that classification.

Table-5: Potential change in belongingness (accuracy) of each instances(particles) of training dataset after using PSO

SL.NO.	P.NO.	I.A.	F.A.	STATUS
1	15	34	85.5	WC
2	34	61.5	95	WC
3	54	29.5	98	WC
4	72	7.5	90	WC
5	92	69.5	98	WC
6	107	3	96.5	WC
7	109	71.5	97	WC
8	110	41	96	WC
9	112	54.5	97	WC
10	116	61.5	97.5	WC
11	117	13	94	WC
12	121	7.5	96	WC
13	122	18	80	MC
14	124	39.5	54.5	MC
15	125	31	97.5	WC
16	129	51	98	WC
17	132	74	84.5	WC
18	135	59.5	95.5	WC
19	138	24.5	86.5	WC
20	147	57.5	57.5	MC
21	150	66	98	WC
22	155	62	98	WC
23	156	31.5	95	WC
24	158	61.5	97	WC
25	160	47	80.5	WC
26	163	9	97.5	WC
27	166	19	94	WC
28	169	27.5	97.5	WC
29	170	69.5	93	WC
30	172	73.5	90.5	WC
31	175	56	56	MC
32	180	15	97	WC
33	186	38	89.5	WC
34	189	65	84	WC
35	193	60.5	64	MC
36	197	15.5	97.5	WC
37	198	4	96.5	WC

In the above table-5 P.NO is the particle number, I.A is the initial accuracy, F.A is the final accuracy, WC stand for well classified and MC stand for miss classified.



Fgure-8: The percentage of accuracy of classification of testing dataset without using PSO.

	Class1	Class2
Class1	97	3
Class2	33	67

Figure-9: Confusion matrix contains information about actual and predicted classification of testing dataset without using PSO.

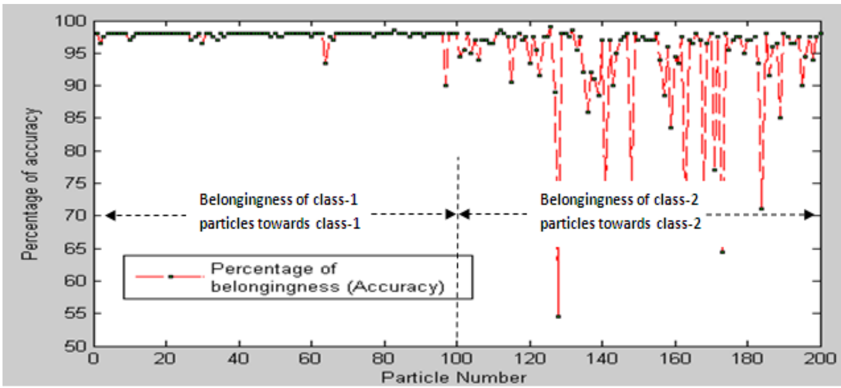


Figure-10: The percentage of accuracy of classification of testing dataset using PSO.

	Class1	Class2
Class1	100	0
Class2	8	92

Figure-11: Confusion matrix contains information about actual and predicted classification of testing dataset using PSO.

4. PSO Variants Verses Accuracy in Classification

The classification accuracy of instances of class1 and class2 varies according to different values of variants (C_1 and C_2) given in the figure-12. We observed that for the initial value C_1 and C_2 100% of classification accuracy of class2 data where as near about 97% of classification accuracy of class1 data is achieved. When the value of C_1 and C_2 tends 2,100% of classification accuracy of class1 data is noted. Where as near about 96% of classification accuracy of class2 data is observed. The values of the C_1 and C_2 will be chosen according to our requirement. If we want to achieve the 100% classification accuracy of class1, we need set the value of C_1 variant near about 2 and if we want to get the classification accuracy 100% for class2, we set the value of C_2 variant near about 1. Depending of the setting of the parameters C_1 and C_2 the performance of classification of PSO based classifier will be enhanced. In case of dataset having more than one class label, value of C_1 and C_2 will be set by looking the accuracy statistics from the output of PSO based classifier.Figure-12 shows variations of accuracy for different value of variants of PSO ie C_1 and C_2.Table-6 describes accuracy chat of modified PSO based classifier for different value of C_1 and C_2.

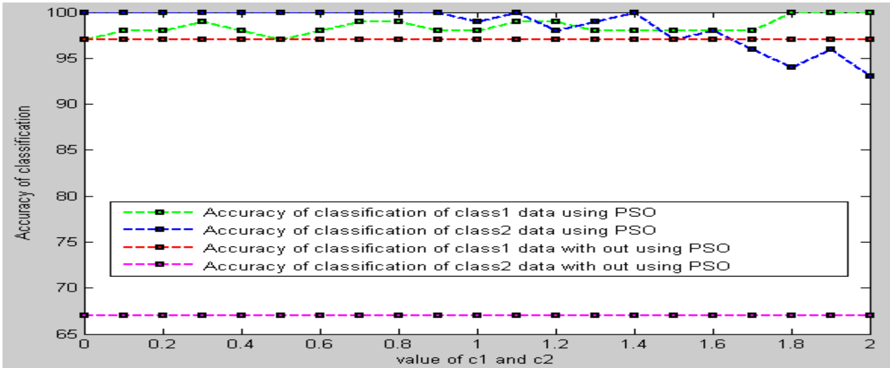


Figure-12: Variation of accuracy of classification for different values of C1 & C2.

Table-6: Variation of accuracy of classification for different values of C1 and C2 training dataset.

C1	C2	Accuracy of classification of class-1 data	Accuracy of classification of classs2 data
0	0	97%	100%
0.1	0.1	98%	100%
0.2	0.2	98%	100%
0.3	0.3	99%	100%
0.4	0.4	98%	100%
0.5	0.5	97%	100%
0.6	0.6	98%	100%
0.7	0.7	99%	100%
0.8	0.8	99%	100%
0.9	0.9	98%	100%
1.0	1.0	98%	99%
1.1	1.1	99%	100%
1.2	1.2	99%	98%
1.3	1.3	98%	99%
1.4	1.4	98%	100%
1.5	1.5	98%	97%
1.6	1.6	98%	98%
1.7	1.7	98%	96%
1.8	1.8	100%	94%
1.9	1.9	100%	96%
2.0	2.0	100%	93%

5. Conclusion

Purpose of this paper is to set the PSO parameters (C_1 and C_2) in the way such that PSO based classifier will give the best result for classification. Values of PSO variants will be set depending upon the context. We have developed statistic for PSO based classifier for classification of multidimensional based cancer dataset. Our purpose model and PSO variants setting mechanism can be helpful for classification of multidimensional dataset of various domains using PSO based classifier. PSO is a one of the most efficient optimization technique and performance of PSO varies depending upon PSO variants and parameters(C_1,C_2), but to achieve 100% accuracy in classification using PSO based classifier is uncertain and time of convergence is also uncertain. We have analyzed PSO based classifier using 100 number of iteration and different number of iteration effects the accuracy of classification. Although values PSO variants (C_1, C_2) have impact on accuracy of classification, still accuracy depends upon number of iterations. Our PSO based classification model can be used in various domain where classification of multidirectional real dataset (cancer data, diabetics data) is required. Also this model can be used in many KDD based application like feature selection and clustering.

6. Future Works

Further our PSO based classifier model can be used to identify noise instances and can be used to accelerate accuracy of classification by using proper values of PSO parameters and PSO variants. Our PSO based classification model can be used in different KDD based application(feature selection and clustering).

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proc.IEEE Int. Conf. Neural Networks, 1995, pp. 1942– 1948.

[2] D.E Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Reading, MA: Addison-Wesley, 1989.

[3] M. Clerc and J. Kennedy, "The Particle Swarm-Explosion, Stability, and Convergence in a multi-dimensional complex space", IEEE Trans.Evol.Comput, Vol.6, pp.58-73, Feb.2002.

[4] Ting-Yu Chen, Tzu-Ming chi, "On the improvements of the Particle Swarm Optimization Algorithm", Advances in Engineering software, vol-41, 2009.

[5] W.F Abd-EL-Wahed, A.A Mousa, M.A.EL-Shorbagy,"Integrating Particle Swarm Optimization With Genetic Algorithms For Solving Nonlinear Optimization problems", Journal Of Computational and Applied mathematics,2010.

[6] Alejandro Cervantes,In'es Galv'an, and Pedro Isasi, " An Adaptive Michigan Approach PSO for Nearest Prototype Classification", Spanish founded research MEC project PLINK::UC3M,Ref: TIN2005-08818-C04-02 and CAM project UC3M-TEC-05-029.

[7] S.Kannana, S. Mary Raja Slochanal,P.Subbaraj, Narayana Prasad Padhy ,"Application of Particle Swarm Optimization Technique and its Variants to Generation expansion Planning Problem", Electric Power System Research 70(2004)203-210.

[8] J.Liang, A.K Qin, Student Member , IEEE, Ponnuthurai nagaratnam suganthan ,Seniour Member,IEEE and S.Baskar, " Comprehensive Learning Particle Swarm Optimizerfor Global Optimization of Multimodal Functions",IEEE Transaction on Evolutionary Computation,Vol-10,No-3,June2006.

[9] Stefan Janson and Martin Middendorf, Member,IEEE, "A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant", IEEE Transactions on Systems, Man and Cybernetic-PartB: Cybernetics, Vol.35, No.6, DEC2005.

[10] Teruyoshi Yamaguchi and Keiichiro Yasuda, "Adaptive Particle Swarm Optimization Self-coordinating Mechanism with Updating Information ," IEEE Transactions on Systems, Man and Cybernetics, October 8-11,2006, Taipei,Taiwan.

[11] Serkan Kiranyaz , Jenni Pulkkinen, Moncef Gabbouj," Multi-dimensional particle swarm optimization in dynamic environments", Expert Systems with Applications 38 (2011) 2212–2223