International Journal of Computational Intelligence and Informatics, Vol. 2: No. 4, January - March 2013

Significant Feature Set Driven and Optimized FFN for Enhanced Classification

Asha Gowda Karegowda
Dept. of Master of Computer Applications
Siddaganga Institute of Technology,
Tumkur, India
ashagksit@gmail.com

M A Jayaram
Dept. of Master of Computer Applications
Siddaganga Institute of Technology,
Tumkur, India

***Abstract*- Neural Networks augmented with back propagation learning is one the extensively used data classification tools. In this paper, a novel classification scheme is elaborated. The method evolved has two steps: In the first step, significant feature selection is made by using decision tree and GA-CFS (genetic algorithm based correlation based feature selection). In the second step, the connection weights of feed forward network (FFN) are optimized using Particle swarm optimization (PSO) and GA. To convalidate the efficacy of the method, it was applied to four benchmark datasets namely diabetes, iris, ionosphere and heart statlog. PSO showed best classification accuracy in the range of 86%-97% for all the datasets considered when compared with BPN and GA based networks. The topology of the PSO optimized FFN was also modest, with a few neurons in the hidden layer.**

Keywords- Particle Swarm Optimization, Genetic Algorithm, Feed forward neural network, backpropagation, feature selection, FFN Connection weights optimization

I. INTRODUCTION

FFN is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Developing a neural network involves first training the network to carry out the desired computations. In Supervised learning, the network is trained by providing it with input and matching output patterns (training data). A commonly used cost is the mean-squared error (MSE), which tries to minimize the average squared error between the network's output, and the expected output. Neural networks have been criticized for their poor interpretability, since it is difficult for humans to interpret the symbolic meaning behind the learned weights. Advantages of neural networks, however, include their high tolerance to noisy data and their ability to classify patterns on which they have not been trained. The most common method adopted for training of FFN is back propagation method (BPN). In BPN model each node is connected to all nodes in the adjoining layer and each connection has an unbounded positive or negative weight associated with it. Back propagation learning works by making modifications in weight values starting at the output layer then moving backward through the hidden layers of the network [14]. BPN uses the gradient-based approach, which either trains slowly or may get struck with local minimum [10, 23, 26, 27, 30, 36]. There are several variants and extensions of BP used for training neural network: gradient descent with momentum, scaled conjugate gradient (SCG), resilient propagation (RPROP), BFGS quasi-Newton, and Levenberg-MarquarJ48 (LM) algorithms [12]. In addition, one may apply the commonly used optimization methods such as Genetic Algorithms (GAs), Particle swarm optimization (PSO), Artificial Bee Colony (ABC) Optimization Algorithm and Ant Colony optimization algorithm for determining not only the connection weights, but also for optimizing various parameters of NN such as number of hidden layers, number of nodes in hidden layers, relevant feature subsets, the learning rate and the momentum.

This paper presents the application of two evolutionary algorithms namely PSO and GA for optimizing network connection weights of FFN. Computational work has been carried out on UCI machine learning benchmark datasets. Section 2 elaborates on GA and optimizing connection weights of FFN using GA. The applications of PSO for optimizing connection weights of FFN are explained in Section 3. Section 4 describes the two filters: GA-Correlation based feature selection (GA-CFS) and decision tree used for identifying the significant inputs for FFN. Computational results and conclusions are presented in Section 5 and Section 6 respectively.

II. GENETIC ALGORITHM

GA is a stochastic general search method, capable of effectively exploring large search space, which is usually required in case of attribute selection. Further, unlike many search algorithms, which performs a local, greedy

search, GAs performs the global search. GA is an optimization technique inspired by natural selection and natural genetics [11]. GA is mainly composed of three operators: reproduction, crossover, and mutation. As a first step of GA, an initial population of individuals is generated at random or heuristically. In each generation, the population is evaluated using fitness function. In the selection process (reproduction operation), the high fitness chromosomes are used to eliminate low fitness chromosomes. But, selection alone does not produce any new individuals into the population. Hence selection is followed by crossover and mutation operations. The new population generated undergoes further selection, crossover and mutation till the termination criterion is not satisfied. Convergence of the genetic algorithm depends on the various criterions like fitness value achieved or number of generations, as specified by the user [11, 32].

A. GA optimized FFN connection weights (GAFFN)

GA has been used for optimizing the NN parameters including, architecture, connections weights, significant feature selection, activation function, training algorithm and numbers of iterations [27]. GA has been used to search optimal hidden-layer architectures, connectivity, and training parameters (learning rate and momentum factor) of NN for predicting community-acquired pneumonia among patients with respiratory complaints [29]. Jihoon et al. [16] have proposed an approach to the multi-criteria optimization problem of feature subset selection using GA with NN. GA has been used to optimize the connection weights of NN and has been applied for predicting stroke disease [36]. GA has been used to optimize the ANN parameters namely: learning rate, momentum coefficient, activation function, number of hidden layers and number of nodes for worker assignment into virtual manufacturing cells (VMC) application [26]. GAFFN model has been experimented for the study of the heat transport characteristics of a Nano fluid thermo syphon in a magnetic field where, GA is used to optimize the number of neurons in the hidden layer, the coefficient of the learning rate and the momentum factor of NN [35]. Application of GA for optimizing the connection weights of FFN for diagnosis of PIMA diabetic dataset is reported in [4]. The functioning of proposed hybrid GAFFN [4] is explained as follows.

a) The original population is a set of randomly generated *N* chromosomes. For a FFN with single hidden layer with *m* hidden nodes, *n* inputs nodes and *p* output nodes, the number of connection weights is equal to *(n+1)* m +(m+1)*p*. Each chromosome is made up of number of genes equal to total number of connection weights of FFN. Genes are represented by real number encoding method.
b) Repeat steps (c) - (f) until termination condition (80% of the chromosomes converge to the same fitness value or maximum generation reached) is reached.
c) Fitness of each chromosome is computed by maximum optimization method: Fitness (*Ci*) = 1 /*E*, for each chromosome *Ci* of the population, where *E* is the error computed as mean square error (*MSE*) at the output layer.
d) The best-fit chromosomes (lowest *MSE*) replace the worst fit chromosomes (Reproduction step).
e) Crossover step is implemented using single point crossover, two-point crossover and multi point crossover. In addition, a new type of crossover called mixed crossover is used. In mixed crossover, given *M* number of generation, multipoint crossover is applied for the first 60% of generation, followed by two point crossover for the next 20% generation and finally one point crossover for the remaining generations.
f) Mutation is applied by changing the weights of randomly selected chromosomes by multiplying it with a random number to generate the new population.

The weights represented by the fittest chromosome (with least *MSE*) in the final population are the optimized connection weights of the FFN. Functioning of GAFFN is shown in Fig. 1.

III. PARTICLE SWARM OPTIMIZATION ALGORITHM (PSO)

PSO [7] is a population based stochastic optimization technique, inspired by social behavior of bird flocking or fish schooling. A basic variant of PSO algorithm works by having a population (called swarm) of candidate solutions (called particles). The movements of particles are directed by their own best-known position: *Pbest* in the search-space as well as the entire swarm's best position: *Gbest*. When improved positions are being discovered, they in turn guide the movements of the swarm. The process involves both social interactions and intelligence so that the swarms learn from their own experience (local search) and also from the experience of neighbor swarms (global search) [18].

The main parameters used in the PSO algorithm are the population size (number of particles), number of cycles, maximum change of a particle velocity *Vmax*, inertia weight w and constants *c1* and c2. *c1* and *c2* are two positive constants usually set to 2; *rand1()* and *rand2()* are two random functions in the range [0,1], *Vmax* is un upper limit on the maximum change of particle velocity [7] and *w* is an inertia weight employed as an improvement proposed by [37] to control the impact of previous velocity on current velocity. *w* plays the role of balancing the global search and the local search and decreases linearly with time [37]. User specified value of *Vmax* (upper and lower bounds) is used to control the change in particle velocity. The PSO process is initialized

with a group of *N* random particles (solutions). Each particle is represented by its position as a point in *M* dimensional space, where *M* is the number of variables. Each particle *i* monitors its current position *Xi*, particles best position *Pi* and its velocity *Vi*. In each cycle, both particles best position and global best position are used to find the current velocity representing both cognition and social collaboration among the particles [17] by using equation 1. The current velocity is used to update the position of particle using equation 2.

$$Vcurrent = w * Vprevious + c1 * rand1() * Pbest - XPrevious + c2 * rand2() * (Gbest - Xprevious) \quad \text{eq(1)}$$

$$Xcurrent = Xprevious + Vcurrent \quad \text{eq(2)}$$

where *Vprevious* and *Vcurrent* are the previous and current velocities of the particle respectively. *Xprevious* and *Xcurrrent* are previous and current position of the particle. *c1* and *c2* are the acceleration constants, *w* is an inertial weight, *rand1* and *rand2* are the random variables with values between 0 and 1. The process is repeated until specified number of iterations is exceeded or the desired fitness score is achieved. Shih et.al has adopted PSO based feature selection to enhance the classification accuracy of Linear discriminate analysis (PSOLDA) [38]. Some of major differences between PSO and GA are as follows [9, 10, 23, 28].

a) The chromosomes under goes the process of reproduction, crossover and mutation to generate the new set of high fitness chromosomes in next iteration. As opposed to GA, the evolutionary process in the PSO does not create the new particles from parent particles. Instead the particles in PSO move towards the best possible solution by updating the particles using the local best (*Pbest*) and global best (*Gbest*) solution.

b) One of the major downsides of the GA is their lack of memory (i.e. GA's crossover and mutation operations may at times loose the best solution achieved so far), which limits the search and convergence capability of the algorithms. PSO algorithm emerges as a powerful stochastic optimization technique, in which particles have memory (stores the previous iterations local best (*Pbest*) and global best (*Gbest*) solution) and work collectively using local best and (*Pbest*) and global best (*Gbest*) solution to move towards a solution region containing the global or a near-optimal solution.

A. PSO optimized FFN connection weights(PSOFFN)

PSO has been used to optimize the neural network parameters: number of hidden layers, number of nodes in hidden layer, the input neurons and connection weights. PSOFFN has been used to optimize the weights, transfer function and topology of FFN constructed for reactive power control [30]. Comparison of hybrid GAFFN and PSOFFN has been carried out for Tennessee Eastman (TE) chemical process reactor by optimizing the network weights [23]. The hybrid PSOFFN has been modeled to train perceptrons in predicting the outcome of construction claims in Hong Kong [6]. PSO based ANN has been used distinguish between normal subjects and those with tremor (Parkinson's disease) [9]. Comparative study of variants of PSO such as multi-start PSO, Guaranteed convergence PSO, conventional BPN method and GA-based techniques have been carried out for medical datasets: breast, diabetes and Hepatitis [10]. PSOFFN has been used has been used for medical diagnosis problem of breast cancer, heart disease and diabetes [25]. PSO has been used to optimize both architecture and connection weights of neural network [21] for classification of diabetes and heart dataset. It is found that PSO algorithm promises global optimum with a large probability and high convergence rate [6, 8, 10, 19, 23, 30]. The significant step of the PSO is the representation of the particles. For a FFN with single hidden layer with m nodes in hidden layer, *n* inputs nodes and *p* output nodes, the number of connection weights is given by *(n+1)* m +(m+1)* p*. The total number of connection weights of the FFN decides the number of dimensions of the PSO particle. The proposed PSOFFN algorithm is explained below.

a) Initialize the original population as set of N particles (each particle representing connection Weights of NN), which is generated randomly.
b) Train the NN using particle (set connection weights using each particle).
c) Compute the learning error at output layer of NN. Fitness of each particle is computed by maximum optimization method. Compute the fitness is given by Fitness (*Pi*) = *1/ E* for each particle of the population, where *E* is the error computed as *MSE* at the output layer of NN as the difference between expected and estimated output.
d) Compare the particles current fitness value with particles *Pbest*. If the current fitness value of particle is better than the previous *Pbest* then set *Pbest* as current fitness value. IF the current best fitness value is better than the previous *Gbest* then set *Gbest* as best current fitness value.
e) Compute the velocity and update position of each particle based on *Gbest* value (lowest learning error found in entire learning process so far) and *Pbest* value (each particles lowest learning error so far) using equations 1 and 2.
f) Repeat steps (b) - (e) until terminating condition is reached (user defined maximum iterations or minimum error criterion).

The *Gbest* positioned particle, represents the optimized connection weights for FFN. The performance of PSOFFN is measured using correctly classified test data. Functioning of PSOFFN is shown in Figure. 2.

IV. FEATURE SELECTION: FILTER APPROACH

The high dimension data makes testing and training of general classification methods difficult. Feature selection is an essential pre-processing method to remove irrelevant and redundant data. Feature subset selection not only improves classification accuracy but also reduces the computational time of classifier [24, 39]. In this work, two filter methods namely: GA-CFS and decision tree has been used.

A. Decision tree

The decision tree C4.5 (Weka j48) [15] has been investigated for exploring the significant attributes [2, 3, 4] for benchmark data sets. A decision tree [31] is a simple tree like structure where non-terminal nodes represent tests on one or more attributes and terminal nodes reflect decision outcomes. The non-terminal nodes in the decision tree represents the significant attributes and given as input to feed forward neural network.

B. GA-CFS filter

Weka's [15] GA is used as random search method to find the significant attribute set [5]. The chromosomes are binary encoded where in, each chromosome is represented as string of zeroes (attribute is not significant) and ones (attribute is significant). The randomly generated population undergoes the selection, crossover and mutation process until the termination condition is not met. The fitness of the chromosome is estimated using correlation based feature selection (CFS). Equation for CFS is given by equation $r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{k+(k*(k-1))\overline{r_{ii}}}}$, where r_{zc} is the correlation between the summed feature subsets and the class variable, k is the number of subset features, $\overline{r_{zi}}$ is the average of the correlations between the subset features an the class variable, and $\overline{r_{ii}}$ is the average inter-correlation between subset features [22]. The process of selection, crossover and mutation is repeated for quantified number of generations. The fittest chromosome represents the set of significant features.

V. RESULTS AND DISCUSSIONS

The model development involved two stages (i) significant feature set determination assisted by GA- CFS and decision tree (ii) optimization of connection weights of FFN using PSO and GA. The performance of BPN, GAFFN and PSOFFN is evaluated using classification accuracy computed as (Total number of correctly classified test samples) / (Total number of test samples). Experiments are conducted on four datasets namely Heart-statlog, diabetes, iris and Ionosphere from UCI Machine Learning Repository.

In the first stage, the GA-CFS filter and decision tree C4.5 are applied to the datasets for feature selection. The number of features selected by GA-CFS and decision tree for the four datasets is depicted in Table 1. For feature extraction using GA-CFS, the GA parameters: population size, number of generations, crossover rate and mutation is set to 20, 20, 0.6 and 0.033 respectively. *K*-fold cross-validation with k = 10 is used for both decision tree and GA-CFS feature selection process.

In the second stage, 60% and 40% of datasets were used for training and testing respectively. Further, investigations were done by varying the number of epochs and the topology of BPN, GAFFN and PSOFFN with (i) all input features and (ii) with significant features as shown in Table 1. For GAFFN, the chromosomes are encoded using real numbers. Four types of crossover operations were experimented: single point, two point, multiple point and mixture cross over with different population size and number of generations. The termination condition used is that almost eighty percent of the chromosomes represent the same connection weights. GAFFN was experimented by varying the size of population size with 20-60, 2-20 numbers of nodes in the hidden layer and with 50-200 numbers of generations. Among the various topologies experimented, the best performance of GAFFN with all inputs, GA-CFS and Decision tree identified inputs is shown in Table 1. With the inputs identified by decision tree, the single point and mixture cross over resulted in slight improved accuracy compared to two points and multiple crossover. GA-CFS identified features resulted in almost same classification accuracy for all the four types of cross overs operations.

In addition to optimizing the connection weights of FFN using GA, similar work was endeavored using PSO. Experiments were piloted to find the most promising configuration of PSOFFN by varying values of *c1* and *c2*, number of hidden neurons, number of PSO particles and number of iterations. It was found that the acceleration constants *c1* and *c2* for *Gbest* and *Pbest*, number of particles, number of hidden neurons and number of iterations for PSOFFN are interrelated. The outcomes of each of this parameter of PSOFFN are presented below.

a) Acceleration constants *c1* and *c2*: The acceleration constant *c1* and *c2* affect the influence of the global best and local best solution of the particle. The performance of PSOFFN was evaluated with *c1* and *c2* ranging from 1.2 to 2.2. The PSOFFN performance is found to be better for *c1* and *c2* in the range of 1.8 to 2.0 with all inputs as well as with reduced significant attributes. The number of nodes in hidden layer was varied

from 2 to 20. Other parameters were kept constant with the 30 particles, maximum number of 200 iterations. The PSOFFN took longer processing time for lower values of *c1* and *c2* while, the processing time of PSOFFN gradually declines with the increase in values of *c1* and *c2*, since higher values of *c1* and *c2* produce large variation in the position of the particles.

b) Number of particles: Number of particles in the PSO population symbolizes the number of possible solution that is covered in the problem. Number of particles was varied from 5 to 40. The PSOFFN performance gradually improved with the increase in the number of particles. The performance was best with 15 to 30 particles with PSOFFN using all inputs as well as with filter identified significant inputs. The classification accuracy of PSOFNN, with number of particles larger than 30, remained almost same and in some case got deteriorated. Furthermore, as the number of particles is increased, PSOFNN took longer processing time.

c) Maximum number of iterations: The performance of PSOFFN was also investigated by varying the number of maximum iterations (terminating criterion) ranging from 50 to 250. The PSOFFN performance progressively enhanced with the increase in the number of iterations but did not show markable improvement beyond 200 iterations.

d) Number of nodes in hidden layer: Experiments were conducted by ranging the number of nodes in the hidden layer from 2 to 20. As the number of hidden nodes is increased, the PSOFFN takes longer processing time; since number of nodes decides the dimension of each particle.

For iris dataset the BPN accuracy with all inputs resulted in same accuracy. Similar results are observed for GAFFN with all and reduced inputs. However PSOFFN showed slight improved classification with reduced set when compared to all inputs. For the remaining datasets: diabetes, heart and ionosphere, the performance of BPN, GAFFN and PSOFFN with reduced set of inputs showed a markable improvement when compared to with all inputs. Input features selected by both decision tree and GA-CFS resulted in almost identical classification precision for PSOFFN. Input features selected by GA-CFS stemmed in enhanced classification accuracy when compared with features identified by decision tree for GAFFN as well as for BPN. Figure 3 clearly proves that significant inputs identified by GA-CFS and decision tree with BPN leads to improvised categorization accuracy, compared to results produced by BPN with all the inputs. Relative performance of FFN trained using BPN, GA and PSO with all inputs and significant inputs identified by decision tree and GA-CFS for the four benchmark datasets is compared with the performance of various classifiers as presented in Table 2. The GAFFN and PSOFFN showed improved classification compared to the earlier reported work.

VI. CONCLUSIONS

This paper discussed the applications of significant feature detection algorithms and consequent optimization of FFN weights using two evolutionary algorithms viz., GA and PSO. The two schemes were used in a cascaded fashion for the classification task. Results have shown that, PSO can classify the data with remarkable classification accuracy when compared to GA.ared to the earlier reported work.

REFERENCES

[1] Alexander K.Seewald, Johannes Fürnkranz, "An Evaluation of grading classifiers", Proc. of 4th International Conference on Advances in Intelligent Data Analysis, London,UK, Springer, Sept.2001, pp.115-124.

[2] Asha Gowda Karegowda, M.A.Jayaram, A.S.Manjunath, "Integrating decision tree and ANN for categorization of diabetes data", International Conference on Computer Aided Engineering, IIIT Madras, India, Dec 2007, pp.875-881.

[3] Asha Gowda Karegowda, M.A.Jayaram, A.S.Manjunath, "Comparative study of attribute selection using Gain Ratio and Correlation based feature selection", International Journal of Information Technology and Knowledge Management, vol. 3(2), 2010, pp.271-277.

[4] Asha Gowda Karegowda, M.A.Jayaram, A.S.Manjunath, "Application of genetic algorithm optimized neural network connection weights for medical diagnosis of PIMA Indian diabetes", International Journal of Soft Computing, 2, 2011, pp.15-23.

[5] Asha Gowda Karegowda, M.A.Jayaram, A.S.Manjunath, "Feature subset selection using cascaded GA and CFS: a filter approach in supervised learning:", International Journal on Computer Applications, 23(2), 2011, pp.1-10.

[6] Chau K.W., "Applications of a PSO-based neural network in analysis of outcomes of construction claims", Automation in Construction, vol. 16(227), 2011, pp.642-646.

[7] Eberhart R.C., Kennedy J., "A new optimizer using particle swarm theory", Proc. of the Sixth International Symposium on Micro machine and Human Science, Nagoya, Japan, 1995, pp.39-43.

[8] Eberhart R.C., Kennedy J., "Particle Swarm Optimization", Proc. of IEEE International Conference on Neural Networks (Perth, Australia), 1995, pp.1942-1948.

[9] Eberhart R.C., Hu X., "Human tremor analysis using particle swarm optimization", Proc. of IEEE Congress on Evolutionary Computation, 1999, pp.1927-1930.

[10] Eberhart R.C., Shi Y., "Comparison between genetic algorithms and particle swarm optimization", Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming , Springer-Verlag, 1998.

[11] Goldberg D., " Genetic algorithms in search, optimization, and machine learning", Addision-Wesley Publishing Co., 1989.

[12] Gudise V.G., Venayagamoorthy, G.K., "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks", IEEE Proc. of the Swarm Intelligence Symposium, 2003, pp. 110- 117.

[13] Hai H. Dam, Hussein A. Abbas, "Neural-based learning classifier systems", IEEE Transactions of Knowledge and Data Engineering, vol.20(1), Jan 2008.

[14] Haykin S. , "Neural Networks- A comprehensive foundation", Macmillan Press, New York, 1994.

[15] Ian H.Witten, Eibe Frank, "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, 2nd edition, 2005.

[16] Jihoon Yang, Vasant Honavar, "Feature subset selection using a genetic algorithm", IEEE Intelligent Systems, vol.13(2), 1998, pp.44-49.

[17] Kennedy J.,"The Particle Swarm: social adaptation of knowledge", Proc. of IEEE International Conference on Evolutionary Computation, 1997, pp.303-308.

[18] Konstantinos E., MichelP., Vrahatis N., "Particle swarm optimization and intelligence : advance and applications", Information Science Reference, Hershey, New york, 2010.

[19] Kennedy J., Eberhart R.C., Shi Y., "Swarm Intelligence", Morgan Kaufmann Publishers, San Francisco, 2001.

[20] Sotiris B. Kotsiantis, I.D. Zaharakis, P.E. Pintelas, "Machine learning : A review of classifications and combining techniques", Artificial Intelligence Review, vol.26, 2006, pp.159-190.

[21] Marcio Carvalho, Teresa B. Ludermir, "Particle Swarm Optimization of neural network architecture and weights", Proc. of the IEEE Seventh International Conference on Hybrid Intelligent System, 2007, pp.336-339.

[22] Mark A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning", Proc. of 17th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2000, pp.359-366.

[23] Malik Braik, Alaa Sheta, Amani Arieqat, "A comparison between GAs and PSOs in training ANN to model the TE chemical process reactor", Symposium on Swarm Intelligence Algorithms and Applications, 2008, pp.24-31.

[24] Manoranjan Dash, Huan Liu, "Feature selection for classification", Intelligent Data Analysis, vol. 1, 1997, pp.131-156.

[25] Mazurowski MA, Habas PA, Zurada JM, Lo JY, Baker JA, Tourassi GD, "Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance", Neural Networks, vol. 21, 2008, pp.427–436.

[26] Murali R.V., Puri A.B., Prabhakaran G., "GA-Driven ANN model for worker assignment into virtual manufacturing cells", Proc. of the World Congress on Engineering, London, U.K., 2010.

[27] Osman Ahmed, Mohd Nordin, Suziah Sulaiman, Wan Fatimah, "Study of genetic algorithm to fully automate the design and training of artificial neural network", International Journal of Computer Science and Network Security, vol. 9(1), 2009, pp.217-226.

[28] Panduro M.A., Brizuela C.A., Balderas L.I., Acosta D.A., "A comparison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scan able circular antenna arrays", Progress In Electromagnetics Research, vol. 13, 2009, pp.171-186.

[29] Paul S. Heckerling, Ben S.Gerber, Thomas G.Tape, Robert S.Wigton, "Use of genetic algorithms for neural networks to predict community-acquired pneumonia", {\it Artificial Intelligence in medicine}, vol. 30(1), 2004, pp.71-84.

[30] Paulo F. Ribeiro, W.Kyle Schlansker, "A hybrid particle swarm and neural network approach for reactive power control", http://webapps.calvin.edu~pribeiro/courses/engr302/Samples /ReactivePower-PSO-wks.pdf}, 2010.

[31] Quinlan J.R., C4.5: Programs for Machine Learning, Morgan Kaufmann Publication, 1993.

[32] Rajasekaran S., Vijayalakshmi Pai G.A., "Genetic algorithm based weight determination for backpropagation networks", Proc. of the 4th International Conference on Advanced Computing, Bangalore, 1996, pp.73-79.

[33] Satchidananda Dehuri, Sung-Bae Cho, "A comprehensive survey on functional link neural networks and an adoptive POS-BP learning for CFLNL", Neural Network Computing and Applications, vol. 19(2), 2010, pp.187-205.

[34] Satchidananda Dehuri, Rahul Roy, Sung-Bae Cho, Ashish Ghosh, "An improved swarm optimized functional link artificial neural network (ISO-FLANN) for classification", The Journal of Systems and Software, vol. 85, 2012, pp.1333-1345.

[35] Salehi H., Zeinali Heris S., Koolivand Salooki M., Noei S.H., "Designing a NN for closed themosyphon with nanofluid using a GA", Brazilian Journal of Chemical Engineering, vol. 28(1), 2011, pp. 157-168.

[36] Shanti D., Sahoo G., Saravanan N., "Evolving connection weights of ANN using GA with application to the prediction of stroke disease", International Journal of Soft Computing, vol. 4(2), 2009, pp.95-102.

[37] Shi Y., Eberhart R.C., "A modified particle swarm optimizer", Proc. of IEEE International Conference on Evolutionary Computation, 1998, pp.69-73.

[38] Shih-Wei Lin, Shih-Chieh Chen, "PSOLDA: A particle swarm optimization approach for enhancing classification accuracy rate of linear discriminant analysis", Applied Soft Computing, vol. 9, 2009, pp.1008-1015.

[39] Saeys Y., Inza I., Larrañaga P., "A review of feature selection techniques in bioinformatics", Bioinformatics, vol. 23(19), 2007, pp.2507-2517.

Table 1: Comparison of BPN, GAFFN and PSOFFN performance various datasets with different topologies

Data Set	Feature selection method	Number of Features	BPN		GAFFN		PSOFFN	
			Topology	Accuracy	Topology	Accuracy	Topology	Accuracy
Diabetic	All	8	8-24-1	72.88	8-20-1	77.707	8-4-1	85.957
	GA-CFS	4	4-8-1	79.5	4-10-1	84.713	4-8-1	86.383
	DT	5	5-15-1	78.21	5-15-1	84.076	5-2-1	86.809
Iris	All	4	4-5-3	95.00	4-10-3	96.66	4-10-3	96.667
	GA-CFS	2	2-4-3	95.00	2-10-3	96.66	2-8-3	97.778
	DT	2	2-4-3	95.00	2-10-3	96.66	2-8-3	97.778
Ionosphere	All	34	34-19-2	83.57	34-12 - 2	85.802	34-10-2	90.00
	GA-CFS	14	14-9-2	85.71	14-10 - 2	86.42	14-6-2	93.839
	DT	14	14-9-2	85.00	14-12-2	87.037	14-8-2	94.313
Heart Statlog	All	13	13-8-2	75.9259	13-10-2	83.57	13-7-2	85.802
	GA-CFS	7	7-5-2	77.778	7-15-2	85.802	7-8-2	88.272
	DT	10	10-7-2	82.407	10-10-2	85.802	10-5-2	87.037

Table 2. Classification accuracy of the proposed model compared with different machine learning methods

Classifier Algorithm	Classification Accuracy				Reference
	Diabetic	Iris	Ionosphere	Heart Statlog	
Decision tree	74.32±1.18	92.80±0.93	89.74±1.03	82.2±1.80	Alaxander et al.(2001) [1]
J48	73.74 ±0.79	94.67±0.70	89.74±0.74	78.67±1.49	
Kernel Density	71.41±0.51	95.20±0.53	89.00±0.31	76.56±0.94	
Kstar	70.29±.43	94.67±0.00	84.02±0.70	76.81±0.77	
Multilayer Regression	76.97±0.46	84.27±0.78	86.55±0.48	83.78±0.89	
Naïve Bayes	75.31±0.28	95.93±0.38	91.77±0.48	84.63±0.66	
Naïve Bayes	75.75±5.32	95.53±5.02	82.17±6.14	83.5±5.98	Kotsiontis et al. 2006 [20]
C4.5	74.49±5.27	94.73±5.30	89.74±4.38	78.15±7.42	
3NN	73.86±4.55	95.20±5.11	86.02±4.31	79.11±6.77	
RIPPER	75.22±4.86	93.93±6.57	89.90±4.63	78.7±6.62	
BP	77.04±4.85	84.80±7.1	87.07±5.52	83.30±6.2	
SMO	77.07±4.14	84.87±7.63	87.93±4.69	83.81±5.59	
PSO-PSO	75.6354±3.7			80.113±2.147	Marcio et al.(2007)[21]
PSO-PSO-WD	76.4583±3.159			81.902±3.057	
Evolutionary Prog.	77.621±.014			83.235±2.029	
GA (Conn. Matrix)	75.44±1.65			76.78±7.87	
GA(Neural Cross)	78.58±2.19			85.1±2.78	
UCS with GA	74.8±4.4	94.9±4.2	72.9±5.1	84.8±11.6	Hai et al (2008)[13]
NLCS with GA	76.5±4.2	94.9+6.7	87.4±6.5	61.1 ±7.1	
HCFLNN	79.82	98.74			Satchidananda et al. (2009)[33]
FLNN	78.12	98.66			
ISO-FLANN	79.63	99.03	90.38		Satchidananda et al. (2012)[34]
FLANN	78.82	97.33	80.94		
MLP	77.19	94.00	73.28		
SVM	75.37	91.70	83.74		
FSN	76.39	96.00	87.50		
LDA - all features	76.4	98.0	86.5		Shih-Wei et al.2009[38]
LDA- forward feature selection	74.8	96.3	85.3		
LDA- backward feature selection	76.5	93.7	90.0		
LDA- PCA based feature selection	75.9	90.0	86.90		
LDA- exhaustive feature selection	76.7	97.0			
PSOLDA	76.7	97.0	92.2		
BPN	72.88	95.00	83.57	75.9259	This paper
BPN-GACFS	79.5	95.00	85.71	77.778	
BPN-DT	78.21	95.00	85.00	82.407	
GAFFN	77.707	96.66	85.802	83.57	
GAFFN-GACFS	84.713	96.66	86.42	85.802	
GAFFN-DT	84.076	96.66	87.037	85.802	
PSOFFN	85.957	96.667	90.00	85.802	
PSOFFN-GACFS	86.383	97.778	93.839	88.272	
PSOFFN-DT	86.809	97.778	94.313	87.037	

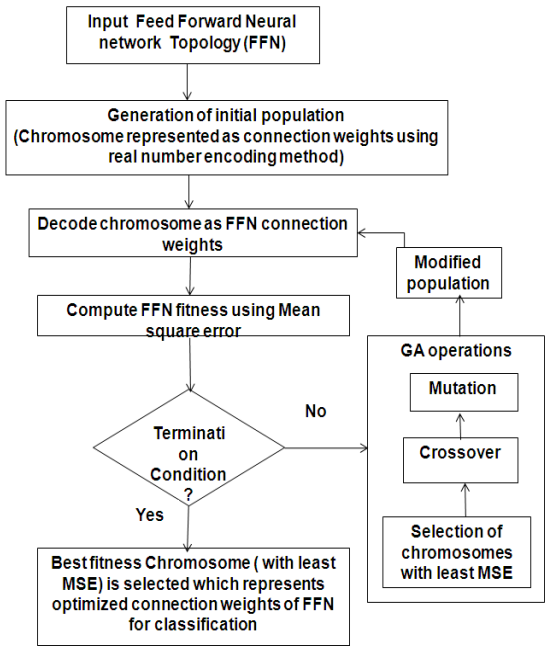

Figure 1: Working of GA optimized FFN connection weights

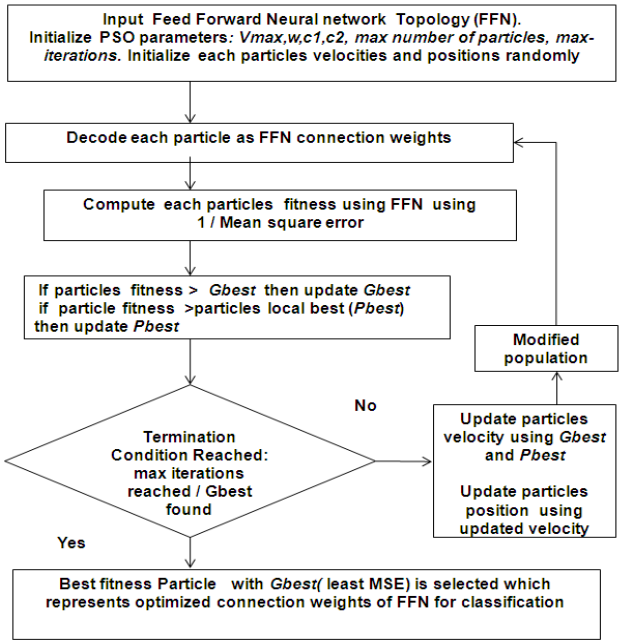

Figure 2: Working of PSO optimized FFN connection weights

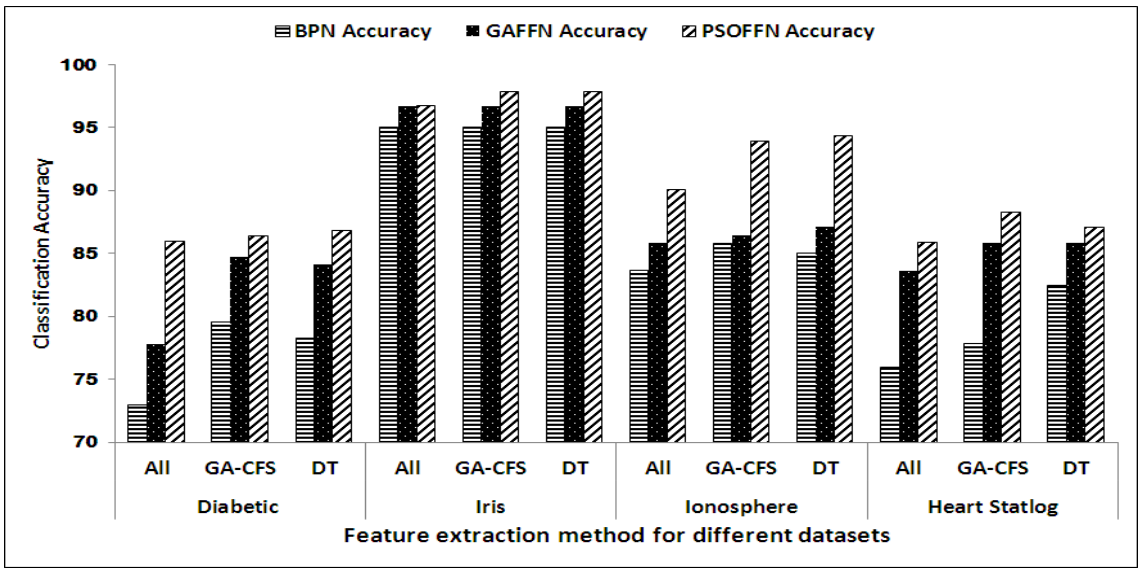

Figure 3: Comparison of BPN, GAFFN and PSOFFN performance with all features, GA-CFS and Decision tree identified features for various datasets