Neural Process Lett
DOI 10.1007/s11063-015-9467-9

Hierarchical Parallel Genetic Optimization Fuzzy ARTMAP Ensemble

Wei Shiung Liew[1] · **Manjeevan Seera**[2] · **Chu Kiong Loo**[1]

Abstract In this paper, a framework for designing optimum pattern classifiers is proposed. The fuzzy ARTMAP (FAM) is first used as a base classifier. Multiple FAM classifiers form an ensemble to improve classification accuracy. Multi-objective genetic algorithms (GAs) are then used to search for the best combinations of variables, for the FAM classifiers. Based on the population of potential solutions, another GA selects the best combination of FAM classifiers to create an ensemble. Individual decisions are combined using a probabilistic voting scheme. To increase the inter-classifier diversity, a hierarchical parallel GA variant and a negative correlation method is employed during the genetic optimization phase for the ensemble evaluation. The proposed framework is evaluated using benchmark and real-world data sets, and the results compared with literature. Results positively indicate the proposed framework is effective in undertaking data classification tasks.

Keywords Fuzzy ARTMAP · Genetic algorithms · Pattern classification · Voting ensemble

1 Introduction

The fuzzy ARTMAP (FAM) neural network is capable of establishing arbitrary mappings between a multi-dimensional analog input space and a multi-dimensional analog output space. It is able to learn a pattern classification task relatively quickly while incorporating

✉ Manjeevan Seera
mseera@gmail.com

Wei Shiung Liew
liew.wei.shiung@gmail.com

Chu Kiong Loo
ckloo.um@um.edu.my

1 Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

2 Faculty of Engineering, Computing and Science, Swinburne University of Technology (Sarawak Campus), Kuching, Sarawak, Malaysia

Published online: 22 August 2015

new information into its existing knowledge base without the need to retrain the previously learned information. The classification accuracy of FAM is reliant on its parameter settings, as well as the order of the training data. Finding the best combination for the highest classification accuracy is essentially an optimization problem.

An optimization task requires a large number of computations to validate every possible permutation, particularly when it involves multiple parameters, with a wide range of values. Genetic algorithms (GAs) are commonly used for its ability to rapidly converge to optimum solutions. Each potential solution is encoded as a chromosome, consisting of all the parameters to be optimized. Each chromosome is then evaluated using a fitness function to determine its effectiveness. Over multiple generations, competitive eliminations will reject chromosomes with sub-par combinations, leaving only the fittest survivors with the best combination of parameter settings. Genetic reproduction then generates variants of the survivors for another round of evaluation and eliminations.

There are limits to what a single neural network classifier can achieve regardless of optimization. As shown in [1], overtraining a FAM may degrade its performance. Alternatively, multiple sub-optimum FAMs can be combined in an ensemble. Ensembles of classifiers operate on the assumption that a consensus by a committee of experts is less likely to be wrong than by a single expert, with higher accuracy rates [2]. Using GAs to develop individual FAMs may lead to reduced genetic variability in the final population of chromosomes. As such, the hierarchical fair-competition parallel GAs (HFCPGAs) was proposed in [3]. The concept is based on having multiple populations of candidates searching in parallel, with limited immigration of solutions between the populations to add genetic diversity while limiting homogeneity.

In this paper, a number of issues are identified. First, finding the optimum combination of parameter settings for individual FAMs to improve the classification accuracy. Second, finding a particular group of FAMs that would give the highest classification accuracy when being combined. Third, improving the diversity of the population and subsequently the ensemble to ensure that the ensemble can achieve the best possible accuracy with the least number of FAM members. While these issues have been addressed individually by various methods, in this paper, we integrate the disparate components into a single unified framework.

The main contribution of this paper is a comprehensive development of an optimized pattern classification system. A variant GA method known as HFCPGA is used in generating a diverse group of FAMs by optimizing training order, feature subset selection, and FAM parameters for individuals. Using the population of optimized FAMs, a second GA is used to determine the optimum combination of individuals in terms of ensemble accuracy and number of ensemble members. Each ensemble is constructed and evaluated using the negative correlation method to identify and remove redundant individuals, followed by a probabilistic voting to combine and integrate the results from individual FAMs into a single ensemble output.

This paper is organized as follow. A literature review on multi-objective and hybrid GAs is first presented in Sect. 2. Inner workings of the FAM classifier, genetic optimization methods, negative correlated ensemble, and the probabilistic voting scheme are described in Sect. 3. In Sects. 4 and 5, experiments using a number of benchmark and real-world data sets are presented, with the results analyzed and discussed. Concluding remarks and suggestions for further work are finally given in Sect. 6.

2 Literature Review

In this section, a literature review on multi-objective (MO) GAs is first presented. This is followed by hybrid GAs. A summary is given at the end.

2.1 Multi-objective Genetic Algorithms

For optimizing a neural network structure in dynamic system modeling, MO GA is proposed in [4]. The aim in [4] is to meet the objectives with good accuracy, while having a minimum model structure. Simulation results show the algorithm is able to accurately identify the examples [4]. A GA-based MO optimization for a neural network (GA-MOO-NN) classifier is proposed for automated diagnosis of breast cancer [5]. GA-MOO-NN simultaneously searches for the most significant feature subsets, while optimizing the network architecture [5]. Experimental results indicate the algorithm outperformed other systems [5]. A MO hybrid GA (MO-HGA) that uses variable neighborhood descent algorithm for local search is proposed in [6]. MO-HGA is used in addressing the thin-film transistor-liquid crystal display module assembly scheduling problem [6]. Experiments based on empirical data are conducted, with the proposed approach acquiring good results as compared to conventional approaches [6].

A MO GA approach in designing electrical distribution networks is presented in [7]. The objectives are monetary cost index and a system failure index [7]. Information acquired from the Pareto-optimal solution is proven to be helpful in decision-making process of the distribution network evolution planning [7]. For sizing and sitting of distributed generation resources into existing distribution networks, a MO method is proposed [8]. The method based on GA allows the planner in deciding the best compromise between cost of power loss, network upgrading, energy not supplied, and energy required by the customers [8]. Effectiveness of the proposed method is proven based on various examples [8].

2.2 Hybrid Genetic Algorithms

A hybrid self-organizing fuzzy neural network with GA (SOFNNGA) for implementation of a Takagi-Sugeno type fuzzy model is proposed in [9]. The hybrid algorithm is based on GA, while recursive least squares estimation is utilized in adjusting parameters and number of fuzzy rules [9]. Good performances of the proposed algorithm is acquired from simulation results [9]. As the Elman neural network contains features of a backpropagation neural network, it inherits problems such as uncertain number of hidden layer neurons, which affects the processing accuracy [10]. In [10], GA is used in optimizing the weights, thresholds, and hidden layer neurons in the Elman network. Results show the training speed and generalization ability is improved in the new model [10].

A hybrid GA method is utilized in finding high-quality solutions for the traveling salesman problem [11]. The method is based on parallel use of multi-population steady-state GA which involves local search heuristics [11]. The large scale traveling salesman problem performed efficiently as compared with other methods [11]. In resolving data mining classification problems, a hybrid genetic based functional link neural network with simultaneous optimization is proposed in [12]. It aims to choose an optimal set of input features using GA by removing features with little to no predictive information [12]. Extensive simulation studies are conducted with results showing the proposed method is robust [12].

In solving global numerical optimization problems with continuous variables, a hybrid Taguchi-GA model is proposed [13]. The model is used in solving fifteen benchmark problems with large dimensions and large numbers of local minima [13]. Experimental results show the proposed model not only finds optimal solutions, but also more robust than algorithms in literature [13]. A stochastic GA is proposed in [14] to improve the search efficiency. The search space is dynamically partitioned into different sections by employing a novel sto-

chastic coding strategy [14]. Experiments on different test functions, on various complexities show the proposed model is able to acquire near-optimal solution in most cases [14].

2.3 Summary

Based on the literature, it can be seen that GA's are well suited in finding optimal solutions for problems with multiple parameters, such as neural network structures. These GA's have been applied in a number of practical applications using benchmark and real data sets. The reported results from the literature shows a positive trend of good accuracy rates in different applications.

3 Classifier Optimization

A number of methods are combined in a single framework to create an ensemble of optimum FAM classifiers which achieves good performances. Given a classification data set, HFCPGA is employed to optimize FAM parameters, feature subsets, and the training order, for training a single FAM. A population of optimum FAM configurations are generated, followed by a second GA for selecting and weighing individual FAMs, which will be grouped into an ensemble. The negative correlation method is used to assemble individual FAMs into an ensemble, while rejecting redundant FAMs. A probabilistic voting system is used to integrate the FAM outputs into a single unified ensemble output. Figure 1 shows the process for creating a population of FAMs with optimized parameters.

3.1 Fuzzy ARTMAP

The FAM architecture is designed based on the adaptive resonance theory of human cognitive information processing [15]. The novelty of the FAM classifier includes the ability

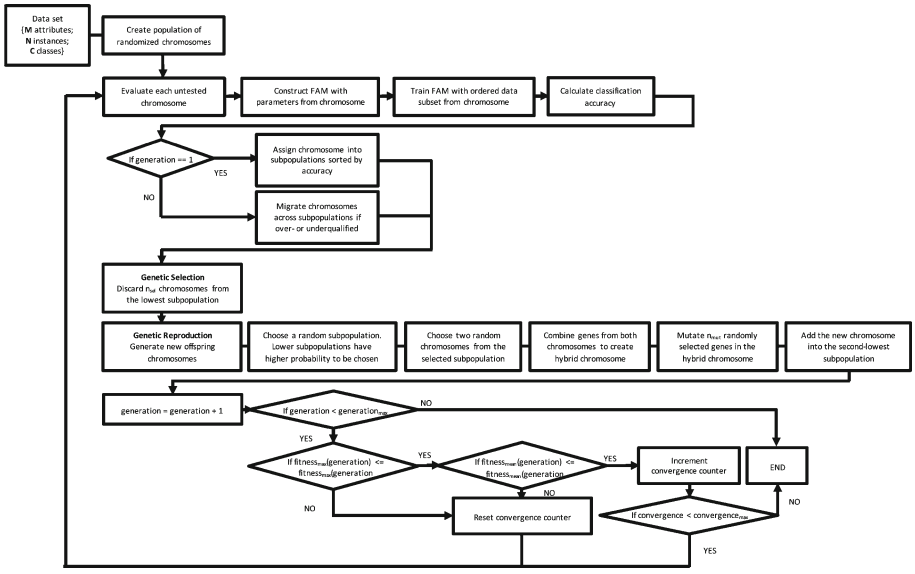

Fig. 1 Flowchart of the methodology for optimizing fuzzy ARTMAP parameters

to organize complex and stable categorical mappings between a series of multidimensional input vectors, and their corresponding category classes. ARTMAP-based classifiers have the *stability-plasticity* characteristics, which allows the classifier to incorporate new information without discarding its current knowledge base.

FAM learns through supervised learning. A single training pattern is represented as a vector of numerical features, and is assigned with a group label. During the supervised learning process, a number of different patterns are presented to FAM for training. FAM consists of two modules connected through a mapping field. The training patterns are given to the input module, while the group labels are given to output module. Association is formed between the training pattern and the label by adjusting the weights of the nodes in the modules and the mapping field. When a pattern is presented to the input to be classified, FAM selects the label with the strongest association. Further details on FAM can be found in [16].

One of the issues in FAM is known as category proliferation. Noise in data sets combined with an inflexible selection process may cause FAM to create a profusion of nodes to match every single training pattern, which does not conform to existing templates. The increased number of nodes results in a large and complex set of fuzzy categories, leading to difficulties in classifying other unknown patterns and reducing the classification accuracy. In the literature, this problem is addressed in a number of ways, such as a modified neural network architecture to mitigate or bypass entirely the category proliferation problem. For example, the Gaussian ARTMAP [17], dARTMAP [18], μARTMAP [19], and TPPFAM [20] all exhibits noise resistance traits, while the Biased ARTMAP [21] modifies the supervised learning process.

The category proliferation issue can be mitigated in FAM by manipulating the order in which training patterns are presented during training. Dagher et al. [22] used a min-max clustering method to identify the training order that would result in a FAM with the best classification accuracy, while Palaniappan and Eswaran [23] used a GA for the same purpose. In addition, the parameters of FAM can be tuned to maximize the classifier performance. Granger et al. [24] used a Particle Swarm Optimization method to determine the best settings for baseline vigilance, choice parameter, learning rate, and match tracking parameter. Similarly, Mohamed et al. [25] used GA to achieve the same objective. In this paper, GA is proposed as a search heuristic to determine the optimum combination of training order and parameter settings, in order to maximize the classification accuracy of a given data set.

3.2 FAM Optimization Using Hierarchical Fair-Competition Parallel Genetic Algorithms

GAs mimics the principles of natural evolution to seek for optimum solutions for a given problem. A popular application for GAs is multi-parameter optimization, where each parameter is represented as a single gene. A chromosome consisting of a number of genes embodies one specific combination of parameter settings. The effectiveness of the chromosome is evaluated by implementing the parameters defined in the genes, as in context of the problem. Given a population of chromosomes, high-fitness solutions are kept to generate new variants for the next round of competitive eliminations, while low-fitness solutions are discarded. At the end of the optimization sequence, the final population consists of the best-performing combination of parameter settings for high-accuracy FAMs.

There is a tendency for solutions to converge and homogenize in GAs after many iterations of competitive eliminations and genetic reproduction. In this case, homogenization is an undesirable side-effect in the intention to create an ensemble of FAM classifiers. Ensembles operate on the rationale that multiple classifiers are more accurate than any individual classifier by combining complementary and conflicting information from different perspectives

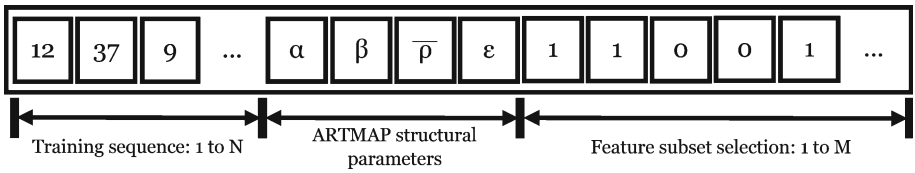

Fig. 2 Structure of a single chromosome, encoding all three parameters for optimizing the performance of a single FAM neural network: the sequence of training data, training feature subset selection, and FAM network parameters

in order to derive the best result [26]. Therefore, an effective ensemble must be composed of a diverse selection of classifiers. To solve the issue of genetic convergence, a variant GA method is applied.

The HFCPGA [3] differs from a simple GA in several aspects. The population of chromosomes are first divided into multiple subpopulations by fitness. All genetic operations such as mutation and reproduction are restricted to chromosomes within each subpopulation. This is to promote fair competition between chromosomes with similar fitness and allowing low-fitness solutions to thrive in their isolated ecosystem. Subsequently, any high fitness chromosomes that may be produced in low-fitness subpopulations are migrated to higher levels. Multiple parallel evolution reduces genetic convergence, thus improving the overall population diversity. At the end of the optimization step, chromosomes in the subpopulations are used as candidates to be assembled into an ensemble of FAMs. The optimum combination of classifiers are selected using a method described in the next section.

Figure 2 shows how all the optimization parameters are encoded as a single chromosome string. The training data sequence is encoded as a string of progressive integers from 1 to N, where N is the number of instances from the data set to be used for training the FAM. Next, feature selection is encoded as a binary string to denote inclusion or exclusion of a particular feature during FAM training. Finally, the FAM parameters consist of the baseline vigilance, learning rate, choice parameter, and match tracking parameters.

The process for optimizing FAM is as follows. A single chromosome is parsed to determine the sequence of the training data to be presented, the features to include and exclude from training, and the parameters used for constructing the FAM. After the construction and training the FAM, the fitness of the FAM is tested in terms of classification accuracy, which is then assigned to the chromosome. Over each generation of the GA, chromosomes with the lowest fitness are discarded and replaced by genetic offspring of the surviving chromosomes. This process is repeated for a number of generations or until a stopping criterion is achieved. The final population of chromosomes would achieve good classification accuracy by keeping and propagating desirable genetic traits. Following this step, the next GA is executed to select the combination of individual FAMs to be assembled into a classifier ensemble.

4 Ensemble Optimization

Ensembles of classifiers typically achieve better classification rates than any single individual classifier by voting. Whereas one classifier may make an erroneous prediction, the other classifiers in the ensemble will most likely select the correct answer. Given a large pool of FAMs generated in the previous step, an ensemble can be created by selecting only a few individuals. The problem lies in determining the best FAMs combinations that results in the best classification accuracy, while minimizing number of individuals in the ensemble.

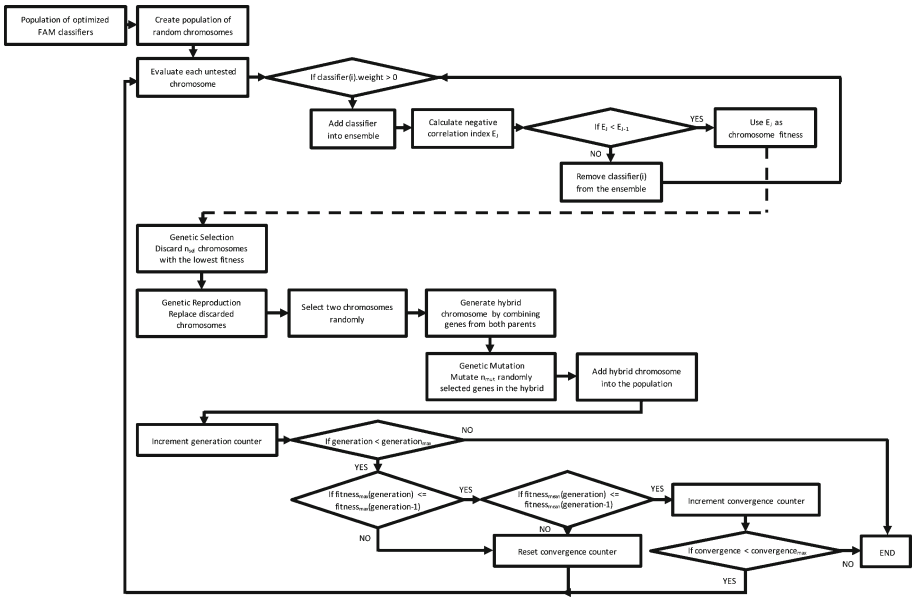

Fig. 3 Flowchart of the methodology for assembling the optimum ensemble of FAMs

A second GA is employed for this purpose. The selection of individual FAMs is designed as a binary problem, and the chromosome for a single ensemble solution is encoded as a string of binary numbers. A "0" indicates that a particular FAM is excluded from the ensemble while a "1" indicates inclusion. The fitness of each chromosome is tested using the overall classification accuracy of the ensemble. Figure 3 shows the process for assembling the optimum ensemble of FAMs.

4.1 Negative Correlation

In theory, given an infinite number of unique FAMs with above average classification ability, any ensemble can achieve perfect classification accuracy by a simple majority voting. The goal therefore is to achieve maximum ensemble classification accuracy while minimizing the number of utilized FAMs. As shown in previous work [27], the mere combination of top-performing FAMs is not enough to produce a better outcome.

A technique, known as negative correlation [28,29] is applied to minimize generalization error of the combined classifiers by removing redundant classifiers. This method can be summarized as follows. As each individual FAM enters the ensemble, the new ensemble is then tested and compared against the performance of the ensemble prior to the addition of the new classifier. If the new ensemble performs worse, then the individual is rejected.

4.2 Probabilistic Ensemble Voting

In theory, ensembles of individual classifiers are able to achieve higher classification accuracy than any of its constituents, by combining complementary information from multiple perspectives. The probabilistic voting strategy used here is partly based on the methods developed by Lin et al. [30] and Loo and Rao [31], and is briefly explained.

Table 1 Data sets used for benchmarking

Data set	Instances	Features	Classes
Cancer	699	9	2
Diabetes	768	8	2
Glass	214	9	6
Heart	270	13	2
Ionosphere	351	34	2
Iris	150	4	3
Segmentation	2310	19	7
Wine	178	13	3

A single instance represents a single exemplar or pattern which consists of a number of numerical attributes or features. Each instance is grouped into any one class category representing a template

The probabilistic voting strategy hinges on the assumption that classifier decisions are independent, and that each classifier has a better than average classification rate, with misclassifications distributed evenly among the residual classes. Given an ensemble of classifiers, each with *a priori* probability for correctly classifying an object. Given an object to be classified, each classifier outputs a number of probable outcomes, where each possible class in which the object can belong to was assigned a probability for being the correct outcome.

Voting is performed by weighting each classifier's output by its *a priori* probability, thus assigning a greater reliability for classifiers with good track record. The most likely outcome is computed by selecting the class with the highest weightage and probability from the combined output of every classifier in the ensemble. In the case where the most reliable classifier makes a mistake, the correct outcome may still be selected given enough support from lesser classifiers. This model has an advantage over winner-take-all by having the ability to analyze the reliability of the best class prediction relative to all other classes. Using the probabilistic voting model, uncertain and weak predictions can be differentiated from unanimous and strong predictions for a more rigorous inspection.

5 Benchmark Data

In this section, a series of experiments are designed to test the performance of the ensembles of FAMs. For testing purposes, the Sonar data set from the UCI Machine Learning Repository [32] is used, while benchmarking is performed using a number of popular data sets. Table 1 presents the list of data sets used in this experiment. Experiments are performed ten times, with the results averaged.

The classification performance of the proposed framework is compared against other contemporary classifier ensemble optimization methods in literature. The literature presented here is selected based on commonalities such as using data sets from the UCI Machine Learning Repository [32]. Accuracy is defined as the ratio of classifications that are assigned to the correct class, over total number of classifications. Table 2 shows a summary of the literature used for comparison. Classification accuracy of the ensemble of FAMs generated using the proposed methodology, as compared to other methods in the literature is shown in Table 3. The HFC-PGA method scored four wins and four losses in the eight compared data sets. Against the best methodology however [40], it scored only one win and three losses.

Table 4 shows the effects of each implemented methodology in the proposed HFCPGA algorithm, using a few of the UCI data sets as examples. For all of the following results,

Table 2 Comparison of ensemble classification methods from literature

Reference	Classifier	Ensemble method
[33]	C4.5 decision trees	Adaboost to create initial ensemble of classifiers. Semi-definite programming used for optimal classifier pruning
		Majority voting used for combination
[34]	Various includes C4.5, random forest, and probabilistic Naive Bayes	GA used for selecting pre-trained classifiers
[35]	Radial basis function	Multi-objective evolution used for optimizing novel objectives based on negative correlation and regularization
[36]	k-Nearest neighbour (kNN), neural network, and SVM	Novel clustering method used to partition data set before training classifiers. Confidence matrices of the base classifiers were combined as input to the fusion classifier
[37]	Multiple, including kNN, linear/quadratic discriminant, and backpropagation	Measure of competence computed and used for selecting classifiers for an ensemble
		Majority voting used for combination
[38]	Decision tree, SVM, and kNN	Double layer voting method. In the first layer, basic ensembles, each consisting of all three classifiers, learn from a training subset
		Output from all basic ensembles are weighted and combined with majority voting
[39]	Extreme learning machine (ELM)	GA used to generate optimum ELM classifiers and select individuals to form a classifier ensemble
[40]	ELM	Novel risk-sensitive hinge loss error function used for selecting and weighting the ELM classifiers in the ensemble

Table 3 Comparison of ensemble classification methods from literature with the proposed HFCPGA-optimized FAM ensemble method, in terms of classification accuracy

Ref	Cancer	Diabetes	Glass	Heart	Iono	Iris	Segment	Wine
[33]	–	0.7466	0.8747	0.7987	0.9254	0.8780	–	–
[34]	–	0.7681	0.7290	–	0.9342	–	–	–
[35]	–	0.7680	–	0.8440	–		**0.9740**	–
[36]	0.9772	0.7108	–	–	0.8909	0.9600	0.9597	0.9905
[37]	0.9625	–	0.7090	–	0.8821	0.9627	0.9639	0.9764
[38]	–	–	–	–	0.8999	0.9521	–	0.9817
[39]	–	0.8457	–	–	–	–	0.9586	–
[40]	**1.000**	**0.8460**	–	**0.9011**	0.9400	–	–	–
HFCPGA	0.9800	0.7617	**0.8832**	0.8778	**0.9658**	**0.9867**	0.9688	**1.000**

The bold values represent the best classification result for that particular data set in the comparison between the proposed methodology and the compared methods [33–40] in literature

Table 4 Comparison of individual and ensemble classification accuracy as each methodology is implemented

	Cancer	Glass	Heart	Iono	Iris	Wine
Base FAM	0.9057	0.7215	0.7632	0.5343	0.9390	0.9624
Base FAM + GA	0.9701	0.7756	0.7917	0.7066	0.9428	0.9653
Ensemble FAM + GA	0.9771	0.7797	0.8222	0.7516	0.9600	0.9944
Base FAM + GA + featsel	0.9538	0.7334	0.8034	0.9311	0.9699	0.9764
Ensemble FAM + GA + featsel	0.9575	0.7944	0.8407	0.9544	0.9800	0.9943
Ensemble FAM + GA + featsel + negcorr	0.9785	0.8271	0.8444	0.9573	0.9867	1.0000
Ensemble FAM + GA + featsel + negcorr + probvot	0.9800	0.8832	0.8778	0.9658	0.9867	1.0000
F1-Score	0.9778	0.8759	0.8754	0.9621	0.9867	1.0000
Area under curve	0.9788	0.9311	0.8733	0.9541	0.9900	1.0000

the base FAM is trained and tested using the ten-fold cross-validation method, and average accuracies of the testing results are reported. Accuracy in this case is the ratio of correctly classified data over all classifications performed. Initially, a population of FAMs is created using randomized parameters, and the mean classification accuracy of each individual is shown in the first row. GA is then applied to optimize the training sequence and ARTMAP parameters of the base FAM for 100 generations. The performance of the individual FAMs are given in the second row. Then, GA is used to assemble an ensemble of FAMs by maximizing overall classification accuracy and minimizing the number of individuals in the ensemble. Results are presented in the third row.

Feature selection is included as one of the GA optimization parameters for individual FAMs. The mean classification accuracy of individual FAMs are presented in row four, and ensemble accuracy in row five. In two of the data sets, the performance of the feature-selected individual FAMs were lower than the FAMs with all features included. However for the other data sets, the feature-selected FAMs displayed better performance than when all features were included. This may indicate the need for a more intelligent feature selection method than just relying on blind selection using the GA, as the current methodology cannot accurately identify which features are noisy. Negative correlation is implemented as part of the ensemble optimization process in order to remove individual FAMs that are redundant from the ensemble. The average classification accuracy are presented in row six. While not shown in the table, the negative correlation method was able to reduce the number of individuals in the ensemble for three of the six tested data sets.

For all of these tests, the ensemble used majority voting to combine multiple FAM outputs into a single classification output. Row seven presents the average classification accuracy when probabilistic voting is used for combination instead. This final step represents the proposed framework in its entirety, encompassing GA optimization of FAM ensembles, with feature subset selection, negative correlation, and probabilistic voting.

Statistical measures of F1-score and area under the curve (AUC) for the classification results of the final ensemble method are presented. The F1-score is a measure of a test's accuracy as a weighted average of the precision (fraction of classified instances that are relevant) and recall (fraction of relevant instances that are classified). An F1-score of 1 indicates its best value, while 0 indicating its worst value. The AUC is derived from the plot of a receiver operating characteristic (ROC) curve, representing the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one.

5.1 Computation Complexity and Running Time

The computation complexity of the algorithm is broken down as follows:

1. During initialization, a number of FAMs (M) are generated, one for every chromosome (N_{pop}) in each subpopulation (N_{subpop}): $t_{init} = M \times t_{create}$.
2. For each FAM, ten-fold cross-validation is used for training and testing. Given the number of data samples in the data set (D), 90 % of the samples are reserved for training ($0.9D$) while the remaining 10 % are used for testing ($0.1D$).
3. Training and testing are repeated ten times, each time using a different test set. The total time taken to train and test a single FAM is: $t_{fam} = 10 \times ((t_{train} \times 0.9D) + (t_{test} \times 0.1D))$.
4. The total time taken for the first generation is: $t_{gen=1} = M \times (t_{create} + t_{fam})$.
5. For each subsequent generation, FAMs with the lowest fitness are discarded. The number of discarded FAMs is equal to the rejection rate multiplied by the total number of FAMs: $N_{rej} = M \times p_{rej}$.

Table 5 Analysis of runtimes using artificially generated data sets of various sizes

Data set instances	Total runtime (mins)	Runtime per instance (mins)
50	5	0.1186
100	16	0.1651
150	30	0.2042
200	46	0.2307
250	72	0.2900
300	107	0.3588
350	133	0.3803
400	183	0.4588
450	213	0.4748
500	280	0.5600

6. For each rejected chromosome, a new FAM is generated through genetic reproduction, and then trained and tested.
7. For each generation after the first, the total time taken is: $t_{gen} = N_{rej} \times (t_{create} + t_{fam})$.
8. This is repeated for G generations, until the stopping criteria or the maximum number of generations is achieved.

The total computational complexity can thus be summarized as:

- First generation: $M \times (t_{create} + t_{fam})$.
- Subsequent generations: $G \times (M \times p_{rej}) \times (t_{create} + t_{fam})$.

The HFCPGA-FAM methodology is capable of generating optimum FAM ensembles through two evolutionary phases. The drawback of the system, however, is the significant amount of computation processes required for the training and testing the FAMs using ten-fold cross-validation method. Runtimes increased almost exponentially as the number of training data is increased. In Table 5, the runtimes of the algorithm are presented using a variety of data set sizes. For a fair comparison, all other parameters are set equally, and the algorithm ran for 100 generations each. The experiment was tested using MATLAB, running on an Intel i7-3770 3.4 GHz processor with 12 GB RAM. MATLAB's Parallel Processing Toolbox is utilized to run multiple t_{fam} operations in parallel, in order to reduce the time taken.

5.2 Number of Subpopulations

The HFCPGA established multiple subpopulations to improve the diversity of solutions generated during the classifier optimization process. In this experiment, the number of subpopulations and chromosome migrations between subpopulations are manipulated and tested to determine their effectiveness in terms of average classification accuracy, population diversity, and the performance of the final classifier ensemble.

Table 6 shows the generated population of FAMs and ensembles when the number of subpopulations and migrations are manipulated for the HFCPGA. When the number of subpopulations is set to 1, no migrations are required. Each setting is evaluated in terms of mean accuracy of individual FAMs in the population, the best ensemble result, and the diversity of the population and the ensemble. A stopping criterion is used to stop the experiment

Table 6 Performing GA with varying number of subpopulations and chromosome migrations

Subpop.	Migrations	Population		Ensemble		Convergence
		MeanAcc	Diversity	MaxAcc	Diversity	
1	–	0.5057	0.3626	0.8100	0.3689	565
2	1	0.5562	0.3463	0.8080	0.2909	1160
2	2	0.5560	0.3068	0.8080	0.2868	400
2	5	0.5133	0.3359	0.8140	0.3387	550
2	10	0.4751	0.2959	0.7940	0.3609	520
5	1	0.6889	0.2543	0.8700	0.2813	2280
5	2	0.6527	0.3294	0.8460	0.3024	995
5	5	0.6622	0.2635	0.8660	0.2614	2130
5	10	0.6945	0.2872	0.8380	0.2370	1725
10	1	0.7690	0.2293	0.9080	0.2347	2365
10	2	0.7327	0.2385	0.8920	0.2308	1720
10	5	0.7509	0.2308	0.8780	0.1811	1870
10	10	0.7345	0.2382	0.8900	0.2457	1715

when three consecutive generations elapsed without any increase in the average and best accuracy across the subpopulations. The experiment is then performed ten times and the results averaged. Convergence in this experiment is measured in terms of the total number of chromosomes generated and tested.

From Table 6, increasing the number of subpopulations generally resulted in longer experiments as the stopping criterion is more difficult to achieve. However, comparing experiments with different number of chromosome searches may be unfair, as the experiment with a higher number of chromosome searches has a better chance of generating good results. For fairness, the experiment was repeated by removing the stopping criterion and running each setting for 100 generations, with results presented in Table 7. One observation was that changing the number of subpopulations and migrations had a negligible effect (about 1 %) on the total runtime.

The differences in results in both tables appears to highlight the potential weakness of the stopping criterion currently used for the framework, which terminates the optimization step whenever the population fitness stagnates for a set number of consecutive generations. The single population GA, while able to converge quickly on the local optima, may stagnate for a long time before being able to locate the proper global optima. HFCPGA, however, is slow to converge but is able to locate the global optima consistently. In the second table, while the average population fitness of the single GA is the worst, the ensemble accuracy is better than some of the HFCPGA, possibly due to better population diversity. However, as seen in the first table, ensemble accuracy is roughly inversely proportional to population diversity. One possibility exists that is there is a more complex relationship between the ensemble accuracy, population diversity, and the quality of the population as characterized by the number of chromosomes searched.

5.3 ARTMAP Parameter Tuning

In the literature [24,25], tuning the FAM parameters produced better classification results. In this experiment, a number of FAM parameter settings are incrementally adjusted and

Table 7 Performing GA with varying number of subpopulations and chromosome migrations, at Generation = 100

Subpop.	Migrations	Population		Ensemble	
		MeanAcc	Diversity	MaxAcc	Diversity
1	–	0.4213	0.3019	0.8420	0.2775
2	1	0.4787	0.2817	0.8480	0.2642
2	2	0.5257	0.2808	0.8560	0.2535
2	5	0.5478	0.2553	0.8540	0.2587
2	10	0.5043	0.2848	0.8340	0.2997
5	1	0.5839	0.2528	0.8400	0.1996
5	2	0.6012	0.2421	0.8160	0.1246
5	5	0.5482	0.2518	0.8480	0.2222
5	10	0.5976	0.2409	0.8400	0.1759
10	1	0.5451	0.2553	0.8420	0.2485
10	2	0.5605	0.2490	0.8340	0.2390
10	5	0.5539	0.2535	0.8200	0.2321
10	10	0.5443	0.2575	0.8400	0.2793

Table 8 Mean and range of FAM parameters from top FAMs in terms of classification accuracy

Parameter	5 %	10 %	20 %
Choice parameter	0.0182 [0, 0.1]	0.0246 [0, 0.1]	0.0213 [0, 0.1]
Learning rate	0.2600 [0.1, 0.5]	0.4908 [0.1, 1.0]	0.3055 [0, 1.0]
Match tracking parameter	0.7500 [0.5, 1.0]	0.7229 [0.5, 1.0]	0.4516 [0.5, 1.0]
Baseline vigilance	0.0400 [0, 0.1]	0.0826 [0, 0.5]	0.1121 [0, 0.5]

subsequently evaluated in terms of classifier accuracy to determine the optimum combination of the ARTMAP parameter settings: choice parameter α, learning rate β, match tracking parameter ϵ, and baseline vigilance $\bar{\rho}$. All other variables such as training sequence remain unchanged.

As there are many combinations of FAM parameter settings to be presented, the results are summarized as follows. A total of 1536 combinations of FAM parameter settings are tested, with the average and best individual FAM classification accuracy of 0.5449 and 0.8090 respectively. Out of the 1536 combinations, approximately 5.2 % achieved the maximum classification accuracy while 36.91 % of the combinations are higher than the mean accuracy. Table 8 shows the optimum range and mean value for each FAM parameter obtained by analyzing the top 5, 10, and 20 % of the FAM combinations in terms of accuracy.

The effect of tuning the choice parameter is presented in detail in [41]. The choice parameter determines the order in which nodes are selected to match against the input training pattern. As explained in [42], a high choice parameter means that the FAM is more prone to memorizing every single pattern instead of learning to associate similar patterns into clusters. As shown in the table, the optimum setting for choice is around 0.02. The learning rate determines how quickly the weights of the nodes are adjusted in response to incoming training

patterns. A lower learning rate is preferable to implement slower incremental learning. Under fast learning conditions, newer training patterns may distort previously learned patterns.

In ARTMAP networks, match tracking raises the vigilance parameter whenever the training pattern is misclassified, triggering a search for another node that correctly predicts the pattern, or select an uncommitted node instead. Extensive testing on the effect of the match tracking parameter is performed in [43]. The optimum parameter setting appeared to be dependent on the amount of inter-class overlap in the training data set, and may thus vary if a different data set is used. However, in most other literatures, the default recommended settings for match tracking is a small non-zero value. For the data set used in this experiment, good results are obtained by setting a higher value for the match tracking parameter than the default used in most experiments in literature involving ARTMAPs.

The baseline vigilance controls the FAM's generalization ability. A high vigilance setting forces FAM to learn very specific exemplars, while a low vigilance allows learning from more abstract patterns. The results in the table indicate that a low vigilance setting benefits FAM for learning this particular data set. When tested with a different data set however, the top 5 % of the best results yielded a different set of ranges for the parameters: choice [0, 0.1], learning [0.5, 1.0], match tracking [0, 0.1], and vigilance [0, 0.5]. This indicates that the optimum settings for the FAM parameters vary if different data sets are used. It may be that the optimum parameter combinations for each data set can only be determined through validation.

6 Real-World Data

The data set consists of physiological signal recordings of electrocardiogram (ECG) and salivary measurements of alpha-amylase and cortisol, obtained from subjects under various stimuli designed to provoke varying levels of stress. The experimental design is based on the trier social stress test (TSST) outlined in [44]. The test was originally designed to study the physiological reaction of participants under controlled stress situations.

Our experiment is divided into two parts: control and stress. Subjects participated in both sessions at different times, and the ECG and salivary measurements are obtained immediately after each task. Post-processing is intended to observe any correlation between stress indicators derived from ECG and the corresponding salivary measurements, which is considered as an objective stress indicator. In total, twenty-two subjects each participated in two sessions, one stress and one control. Four ECG recordings and six salivary measurements were taken from each participant during each session, as illustrated in Fig. 4, for a total of 176 ECG recordings and 264 salivary measurements. Further details on the experimental procedure is given in [27].

The concentrations of salivary alpha-amylase and cortisol are normalized within subject to a range of 0 to 1. The bootstrapped mean is calculated for each session, using 1000 samples at 95 % confidence interval, and the results are presented in Table 9. For the control sessions, alpha-amylase spiked after 10 min of rest, followed by a dip after 10 min, and spiked again 10 min later. Cortisol however showed a steady decline during the resting interval. For stress sessions, alpha-amylase peaked just after the mental arithmetic session followed by a gradual decrease. Cortisol peaked at the 10- and 20-min mark of the resting interval, displaying a slight lag in reaction compared to alpha-amylase.

The Wilcoxon signed rank test is used to determine whether the alpha-amylase and cortisol shows significant difference between control and stress tasks. Significance is computed

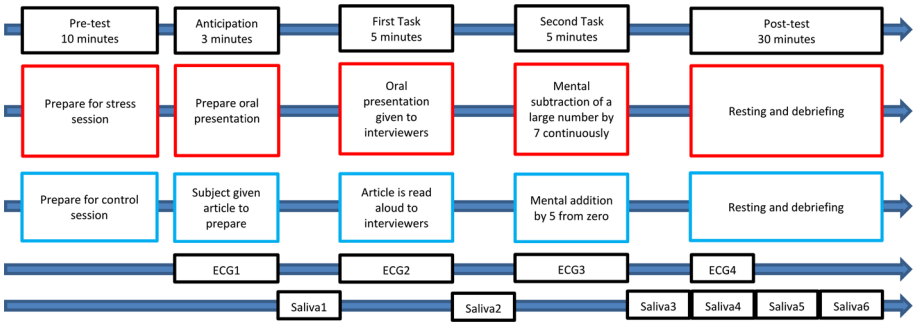

Fig. 4 Proceedings of the TSST experiment to acquire ECG and salivary measurements while subjects performed various tasks. For each session, four ECG and six salivary samples were acquired

Table 9 Bootstrapped mean of alpha-amylase and cortisol concentrations for each TSST session

TSST session	Alpha-amylase	Cortisol
Control: reading silent	0.3803	0.3121
Control: reading aloud	0.3373	0.2671
Control: mental arithmetic	0.3985	0.3369
Control: rest 10 min	0.4687	0.3960
Control: rest 20 min	0.2283	0.3888
Control: rest 30 min	0.4738	0.3474
Stress: prepare interview	0.4024	0.2779
Stress: present interview	0.4248	0.3101
Stress: mental arithmetic	0.5339	0.4628
Stress: rest 10 min	0.4119	0.6172
Stress: rest 20 min	0.4085	0.6025
Stress: rest 30 min	0.3872	0.4915

using the normalized measurements for all subjects, divided according to tasks. The average p-value for alpha-amylase is 0.3808, while cortisol is 0.3341. Both salivary measurements showed no significant difference between control and stress tasks. Feature extraction is performed for each ECG recording. In addition to standard HRV features elaborated in [45], principal dynamic modes (PDM) features are used in an attempt to derive representations of sympathetic and parasympathetic activity. PDM features are computed according to methods outlined in [46] and [47].

It is known that salivary responses have a degree of lag starting from the onset of the stimuli, theorized in [48]. In contrast, any stressful stimulus would provoke an immediate measurable response in cardiac activity. Using the measurements obtained from the experiment, a number of data sets are created by pairing ECG feature vectors to lagged salivary measurements. Referring to Fig. 4, a total of six data sets are created by pairing ECG features to lagged salivary measurements of alpha-amylase and cortisol, as shown in Table 10. Using the salivary measurements as a reference, the data sets are formatted into a binary classification problem: low stress and high stress. The normalized mean of alpha-amylase and cortisol for each data set is used as the threshold value dividing the two classes.

The correlation coefficients between each feature and the derived class labels are computed. The best and worst correlated features for each data set, as well as the mean p-values

Table 10 Data sets created by pairing ECG features to salivary alpha-amylase and cortisol measurements with varying lag

Data set	Pairings				Normalized mean	Class distribution
DatasetAmy0	ECG1	ECG2	ECG3	ECG4	0.4189	97 Low, 79 High
	Amy1	Amy2	Amy3	Amy4		
DatasetAmy1	ECG1	ECG2	ECG3	ECG4	0.4012	96 Low, 80 High
	Amy2	Amy3	Amy4	Amy5		
DatasetAmy2	ECG1	ECG2	ECG3	ECG4	0.4133	98 Low, 78 High
	Amy3	Amy4	Amy5	Amy6		
DatasetCor0	ECG1	ECG2	ECG3	ECG4	0.3731	104 Low, 72 High
	Cor1	Cor2	Cor3	Cor4		
DatasetCor1	ECG1	ECG2	ECG3	ECG4	0.4228	96 Low, 80 High
	Cor2	Cor3	Cor4	Cor5		
DatasetCor2	ECG1	ECG2	ECG3	ECG4	0.4560	90 Low, 86 High
	Cor3	Cor4	Cor5	Cor6		

The normalized mean of the salivary measurements are presented as well as the class distribution of Low and High Stress classes

Table 11 p value correlation coefficients between features and class labels

Data set	Best correlations	Mean p values
DatasetAmy0	NN50 count ($p = 0.0240$), NN20 count ($p = 0.0240$)	0.3936
DatasetAmy1	Total power ($p = 0.0866$), RMSSD ($p = 0.0914$)	0.4317
DatasetAmy2	LF/HF ($p = 0.1523$), Zhong PDM1 ($p = 0.3198$)	0.4684
DatasetCor0	LF/HF ($p < 0.0001$), HF ($p = 0.0027$)	0.4509
DatasetCor1	LF/HF ($p = 0.0110$), HF ($p = 0.0157$)	0.2854
DatasetCor2	HF (p=0.0025), SDSD ($p = 0.0133$)	0.2141

for all features in the data set is given in Table 11. Of the six data sets, the lagged salivary cortisol data sets showed the best correlation to their class labels. The best correlated features in all six data sets are SDSD (average $p = 0.2544$) and the LF/HF ratio (average $p = 0.1662$), which is surprisingly more correlated to the labels than the PDM features (average $p = 0.4750$).

A number of data sets are selected for testing the proposed framework. The first data set is an attempt to correlate heart rate variability to stress indicators derived from salivary measurements, similar to the data set presented in [27]. To that end, six data sets are created, three for alpha-amylase and three for cortisol, with varying levels of lag between the HRV features and the salivary measurements. The experiment is performed ten times for each data set and the results averaged. The reported results are ratio of correctly classified patterns over all classified patterns, using ten-fold cross-validation. The results of the experiment is presented in Table 12. The population is evaluated at the end of the FAM optimization step, in terms of average and best classifier accuracy of the individual FAM. Subsequently, the best ensemble is selected based on having the highest overall accuracy with the lowest number of member FAMs. The experiment is performed ten times and the results averaged. For the statistical measures of F1-score and AUC, it is assumed that Low is Positive and High is Negative.

Table 12 Mean and best classification accuracy of the population of FAMs, and the performance of the generated ensemble in terms of accuracy, F1-scores, and AUC

Data set	FAM		Ensemble			
	Mean	Best	Acc.	Size	F1-score	AUC
DatasetAmy0	0.6718	0.7642	0.8235	6.10	0.7808	0.7597
DatasetAmy1	0.6318	0.6928	0.7445	6.70	0.6219	0.6224
DatasetAmy2	0.6337	0.6785	0.7318	6.40	0.6894	0.6826
DatasetCor0	0.7814	0.8266	0.8648	6.20	0.8489	0.8457
DatasetCor1	0.7669	0.8200	0.8843	8.20	0.8793	0.8730
DatasetCor2	0.7104	0.7400	0.7916	8.20	0.7913	0.7912

In terms of overall classification accuracy, Cortisol data sets are shown to be better than Alpha-amylase for stress classification, although it should be noted that DatasetCor1 and DatasetCor2 have a larger ensemble size as well. Statistic-wise, the Cortisol data sets and DatasetAmy0 show good F-scores and AUC. While DatasetCor0 has the best average individual FAM classification accuracy, it appeared that DatasetCor1 is able to produce a better ensemble with a trade-off of having a larger ensemble.

7 Conclusions

In this paper, a framework of methods for generating an optimum ensemble of FAM classifiers has been presented. A two-step GA method was used, first to generate optimum configurations of the FAM classifier, and the second to select combinations of FAMs to be integrated into an ensemble. Negative correlation was used to remove redundant FAMs from the ensemble. The output from the FAMs in the ensemble was combined by means of a probabilistic voting strategy. The proposed framework was compared with other classifier ensemble methods in literature, and showed good performances in multiple benchmark data sets. In particular, the framework scored highest in four data sets over all other ensemble methods, and second place in three other data sets. The framework was then applied towards correlating stress measures between heart rate measurements and salivary measurements, which was able to produce classifier ensembles that can accurately classify low or high salivary stress from the heart rate.

Future work will look into implementing a method for subsampling larger data sets, in order to improve the algorithm speed. The framework in its current incarnation may be impractical when using data sets with a very large number of training instances, especially for locating an optimum sequence of training involving the entire data set. A potential solution would be to implement a method for scaling the GA to allow selection of a smaller subset of the data set to train a single FAM.

Acknowledgments This research is supported by University of Malaya High Impact Research Grant UM.C/625/1/HIR/MOHE /FCSIT/10.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Georgiopoulos M, Koufakou A, Anagnostopoulos GC, Kasparis T (2001) Overtraining in fuzzy ARTMAP: myth or reality? In: Proceedings of the 2001 international joint conference on neural networks (IJCNN 2001), vol 2. pp 1186–1190
2. Dvzeroski S, Zenko B (2004) Is combining classifiers with stacking better than selecting the best one? Mach Learn 54(3):255–273
3. Hu JJ, Goodman ED (2002) The hierarchical fair competition (HFC) model for parallel evolutionary algorithms. In: Proceedings of the 2002 IEEE congress on evolutionary computation. IEEE, pp 49–54
4. Loghmanian SMR, Jamaluddin H, Ahmad R, Yusof R, Khalid M (2012) Structure optimization of neural network for dynamic system modeling using multi-objective genetic algorithm. Neural Comput Appl 21(6):1281–1295
5. Ahmad F, Isa NAM, Hussain Z, Sulaiman SN (2013) A genetic algorithm-based multi-objective optimization of an artificial neural network classifier for breast cancer diagnosis. Neural Comput Appl 23(5):1427–1435
6. Chou CW, Chien CF, Gen M (2014) A multiobjective hybrid genetic algorithm for TFT-LCD module assembly scheduling. IEEE Trans Autom Sci Eng 11(3):692–705
7. Carrano EG, Soares LA, Takahashi RH, Saldanha RR, Neto OM (2006) Electric distribution network multiobjective design using a problem-specific genetic algorithm. IEEE Trans Power Deliv 21(2):995–1005
8. Celli G, Ghiani E, Mocci S, Pilo F (2005) A multiobjective evolutionary algorithm for the sizing and siting of distributed generation. IEEE Trans Power Syst 20(2):750–757
9. Leng G, McGinnity TM, Prasad G (2006) Design for self-organizing fuzzy neural networks based on genetic algorithms. IEEE Trans Fuzzy Syst 14(6):755–766
10. Ding S, Zhang Y, Chen J, Jia W (2013) Research on using genetic algorithms to optimize Elman neural networks. Neural Comput Appl 23(2):293–297
11. Nguyen HD, Yoshihara I, Yamamori K, Yasunaga M (2007) Implementation of an effective hybrid GA for large-scale traveling salesman problems. IEEE Trans Syst Man Cybern Part B 37(1):92–99
12. Dehuri S, Cho SB (2010) A hybrid genetic based functional link artificial neural network with a statistical comparison of classifiers over multiple datasets. Neural Comput Appl 19(2):317–328
13. Tsai JT, Liu TK, Chou JH (2004) Hybrid Taguchi-genetic algorithm for global numerical optimization. IEEE Trans Evol Comput 8(4):365–377
14. Tu Z, Lu Y (2004) A robust stochastic genetic algorithm (StGA) for global numerical optimization. IEEE Trans Evol Comput 8(5):456–470
15. Carpenter GA, Grossberg S, Reynolds J (1991) ARTMAP: a self-organizing neural network architecture for fast supervised learning and pattern recognition. In: Proceedings of the 1991 international joint conference on neural networks (IJCNN 1991) vol 1. pp 863–868
16. Carpenter GA, Grossberg S, Marzukon N, Reynolds JH, Rosen DB (1992) Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans Neural Netw 3(5):698–713
17. Williamson JR (1996) Gaussian ARTMAP: a neural network for fast incremental learning of noisy multidimensional maps. Neural Netw 9(5):881–897
18. Carpenter GA, Milenova BL, Noeske BW (1998) Distributed ARTMAP: a neural network for fast distributed supervised learning. Neural Netw 11(5):793–813
19. Gómez-Sánchez E, Dimitriadis YA, Cano-Izquierdo JM, López-Coronado J (2002) muARTMAP: use of mutual information for category reduction in Fuzzy ARTMAP. IEEE Trans Neural Netw 13(1):58–69
20. Zhang Y, Ji H, Zhang W (2014) TPPFAM: use of threshold and posterior probability for category reduction in fuzzy ARTMAP. Neurocomputing 124:63–71
21. Carpenter GA, Gaddam SC (2010) Biased ART: a neural architecture that shifts attention toward previously disregarded features following an incorrect prediction. Neural Netw 23(3):435–451
22. Dagher I, Georgiopoulos M, Heileman GL, Bebis G (1999) An ordering algorithm for pattern presentation in fuzzy ARTMAP that tends to improve generalization performance. IEEE Trans Neural Netw 10(4):768–778
23. Palaniappan R, Eswaran C (2009) Using genetic algorithm to select the presentation order of training patterns that improves simplified fuzzy ARTMAP classification performance. Appl Soft Comput 9(1):100–106
24. Granger E, Henniges P, Oliveira L, Sabourin R (2006) Particle swarm optimization of fuzzy ARTMAP parameters. In: Proceedings of the 2006 international joint conference on neural networks. pp 2060–2067
25. Mohamed MA, Hegazy AEF, Badr AA (2011) Evolutionary fuzzy ARTMAP approach for breast cancer diagnosis. Int J Comput Sci Netw Secur 11(4):77–84

26. Hansen LK, Salamon P (1990) Neural network ensembles. IEEE Trans Pattern Anal Mach Intel 12(10):993–1001
27. Loo CK, Cheong SF, Seldon M, Mand A, Muthu K, Liew WS, Lim E (2012) Genetic-optimized classifier ensemble for cortisol salivary measurement mapping to electrocardiogram features for stress evaluation. In: PRICAI 2012: trends in artificial intelligence. pp 274–284
28. Lee H, Kim E, Pedrycz W (2012) A new selective neural network ensemble with negative correlation. Appl Intell 37(4):488–498
29. Liu Y, Yao X, Higuchi T (2000) Evolutionary ensembles with negative correlation learning. IEEE Trans Evol Comput 4(4):380–387
30. Lin X, Yacoub S, Burns J, Simske S (2003) Performance analysis of pattern classifier combination by plurality voting. Pattern Recogn Lett 24(12):1959–1969
31. Loo CK, Rao MVC (2005) Accurate and reliable diagnosis and classification using probabilistic ensemble simplified Fuzzy ARTMAP. IEEE Trans Knowl Data Eng 17(11):1589–1593
32. Frank A, Asuncion A (2011) UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed March 2012
33. Zhang Y, Burer S, Street WN (2006) Ensemble pruning via semi-definite programming. J Mach Learn Res 7:1315–1338
34. Ordóñez FJ, Ledezma A, Sanchis A (2008) Genetic approach for optimizing ensembles of classifiers. In: FLAIRS conference. pp 89–94
35. Chen H, Yao X (2010) Multiobjective neural network ensembles based on regularized negative correlation learning. IEEE Trans Knowl Data Eng 22(12):1738–1751
36. Verma B, Rahman A (2012) Cluster-oriented ensemble classifier: impact of multicluster characterization on ensemble classifier learning. IEEE Trans Knowl Data Eng 24(4):605–618
37. Woloszynski T, Kurzynski M, Podsiadlo P, Stachowiak GW (2012) A measure of competence based on random classification for dynamic ensemble selection. Inf Fusion 13(3):207–213
38. Wei H, Lin X, Xu X, Li L, Zhang W, Wang X (2014) A novel ensemble classifier based on multiple diverse classification methods. 2014 11th international conference on fuzzy systems and knowledge discovery. pp 301–305
39. Xue X, Yao M, Wu Z, Yang J (2014) Genetic ensemble of extreme learning machine. Neurocomputing 129:175–184
40. Sachnev V, Ramasamy S, Sundaram S, Kim HJ, Hwang HJ (2015) A cognitive ensemble of extreme learning machines for steganalysis based on risk-sensitive hinge loss function. Cogn Comput 7(1):103–110
41. Georgiopoulos M, Fernlund H, Bebis G, Heileman GL (1996) Order of search in fuzzy ART and fuzzy ARTMAP: effect of the choice parameter. Neural Netw 9(9):1541–1559
42. Carpenter GA, Gjaja MN (1994) Fuzzy ART choice functions. In: Proceedings of the 1993 world congress on neural networks (WCNN 1994), vol 4. pp 133–142
43. Anagnostopoulos GC, Bharadwaj M, Georgiopoulos M, Verzi SJ, Heileman GL (2003) Exemplar-based pattern recognition via semi-supervised learning. In: Proceedings of the 2003 international joint conference on neural networks (IJCNN 2003), vol 4. pp 2782–2787
44. Kirschbaum C, Pirke K-M, Hellhammer DH (1993) The trier social stress test-a tool for investigating psychobiological stress responses in a laboratory setting. Neuropsychobiology 28:76–81
45. Malik M, Bigger JT, Camm AJ, Kleiger RE, Malliani A, Moss AJ, Schwartz PJ (1996) Heart rate variability standards of measurement, physiological interpretation, and clinical use. Eur Heart J 17(3):354–381
46. Zhong Y, Wang H, Ju KH, Jan K-M, Chon KH (2004) Nonlinear analysis of the separate contributions of autonomic nervous systems to heart rate variability using principal dynamic modes. IEEE Trans Biomed Eng 51(2):255–262
47. Choi J, Gutierrez-Osuna R (2009) Using heart rate monitors to detect mental stress. In: 6th international workshop on wearable and implantable body sensor networks. pp 219–223
48. Engert V, Vogel S, Efanov SI, Duchesne A, Corbo V, Ali N, Pruessner JC (2011) Investigation into the cross-correlation of salivary cortisol and alpha-amylase responses to psychological stress. Psychoneuroendocrinology 36(9):1294–1302