DISEASE DIAGNOSIS FINDING SYSTEM USING DIVERGENCE ACCELERATED PARTICLE SWARM OPTIMIZATION IN BIG DATA

Ms. F. DARISH, Dr. K. MANIKANDAN
PG Scholar, Assistant Professor
Department of Computer Science and Engineering
Sri Krishna College of Engineering and Technology
15epcs005@skcet.ac.in, manikandank@skcet.ac.in

Abstract

Feature selection is popularly used to lighten the processing load in a data mining model. However, when it comes to mining over high dimensional data, the search space from which an optimal feature subset is derived grows exponentially in size, leading to an intractable manner. The feature selection is designed particularly for mining, streaming data on the fly, by using Accelerated Particle Swarm Optimization (APSO) type of swarm search that achieves enhanced analytical accuracy within reasonable processing time. This paper discusses about modifying APSO swarm search. We include a divergence concept to decrease the processing time and provide high accuracy. The difference between the two positions of global best and local best should be less than the divergence. If the difference is greater than the divergence we have to adjust that position. Finally, compare the performance results of the existing APSO based feature selection with the proposed modified APSO feature selection.

Keyword- Feature selection, APSO, swarm search optimization and divergence.

I. INTRODUCTION

One major challenge facing researchers work with big, data is high dimensionality, which occurs when a dataset has a large number of features (independent attributes). The PSO is a population based search algorithm based on the simulation of the social behavior of birds, bees or a school of fishes. PSO originally intends to graphically simulate the graceful and unpredictable choreography of a bird folk. Each individual within the swarm is represented by a vector in multidimensional search space. This vector has also one assigned vector which determines the next movement of the particle and is called the velocity vector. The PSO also determines how to update the velocity of a particle. Each particle updates its velocity based on current velocity and the best position it has explored so far; and also based on the global best position explored by swarm [2].

The main advantage of PSO is that it has less parameters to adjust. Other advantages are PSO does not have any complicated evolutionary operators such as crossover, mutation as in genetic algorithms. It has shortcomings too. PSO gives good results and accuracy for single objective optimization, but for multi-objective problem it stuck in local optima. Another problem in PSO is its nature to a fast and premature convergence in mid optimum points Several PSO variants have been developed to solve this problem [3].

The feature becomes minimal. By the principle of removing redundancy, the feature set may shrink to its most minimal size. The feature selection methods are custom designed for some particular classifier and optimizer [1]. We have investigated the efficiency of a new light-weight feature selection called Swarm Search with Accelerated Particle Swarm Optimization, with the aim of finding the right combination of classification algorithms and the lightweight feature selection algorithms for accurately data mining data streams on the fly.

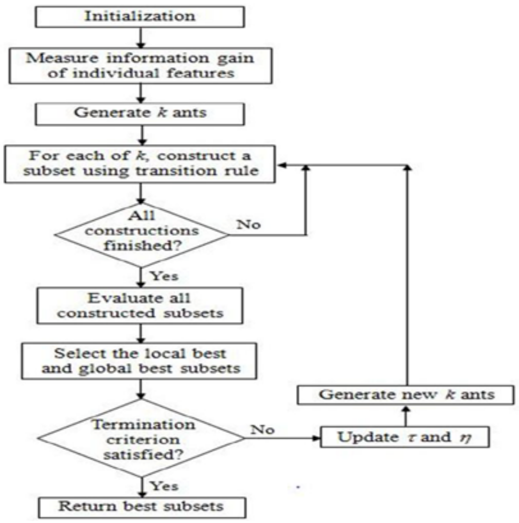

Initialization
Measure information gain of individual features
Generate k ants
For each of k, construct a subset using transition rule
All constructions finished?
No
Yes
Evaluate all constructed subsets
Select the local best and global best subsets
Generate new k ants
Termination criterion satisfied?
No
Update τ and η
Yes
Return best subsets

Fig.1: Ant Colony Optimization Towards Feature Selection

Fig.1 is described normal PSO swarm optimization process towards feature selection subsets are discussed in the above diagram. In this proposed work, the research methods are described with the incremental learning to be provided. In the existing approach divergence of the feature selection is not considered. So it can take more time to execute and less accuracy. To decrease the time complexity we include divergence concept with the existing APSO algorithm. In this approach the difference between the two positions should be less than the divergence. If the difference is greater than the divergence we have to adjust the positions. The advantage and disadvantages that occur in those methods also described [4].

II. RELATED WORK

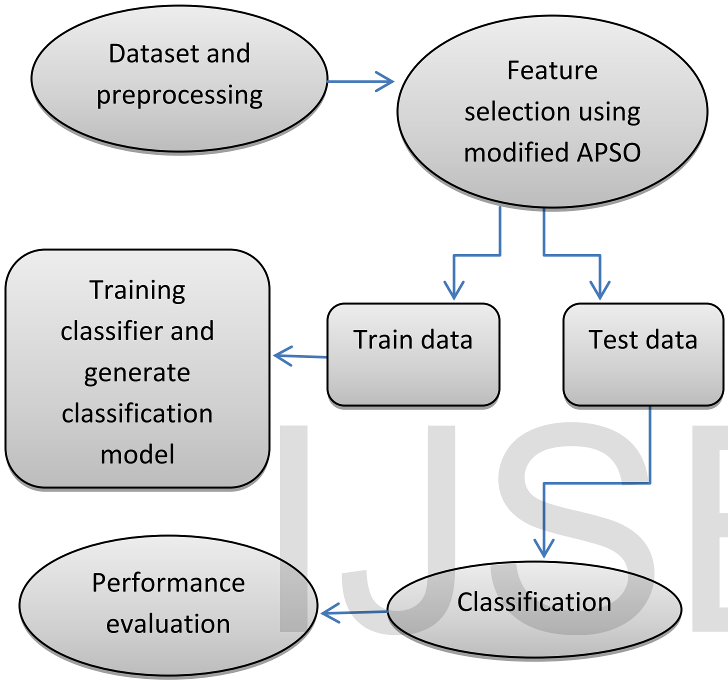
The standard particle swarm optimization uses both the current global best position g^* and the individual best position x_i^* [5]. The feature selection is the main concept of particle swarm search optimization and select the feature is to calculate the best fit of optimization. 5 representative data sets from various domains are downloaded from the UCI archive for experimentation. They are “arcene”, “dexter”, “Dorothea”, “gisette” and “Madelon”. The dataset “arcane” is used to train a classifier for distinguishing anomalous pattern of cancer from the normal patterns.

The “dexter” dataset is a large set of numbers which of each representing certain text words, commonly known as bag-of-word. The dataset “Dorothea” is the structural molecular features certain chemical compounds exist in a particular drug. The dataset “gisette” is used in training a classifier to recognize handwritten numbers and dataset “Madelon” is the vertices of a 5D hypercube, and they are randomly tagged with values of positive 1 or negative 1.

The decision tree construction, for example, heuristic function is an important evaluation method that determines the split attributes for converting leaves into nodes, for instance, information gain used in classification and regression tree algorithms [6] and Hoeffding Tree [7]. In incremental learning, Hoeffding bound (HB) is used to decide whether an attribute should be split to establish new nodes provided that sufficient samples for that attribute have appeared in the data stream. The new approach is designed for incremental decision trees, the pioneer of which is Very Fast Decision Tree (VFDT) and sometimes it is more generally called Hoeffding Tree (HT) [7]. HT is a classical work using HB in the node-splitting test. This is attributed to the statistical property of HB that controls the node-splitting error rate on the fly.

The three training datasets with equal number of examples using web data streams, and built three decision tree models using each training data set. For classifying the test or unseen examples: counts the weighted votes for each decision tree and assigns the class with the maximum weighted vote for that example [11]. The research issues should be addressed in order to realize robust systems that are capable of fulfilling the needs of data stream mining Applications [9].

III. PROPOSED ARCHITECTURE


Dataset and preprocessing
Feature selection using modified APSO
Training classifier and generate classification model
Train data
Test data
Performance evaluation
Classification

IJSER

Fig.2: Architecture of DAPSO

In above fig.2 architecture of the proposed system of data is 5 representative data sets from various domains are downloaded from the UCI archive for experimentation. They are “arcene”, “dexter”, “Dorothea”, “gisette” and “Madelon”. Data preprocessing is an important task in the data mining. The data from the real world entities may contain the missing values, inconsistent, incomplete or contain some errors. Each particle is attracted towards the position of the current global best g*and its own best location o_i^* in history called ‘individual best’ [4], while at the same time it has a tendency to move randomly.

The training data set with selected features are used to generate the classification model. And the testing dataset with selected features are used to classify the features. Classification is used to classify the features. The performance results are harvested in terms of Accuracy, Kappa (Kappa statistics), True Positives rate, False Positive rate, Precision, Recall, F-measure [1], Model building time per run, Pre-processing time, and number of features selected.

III PROPOSED ALGORITHM FOR DIVERGENCE ACCELERATED PARTICLE SWARM OPTIMIZATION

In this proposed work basic PSO algorithm includes the divergence formula and finally calculate the performance evaluation.

Algorithm:

Step1: Initialize all particles i with random positions x^0_i(initial position) in search space as well as random velocities v^0_i(velocity)

Step 2: Initialize the particle's best known position pb^0_i(local position) to its initial position.

Step 3: Calculate the initial swarm's best known position gb^0 (global position)

IJSER

```
Step 7: Update the particle's velocity: $v_i^{t+1}$= a*$v_i^t$+b*$r_p$*($pb^t$-$x_i^t$)+c*$r_g$*($gb^t$-$x_i^t$) ($v_i^t$ -velocity vector,(a,b,c)-parameters, $r_p$ -random local position, $r_g$ –random global position, $x_i^t$ –position vector)

Step 8: Compute the particle's new position: $x_i^{t+1}$= $x_i^t$+ $v_i^{t+1}$ ($x_i^{t+1}$ –adding new position vector, $v_i^{t+1}$ –calculate the new velocity vector)

Step 9: if (fitness($x_i^{t+1}$)< fitness($pb_i^t$)) then ($pb_i^t$ -initial position of local best)

    Step 9.1: Update the particle's best known position: $pb^{t+i}$ = $x_i^{t+1}$($pb^{t+i}$ –adding new local best)

Step 11: end if

Step 12: if (fitness($pb^{t+i}$ ) < fitness($gb^t$)) then

    Step 12.1: Update the swarm's best known position: $gb^{t+1}$= $pb_i^{t+1}$

Step 13: end if
```

```
NP - Population Size

D - Dimension of the problem

Step 16: if $x_j^t$ – $x_k^t$ > divg(P)

    Step 16.1: Update the swarm's best known position:

        $x^{t+1}$ = $x_i^t$ +rand( ($x^t_{min}$ - $x^t_{max}$))

Step 17: end if

Step 18: end for
```

The main goal of this APSO algorithm has included the divergence concept used to calculate the best fit. In this method the difference between the two positions should be less than the divergence. If the difference is greater than the divergence we have to adjust the positions [4]. Then update the best known position of the modified APSO algorithm.

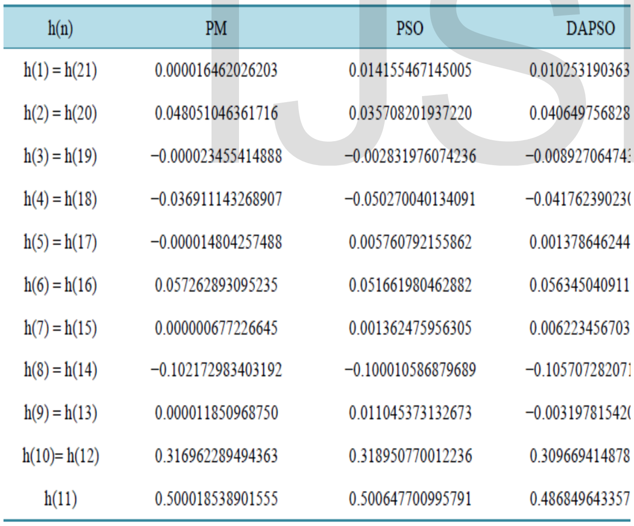
IV. EXPERIMENTAL EVALUATIONS

The convergence behavior of PSO and DAPSO respectively. These plots provide the error fitness value of the algorithms with the number of iterations. PSO converges to the minimum error fitness value of 0.9392 in 55.39 Sec. DAPSO converges to the minimum error fitness value of 1.129 in 35.92 Sec.

Table 1. PSO, DAPSO Parameters.

Parameter	PSO	DAPSO
Swarm Size	75	55
No. of Iteration	500	450
C1	2.05	2.05
C2	2.05	2.05
ω_{max}	0.95	0.95
ω_{min}	0.40	0.40
v_{max}	1	1
AC	-	Rand () * 0.5

Table 2. Optimized Coefficient of Low pass filter using Fitness 1.

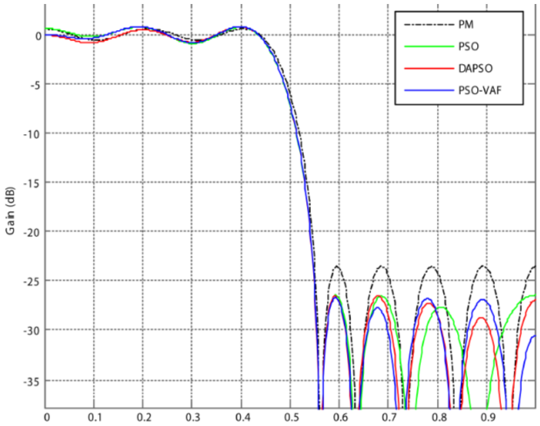


h(n)	PM	PSO	DAPSO
h(1) = h(21)	0.000016462026203	0.014155467145005	0.010253190363
h(2) = h(20)	0.048051046361716	0.035708201937220	0.040649756828
h(3) = h(19)	−0.000023455414888	−0.002831976074236	−0.00892706474
h(4) = h(18)	−0.036911143268907	−0.050270040134091	−0.04176239023
h(5) = h(17)	−0.000014804257488	0.005760792155862	0.001378646244
h(6) = h(16)	0.057262893095235	0.051661980462882	0.056345040911
h(7) = h(15)	0.000000677226645	0.001362475956305	0.006223456703
h(8) = h(14)	−0.102172983403192	−0.100010586879689	−0.10570728207
h(9) = h(13)	0.000011850968750	0.011045373132673	−0.00319781542
h(10)= h(12)	0.316962289494363	0.318950770012236	0.309669414878
h(11)	0.500018538901555	0.500647700995791	0.486849643357

Table 1 and 2 parameters of PSO and DAPSO are existing method using h(n) is position of the best one and the DAPSO is higher than other fitness value.

V. GRAPHICAL ANALYSIS

1. Gain vs. Normalized frequency



Normalized frequency

Fig.3: Frequency Response of Lowpass filter using Fitness 1

In this above fig.3 is the normalized frequency of DAPSO are shows the gain of the PSO and DAPSO. It is more accurate normalized frequency than the PSO algorithm is calculated the best fitness value than the other algorithms.

2. Accuracy

The exact positive and the negatives total is described as the accuracy and it partitioned by the total number of classification attributes (Tp+Tn+Fp+Fn)

Accuracy=Tp +Tn ⁄ Tp+Tn+Fp+Fn …(1)

Where, Tp- True Positive, Tn- True Negative, Fp- False Positive, Fn- False negative.

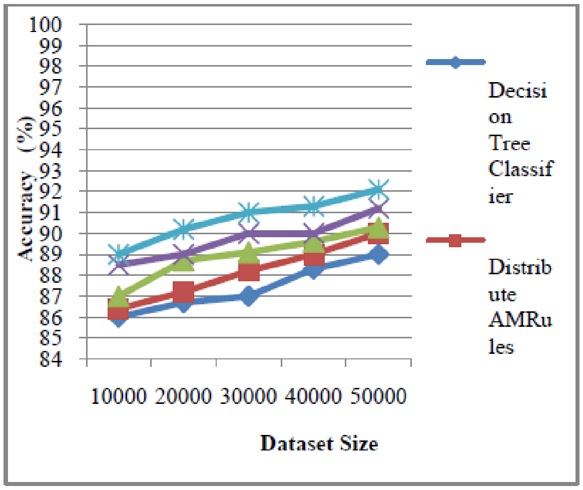


Fig.4: Accuracy Comparison

The above fig.4 shows the size of dataset is taken as X axis and in y axis accuracy is taken. For dataset size 50000, Decision Tree Classifiers, Distribute

IJSER

AMRules, SHARP method, PSO-FCM and APSO achieves accuracy result of 89%, 90%, 90.3 %, 91.2 % and 92.1%. Finally, the APSO approach reaches the high accuracy of the entire size of the data set.

3. Precision

The proportion of exact positives in opposition to both the exact positive and inaccurate positives results for intrusion and the real characteristics is described as Precision. It is described as follows

$$Precision = Tp/Tp+Fp \ldots(2)$$

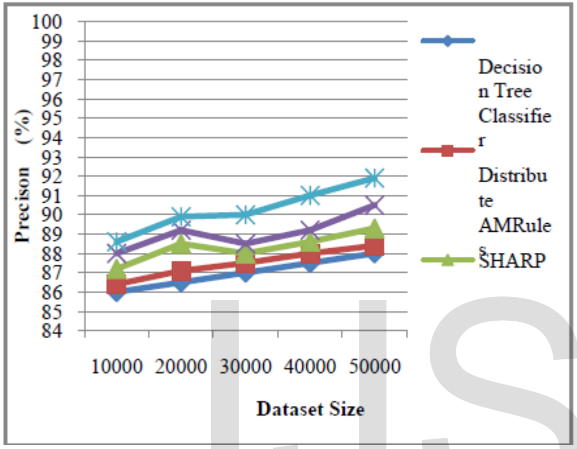


Fig.5: Precision Comparision

IJSER

The above fig.5 shows an X axis the size of the data set is represented and the precision is represented in the Y axis. For dataset size is 50000 of Decision Tree Classifiers, Distribute AM Rules, SHARP method, PSO-FCM and proposed APSO accomplishes a precision outcome of 88%, 88.4%, 89.3 %, 90.5 % and 91.9 % correspondingly. The graph it has been find out the APSO methodology outperforms than that of the other designs and results in precision values.

4. Recall

It measures the proportion of positives that are correctly identified

$$Recall = Tp/Tp+Fn \ldots(3)$$

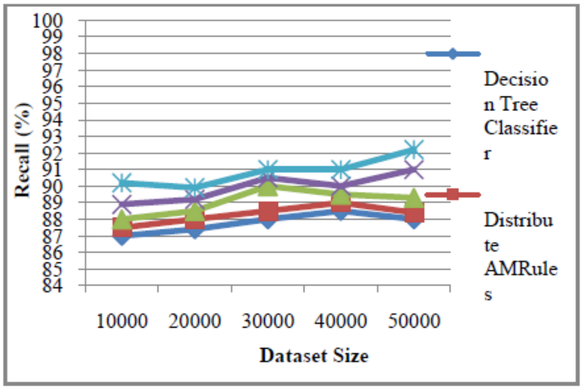


Fig.6: Recall Comparison

The recall is demonstrated in figure 6. For dataset size is 50000 of Decision Tree Classifiers, Distribute AM Rules, SHARP method, PSO-FCM and proposed APSO accomplishes a recall outcome of 88%, 88.4%, 89.3 %, 91 % and 92.2 % correspondingly. Finally, that the APSO methodology has demonstrated the high recall value for the entire size of the data set.

4. No. of iteration vs. Error fitness value

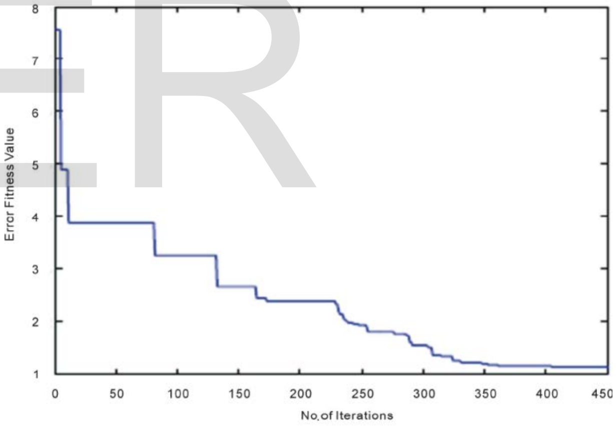
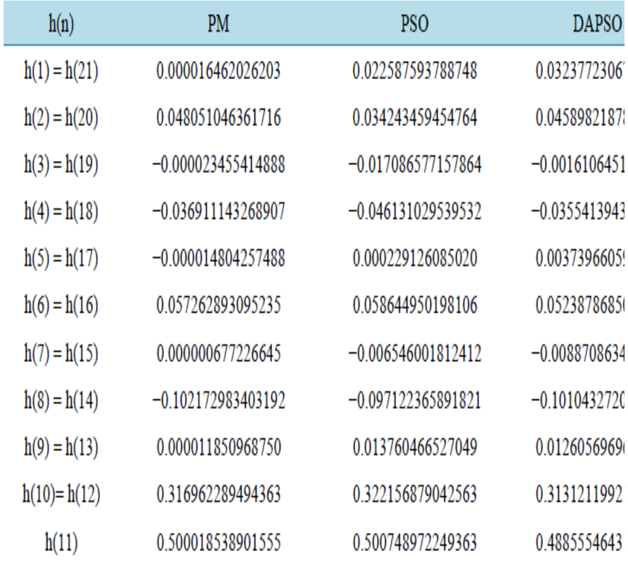


Fig.7: Convergence Plot For DAPSO Using Fitness 2

In the above fig.7 shows the convergence plot of DAPSO are error fitness value to be increased and the number of iterations to be decreased in the plot is a divergence concept to be include than the iterations to be lower than other algorithms.

Table 3: Optimized Coefficient of LP FIR using Fitness 2.



h(n)	PM	PSO	DAPSO
h(1) = h(21)	0.000016462026203	0.022587593788748	0.0323772306
h(2) = h(20)	0.048051046361716	0.034243459454764	0.0458982187
h(3) = h(19)	−0.000023455414888	−0.017086577157864	−0.0016106451
h(4) = h(18)	−0.036911143268907	−0.046131029539532	−0.0355413943
h(5) = h(17)	−0.000014804257488	0.000229126085020	0.0037396605
h(6) = h(16)	0.057262893095235	0.058644950198106	0.0523878685
h(7) = h(15)	0.000000677226645	−0.006546001812412	−0.0088708634
h(8) = h(14)	−0.102172983403192	−0.097122365891821	−0.1010432720
h(9) = h(13)	0.000011850968750	0.013760466527049	0.0126056969
h(10)= h(12)	0.316962289494363	0.322156879042563	0.3131211992
h(11)	0.500018538901555	0.500748972249363	0.4885554643

The above table is calculated the fitness value is higher accuracy than fitness 1 value. In the number of iterations to be calculated on the best fitness value in the optimized coefficient of LP FIR using fitness 2.

5. Overall accuracy, precision and recall comparison

The new APSO methodology, the exact result is acquired for the entire data set size like 10000, 20000,30000,40000,50000 is 90.72%, which is 3.1%,2.1%, 1.8% and 0.9 % higher than existing Decision Tree Classifiers, Distribute AM Rules, SHARP method, PSO-FCM approaches respectively. The proposed APSO approach achieves precision as 90.28%, which is 3.9%, 3.5 %, 2.6 %, and 1.4 % higher than existing Decision Tree Classifiers, Distribute AM Rules, SHARP method, PSO-FCM approaches.

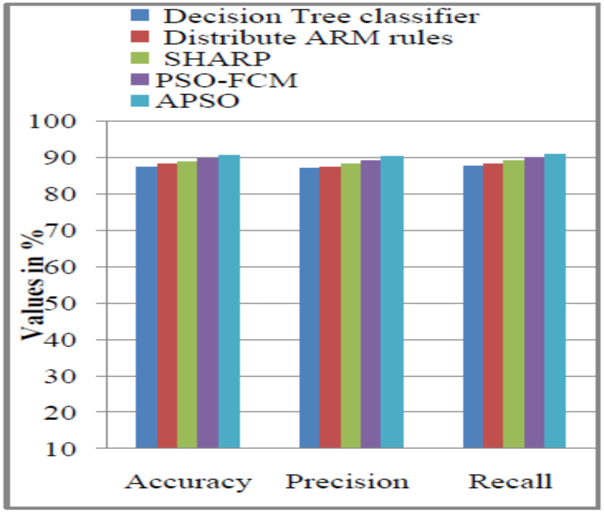


Fig.8: Performance Comparison

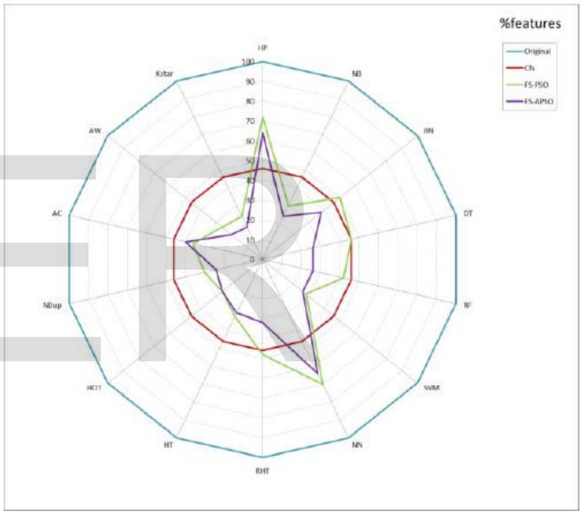


Fig.9: Performance of Feature Selected.

The above fig.8 and fig.9 are showing an APSO approach achieves recall as 90.88%, which is 4.2%, 3.8 %, 2.9%, and 1.2% higher than existing Decision Tree Classifiers, Distribute AM Rules, SHARP method, PSO-FCM approaches respectively. Finally, the APSO methodology is efficiently mining the streaming data from this graph.

V CONCLUSION

Big Data grows continuously with fresh data are being generated at all times; hence it requires an incremental

IJSER

computational approach which is able to monitor large scale of data dynamically. In this paper, we calculated the possibility of using a group of incremental classification algorithm for classifying the collected data streams pertaining to Big Data. In particular the feature selection is designed particularly for mining, streaming data on the fly, by using accelerated particle swarm optimization (APSO) type of swarm search. In this approach divergence of the population is not considered. So it can take more time to execute. To decrease the time, complexity and improve the accuracy. So, we include divergence concept with the existing APSO algorithm. In this approach the difference between the two positions are global best and local best. It should be less than the divergence. If the difference is greater than the divergence we have to adjust the two positions.

VI. REFERENCES

[1] Simon Fong, Raymond Wong, and Athanasios v. Vasilakos, senior member, IEEE "Accelerated PSO swarm search feature selection for data stream mining big data", IEEE transactions on services computing volume: 9, issue: 1, January 2016.

[2] Amreen Khan1, Prof. Dr. N.G.Bawane, Prof. Sonali Bodkhe, "An Analysis of Particle Swarm Optimization with Data Clustering-Technique for Optimization in Data Mining" International Journal on Computer Science and Engineering Vol. 02, No. 07, 2010, 2223-2226.

[3] Sunita Sarkar, Arindam Roy, Bipul Shyam Purkayastha "Application of Particle Swarm Optimization in Data Clustering: A Survey" International Journal of Computer Applications (0975 – 8887) Volume 65– No.25, March 2013

[4] S. Sudha · S. Busker · S. Miruna Joe Amali · S. Krishnaswamy "Protein structure prediction using diversity controlled self-adaptive differential evolution with local search", Springer-Verlag Berlin Heidelberg 2014.

[5] S. Meera, B. Rosiline Jeetha "Survey on Swarm Search Feature Selection for Big Data Stream Mining" International Journal of Computer Applications (0975 – 8887) Volume 158 – No 1, January 2017.

[6] Quinlan, J. R., C4.5: Programs for Machine Learning. Morgan Kauf-Mann Publishers, 1993.

[7] Domingos P., and Hulten G. 2000. "Mining high-speed data streams", in Proc. of 6th ACM SIGKDD international conference on Knowledge dis-covery and data mining (KDD'00), ACM, New York, NY, USA, pp. 71-80.

[8] Mudit Shukla, G. R. Mishra "DAPSO and PSO-VAF in Linear Phase Digital Low Pass FIR Filter Design", Published Online March 2014 in Sci Res. http://www.scirp.org/journal/cs,http://dx.doi.org/10.4236/cs.2014.53008

[9] Mohamed Medhat Gaber, Arkady Zaslavsky, Shonali Krishnaswamy, "Mining data streams: a review", ACM SIGMOD Record, Volume 34 Issue 2, June 2005, pp.18-26.

[10] Wei Fan, Albert Bifet, "Mining Big Data: Current Status, and Forecast to the Future", SIGKDD Explorations, Volume 14, Issue 2, pp.1-5.

[11] Fauzia Yasmeen Tani, Dewan Md. Farid, Mohammad Zahidur Rahman " Ensemble of Decision Tree Classifiers for Mining Web Data Streams", International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 1– No.2, January 2012.

IJSER

IJSER