Gazi University Journal of Science
GU J Sci
28(1):115-132 (2015)

A Hybrid Applied Optimization Algorithm for Training Multi-Layer Neural Networks in Data Classification

H. Hasan ÖRKCÜ[1,♠], Mustafa İsa DOĞAN[1], Mediha ÖRKCÜ[2]

[1]Gazi University, Faculty of Sciences, Department of Statistics, 06500 Teknikokullar, Ankara, TURKEY.
[2]Gazi University, Faculty of Sciences, Department of Mathematics, 06500 Teknikokullar, Ankara, TURKEY

Received: 17/10/2014 Accepted:08/12/2014

ABSTRACT

Backpropagation algorithm is a classical technique used in the training of the artificial neural networks. Since this algorithm has many disadvantages, the training of the neural networks has been implemented with various optimization methods. In this paper, a hybrid intelligent model, i.e., hybridGSA (hybrid Genetic Algorithm and Simulated Annealing), is developed for training artificial neural networks (ANN) and undertaking data classification problems. The hybrid intelligent system aims to exploit the advantages of genetic and simulated annealing algorithms and, at the same time, alleviate their limitations. To evaluate the effectiveness of the hybridGSA method, three benchmark data sets, i.e., Breast Cancer Wisconsin, Pima Indians Diabetes, and Liver Disorders from the *UCI* Repository of Machine Learning, and a simulation experiment are used for evaluation. A comparative analysis on the real data sets and simulation data show that the hybridGSA algorithm may offer efficient alternative to traditional training methods for the classification problem.

Keywords: Artificial neural networks; data classification; training of neural networks; genetic algorithm; simulated annealing.

1. INTRODUCTION

Data classification is one of the fundamental problems in many decision-making tasks. It deals with the identification of patterns and the assignment of new samples into known groups. The classification problem of assigning several observations into different disjoint groups plays an important role in business decision making and many other areas. Many decision-making tasks are instances of classification problem or can be easily formulated into a classification problem, e.g., prediction or forecasting tasks, diagnosis tasks, and pattern recognition.

Several different classification approaches have been proposed in the literature since the earliest work of Fisher [1]. Generally, the major classification methodologies include statistical methods, neural networks, support vector machines, decision trees and mathematical programming approaches. The development of statistical classification models can be tracked back to Fisher's linear discriminant analysis (FLDA) where the ratio of between groups and within group variances is maximized. It is theoretically optimal for situations where the underlying populations are multivariate normal and where all the different groups have equal covariance structures. With multivariate populations having unequal covariance structures quadratic discriminant analysis can be used in [2]. However, when normality assumption is not provided, these well-known statistical classification models are usually unable to provide good or satisfactory classification results. Mathematical programming techniques have also been proposed to various data classification problems. The papers by Fred and Glover [3-4] have triggered a plethora of research contributions on mathematical programming approaches [2-18]. In these mathematical programming techniques, minimization of the sum of deviations approaches, maximum of the minimum deviation approaches, goal

♠Corresponding author, e-mail: hhorkcu@gazi.edu.tr

programming approaches, mixed-integer programming approaches and hyper-box representation approaches are the most commonly used optimization approaches.

Artificial neural networks (ANNs) have popularity in solving several business and technical problems that involve prediction, and have also a wide ranging usage area in the classification problems [19-25]. One of the important issues on the neural networks is training of the networks. Backpropagation algorithm is the most widely used search technique for training neural networks. Backpropagation algorithm has negative features such as being captured in the local solutions and having low classification performance in some cases. In order to prevent these disadvantages, researchers have proposed many alternatives.

Hybrid of methods genetic algorithms and simulated annealing algoritms are used for different purposes, i.e., in network design problems [26], in the job planning and scheduling problems [27], in the regression problems [28], in digital signal processing [29], and in nuclear energy area [30]. Örkcü [28] adapted the binary-coded genetic algorithm and simulated annealing methods for variable selection in multiple linear regression. The hybrid method proposed in this study is similar to Örkcü's method [28]. In this study, hybrid of real-coded genetic algorithms and simulated annealing algorithm (named by hybridGSA) is used for training multi-layer neural networks in the data classification.

In order to compare the hybridGSA training algorithm, three different medical data classification data and large-scale simulation datasets covering the different network architectures are used. The results based on real-data and simulation show that classification success of artificial neural network model trained with hybridGSA algorithm is better than backpropagation and other optimization training methods.

The rest of the paper is organized as follows. Section 2 gives general definition of neural networks and its training concept. Section 3 presents proposed ANN training hybridGSA algorithm for the classification problems respectively based on genetic and simulated annealing algorithms. Section 4 provides experimental results and a comparison against existing backpropagation, real-coded genetic algorithm and other training algorithms using three real benchmark datasets from *UCI* machine learning repository and a simulation data sets. Finally, Section 5 concludes the findings and proposes directions for future work.

2. TRAINING OF THE NEURAL NETWORKS

Artificial neural networks (ANNs) are computer systems developed to mimic the operations of the human brain by mathematically modeling its neuro-physiological structure. Artificial neural networks have been shown to be effective at approximating complex nonlinear functions [31].

Since multi-layer neural networks are by far the most popular and simple neural networks they are considered in this study. In particular, the neurons of a multi-layer feed-forward neural network are organized in three layers: the input units receive information from the outside world, usually in the form of a data file; the intermediate neurons, contained in one or more hidden layers, allow nonlinearity in the data processing, the output layer is used to provide an answer for a given set of input values. In a fully connected artificial neural network, each neuron in a given layer is connected to each neuron in the following layer by an associated numerical weight (w_{ij}). The weight connection two neurons regulate the magnitude of signal that passes between them. In addition, each neuron possesses a numerical bias term corresponding to an input of -1 whose associated weight has the meaning of a threshold value.

One of the important issues on neural networks is the training or learning of the networks. Backpropagation algorithm (BP) is the most widely used search technique for training neural networks. BP is an adapting method to search the weights of a neural network for a correct mapping of set of training input–output pairs using the gradient descent method, which is a widely used technique for supervised neural networks learning in many application areas. One major weakness of the gradient methods is that the derivative information is necessary such that the error function to be minimized has to be continuous and differentiable. Also, the training process is easily trapped in a local optimum especially when the problems are multimodal and the training rules are network structure dependent. Backpropagation algorithm, a gradient method, is the most widely used search technique for training neural networks. Despite their popularity, BP has the drawback of converging to an optimal solution slowly when the gradient search technique is applied. That is, a BP using the gradient search technique has two serious disadvantages: the gradient search technique converges to an optimal solution with inconsistent and unpredictable performance for some applications and when trapped into some local areas, the gradient search technique performs poorly in getting a globally optimal solution [32]. The most major problem during the training process of the neural network is the possible overfitting of training data. That is, during a certain training period, the network no longer improves its ability to solve the problem. In this case, the training stopped in a local minimum, leading to ineffective results and indicating a poor fit of the model.

In order to attempt to prevent these disadvantages, researchers have modified the basic algorithm to try to escape local optima and find the global solution. Numerous modifications have been implemented in order to overcome this problem including modifications of differential scaling (changing the learning rate and momentum values during training) [33].

3. A HYBRID OF GENETIC AND SIMULATED ANNEALING ALGORITHMS

The new approach for training the network proposed in this paper is based on an hybrid of genetic and simulated annealing heuristic optimization procedures called hybridGSA. Firstly, genetic and simulated annealing algorithms were summarized in the subsections.

3.1. Genetic Algorithms

Genetic Algorithms (GAs)were proposed by Holland [34] as a global optimization approach inspired by natural evolution and survival of the fittest. GAs use a solution population (chromosomes) which evolves by means of selection, crossover and mutation operators in [35]. GAs perform search in complex, large and multimodal landscapes, and provide near-optimal solutions for objective or fitness function of an optimization problem.

GAs performs the search process in four stages: initialization, selection, crossover, and mutation in [36]. In the initialization stage, a population of genetic structures, called chromosomes, which are randomly distributed in the solution space, is selected as the starting point of the search. After the initialization space, each chromosome is evaluated using a user-defined fitness function. The role of the fitness function is to numerically encode the performance of the chromosome. The mating convention for reproduction is such that only the high scoring members will preserve and propagate their worthy characteristics from generations to generation and thereby help in continuing the search for an optimal solution. The chromosomes with high performance may be chosen for replication several times whereas poor-performing structures may not be chosen at all. Such a selective process causes the best-performing chromosomes in the population to occupy an increasingly larger proportion over time. The crossover forms a new offspring between two randomly selected “good parents”. The crossover operates by swapping corresponding segments of a string representation of the parents and extends the search for new solution in far-reaching direction. The crossover operator typically serves a dual purpose. First, to effectively reduce the search space to regions of greater promise. Second, to provide a mechanism for allowing offspring to inherit the properties of their parents. The crossover occurs only with some probability, which is called the crossover rate. There are many different types of crossover that can be performed: the one-point, the two-point, and the uniform type [37]. The crossover process is also commonly referred to as "recombination". The mutation is the process that mimics the unpredictable and unexpected developments that occur in biological reproduction. In a GA, mutation is a random perturbation to one or more genes that occurs infrequently during the evolutionary process. The purpose of the mutation operator is to provide a mechanism to escape local optima.

3.2. Simulated Annealing Algorithm

Simulated annealing (SA) is a random-search technique which exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system; it forms the basis of an optimisation technique for combinatorial and other problems. It was Metropolis et al. [38] who first proposed this idea, and 30 years later, Kirpatrick et al. [39] and Kirkpatrick [40] observed that this approach could be used to search for feasible solutions of quite general optimization problems. The main idea is that the SA strategy may help prevent being trapped at poor solutions associated with local optima of the fitness function.

SA consists of several decreasing temperatures. Each temperature (T) has a few iterations. First, the beginning temperature is selected and an initial solution is randomly chosen. The value of the cost function based on the current solution (i.e., the initial solution in this case) will then be calculated. The goal is to minimize the cost function. Afterwards, a new solution from the neighborhood of the current solution will be generated. The new value of the cost function based on the new solution will be calculated and compared to the current cost function value. If the new cost function value is less than the current value, it will be accepted. Otherwise, the new value would be accepted only when the Metropolis's criterion is met [38]. According to Metropolis's criterion, if the difference between the cost function values of the current and the newly generated solutions (ΔE) is equal to or larger than zero, a random number δ in $[0,1]$ is generated from a uniform distribution. If $\exp\left(-(\Delta E)/T\right) \geq \delta$ is met, the newly generated solution is accepted as the current solution. This exponential function is occasionally called a Boltzmann function, thus the operator is also called a Boltzmann-type operator.

The number of new solutions generated at each temperature is the same as the iteration number at the temperature which is constrained by the termination condition. The termination condition could be as simple as a certain number of iterations. After all the iterations at a temperature complete, the temperature would be lowered based on the temperature updating rule. At the updated (and lowered) temperature, all required iterations will have to be completed before moving to the next temperature. This process would repeat until the halting criterion is met. The result of simulated annealing (SA) is related to the number of iterations at each temperature and the speed of reducing temperature. The temperature updating rule can be choosen as $T_1 = \alpha^N T_0$. Where N is number of iterations (generations), T_0 is the initial temperature, T_1 $(T_0 > T_1)$ is the final temperature, α is the cooling ratio. The cooling ratio controls the speed of cooling. The higher the cooling ratio, the faster the temperature cools down.

3.3. HybridGSA

GA has been successfully used in a wide range of differentiable, nondifferentiable, and discontinuous optimization problems encountered in statistics, engineering and economics applications [34-35]. However, conventional genetic algorithms has the disadvantages of slow convergence rate, local optimum, and premature convergence might occur when the GA cannot find the optimal solution due to loss of some important characters (genes) in candidate solutions [28].

The performance of GA can be improved by introducing more diversity among the chromosomes in the early stage of the solution process so that premature convergence can be eliminated. A hybrid algorithm, which combines aspects of GA and SA is proposed to overcome the limitations of GA. To implement this idea, the Metropolis acceptance test technique from SA is adopted into GA. The new hybrid algorithm, referenced to as hybridGSA, has been shown to overcome the poor convergence properties of GA and outperform GA or SA.

Örkcü [28] adapted the binary-coded genetic algorithm and simulated annealing methods for variable selection in multiple linear regression. The hybrid method proposed in this study has similar features with Örkcü's method. In this study, hybrid of real-coded genetic algorithms and simulated annealing algorithm (named by hybridGSA) is used for training multi-layer neural networks in the data classification.

By adding this Boltzmann-type operator (SA operator), we obtain a hybrid simulated annealing-genetic algorithm (hybridGSA). The steps of hybridGSA are detailed as follows:

1. A population of potential chromosomes (or solutions) are initialized randomly.
2. The fitness of each chromosome in the population are evaluated.
3. The chromosomes for reproduction with probabilities proportional to fitness are selected.
4. The crossover to the selected chromosomes to produce new chromosomes (children) is applied.
5. The mutation operator to the new chromosomes is applied.
6. The new chromosomes are evaluated.
7. The SA operator to decide which two of the parents and children remain is applied.
8. The temperature as given by $T_1 = \alpha^N T_0$ is decreased.
9. If the stopping criterion is reached, return the best solution. Otherwise, go to step 3.

In this study, training of the artificial neural networks is implemented with hybridGSA method and network structure that has been trained with hybridGSA has been used in the solutions of the classification problems.

4. EXPERIMENTAL RESULTS

The comparison of multi-layer network structures that are trained with BP algorithm, hybridGSA and other optimization algorithms is realized in this section via data obtained from the real classification problems in the literature and large-scale simulation datasets covering the different network architectures.

In order to compare performances of classification methods, cross validation techniques are used. In this study, classification performances of methods are compared with *k-fold* cross validation technique. The *k-fold* cross validation technique proposed by Salzberg [41] was employed in the experiments, with $k = 10$. The data set was thus split into 10 portions, with each part of the data sharing the same proportion of each class of data. Nine data portions were used in the training process, while the remaining part was used in the testing process.

4.1. Some real-data classification problem

The ANN-training methods were evaluated using 3 medical datasets, i.e., Breast Cancer Wisconsin, Pima Indians Diabetes, and Liver Disorders, in the *UCI Machine Learning Repository*. These data sets were taken by Seera and Lim [42]. Table 1 lists the numbers of groups, total units, and variables for each *UCI* data set used in this study. Before the experiments, all variables in the data set were first normalized between 0 and 1. They were then used by hybridGSA for training. Network structure containing different hidden layers (1-*3*-1 and 1-*9*-1) are used when multi-layer structure was trained with hybridGSA. Maximum iteration number is chosen as 10000. For backpropagation algorithm, the learning rate and momentum constant are chosen as 0.1 and 0.9 respectively. The activation function in all cases was the standard sigmoid. Number of neurons in the hidden layer is five in all architects. All of the results are obtained by using the MATLAB programming. Each experiment was repeated 30 times.

4.1.1. Breast Cancer Wisconsin data set

This breast cancer data set was created by Wolberg from the University of Wisconsin. The features used were: (1) clump thickness, (2) uniformity of cell size, (3) uniformity of cell shape, (4) marginal adhesion, (5) single epithelial cell size, (6) bare nuclei, (7) bland chromatin, (8) normal nucleoli, and (9) mitoses [42].

Table 1: Results of hybridGSA and comparison with other methods for the Breast Cancer Wisconsin data set.

Source	Method	Accuracy (%)	Ratio	Runs
Luukka [44]	BC FRPCA1	98.19	1:1	30
	BC FRPCA2	98.13	1:1	30
	BC FRPCA3	98.16	1:1	30
Luukka [45]	Sim	97.49	1:1	30
	Sim + F1	97.10	1:1	30
	Sim + F2	97.18	1:1	30
Örkcü & Bal [25]	Binary-coded GA	94.00	9:1	10
	GA	93.10	9:1	10
	Real-coded GA	96.50	9:1	10
Polat et al. [43]	AIRS	97.20	9:1	30
	Fuzzy-AIRS	98.51	9:1	10
Stoean and Stoean [46]	Cooperative coevolution	96.69	2:1	30
	Decompositional	95.93	2:1	30
	Pedagogical	97.07	2:1	30
	SVMs	96.50	2:1	30
Seera & Lim [42]	FMM-CART-RF	97.29	1:1	30
	FMM-CART-RF	97.57	4:1	30
	FMM-CART-RF	98.84	9:1	30
Proposed method	hybridGSA	98.25	3:1	30
	hybridGSA	98.84	9:1	30

Table 1 shows the overall results. Multi-layer artificial neural network structure, trained with hybridGSA and FMM-CART-RF [42] was found to have the highest correctly classification performances as; 98.84% performance in 9:1 train-to-test ratio. The results from five publications, which were based on different train-to-test ratios, were used for comparison, as shown in Table 1. The Fuzzy-AIRS method [43] produced the best accuracy of 98.51%, followed by BC FRPCA1 method [44] at 98.19%, Sim [45]) at 97.49%, Pedagogical [46] at 97.07% and Real-coded GA [25] at 96.50%. Nevertheless, this performance was inferior to that of hybridGSA.

4.1.2. Pima Indians Diabetes data set

The Pima Indian Diabetes data set was concerned with the presence or absence of diabetes among Pima-Indian women living near Phoenix, Arizona. There were eight features in each data sample: (1) number of times pregnant, (2) plasma glucose concentration a 2 h in an oral glucose tolerance test, (3) diastolic blood pressure, (4) triceps skin fold thickness, (5) 2-h serum insulin, (6) body mass index, (7) diabetes pedigree function, (8) age [42]. The results of hybridGSA and other methods are summarized in Table 2. Multi-layer artificial neural network structure, trained with hybridGSA, was found to have the highest correctly classification performances as; 78.45% performance in 9:1 train-to-test ratio. The results from three publications were compared, as shown in Table 2. The best reported accuracy rates were 75.97% from the Sim + F2 method in Luukka [45], 77.60% from the Real-coded GA method in [25] and 78.39% from the FMM-CART-RF method in [42]. Nevertheless, they were inferior to that of hybridGSA (78.45% accuracy).

Table 2: Results of hybridGSA and comparison with other methods for the Pima Indians data set.

Source	Method	Accuracy (%)	Ratio	Runs
Luukka [45]	Sim	75.29	1:1	30
	Sim + F1	75.84	1:1	30
	Sim + F2	75.97	1:1	30
Örkcü & Bal [25]	Binary-coded GA	74.80	9:1	10
	GA	73.80	9:1	10
	Real-coded GA	77.60	9:1	10
Seera & Lim [42]	FMM-CART-RF	76.56	4:1	30
	FMM-CART-RF	76.87	1:1	30
	FMM-CART-RF	78.39	9:1	30
Proposed method	hybridGSA	77.85	3:1	30
	hybridGSA	78.42	9:1	30

4.1.3. Liver Disorders data set

The Liver Disorders data set was prepared by the BUPA Medical Research Company. The features were obtained from blood tests (the first five features) and daily alcohol consumption (the last feature), as follows: (1) mean corpuscular volume, (2) alkaline phosphatase, (3) alamine aminotransferase, (4) aspartate aminotransferase, (5) gamma-glutamyl transpeptidase, and (6) number of half-pint equivalents of alcoholic beverages drunk per day [42].

Table 3 shows the overall the results. Multi-layer artificial neural network structure, trained with FMM-CART-RF and hybridGSA, was found to have the highest correctly classification performances as; 95.01 and 95.00, respectively. Similar other two data sets, hybridGSA and FMM-CART-RF produced the highest accuracy using the train-to-test ratio of 9:1. The results from five publications, which were based on different train-to-test ratios, were used for comparison, as shown in Table 3. Among to other methods, the best reported accuracy rates were 94.29% from the LSSVM with fuzzy weighting in Çomak et al. [47]. Nevertheless, it was inferior to that of hybridGSA (94.58% accuracy) and FMM-CART-RF method (95.01% accuracy).

Table 3: Results of hybridGSA and comparison with other methods for the Liver Disorders data set.

Source	Method	Accuracy (%)	Ratio	Runs
Çomak et al. [47]	LSSVM	60.00	9:1	--
	LSSVM with fuzzy weighting	94.29	9:1	--
Luukka [44]	Liver FRPCA1	68.25	1:1	30
	Liver FRPCA2	67.88	1:1	30
	Liver FRPCA3	70.25	1:1	30
Özşen & Güneş [48]	AWAIS	70.17	9:1	10
	GA-AWAIS	85.21	9:1	10
Polat et al. [43]	AIRS	81.00	9:1	30
	Fuzzy-AIRS	83.38	9:1	30
Seera & Lim [42]	FMM-CART-RF	93.61	1:1	30
	FMM-CART-RF	93.62	4:1	30
	FMM-CART-RF	95.01	9:1	30
Proposed method	hybridGSA	94.02	3:1	30
	hybridGSA	94.58	9:1	30

4.2. Simulation study

In addition to the real data sets, the performances of methods are compared by many different types of artificially generated data sets. For this study, a whole data set regarding the 500 observations from taken *Uniform (0, 1000)* is used and separated into four different grouping cases (two, three, four and five groups). To make each group, all of the 500 observations are classified by these ranks. In other words, in order to examine classification performances of neural network training methods in the two different architects (different from real-word application; *1-3-1* and *1-9-1*) and different four group case (two, three, four and five groups), eight simulated data configurations are generated from continuous uniform distributions in *three* independent variables. Learning rate, momentum constant, activation function and number of neurons in the hidden layer are defined as in the real-world applications.

To run the simulation experiments, 100 data sets are generated for each of the data configurations. That is, 100 data of sample are used or 100 repeat is used. The *10-fold* classification rates obtained from ANN-training approaches are given in *Figure 1 – Figure 8*.

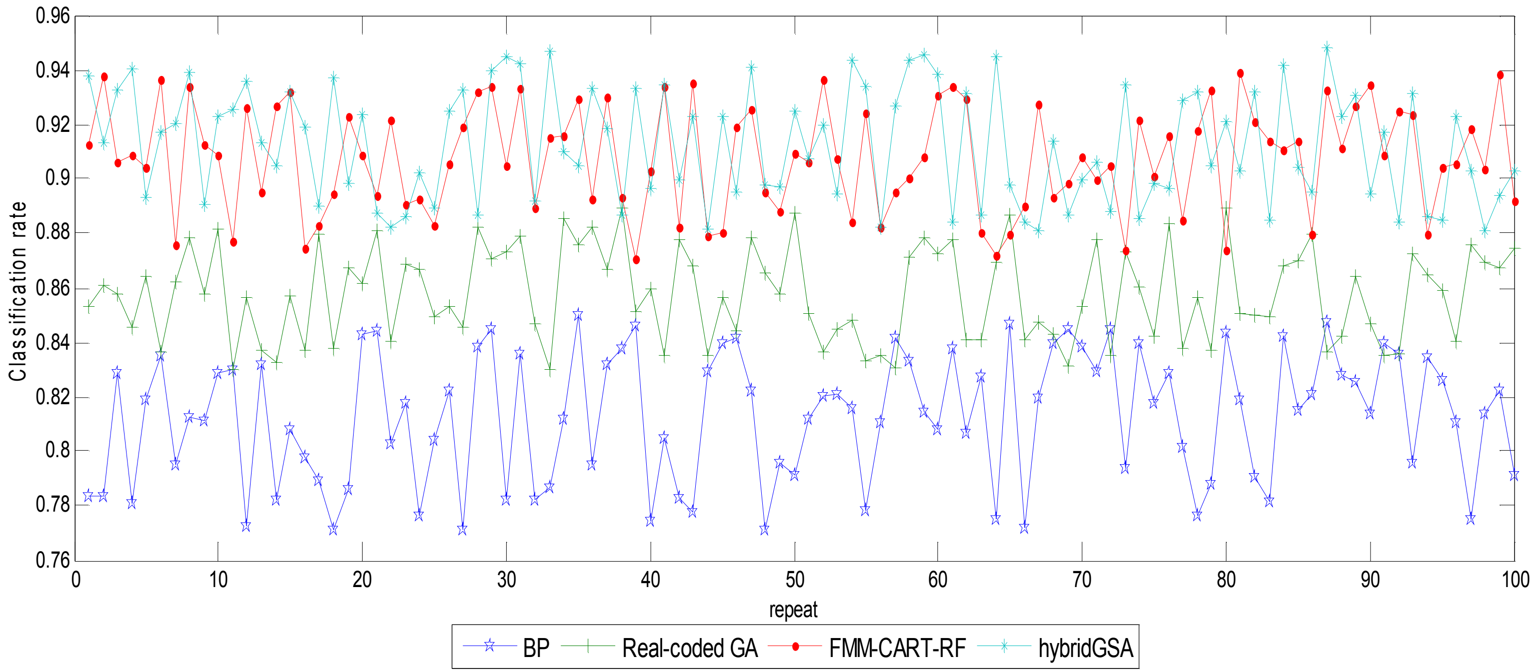

Figure 1: *10-fold* classification rates of methods in *1-3-1* architect in *two-group* case

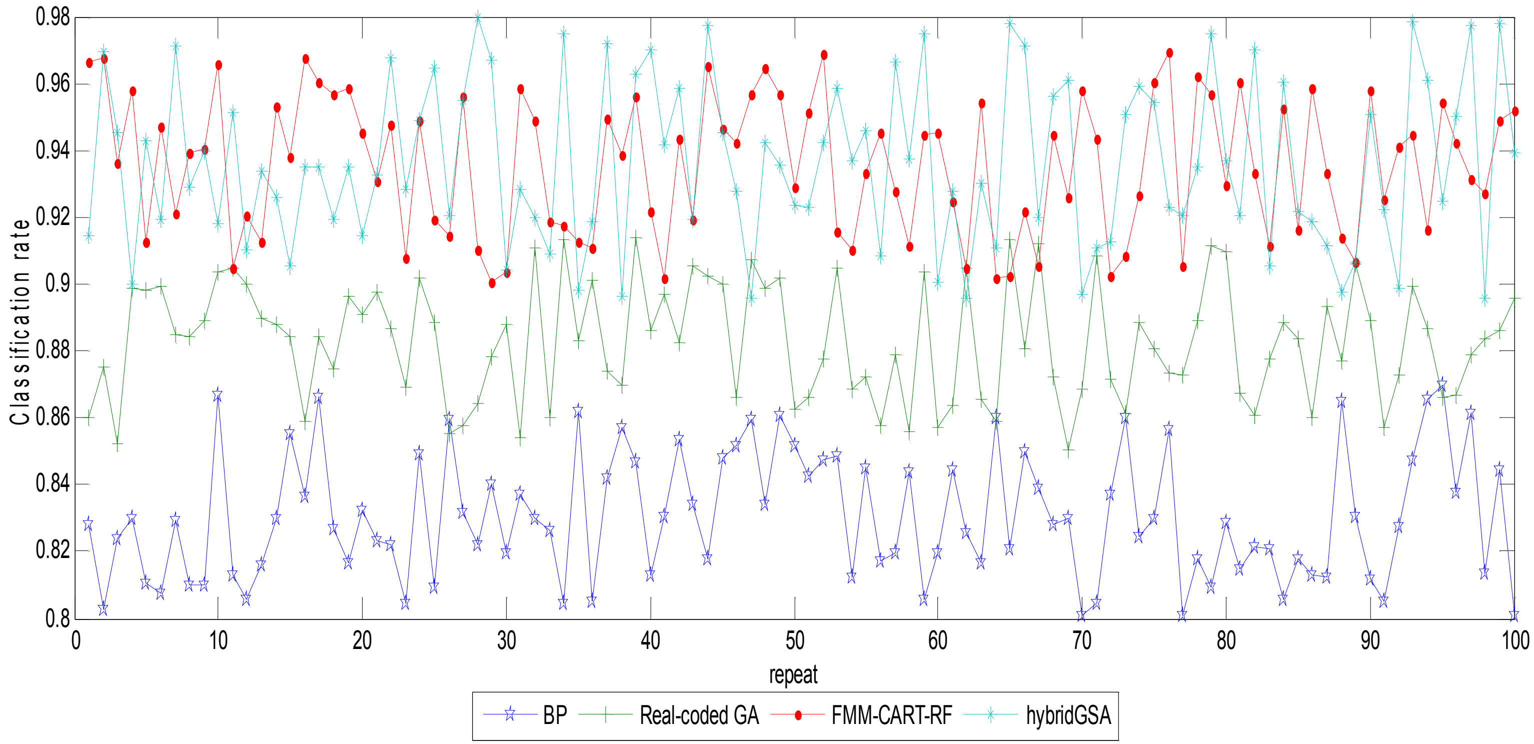

Figure 2: *10-fold* classification rates of methods in *1-9-1* architect in *two-group* case

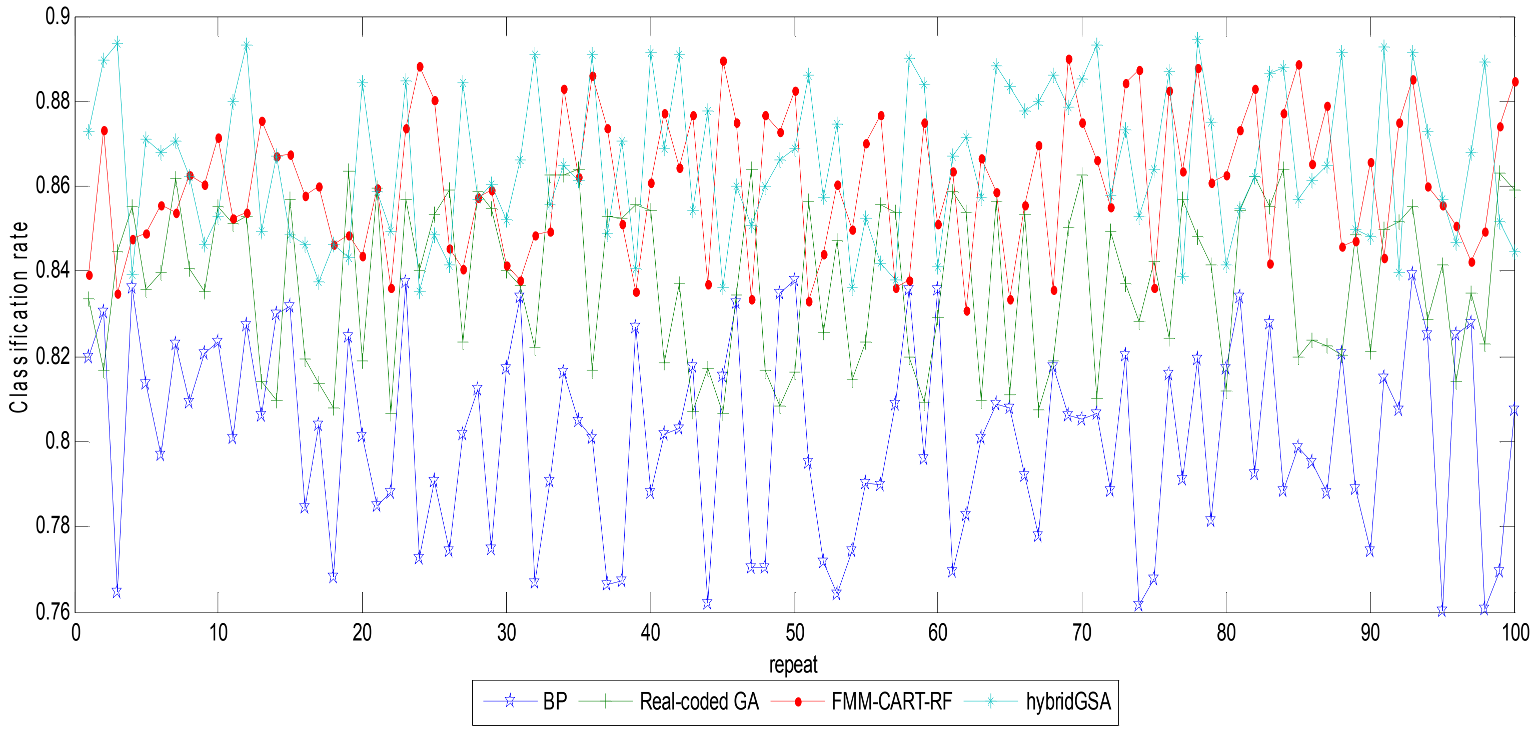

Figure 3: *10-fold* classification rates of methods in *1-3-1* architect in *three-group* case

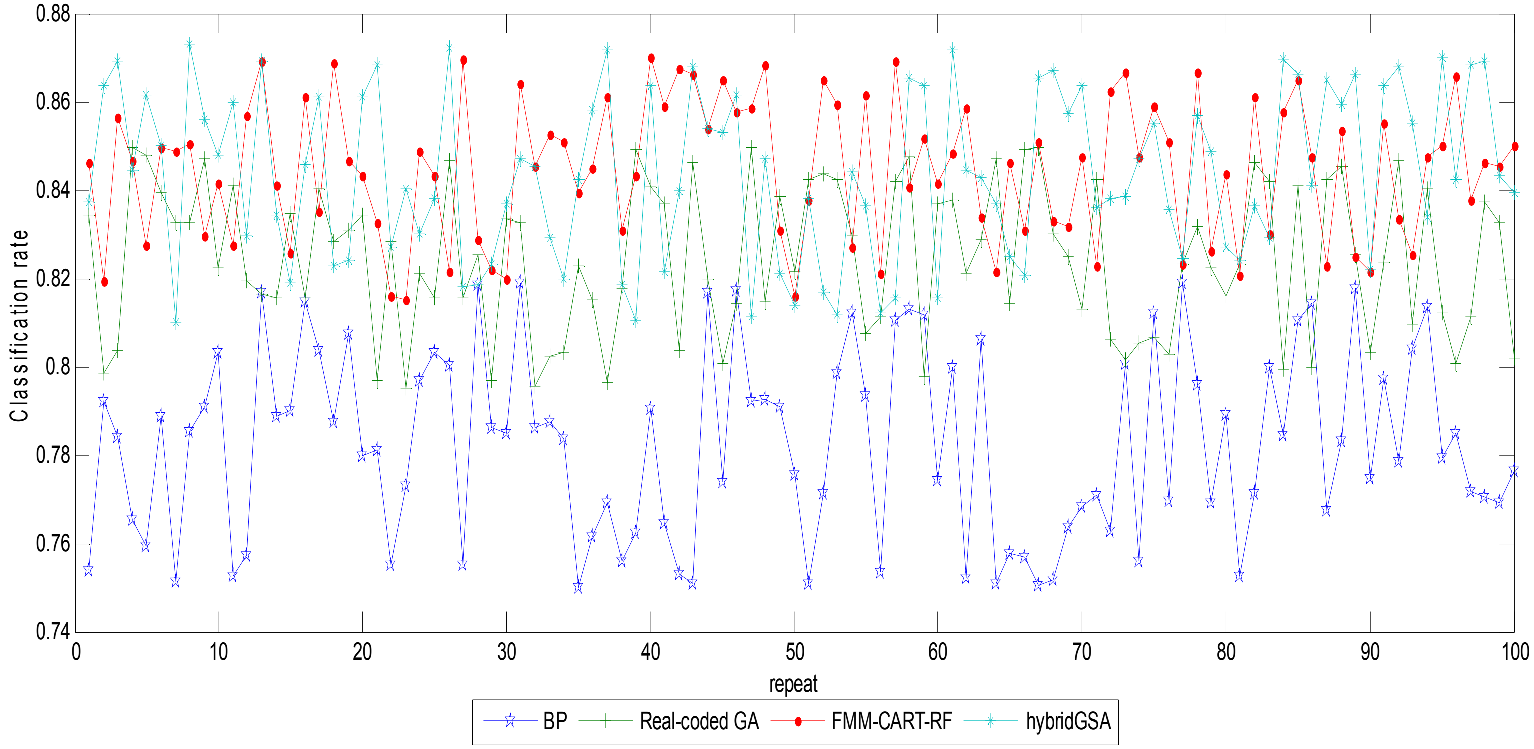

Figure 4: *10-fold* classification rates of methods in *1-9-1* architect in *three-group* case

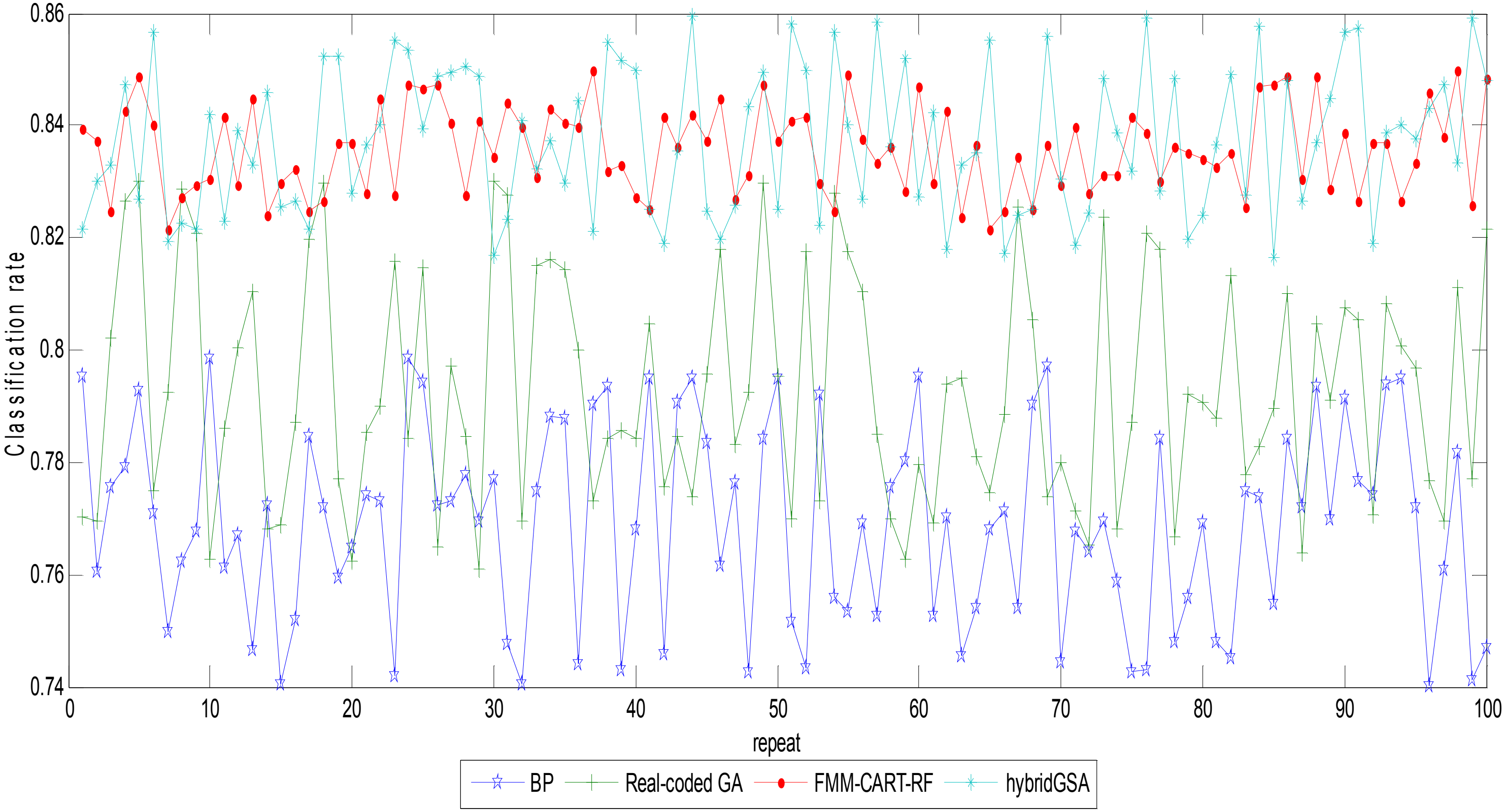

Figure 5: *10-fold* classification rates of methods in *1-3-1* architect in *four-group* case

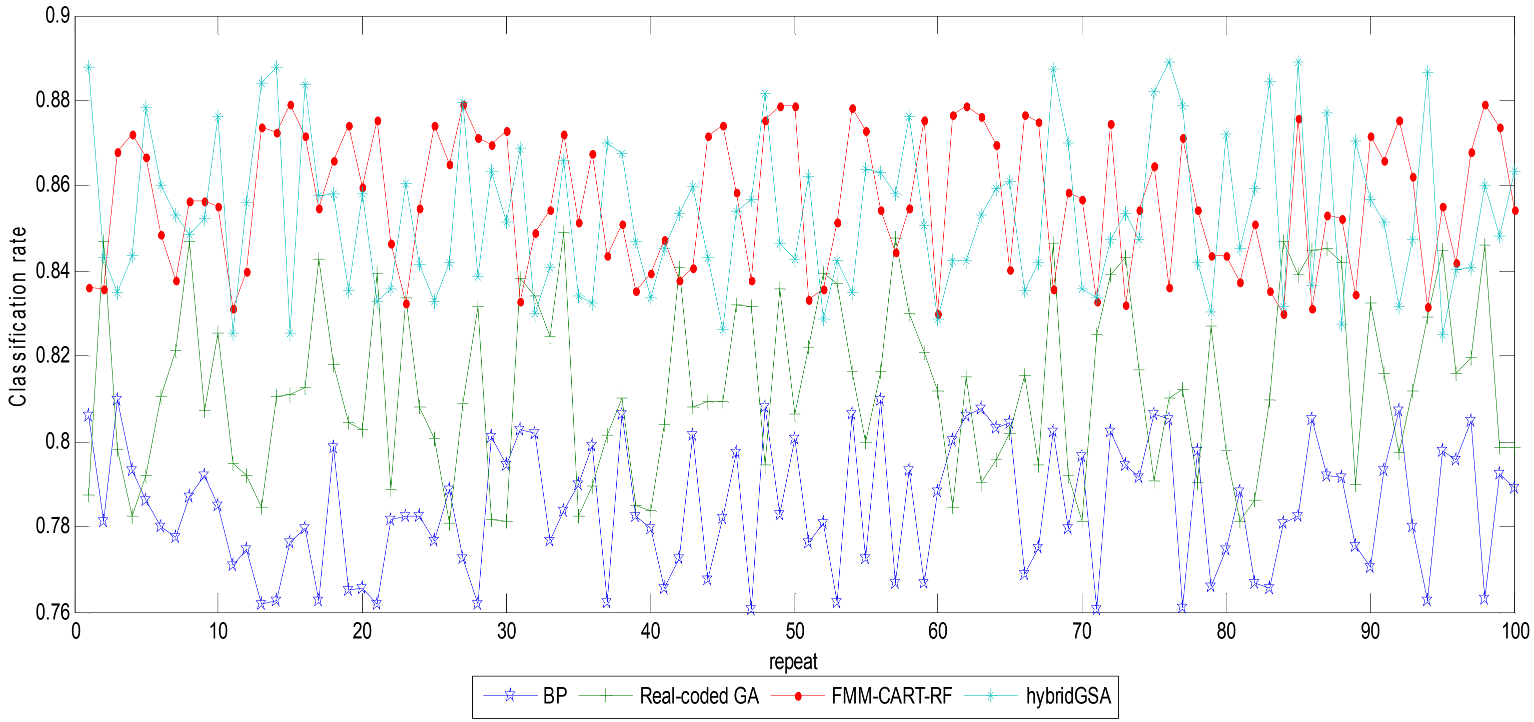

Figure 6: *10-fold* classification rates of methods in *1-9-1* architect in *four-group* case

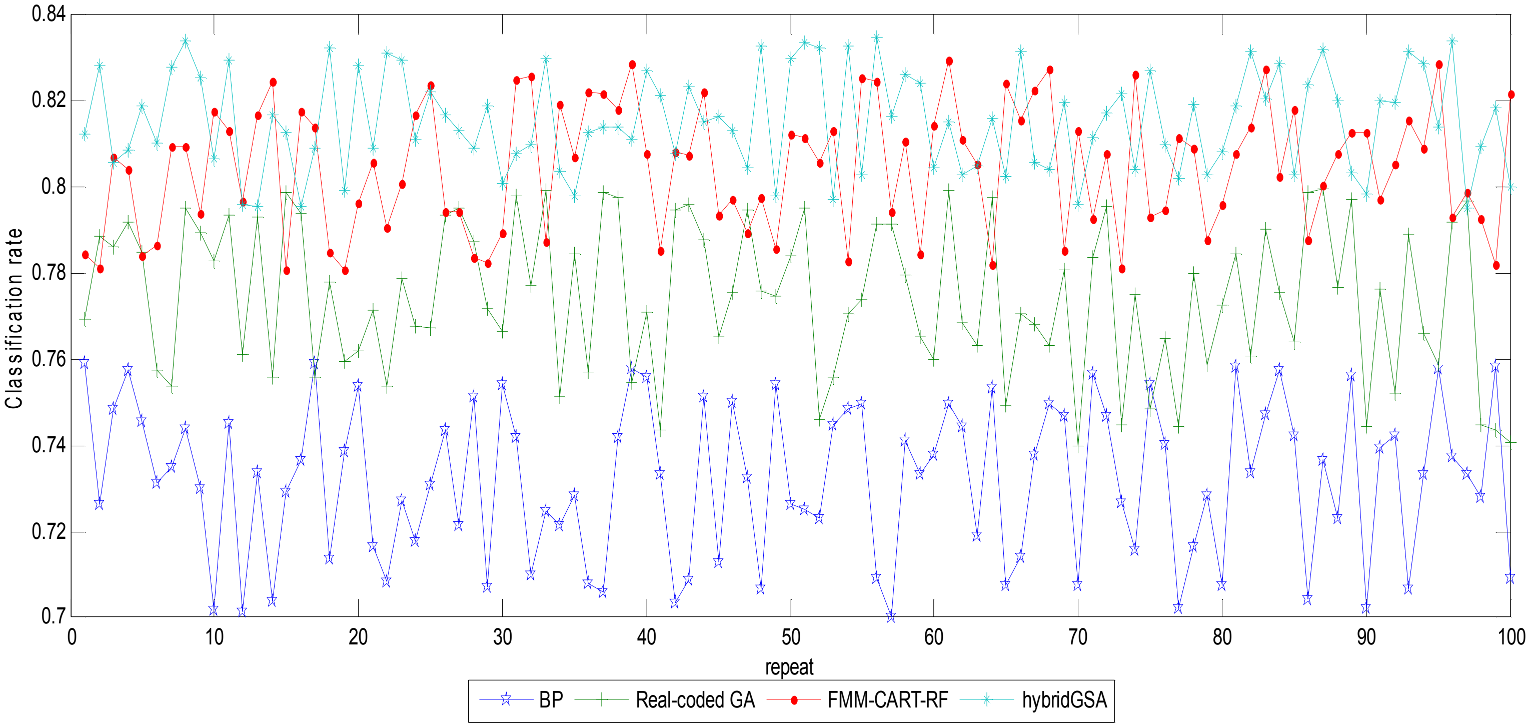

Figure 7: *10-fold* classification rates of methods in *1-3-1* architect in *five-group* case

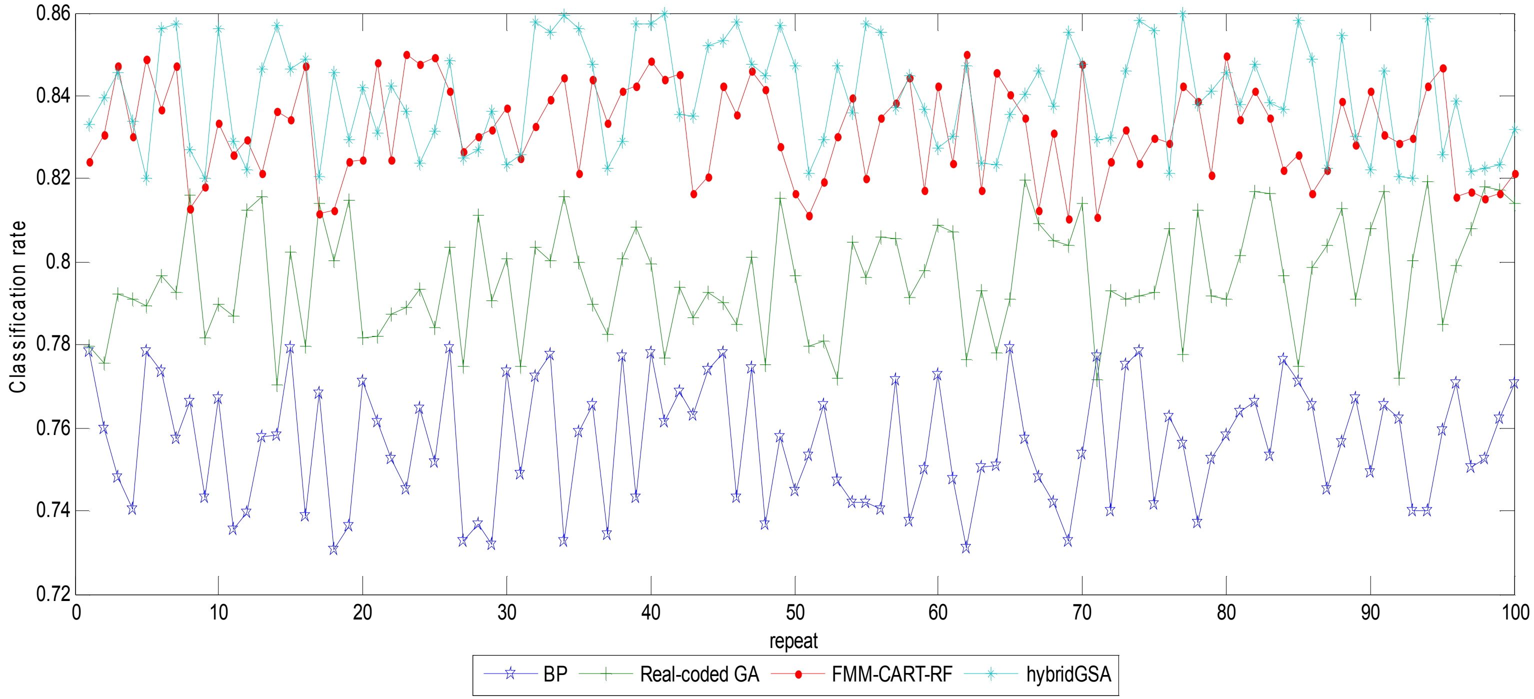

Figure 8: *10-fold* classification rates of methods in *1-9-1* architect in *five-group* case

When these figures are examined, it is observed that the classification success of artificial neural network approach trained with hybridGSA method is better than real-coded genetic algorithm and backpropagation training methods.

5. CONCLUSION

In this paper, a hybrid intelligent model, i.e., hybridGSA, has been proposed for training artificial neural network and undertaking data classification tasks. A series of empirical studies using three benchmark data sets from the *UCI* Machine Learning Repository and a large data simulation experiments have been conducted to evaluate the efficacy of the hybrid model. According to the computational results, it is seen that neural network approach trained with hybridGSA algorithm is indeed capable of providing good classification results than other training methods. The findings reveal that hybridGSA is able to yield better performances in comparison with backpropagation and real coded genetic algorithms. HybridGSA are also comparable with those from other methods published in the literature such as FMM-CART-RF, Fuzzy-AIRS and LSSVM. The hybridGSA method may be efficiently used in the other statistical problems as the chaotic time series, the subset selection problem in multiple linear regression problems, the parameter estimation problem in the nonlinear regression, and data clustering problems.

CONFLICT OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

[1] Fisher, R.A. “The use of multiple measurements in taxonomy problems” ***Annals of Eugenics***, 7, 179–188 (1936).

[2] Xu, G. & Papageorgiou, L.G. “A mixed integer optimization for data classification” ***Computers & Industrial Engineering*** 56(4), 1205-1215 (2009).

[3] Fred, N. & Glover, F. “A linear programming approach to discriminant problem” ***Decision Sciences***, 12, 68–74 (1981).

[4] Fred, N. & Glover, F. “Simple but powerful goal programming models for discriminant problems” ***European Journal of Operational Research***, 7, 44–60 (1981).

[5] Lam, K.F., Choo, E.U., Moy, J.W. “Minimizing deviations from the group mean: A new linear programming approach for the two-group classification problem” ***European Journal of Operational Research***, 88, 358–367 (1996).

[6] Lam, K.F. & Moy, J.W. “Improved linear programming formulations for the multi-group discriminant problem” ***Journal of Operational Research Society***, 47, 1526–1529 (1996).

[7] Lam, K.F. & Moy, J.W. “An experimental comparison of some recently developed linear programming approaches to the discriminant problem” ***Computers and Operations Research*** 24(7), 593-599 (1997).

[8] Sueyoshi, T. “DEA-discriminant analysis in the view of goal programming” ***European Journal of Operational Research***, 115, 564–582 (1999).

[9] Sueyoshi, T. Extended “DEA-Discriminant analysis” ***European Journal of Operational Research***, 131, 324–351 (2001).

[10] Sueyoshi, T. “Mixed integer programming approach of extended DEA-Discriminant Analysis”, ***European Journal of Operational Research***, 152, 45–55 (2004).

[11] Sueyoshi, T. “DEA-Discriminant Analysis: Methodological comparison among eight discriminant analysis approaches” ***European Journal of Operational Research***, 169, 247–272 (2006).

[12] Bal, H., Örkcü, H.H., Çelebioğlu S. “An experimental comparison of the new goal programming and linear programming approaches in the two-group discriminant problems” ***Computers&Industrial Engineering***, 50(3), 296–311 (2006).

[13] Bal, H. & Örkcü, H.H. “Data envelopment analysis approach to two-group classification problems and an experimental comparison with some classification models” ***Hacettepe Journal of Mathematics and Statistics***, 36 (2), 169–180 (2007).

[14] Glen, J.J. “A comparison of standard and two-stage mathematical programming discriminant analysis method” ***European Journal of Operational Research***, 171, 496–515 (2006).

[15] Üney, F. & Türkay, M. “A mixed-integer programming approach to multi-class data classification problem” ***European Journal of Operational Research***, 173, 910–920 (2006).

[16] Bal, H. & Örkcü, H.H. “A new mathematical programming approach to multi-group classification problems” ***Computers&Operations Research***, 38, 105–111 (2011).

[17] Örkcü, H. H., & Bal, H. “A combining mathematical programming method for multi-group data classification” ***Gazi University Journal of Science***, 24(1), 77–84 (2011).

[18] Maskooki, A. “Improving the efficiency of a mixed integer linear programming based approach for multi-class classification problem” ***Computers and Industrial Engineering***, 66, 383–388 (2013).

[19] Denton, J.W., Hung, M.S., Osyk, B.A. “A neural network approach to the classification problem” ***Expert Systems with Applications***, 1(4), 417-424 (1990).

[20] Patwo, E., Hu, M.Y., Hung, M.S. "Two-group classification using neural networks" ***Decision Sciences***, 24(4), 825–845 (1993).

[21] Holmstrom, L., Koistinen, P., Laaksonen, J., Oja, "E. Neural and statistical classifiers taxonomy and two case studies" ***IEEE Trans. Neural Networks***, 8, 5–17 (1997).

[22] Mangiameli, P. & West, D. "An improved neural classification network for the two-group problem" ***Computers and Operations Research***, 26, 443–460 (1999).

[23] Pendharkar, P.C. "A threshold varying artificial neural network approach for classification and its application to bankruptcy prediction problem" ***Computers and Operations Research***, 32, 2561–2582 (2005).

[24] Yim, J. & Mitchell, H. "Comparison of country risk models: hybrid neural network, logit models, discriminant analysis and cluster techniques" ***Expert Systems with Applications***, 26(1), 137-148 (2005).

[25] Örkcü, H. H., & Bal, H. Comparing performances of backpropagation and genetic algorithms in the data classification. ***Expert Systems with Applications***, 38(4), 3703–3709 (2011).

[26] Xu, T., Wei, H., Hu, G. "Study on continuous network design problem using simulated annealing and genetic algorithm" ***Expert Systems with Applications***, 36, 1322-1328 (2009).

[27] Liu, M., Sun, Z.J., Yan, J.W., Kang, J.S. "An adaptive annealing genetic algorithm for the job-shop planning and scheduling problem" ***Expert Systems with Applications***, 38, 9248-9255 (2011).

[28] Örkcü, H. H. "Subset selection in multiple linear regression models: a hybrid of genetic and simulated annealing algorithms" ***Applied Mathematic and Computation***, 219, 11018–11028 (2013).

[29] Hosseini, P. & Shayesteh, M.G. "Efficient contrast enhancement of images using hybrid ant colony optimization, genetic algorithm, and simulated annealing" ***Digital Signal Processing***, 23, 879–893 (2013).

[30] Zameer, A., Mirza, S.M., Mirza, N.M. "Core loading pattern optimization of a typical two-loop 300 MWe PWR using Simulated Annealing (SA), novel crossover Genetic Algorithms (GA) and hybrid GA(SA) schemes" ***Annals of Nuclear Energy***, 65, 122-131 (2014).

[31] Zhang, X.P. "Thresholding neural network for adaptive noise reduction", ***IEEE Transactions on Neural Networks***, 12 (3), 567-584 (2001).

[32] Huang, C.Y., Chen, L.H., Chen, Y.L., Chang, F.M. "Evaluating the process of a genetic algorithm to improve the back-propagation network: A Monte Carlo study" ***Expert Systems with Applications***, 36, 1459–1465 (2009).

[33] Rumelhart, D.E., Hinton, G., Williams, R. "Learning representation by back-propagation errors" ***Nature***, 32(9): 533-536 (1986).

[34] Holland, J. Adaptation in Natural and Artificial Systems. Michigan Press, Michigan, (1975).

[35] Goldberg, D.E. "Genetic algorithms in search, optimization and machine learning", ***Addison Wesley***, Readin, MA (1989).

[36] Gen, M. & Cheng, R. Genetic algorithms and engineering design, ***Wiley***, New York, (1996).

[37] Davis, L., Handbook of genetic algorithms. New York: Van Nostrand Reinhold, (1991).

[38] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E. "Equation of State Calculations by Fast Computing Machines" ***J. Chem. Phys***., 21 (6), 1087-1091 (1953).

[39] Kirkpatrick, S., Gerlatt, C.D., Vecchi, M.P. "Optimization by Simulated Annealing" ***Science***, 220, 671-680 (1983).

[40] Kirkpatrick, S. "Optimization by Simulated Annealing-Quantitative Studies" ***J. Stat. Phys***, 34, 975-986 (1984).

[41] Salzberg, S.L. "On comparing classifiers: pitfalls to avoid and a recommended approach", ***Data Mining and Knowledge Discovery,*** 1, 317-328 (1997).

[42] Seera, M. & Lim, C.P. "A hybrid intelligent system for medical data classification", ***Expert Systems with Applications***, 41, 2239–2249 (2014).

[43] Polat, K., Şahan, S., Kodaz, H., & Güneş, S. "Breast cancer and liver disorders classification using artificial immune recognition system (AIRS) with performance evaluation by fuzzy resource allocation mechanism" ***Expert Systems with Applications***, 32(1), 172–183 (2007).

[44] Luukka, P. "Classification based on fuzzy robust PCA algorithms and similarity classifier" ***Expert Systems with Applications***, 36 (4), 7463–7468 (2009).

[45] Luukka, P. "Feature selection using fuzzy entropy measures with similarity classifier" ***Expert Systems with Applications***, 38 (4), 4600–4607 (2011).

[46] Stoean, R., & Stoean, C. "Modeling medical decision making by support vector machines, explaining by rules of evolutionary algorithms with feature selection" ***Expert Systems with Applications***, 40 (7), 2677–2686 (2013).

[47] Çomak, E., Polat, K., Güneş, S., Arslan, A. "A new medical decision making system: Least square support vector machine (LSSVM) with fuzzy weighting pre-processing" ***Expert Systems with Applications***, 32(2), 409–414 (2007).

[48] Özşen, S., & Güneş, S. "Attribute weighting via genetic algorithms for attribute weighted artificial immune system (AWAIS) and its application to heart disease and liver disorders problems" ***Expert Systems with Applications***, 36(1), 386–392 (2009).