

Improvised Gene Selection Using Particle Swarm Optimization With Decision Tree As Classifier

Pranav Teja Garikapati , Naveen Kumar Penki, Sashank Gogineni

Abstract— Innovation has spread its foundations profound into the lives of a cutting-edge man, and the essential perspective to get the life going, health care is not an exemption. In spite of the fact that numerous researchers are contributing for inquiring about in the field of Bio-Technology, not many contributed for identification of cancer by selecting prominent genes from a microarray gene data set. Here, we intend to perform gene selection using Particle Swarm Optimization in a novel method, by combining it with other existing classification algorithm to arrive at a better accuracy in cancer identification, as compared to the already existing one. In this study, we implemented a novel method for medical problem, it is the integration of particle swarm optimization (PSO) and decision tree (C4.5) named PSO + C4.5 algorithm. To evaluate the effectiveness of PSO + C4.5 algorithm, it is implemented on cancer data set of life sciences obtained from UCI machine learning databases. The fitness value is calculated using C4.5 and this improves the PSO algorithm's efficiency in obtaining the characteristic genes.

Index Terms— Classification, Microarray gene data, PSO, Decision Tree C4.5.

I. INTRODUCTION

SWARM INTELLIGENCE:

Swarm Intelligence (SI) is the collective behavior of decentralized, self-organized systems, natural or artificial. SI systems are typically made up of a population of simple agents or bodies interacting locally with one another and with their environment. The inspiration often comes from nature, especially biological systems. The agents follow very simple rules and although there is no centralized control structure dictating how individual agents should behave local, and to a certain degree random, interactions between such agents lead to the behavior of intelligence global behavior, unknown to the individual agents. Natural examples SI includes ant colonies, bird flocking, animal herding, bacterial growth and fish schooling.

‘Swarm prediction’ has been used in the context of forecasting problems. Swarm intelligence is a heuristic based on the natural analogy. There are many algorithms under swarm intelligence some of the famous ones are Ant colony optimization, artificial bee colony algorithm, charged system search, Firefly algorithm, Gravitational search algorithm, Intelligent water drops, Multi swarm optimization, Particle swarm optimization, River formation dynamics and so on.

From above mentioned algorithms, PSO represents a general form of the swarm intelligence algorithm. Each of the other

algorithm are based on a single natural analogy of a naturally formed swarm culture. Each of the models are suitable for a particular type of application and give best results when the problem space is expressed as a function of the natural search ability of the swarm.

PARTICLE SWARM OPTIMIZATION (PSO):

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. PSO is based on the movement of intelligent swarms. The swarms consists of agents moving in the solution space directed by their own experience as the experience of the swarm. In all the swarm tries to achieve an objective as a whole without any centralized control.

In computer science, Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions and moving these particles around in the search space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement positions in the search space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found. PSO can nevertheless be used an optimization problems that are partially irregular, noisy, change over time, etc.

BIOLOGICAL ANALOGY:

As stated before, PSO simulates the behavior of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what's the best strategy to find the food? The effective one is to follow the bird which is nearest to the food. PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a “bird” in the search space. We call it “particle”. All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particle fly through the problem space by following the current optimum particles.

Pranav Teja Garikapati, University of Texas, Arlington, USA
Naveen Kumar Penki, University of Oklahoma, Norman, USA
Sashank Gogineni, Georgia State University, Atlanta, USA

There are several algorithms developed to identify the cancer affected genes in the given data set, out of which the well-known classifier algorithms are Support Vector Machine (SVM), self-organizing map (SOM), and C4.5, Back Propagation Neural Network (BPNN), SVM, Naive Bayes (NB), CART decision tree, and Artificial Immune Recognition System (AIRS), genetic algorithm, Gene Mining.

II. LITERATURE REVIEW

Xia Li, Shaoqi Rao et.al [1] presented a decision approach, which can efficiently perform multiple gene mining tasks. An application of this approach to analyse two publicly available data sets (colon data and leukaemia data) identified 20 highly significant colon cancer genes and 23 highly significant molecular signatures for refining the acute leukaemia phenotype, most of which have been verified either by biological experiments or by alternative analysis approaches.

Isabelle Guyon et.al [2] addressed the problem of selection of a small subset of genes from broad patterns of gene expression data, recorded on DNA micro-arrays. Using available training examples from cancer and normal patients, it built a classifier suitable for genetic diagnosis, as well as drug discovery. It proposed a new method of gene selection utilizing Support Vector Machine methods based on Recursive Feature Elimination (RFE). It demonstrate experimentally that the genes selected by its techniques yield better classification performance and are biologically relevant to cancer.

Shutao Li et.al [3] presented a hybrid Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) method is used for gene selection, and Support Vector Machine (SVM) is adopted as the classifier. The proposed approach is tested on three benchmark gene expression datasets: Leukaemia, Colon and breast cancer data. Experimental results show that the proposed method can reduce the dimensionality of the dataset, and confirm the most informative gene subset and improve classification accuracy.

Yang Su1 et.al [4] used Rank Gene program for analysing gene expression data and computing diagnostic genes based on their predictive power in distinguishing between different types of samples. This program integrates into one system a variety of popular ranking criteria, ranging from the traditional t-statistic to one-dimensional support vector machines.

AzadehMohammadi et.al [5] have combined the Fisher method and SVMRFE to utilize the advantages of a filtering method as well as an embedded method. Furthermore, they have added a redundancy reduction stage to address the weakness of the Fisher method and SVMRFE. In addition to gene expression values, the proposed method uses Gene Ontology which is a reliable source of information on genes.

Zhi Yan et.al [6] analysed the microRNA (miRNA) expression pattern in gastric cancer with and without recurrence and obtained 17 differentially expressed miRNAs with potential to predict recurrence risk for GC patients. Three different miRNA target gene databases (miRanda, TargetScan and PicTar) were used for searching the potential genes regulated by miRNAs. A combination was performed between miRNA target genes and recurrence-related gene expression profiling. Three bioinformatics algorithms (PAM, SVM and RF) were used to feature recurrence-related gene selection.

Kun-Huang Chen1 et.al [7] developed a novel method utilizing particle swarm optimization combined with a decision tree as the classifier. This study also compares the performance of our proposed method with other well-known benchmark classification methods (support vector machine, self-organizing map, back propagation neural network, C4.5 decision tree, Naive Bayes, CART decision tree, and artificial immune recognition system) and conducts experiments on 11 gene expression cancer datasets.

III. METHODOLOGY

PSO applies the concept of social interaction to problem solving. It uses a number of agents (particles) that constitutes a swarm moving around in the search space looking for the best solution. Each particle is treated as a point in a N-dimensional space which adjusts its movement. According to its own flying exercise (Pbest – personal best) as well as the flying experience of other particles (Gbest - Globalbest). The basic concept of PSO lies in accelerating each particle towards its Pbest and Gbest locations with regard to a random weighted acceleration at each time. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. (The fitness is also stored.) This is called pbest. Another “best” value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called lbest. When a particle takes all the population as its topological neighbors, the best value is a global best and is called gbest. The particle swarm optimization concept consists of, at each time step, changing the velocity of each particle toward its pbest and lbest locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and lbest locations. The modification of the particle’s positions can be mathematically modelled by making use of the following equations:

$$V_{k+1} = wV_k + C_1 \text{rand}_1(P_{\text{best}} - S_k) + C_2 \text{rand}_2(G_{\text{best}} - S_k)$$

$$S_{k+1} = S_k + V_{k+1}$$

Where, S_k is current search point; S_{k+1} is the modified search point; V_k is the current velocity; V_{k+1} is the modified velocity; P_{best} is the velocity based on Pbest; G_{best} = velocity based on Gbest; w is the weighing function; C_j is the weighing factors; rand_j are uniformly distributed random numbers between 0 and 1. In the particle swarm optimization technique, the particles searches the solution in the solution

space within the range [-s,s]. The diagram below illustrates the movement of particles.

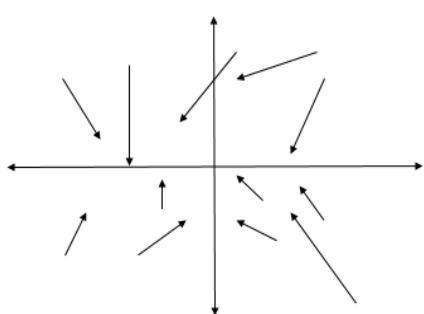


Fig1: Movement of particles in PSO Algorithm

PSEUDO CODE:

The pseudo code describing above methodology is given below:

```

For each particle
    Initialize particle
END
Do
    For each particle
        Calculate fitness value
        If the fitness value is better than the best fitness value
        (pbest) in history
            Set current value as the new pBest
        END
    Choose the particle with the best fitness value of all the
    particles as the gBest
    For each particle
        Calculate particle velocity according equation (a)
        Update particle position according equation (b)
    END

```

While maximum iterations or minimum error criteria is not attained.

Objective Function: Measure of model accuracy

The objective function determines the way the PSO heuristic is modified and also the way the accuracy of the particular model is ascertained. The objective function define the way in which the PSO swarms train the solutions to reach a better solution. There are three objective functions used here:

Total absolute error: It is the absolute value of the difference between the obtained value and the expected value. The value obtained after model function evaluation and normalization is compared with the actual 0 or 1 value.

$$TAE = \sigma(f(x) - (1 \text{ or } 0))$$

Q3 Accuracy: This defines the accuracy of the system in terms of the correct residues predicted of each type.

$$Q3 = (Hc + Cc + Sc) / \text{Total number of residues}$$

Where Hc, Sc, and Cc are the number of correctly predicted helixes, coils and sheets.

Correlation Coefficient: Correlation coefficient is a measure that takes into account not only correct hits but also incorrectly predicted values. It provides a comprehensive way of determining the accuracy of the model. The formula for getting the correlation coefficient is as shown below:

$$C_x = ((P_x N_x) - (N_x^f P_x^f)) / ((N_x + N_x^f)(N_x + P_x^f)(P_x + N_x^f)(P_x + P_x^f))^{1/2}$$

X= {coil, helix, sheet}

Pfx= correctly predicted positives of X

Nx= correctly predicted negatives of X

Pfx= incorrect positive

Nfx= incorrect negative

Perfect prediction gives C(x) =1

Fully imperfect gives C(x) =-1

PSO: Learning and Testing

PSO used for testing and learning, has to be adapted to suit the problem at hand. The formula for PSO depends upon the each parameter that needs to be optimized. The particles are updated by the formula given below:

$$Vaik+1 = wVaik + c1 \text{ rand1} (Pbest - Sik) + c2 \text{ rand2} (Gbest - Sik)$$

$$Saik+1 = Saik + Vaik+1$$

Here, the formula are shown for a single parameter. It is similarly done for other parameters.

A few issues before considering PSO implementations are:

Thresholding and Normalization

The value obtained after using the model function needs to be threshold to obtain a value suitable for comparison.

In the PSO process, the threshold for a successful prediction is taken to be as (NOD-1)

$$\begin{aligned} \text{Normalized value} &= 1, \text{ if } f(x) > (\text{NOD-1}) \\ &0, \text{ if } f(x) < (\text{NOD-1}) \end{aligned}$$

Learning:

Learning is carried out by finding all the suitable parameters from a_1 to a_n , which represent the most optimal values as given by the objective function. The ultimate process leads to finding the parameters of the model function.

Assigning of Secondary Structure in Testing:

The secondary structure of a residue is assigned based on the value of the model functions obtained as well as the rules used for evaluating the effectiveness of the solution to each case.

The flow chart following depicts the sequence of activities that are carried out in the process of determining a possible secondary structure for the test residue.

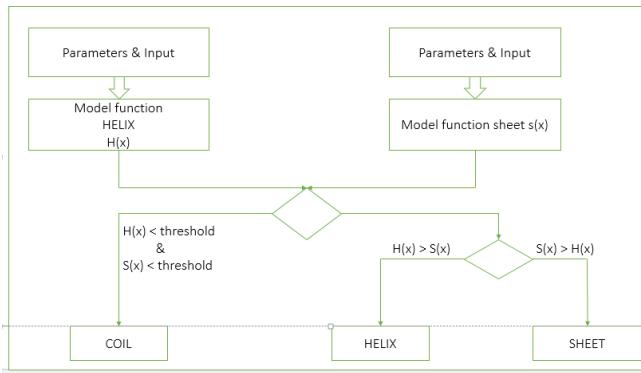


Fig2. Selection of secondary structure in testing phase

Post processing and smoothening:

After the values are obtained from the model function and the secondary structure assignment is made, the sequence has to be smoothened in order to remove isolated values and also for having smooth running data sequence.

Following steps are needed:

Smoothening: In the smoothening process, the values are averaged over a long distance to spread their characteristics in order to have a smooth transition of values. The following formulae are useful for averaging of helix and sheet values:
 $hi = (hi-1+hi+hi+1)/3$ and $ei = (ei-1+ei+ei+1)/3$

From the above equations, h and e represents the adjacent helix and sheet region. This causes the output to be more smooth and uniform at the cost of reduced accuracy.

Rule based Clash Resolving System:

In the rule based clash resolving system, the final smoothed sequence is evaluated to remove isolated values as per the rules of the DSSP code. The following rules were generated for finding the final sequence of secondary structure.

- Rule 1: If Centre is helix followed by two non-helix then convert to coil
- Rule 2: If Centre is helix and next helix, followed by peripheral non helix, convert both i and i+1 to coil
- Rule 3: If i is S and i+1, i-1 are non-sheet structures, convert i to coil.
- Rule 4: If none of the above rules suffice, leave the structure as it is.

Presentation of Output results:

The final output represents the secondary structure of the proteins converted back from the numerical values to the letter code. The accuracy of the system is also given as the output. Accuracy is given as the Q3 measure as follows:

$$Q3 = (Hc + Cc + Sc) / \text{Total number of residues}$$

Where Hc, Sc and Cc are the number of correctly predicted Helices, Coils and sheets. The output is in the form of the predicted secondary structure of the sequence.

Parameters and Configurations:

The following are the typical configuration of swarms for optimization problems:

1. The number of particles: The typical range is 20-40. Actually for most of the problems 10 particles is large enough to get good results. For some difficult or special problems one can try 100 or 200 particles as well.
2. Dimension of particles: It is determined by the problem to be optimized and parameters.
3. Range of particles: It is also determined by the problem to be optimized, you can specify different ranges for different dimension of particles.
4. Vmax: It determines the maximum change one particle can take during one iteration. Usually we set the range of particles as Vmax.
5. Learning factors: c1 and c2 usually equal to 2. However other settings were also used in other papers, but usually c1 equals to c2 and ranges from [0,4].
6. The stop condition: The maximum number of iterations the PSO execute and the minimum error requirement. Usually it is iterated till we get stable parameter values.
7. Global version vs local version: We introduced two versions of PSO, global and local version. Global version is faster but might coverage to local optimum for some problems. Local version is a little bit slower but not easy to be trapped into local optimum. One can use global version to get quick result and use local version to refine the search.
8. Inertia weight: The w value is usually taken to be 0.5.

The main considerations for implementing the PSO are finding and defining the following parts of the program:

1. Methodology for defining the update of velocities.
2. Defining the coordinate system and bounds of the problem.
3. Defining the objective function and fitness function.
4. Defining the model function and output criterion.

C4.5 Algorithm:

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = \{s_1, s_2, \dots\}$ of already classified samples. Each sample s_i consists of a p-dimensional vector $(x_{1,i}, x_{2,i}, \dots, x_{p,i})$, where the x_j represent attributes or features of the sample, as well as the class in which s_i falls.

At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs on the smaller sub lists.

This algorithm has a few base cases:

1. All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
2. None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.

3. Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

Pseudo code

In pseudo code, the general algorithm for building decision trees is:

Check for base cases

For each attribute a

 Find the normalized information gain ratio from splitting on a

 Let a_{best} be the attribute with the highest normalized information gain

 Create a decision node that splits on a_{best}

 Recur on the subsists obtained by splitting on a_{best}, and add those nodes as children of node

Mathematical expression for calculating the fitness values of Particles:

Entropy H(S) is a measure of the amount of uncertainty in the (data) set S (i.e. entropy characterizes the (data) set S)

$$H(S) = - \sum_{x \in X} p(x) \log_2 p(x)$$

Where, S is the current (data) set for which entropy is being calculated (changes every iteration of the ID3 algorithm)

X is set of classes in S

P(x) is the proportion of the number of elements in class x to the number of elements in set S

When H(S) = 0, then the set S is perfectly classified (i.e. all elements in S are of the same class)

IV. IMPLEMENTATION

The algorithm used for selection of cancer genes for identification of cancer is Particle Swarm Optimization, in which the fitness value is evaluated by using the C4.5 Decision Tree. This combination has been found optimum for selection of cancer genes among other techniques already implemented.

The data sets used for testing are obtained from world renowned repository for any type of data records, UCI repository. The data sets are nothing but microarray data that are obtained by analyzing a blood sample and digitizing the patterns in blood. The microarray data set used in this thesis consists of a user id, a class label, and various other entities like the clump, size, epithelial cells etc, that are present in the blood of a human being. Now the job at hand is to analyze that data set and find out what records have characteristics close to the disease causing ones.

Initially, the ID number and class label are removed and the remaining data is given as input to the algorithm. Input parameters like the target, Number of inputs, maximum number of particles, and number of maximum iterations that can be taken to produce the output are given. The algorithm takes these conditions as input, and analyzes the data and

calculates the pbest value for each and every particle and at the end of each iteration, comparison is made with the target value, so as to decide what particles are close to the target.

This processes is backed up by the C4.5 Fitness function, where the entropy value calculated provides the algorithm with a probabilistic value so as to choose what particle, for the remaining particles to follow. Every iteration also updates the velocity of the particles, the velocity specifies to choose a random number, leaving behind a finite number of numbers behind. As the velocity keeps on updating, the efficiency of the algorithm further improves, and the target is reached in a much lesser time period. After the target is reached, the records that have a value closest to the target are returned as output. The people with these records are having a higher chance of getting the disease, basing on the history of patients.

PSO algorithm with C4.5 as classifier:

Input

$c_1, c_2, r_1, r_2, w, v_{min}, v_{max}$

Output

GB

The fitness value of GB

begin

 Initialize Population

while (max iteration or convergence criteria is not met) do

for i = 1 to numbers of particles

 Evaluate fitness value of the particle by C4.5

if the fitness value of X_i is greater than that of PB_i
 then $PB_i = X_i$

if the fitness value of X_i is greater than that of PB_i
 then $GB = X_i$

end if

for d = 1 to no of genes

$v_{id}^{new} = w \times v_{id}^{old} + c_1, r_1 (pb_{id}^{old} - x_{id}^{old}) + c_2, r_2 (pb_{id}^{old} - x_{id}^{old})$

if $v_{id}^{old} > v_{max}$ **then** $v_{id}^{new} = v_{max}$

if $v_{id}^{old} < v_{min}$ **then** $v_{id}^{new} = v_{min}$

if $sigmoid(v_{id}^{new}) > U(0,1)$

then

$x_{id}^{new} = 1$

else

$x_{id}^{new} = 0$

endif

nextd

nexti

end while

end

Improvements from ID3 algorithm:

C4.5 made a number of improvements to ID3. Some are:

1. Handling both continuous and discrete attributes - In order to handle continuous attributes, C4.5 creates a threshold and

then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it.

2. Handling training data with missing attribute values - C4.5 allows attribute values to be marked as ? for missing. Missing attribute values are simply not used in gain and entropy calculations.

3. Handling attributes with differing costs. Pruning trees after creation - C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes.

RESULTS:

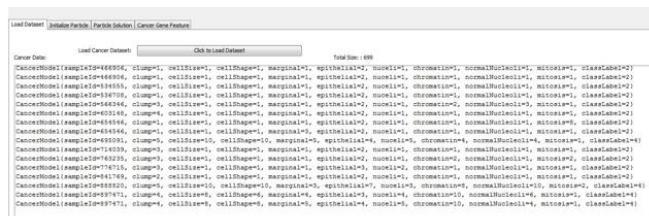


Fig3: Loading Cancer Dataset

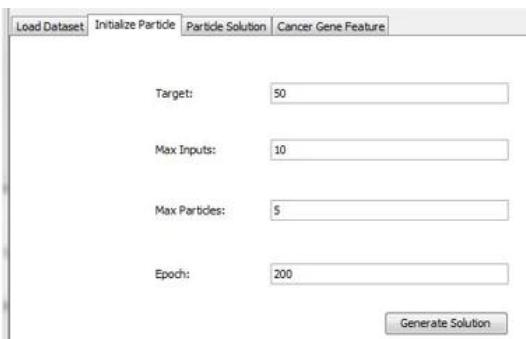


Fig4: Initializing the Particle

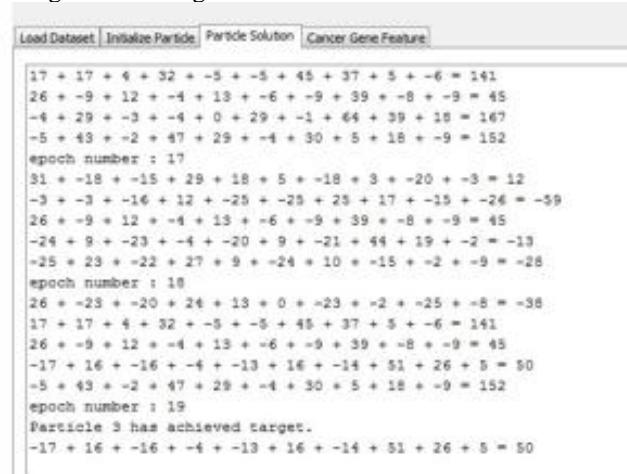


Fig5: Particle Solution

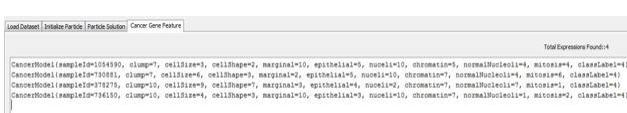


Fig6: Featured selected cancer Gene with target value

V. CONCLUSION

Genes play a vital part in analysis and pathogenesis of different types of tumors. Various cancer datasets from the UCI repository with gene data have been used in evaluating the performance of algorithms. From our research, it's evident that PSODT algorithm gives mere best results. Here we have made some changes to existing algorithm and gives better results when used with C4.5 Decision Tree as a classifier evaluating fitness value. And, entropy value gives a definitive probability in exact prediction of which particle is closest to the target and that every particle should forage in that direction. This combination of algorithms helped in achieving classification in lesser iterations than before with much promising accuracy.

This proposed method has incorporated with the nonlinear search capability of PSO and linearly separable advantage of DT to apply to microarray cancer datasets for gene selection. The precision of results can be well differentiated when large datasets are given as input. Further enhancements can be suggested as follows, In PSO, developing self-adaptation parameters of particle size, number of iterations, and constant weight factors may result in better accurate solutions. Secondly, adding hybrid search algorithms to PSO algorithm may improve its performance. Finally, the improvement in the execution time for huge datasets could be treated as a research subject in the future.

REFERENCES

REFERENCES

- [1] Li X, Rao S, Wang Y, and GONG B: "Gene mining: A novel and powerful ensemble decision approach to hunting for disease genes using microarray expression profiling" *Nucleic Acids Res* 2004, 32:2685-2694.
- [2] Isabelle Guyon, Jason Weston, Stephen Barnhill Barnhill Bioinformatics, Savannah, Georgia, USA. Vladimir Vapnik, AT & T Labs, Red Bank, New Jersey, USA: "Gene Selection for Cancer Classification using Support Vector Machines. Machine Learning", 46, 389–422, 2002 c 2002 Kluwer Academic Publishers. Manufactured in The Netherlands.
- [3] Li S, Wu X, Tan M: "Gene selection using hybrid particle swarm optimization and genetic algorithm". *Soft Computing* 2008, 12:1039-1048.
- [4] Su Y, Murali TM, et al.: "RankGene: Identification of diagnostic genes based on expression data". *Bioinformatics* 2003, 19:1578-1579.
- [5] AzadehMohammadi, Mohammad H Saraee, MansoorSalehi: "Identification of disease-causing genes using microarray data mining and Gene Ontology". *BMC Medical Genomics* 2011.
- [6] Zhi Yan, YiminXiong, Weitian Xu, Min Li, Yi Cheng, Fang Chen, Shifang Ding, Hualin Xu, and Guorong Zheng: "Identification of recurrence-related genes by integrating microRNA and gene expression profiling of gastric cancer". Wuhan General Hospital of Guangzhou Command, Wuchang, Wuhan 430070, P.R. China. Received July 2, 2012; Accepted August 17, 2012.
- [7] Kun-Huang Chen, Kung-JengWang, Min-Lung Tsai, Kung-Min Wang, Angelia Melani AdrianI, Wei-ChungCheng, Tzu-Sen Yang, Nai-Chia Teng, Kuo-Pin Tan and Ku-Shang Chang: "Gene selection for cancer identification: a decision tree model empowered by particle swarm optimization algorithm". Chen et al. *BMC Bioinformatics* 2014.

Pranav TejaGarikapati has completed his Master of Science in computer science from The University of Texas, Arlington, USA and his Bachelors of Technology in Computer Science & Engineering at GITAM University, Visakhapatnam. His research areas include Data Mining, Network Security and Software Engineering.

Naveen Kumar Penki is pursuing his Master of Science, computer science at The University of Oklahoma, Norman, USA. He completed his Bachelors of Technology in Computer Science & Engineering from GITAM University, Visakhapatnam, India. His research areas include Data Mining, Cryptography and Data Analytics.

SashankGogineni has completed his masters from Georgia State University, Georgia, USA. He completed his bachelors of Technology in Computer Science & Engineering from GITAM University, Visakhapatnam. His research areas include Data Mining and Image processing.