International Journal of Innovative Computing, Information and Control
Volume **8**, Number **11**, November **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 7543–7560

GENERAL PATTERN LEARNING AND RECOGNITION USING GENETICALLY-OPTIMIZED TRAINING OF A BIASED ARTMAP ENSEMBLE VOTING SYSTEM

Chu Kiong Loo[1], Wei Shiung Liew[1] and Md. Shohel Sayeed[2]

[1]Faculty of Computer Science and Information Technology
University of Malaya
Kuala Lumpur 50603, Malaysia
ckloo.um@um.edu.my; liew.wei.shiung@gmail.com

[2]Faculty of Information Science and Technology
Multimedia University
Jalan Ayer Keroh Lama 75450 Bukit Beruang, Melaka, Malaysia
shohel.sayeed@mmu.edu.my

Received July 2011; revised November 2011

Abstract. *This paper is an attempt to create a method and system for generating an optimal machine-based pattern recognition system. If successful, this allows any given classifier to improve its classification accuracy by introducing a genetic optimization pre-processing method combined with a probabilistic ensemble voting system. Several problems were identified, and the solutions proposed were based on literature review on similar fields of research. The proposed system utilizes a Fuzzy ARTMAP variant, Biased ARTMAP, as the core pattern learning and classification method for extracted features due to its ability for incremental learning and a biasing parameter which improves its online learning capability over the traditional Fuzzy ARTMAP. One weakness in the ARTMAP system is the effect of the training data sequence on the ARTMAPs learning processes, and consequently, its classification accuracy. A genetic permutation method is proposed to solve this problem by optimizing the training data sequence over several generations of genetic mating and mutation operations. The best training sequences are selected to train multiple Biased ARTMAPs and combined in a probabilistic voting system to determine the final class prediction. Classification performance of the voting system can be improved by implementing a reliability threshold to filter unreliable predictions from the final results. Genetic optimization of the training process combined with the probabilistic voting system improved the Biased ARTMAPs classification accuracy to 75% – 87%, up from 67% using only the Biased ARTMAP system.*

Keywords: Biased ARTMAP, Pattern recognition, Genetic optimization, Probabilistic voting

1. **Introduction.** In 1965, Gordon Moore states that the number of transistors that can be placed in an integrated circuit doubles roughly every 2-3 years for the next few decades [1]. This axiom has proved correct, as computing power has increased in leaps and bounds over the machines of the previous generation. Software developers raced to design and develop new ways to take advantage of the massive amounts of computing power that is available at a relatively inexpensive cost. Examples include extremely realistic computer-generated imagery (CGI) being used to develop high budget movies and computer games, massively distributed computing networks to compute solutions to research projects, and even personal computers being used for high-quality work using consumer-level applications.

The field of artificial intelligence (AI) benefits greatly from the increase in available computing power. The increasing complexity in software and application development forces developers to rely more on automated algorithms to perform complex computations. One objective in the field of AI is to program machines to imitate human thought processes. An example is to train a machine in optical character recognition skills [2] to simulate the method in which humans are able to solve CAPTCHA tests. Similarly, the field of content-based image retrieval (CBIR) such as facial recognition has a wide array of potential applications as algorithms become increasingly sophisticated in data mining methods from still images and video recordings.

One essential component to implement an algorithm with pattern recognition capabilities is the learning process in which the chosen learning/classification unit is trained to recognize specific patterns. The ARTMAP neural network model, developed by Grossberg and Carpenter [3], was designed to model some aspects of the human thought processes, including pattern recognition and classification. ARTMAPs rely on a pattern matching process which compares an input with the internal memory. When a pattern is successfully matched in the memory, the memorized pattern is reinforced using data incorporated from the incoming pattern. Otherwise, the system uses the input to create a new pattern in the memory. ARTMAP-based systems are capable of incremental learning which makes it suitable for real-time applications.

The design of the ARTMAPs learning processes creates a situation in which certain sequences of data presentation with a specific featural attention can distort the pattern learning process, reducing the systems recognition effectiveness. Under the controlled conditions of offline training, repeated presentation of the same training data will eventually correct such distortions. Real-time learning provides no such accommodation, and the network must be capable of learning completely new patterns as they are presented. In response, a method is devised to overcome the problem of overemphasis on early critical features using a biasing method [4] to selectively ignore previously activated patterns whenever the system makes a predictive error. During future pattern searching, these features which activated a predictive error will be given less priority. The strength of the biasing is controlled by an attention parameter λ, where the unbiased ARTMAP network ($\lambda = 0$) is identical to a Fuzzy ARTMAP neural network model. For any given application, an optimal value of λ can be determined via validation.

Using the Biased ARTMAP, the systems classification performance was thus dependent on two factors: the strength of the biasing from the attention parameter and the ordering of the training data presentation. For any value of the attention parameter, a specific sequence of training data presentation exists that will yield the best classification accuracy. A paper by Kuan et al. [7] studied three operating strategies applied to Fuzzy ARTMAP networks: an averaging method derives the average classification performance from a pool of randomly-trained networks; a voting strategy to obtain class predictions via majority voting from a pool of randomly-trained networks; and an ordering algorithm using max-min clustering method to determine a single training sequence with the best generalization and classification performance. Our experiment was an attempt to combine a pre-processing ordering method to overcome the ARTMAPs training sequence dependency, and a post-processing voting method to overcome the limitations of a single learning system.

A genetic permutation algorithm [5] was selected to compute the optimal combination of training sequences and attention parameter for the Biased ARTMAP for any given training data set. A genetic algorithm (GA) is a search heuristic which mimics natural evolution to generate solutions for optimization and search problems. Genetic permutation in this experiment tests a population of randomly generated training sequences using the Biased ARTMAP for fitness selection. After each generation, the best training sequences are kept and used to generate variations for the next iteration of testing. Using this method, an optimal training sequence can be derived more efficiently than performing trial-and-error on every possible permutation.

An example of a similar attempt to integrate GA with neural networks to create an optimized learning and classification system was performed by Hsieh et al. [18] where the genetic neural networks method was found to be slightly better at predicting a financial crisis compared with case-based reasoning, backpropagation neural networks, logistic regression analysis, and quadratic discriminant analysis. Another paper by Hengpraprohm and Chongstitvatana [19] uses a GA variant, Genetic Programming, to optimize the construction of a classifier ensemble using K-means clustering and SNR feature selection. Each of the constructed classifiers evaluated a different selection of features to improve ensemble classifier diversity, and used a weighted voting approach to determine the final class prediction.

This experiment proposed using a probabilistic voting strategy [6] for the classifier ensemble. Using N classifiers, the Nth best training sequences obtained from the genetic optimization will each be used to train a different Biased ARTMAP classifier. The proposed voting strategy calculates the recognition rates of plurality voting techniques while taking into consideration each classifiers measure of reliability, the probability of a decision to be classified correctly given a specific input pattern. Implementing a minimum reliability filter can reduce classification error by excluding class predictions which have a low reliability metric.

To summarize, this paper is a proposed system for pattern learning and recognition, using Biased ARTMAP as the primary pattern learning and classification method. The learning phase will be optimized using genetic permutation algorithm to determine the best combinations of biasing value and presentation sequence of training data. Multiple classifiers will be trained using the results of the optimization exercise, and used as independent voters in a probabilistic voting strategy to determine the final predictions of any given test input. Reliability of each class prediction can be computed as a single metric, and unreliable predictions are filtered from the final prediction results using a reliability threshold. The final incarnation of the classifier ensemble will be used as a prototype pattern recognition system that is capable of continuous incremental improvement of its pattern recognition effectiveness via online learning.

Section 2 will elaborate on the theory and specifics on each component in the system, specifically the genetic permutation process, the Biased ARTMAP method, and the explanation for the probabilistic voting strategy. Section 3 explains the data set used to test the performance of the system, as well as the experimental methods and the subsequent results. Section 4 will outline the conclusions derived from the experimental study and results, as well as some suggestions on how future incarnations of the system can be improved.

This paper is motivated by a desire to create an optimal method for offline training of a classifier, or ensemble of classifiers to be used for real-time pattern classification. Furthermore, the system was designed to be modular, so that the different subsystems (preprocessing training optimization, pattern classifier, and post-processing voting system) can be modified and/or completely replaced by superior alternatives. When completed, this system can be repurposed into a general pattern classification method for future research as well as a method to compare different schemes for training data ordering, pattern learning and classification, and voting strategies.

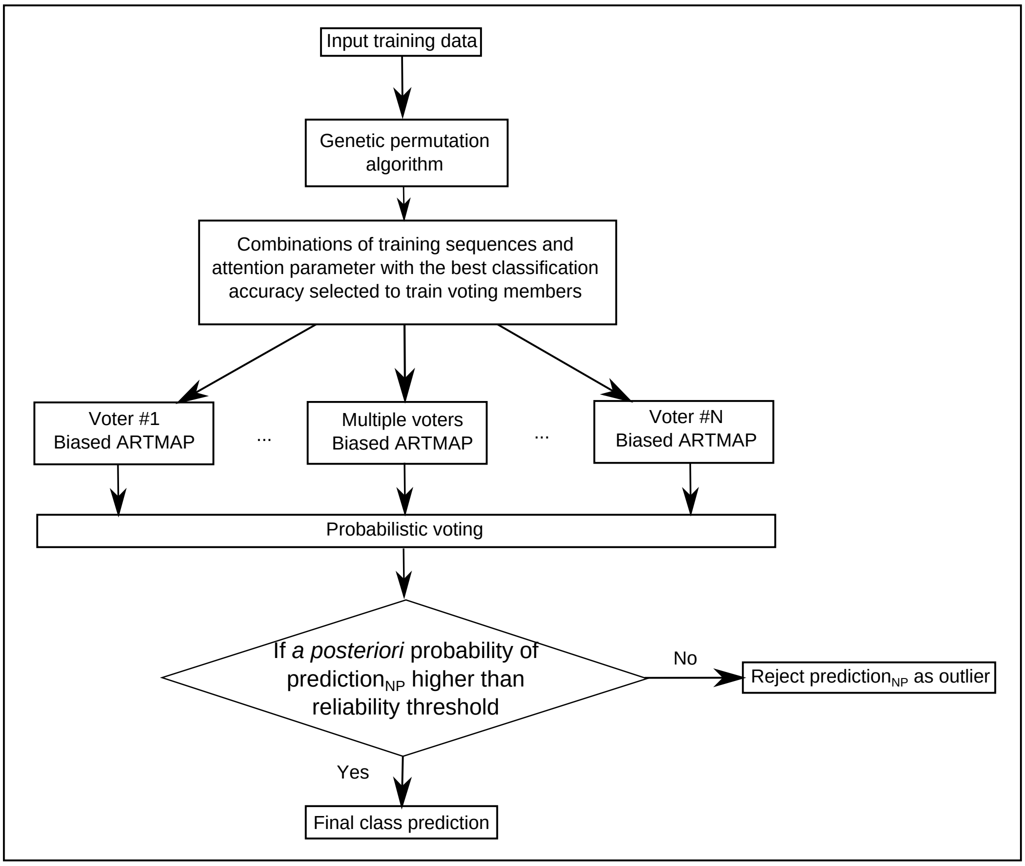
Input training data
Genetic permutation algorithm
Combinations of training sequences and attention parameter with the best classification accuracy selected to train voting members
Voter #1 Biased ARTMAP
...
Multiple voters Biased ARTMAP
...
Voter #N Biased ARTMAP
Probabilistic voting
If a posteriori probability of prediction$_{NP}$ higher than reliability threshold
No
Reject prediction$_{NP}$ as outlier
Yes
Final class prediction

Figure 1. Flowchart of the genetic algorithm for optimal training sequence ordering

2. **Genetic Ensemble Biased ARTMAP.** The block diagram illustrates the various modules that make up the Genetic Ensemble Biased ARTMAP system. The system can be divided into three main modules: the genetic permutation module is considered a pre-processing step to prepare the training method based on the given training data set; the Biased ARTMAP module may consist of a single or multiple ARTMAPs, each trained using a different training method prepared by the genetic optimization process. The results of the testing prediction from the ARTMAP(s) are combined in a probabilistic ensemble voting system. Testing predictions can be filtered using a reliability metric computed using *a posteriori* probability.

2.1. **Genetic permutation for optimal pattern ordering.** The objective for this module is to utilize genetic algorithm to derive, through mutation and fitness selection, the most effective training sequences of any given training data set for the Biased ARTMAP, as an alternative to trial-and-error testing of every possible permutation. The training data set, consisting of M features by NP data samples, can be encoded as a single chromosome consisting of $1 \times NP$ genes. A group of twenty randomly generated chromosomes were initialized, and were subjected to fitness testing. The least-fit chromosomes were discarded and a genetic reproduction method was used to repopulate the group. Gene mutation was also introduced to reduce the probability of early convergence. Over twenty generations, the chromosomes were serially evolved and mutated from among the survivors of each generations fitness selection. The final generation consisted of twenty training sequences which yielded the best classification accuracy when used for training the ARTMAP.

The steps involved in the genetic optimization process are [8]:

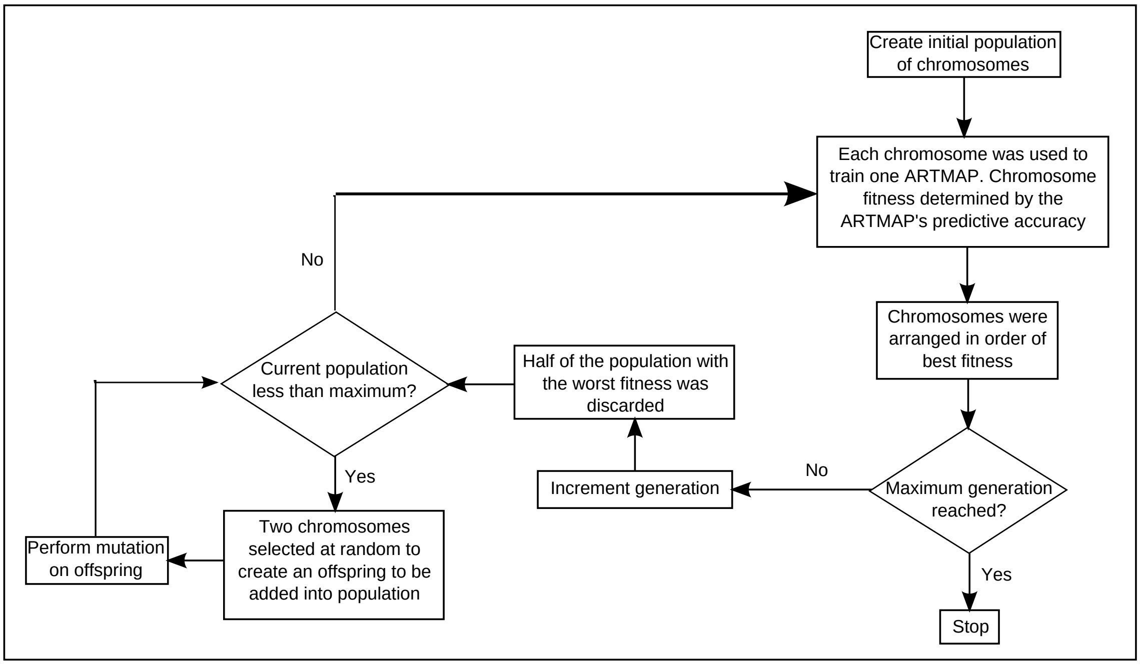
Create initial population of chromosomes
Each chromosome was used to train one ARTMAP. Chromosome fitness determined by the ARTMAP's predictive accuracy
Chromosomes were arranged in order of best fitness
Maximum generation reached?
Yes
Stop
No
Increment generation
Half of the population with the worst fitness was discarded
Current population less than maximum?
No
Yes
Two chromosomes selected at random to create an offspring to be added into population
Perform mutation on offspring

Figure 2. Flowchart of the genetic algorithm for optimal training sequence ordering

- **Initialization**.
  The number of genes in each chromosome was set to the number of patterns, NP, in the given dataset. Each gene was randomly set from integers 1 to NP without repetition. One chromosome was generated as a single member of the population, and repeated until the population cap was reached.
- **Fitness testing**.
  To calculate the fitness value of each chromosome, a single-voter Biased ARTMAP was trained and tested with 5-fold cross-validation using the chromosomes gene sequence to determine the training sequence presentation. Fitness value of the chromosome was calculated as the percentage of correctly classified patterns over the total number of tested patterns. Chromosomes are then sorted according to fitness, and half of the most fit chromosomes was kept for the next generation.
- **Reproduction and mutation operators**.
  Mating process was performed to repopulate the chromosome pool and replace the discarded chromosomes with offspring of fit parents. Two chromosomes were randomly chosen from among the survivors to generate the genetic traits for two offspring. Common genetic features between the two parents were passed down to both offspring, while uncommon features were assigned to one or the other offspring. The process was repeated until the population reaches the cap. Each generated offspring was subject to a mutation process, where two genes were randomly chosen and swapped if a random generated number [0, 1] is less than the user-defined rate of mutation, p_m. For this experiment, rate of mutation is set to 0.2, which mutates one gene out of every five.
- **Genetic permutation iteration**.
  The process of fitness testing and selection, mating, and mutation ensured that each successive population of chromosomes was more fit than the previous generation. After 20 generations, the resultant is a population of twenty chromosomes for each value of λ tested.

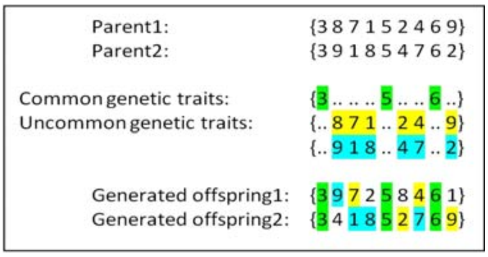
Parent1: {3 8 7 1 5 2 4 6 9}
Parent2: {3 9 1 8 5 4 7 6 2}
Common genetic traits: {3.. .. .. 5.. .. 6 ..}
Uncommon genetic traits: {.. 8 7 1 .. 2 4 .. 9}
{.. 9 1 8 .. 4 7 .. 2}
Generated offspring1: {3 9 7 2 5 8 4 6 1}
Generated offspring2: {3 4 1 8 5 2 7 6 9}

FIGURE 3. Genetic reproduction method for generating offspring with inherited traits. Example using two 9-gene chromosomes with several shared genetic traits.

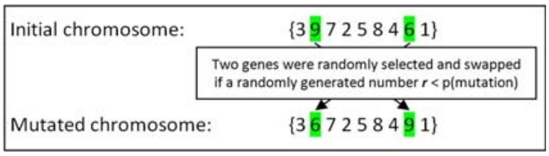
Initial chromosome: {3 9 7 2 5 8 4 6 1}
Two genes were randomly selected and swapped if a randomly generated number r < p(mutation)
Mutated chromosome: {3 6 7 2 5 8 4 9 1}

FIGURE 4. Simple genetic mutation

TABLE 1. Summary of genetic algorithm parameters imposed on this experiment

Gene coding	Integer coding in the range $[1, NP]$ where NP is the number of optimization variables, in this case, the number of training samples in the data set.
Fitness function	Percentage of correctly classified data samples using single-voter Biased ARTMAP.
Population size	20 chromosomes, each representing one training sequence.
Number of genes	One gene per data sample in the data set (100).
Selection	Chromosomes arranged by fitness. 50% of the least fit chromosomes were discarded.
Reproduction	Two chromosomes selected at random to generate one offspring. Repeated until number of offspring replaces all discarded chromosomes.
Mutation	Probability for one gene to be swapped with a randomly selected gene. Set to 0.2.
Convergence	20 successive iterations of fitness selection, reproduction, and mutation of the initial population of chromosomes.

2.2. Biased ARTMAP.

2.2.1. *Adaptive resonance theory.* Adaptive resonance theory (ART) was developed as a theory of human cognitive information processing [3], the design principles of which led to the development of real-time neural network models that perform supervised and unsupervised learning, pattern recognition and prediction. The ARTMAP model [9] is a hierarchical network architecture that can organize stable categorical mappings between M-dimensional input vectors and N-dimensional output vectors. Characteristics of ARTMAP neural network models include complement coding and match tracking. Under supervised learning conditions, ARTMAP's internal control mechanisms create stable recognition categories of optimal size by maximizing predictive generalization while minimizing predictive error.

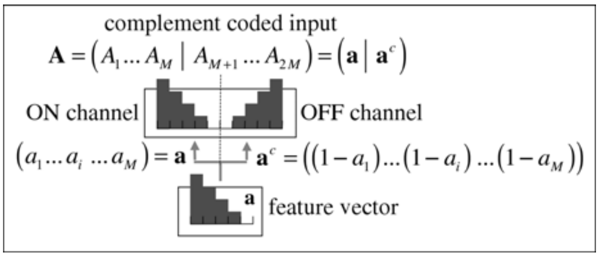
complement coded input
A = (A_1 ... A_M | A_{M+1} ... A_{2M}) = (a | a^c)
ON channel
OFF channel
(a_1 ... a_i ... a_M) = a
a^c = ((1 − a_1)...(1 − a_i)...(1 − a_M))
a
feature vector

FIGURE 5. Input vector A consists of original feature vector a and its complement a^c. Vector A represents the degree in which a feature i is present (a_i) as well as the degree in which the same feature is absent $(1a_i)$ [4].

2.2.2. *Complement coding.* Complement coding is a preprocessing step in which an M-dimensional input feature vector a is recoded into a 2M-dimensional input vector A, consisting of the original feature vector a, and its complement a_c. This method allows the ARTMAP to encode within its critical feature patterns of memory features that are consistently present on an equal basis with features that are consistently absent. Complement coding is implemented by first scaling the initial component features a_i of a feature vector a to $[0 <= a_i <= 1]$. For each feature **i**, activity a_i determines its complementary activity as $(1 - a_i)$.

2.2.3. *Search and match tracking.* Whenever a new input feature vector A is presented, the search process attempts to match A to the critical feature pattern of the currently activated node. Matching criterion is determined by a vigilance parameter ρ. Setting the initial baseline vigilance $\bar{\rho}$ to zero allows more generalization in the learning process, while setting a high $\bar{\rho}$ ensures a more specific exemplar-like learning. If the system does not find a match, an uncommitted node is encoded with the current input feature pattern. If the system matches and correctly predicts an input A, the critical features of the activated node are weighted heavier. If the prediction is incorrect, ρ is raised and search is repeated. This continues until a correct prediction is reached, to which ρ is reset to its baseline value for the next input A. Match tracking controls the degree to which ρ is increased to implement the design goals of maximum generalization and minimum predictive error by implementing a degree of flexibility in the matching criterion.

2.2.4. *Biasing mechanism.* Under online fast learning conditions, the ARTMAPs critical featural attention may be distorted by certain sequences of input presentation, causing less-suitable critical features to be overemphasised in future searching procedures. The Biased ARTMAP variant [12] introduces a new medium-term memory that would enable the network to shift attention among input features whenever a predictive error was generated.

For any given input, the Biased ARTMAP tracks attended features that caused predictive errors and reduces the activations of these features during future searching. The strength of the biasing is determined by an attention parameter. The optimal attention parameter for any given input can be determined by validation, but a default value of 10 produced near-optimal results on small-scale and large-scale computational examples. Biasing the input features will allow the network to activate a previously inactive node in response to a mismatch reset, instead of reactivating the same node which caused the mismatch.

Table 2. Biased ARTMAP parameters

$\lambda \geq 0$	Bias parameter. Optimal results in most applications with $\lambda = 10$. Study on effectiveness of biasing within [0, 10]
$e \equiv (e_1, \ldots, e_i, \ldots, e_{2M})$	Each biasing variable e_i represents strength of bias against feature i
$\tilde{A} \equiv [A - e]^+$	Input vector is modified by biasing weights
$\tilde{w}_J \equiv [w_J - e]^+ \equiv (\tilde{u}_J \vert \tilde{v}_J^c)$	Weights consist of input weights u_J and its complement v_J^c, modified by biasing vector
$\bar{\rho} \in [0, 1]$	Baseline vigilance set to zero for maximum generalization
$\tilde{x} \equiv [x - e]^+ = \tilde{w}_J \wedge \tilde{A}$	Biased matched vector
$r = \{0, 1\}$	Mismatch reset is triggered if $\rho\vert A\vert - \vert x\vert > 0$
$R = \{0, 1\}$	Error detection when an active node J makes a predictive error
$\Gamma \gg 1$	Fast integration of medium-term memory variables ρ and e_i after a predictive error
$\beta \in [0, 1]$	Learning rate parameter set to 1 for fast learning
$\varepsilon = 0^-$	Match tracking parameter. On the time scale of search in the medium-term memory, ρ will decay by ε
$\alpha = 0^+$	Choice parameter

Table 3. Initial parameters when a new input is presented

$e = 0$	Initial biasing weights set to zero
$\tilde{A} = A \equiv (a\vert a^c)$	Initial input vector is unbiased, consisting of original feature vector a and its complement a^c
$\rho = \bar{\rho}$	Vigilance set to baseline value
$r = R = 0$	Reset signal r not triggered until predictive error R is made
$w_{iJ} = 1$	Initial weights set to 1

The steps involved in a single iteration of the Biased ARTMAP are illustrated as follows:

- **New training input**.
  When a new training input **a** is presented, initial parameters are set as follows:
- **Pattern matching**.
  For each input pattern A presented, the system chooses a category node J in the coding field F_2 from among the nodes j that has not been reset, to maximize the choice-by-difference F_0-to-F_2 signal function:

$$T_j = |A \wedge w_j| + (1 - \alpha)(M - |w_j|) \tag{1}$$

  The matched pattern $x = |A \wedge w_j|$ is further modified with a biasing vector into $\tilde{x} = [x - e]^+$. The biasing vector e_i is initialized to zero.
- **Search and match tracking**.

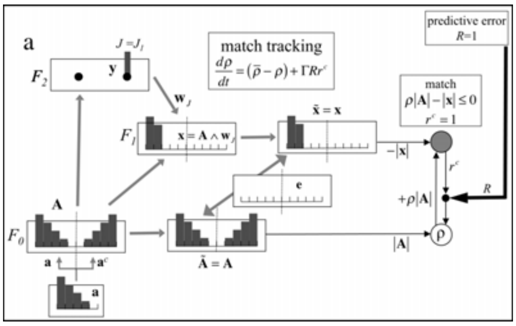
a
predictive error
R=1
J=J1
F2
y
match tracking
$\frac{d\rho}{dt}=(\bar{\rho}-\rho)+\Gamma Rr^c$
match
$\rho|\mathbf{A}|-|\mathbf{x}|\le 0$
$r^c=1$
$\mathbf{w}_J$
$\tilde{\mathbf{x}}=\mathbf{x}$
F1
$\mathbf{x}=\mathbf{A}\wedge\mathbf{w}_J$
$-|\mathbf{x}|$
r^c
e
$+\rho|\mathbf{A}|$
R
A
F0
ρ
$|\mathbf{A}|$
a
$\mathbf{a}^c$
$\tilde{\mathbf{A}}=\mathbf{A}$
a

FIGURE 6. Match tracking with a biasing module to bias certain features in feature vector A. Initial values of all biasing vectors e_i are set to 0 and changes whenever predictive error occurs [4].

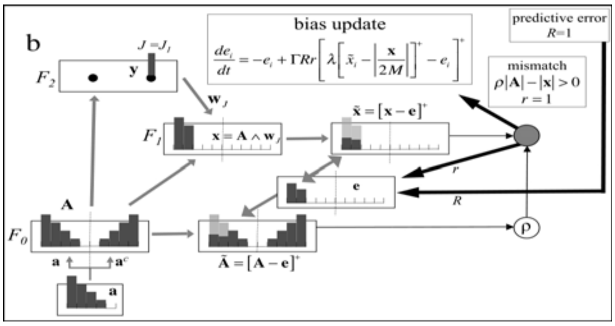
b
bias update
$\frac{de_i}{dt}=-e_i+\Gamma Rr\left[\lambda\left[\tilde{x}_i-\left|\frac{\mathbf{x}}{2M}\right|\right]^+-e_i\right]^+$
predictive error
R=1
J=J1
F2
y
mismatch
$\rho|\mathbf{A}|-|\mathbf{x}|>0$
$r=1$
$\mathbf{w}_J$
$\tilde{\mathbf{x}}=[\mathbf{x}-\mathbf{e}]^+$
F1
$\mathbf{x}=\mathbf{A}\wedge\mathbf{w}_J$
r
e
R
A
F0
ρ
a
$\mathbf{a}^c$
$\tilde{\mathbf{A}}=[\mathbf{A}-\mathbf{e}]^+$
a

FIGURE 7. Biasing vector e_i is updated according to the biasing equation. Degree of biasing is determined by free parameter λ. Biasing affects all currently attended input features, which are $i = 1$ and $i = 2$ [4].

If node **J** fails to meet the matching criterion $\left(\frac{|\tilde{x}|}{|\tilde{A}|} > \rho\right)$, then node **J** is shut off for the duration of the current search cycle. Input **A** then chooses another node **J**.

If node **J** meets the minimum vigilance parameter, then an output prediction is made. If the prediction is correct, or if **J** is an uncommitted node, the system performs learning. If the prediction is incorrect, vigilance ρ is increased enough to reset **J**. The rate of increase is determined by the match tracking equation:

$$\frac{d\rho}{dt} = -(\rho - \bar{\rho}) + \mathit{\Gamma} Rr^c \tag{2}$$

$\bar{\rho}$ is commonly set to zero for maximum generalization during the learning process. When an incorrect prediction is made, $\mathbf{R} = 1$ When ρ is raised high enough to cause node **J** to mismatch reset ($\mathbf{r} = 1$), ρ stops increasing, at which point ρ will be incrementally larger than the match value $\frac{|\tilde{x}|}{|\tilde{A}|}$.

On the time-scale of search, ρ will decay by match tracking parameter ($\varepsilon = 10^{-5}$) before the next coding node is chosen.

- **Bias update**.

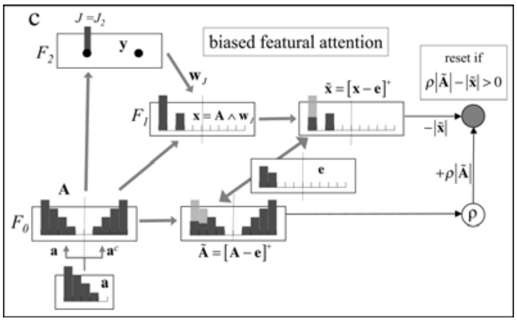
C
$J=J_2$
F_2
y
biased featural attention
reset if
$\rho|\tilde{\mathbf{A}}|-|\tilde{\mathbf{x}}|>0$
$\mathbf{w}_J$
$\tilde{\mathbf{x}}=[\mathbf{x}-\mathbf{e}]^+$
F_1
$\mathbf{x}=\mathbf{A}\wedge\mathbf{w}_J$
$-|\tilde{\mathbf{x}}|$
e
$+\rho|\tilde{\mathbf{A}}|$
A
F_0
ρ
a
$\mathbf{a}^c$
$\tilde{\mathbf{A}}=[\mathbf{A}-\mathbf{e}]^+$
a

FIGURE 8. A new category node $J = J_2$ is activated following mismatch reset. With biasing against features $i = 1$ and $i = 2$, now feature $i = 3$ is given more attention [4].

When an incorrect prediction ($\mathbf{R} = 1$) and mismatch reset ($\mathbf{r} = 1$) occur, bias vector components e_i ($i = 1, 2, \ldots, 2M$) are increased according to the biasing equations:

$$\frac{de_i}{dt} = -e_i + \mathit{\Gamma} Rr \left[\lambda \left[[x_i - e_i]^+ - \frac{|x|}{2M} \right]^+ - e_i \right]^+ \tag{3}$$

$\mathit{\Gamma}$ is assumed to be large enough so that e_i reaches equilibrium before node $\mathbf{J}$ is shut off, which will switch the predictive error signal $\mathbf{R}$ back to zero.

On the time-scale of search, e_i decays by ε before the next coding node is chosen.

Biasing vector e_i is unchanged if

$$\lambda \left[[x_i - e_i]^+ - \frac{|x|}{2M} \right] \leq 0 \tag{4}$$

or if

$$e_i > \lambda \left[[x_i - e_i]^+ - \frac{|x|}{2M} \right] > 0 \tag{5}$$

However, if the matched pattern

$$x_i > e_i^{old} \tag{6}$$

and

$$\lambda \left[[x_i - e_i^{old}]^+ - \frac{|x|}{2M} \right] > 0 \tag{7}$$

then e_i is updated according to the equation:

$$e_i^{new} = \left(\frac{x_i - \frac{|x|}{2M}}{1 + \lambda^{-1}} \right) \tag{8}$$

- **Perform ARTMAP learning**.

The weights of the activated node $\mathbf{J}$ is updated according to the weight equation:

$$w_j^{new} = (1 - \beta) w_j^{old} + \beta \left(w_j^{old} \wedge A \right) \tag{9}$$

With fast learning ($\beta = 1$):

$$w_j^{new} = w_j^{old} \wedge A \tag{10}$$

The time-scale of learning is assumed to be greater than the medium-term memory time-scale. Thus, ρ and e_i decay to their initial values as their weights are updated. Biasing during search affects the choice of the final node $\mathbf{J}$ but will not affect learning.

The next input **a** is then selected to be presented.

2.3. **Probabilistic voting and reliability test.** The probabilistic ensemble voting strategy used here is based on research by Lin et al. [12] and Loo and Rao [6]. N classifiers $(E_1, E_2, \ldots, E_N)$ are employed for an M-class pattern recognition task, in which an input object X is classified into one of the M classes $(C_1, C_2, \ldots, C_M)$. Classifier E_i keeps a constant recognition rate p_i for any input object X:

$$P\left(E_i(X) = C(X)\right) = p_i \tag{11}$$

$C(X)$ is the true class in which input X belongs to, and $E_i(X)$ is the class selected by classifier E_i. All other classes have equal probability of being chosen in case of incorrect classification.

$$P(E_i(X) = C_j) = \frac{1 - p_i}{M - 1} = e_i \tag{12}$$

where $(j = 1, 2, \ldots, M)$ and $C_j \neq C(X)$. Each classifier is assumed to make its decision independently. In order to minimize the error rate of the combination system, the class with the largest *a posteriori* probability should be selected according to Bayes' rule. Since the probability for all other classes to be chosen is equal, the effective decision can be summarized as:

$$D_j(X) = \ln P(C(X) = C_j) + \Sigma_{i=1}^{N} \ln \left(\frac{(M-1)p_i}{1 - p_i} \right) \delta_{ij}(X) \tag{13}$$

The above equation is a generic form of plurality voting rule. The class C_j that maximizes $D_j(X)$ is selected. Each classifier can have a different weight and each class has a constant representing its a priori probability. From the above analysis, plurality voting as shown is equivalent to the Bayesian criterion under the following conditions:

- The classifiers' decisions are independent of each other.
- Misclassifications are evenly distributed among the $M - 1$ residual classes.
- In case of a tie, the class with the maximum support is chosen arbitrarily.
- Input objects are evenly distributed among all the classes.

The independence assumption is not easy to meet in practical pattern recognition applications. More commonly, all of the classifiers are prone to make mistakes simultaneously on some very difficult samples. Taking this factor into account, a modified model is proposed, composed of both the independent situations: the N classifiers will simultaneously misrecognize a sample with a probability of α. Otherwise with a probability of $(1 - \alpha)$, the N classifiers will perform independently. Under this model, the overall recognition rate of classifier E_i is $(1 - \alpha)p_i$. The p_i in the above equation is replaced with $(1 - \alpha)p_i$.

Using a quantitative analysis of the probabilistic voting systems performance, the following observations were made:

- The voting systems recognition rate increases or remains constant when additional classifiers or data samples were given, due to the implementation of reverse probabilistic rule whenever recognition rate falls below average. Average recognition rate is given as $p = \frac{1}{M}$, where M is the number of classes in the data set. If $p \neq \frac{1}{M}$, recognition rate will approach 1 with a sufficiently large N. Reverse probabilistic rule is employed whenever $p < \frac{1}{M}$. When $p = \frac{1}{M}$, class selection is random if there is a tie between multiple eligible classes.
- When individual classifiers perform above average $(p > \frac{1}{M})$, recognition rates increases with a larger M. With more classes, the erroneous predictions will be scattered equally among $M - 1$ incorrect classes, giving the chance for the correct class to stand out.

- If multiple comparable classifiers have the same recognition rates, combining three or more classifiers may give better results. Increasing the number of classifiers beyond three will increase computational complexity and may not be required. If the classifiers have different recognition rates, weights are assigned to the best classifier, in which case combining multiple classifiers may be redundant as the final decision can be dominated by only the best performing classifier.

The study by Loo and Rao [6] implements a method to measure the reliability of a class prediction computed from the probabilistic voting results. Reliability of a class prediction is computed by defining the a posteriori probability of the winning class given the predictions of N classifiers. Classification reliability in this case is decided by the vote difference between the winning class and the other classes. The desired reliability of a classification system can be enforced by requiring that each and every input objects winning class to have at least r more votes than the closest competing class, failure of which the classification of the input object is rejected due to unreliability of the prediction. A system with $r = 0$ is known as simple probabilistic voting, while $r > 0$ is strict probabilistic voting. Thus, the classification performance of a system may be artificially increased by setting a high reliability threshold r at the expense of rejecting a higher number of input data samples.

3. **The Experiment.**

The experiment is tested using the dataset collected by Wagner et al. [13]. To induce the subject to feel different emotions, four music songs were used, selected by the subjects themselves, in respect of the targeted emotion classes of anger, joy, pleasure, and sadness. An advantage of this method is that most people associate different moods with specific songs.

While the subject listens to the music, biosensors are used to measure electromyogram (EMG), electrocardiogram (ECG), skin conductivity (SC) and respiratory change (RSP). Overall, 25 recordings for each emotion class were collected, for a total of 100 training data samples per emotion class, each being a four-channel digital signal recording. A label set is created to segment the data into a four-emotion classification problem: (1: Anger) (2: Joy) (3: Pleasure) (4: Sadness).

Before the signals can be analyzed, feature extraction was performed to reduce the dimensionality of the raw signal measurements into several parameters representative of the entire signal. This has the advantage of reducing the computational costs. Feature extraction was performed using an algorithm designed by Wagner et al. [14]. A total of 211 features were extracted from each recording, including several features not available

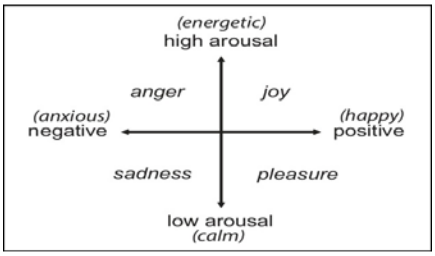

Figure 9. Targeted emotion classes in regard to a common representation of four distinct human emotions in two dimensions of arousal and valence [13]

in the original toolbox algorithm. The features of principal dynamic modes [15, 16] were included to provide nonlinear analysis to the overall feature set.

A genetic optimization algorithm was employed to optimize the training sequence to be presented to the Biased ARTMAP. The initial 20 chromosomes were generated at random, each one encoded as a single complete training sequence. Fitness testing was performed using a single-voter Biased ARTMAP, with fitness defined as percentage of correct classifications. The population was sorted by fitness, and 50% of the least fit chromosomes were discarded from the population. The remaining chromosomes were used to generate new chromosomes to replace the discarded chromosomes using mating and mutation operations. Rate of mutation was set to 20% of the total number of genes in each chromosome.

The genetic selection process was iterated for 20 generations, and repeated with a random population for each value of the attention parameter from 0 to 10. This genetic optimization exercise generated a total of 220 chromosomes, which were then arranged in order of fitness. The chromosome with the best fitness was used for training the first voter, and each subsequent voter was trained using the next-best chromosome. Each ARTMAP was configured for fast learning and maximum generalization.

Training and testing was performed using leave-one-out method, and the final class prediction was determined by probabilistic ensemble voting. The classification performance of the classifier ensemble was defined as the percentage of correctly classified data samples.

3.1. **Experiment results.** The first test compares the classification performance of the Biased ARTMAP against Fuzzy ARTMAP. The final generated population was used for generating a bootstrapped mean of the ARTMAPs classification performance. Bootstrapping was performed using 1000 resamplings with 95% confidence.

For this case, the classification performance was not significantly impacted by modifying the biasing parameter. One hypothesis is that genetic ordering compensation inadvertently solves the problem of early featural distortion which the Biased ARTMAP was designed to solve. Nevertheless, the above results were obtained from offline learning, and the biasing technique will be more useful during online learning.

Table 4. Classification performance of ARTMAPs with different biasing intensity

Attention parameter λ	Minimum	Maximum	Bootstrapped mean
0 (Fuzzy ARTMAP)	59	76	$66.06^{-2.66}_{+1.79}$
1	59	75	$66.75^{-1.95}_{+1.95}$
2	61	76	$66.97^{-1.17}_{+1.33}$
3	59	78	$67.11^{-1.96}_{+2.29}$
4	59	77	$66.90^{-1.90}_{+2.00}$
5	62	76	$67.06^{-1.71}_{+2.04}$
6	54	78	$67.28^{-1.38}_{+2.12}$
7	60	76	$67.11^{-1.86}_{+1.79}$
8	60	77	$67.05^{-2.55}_{+1.65}$
9	61	76	$66.56^{-1.66}_{+1.89}$
10	59	77	$66.15^{-2.05}_{+2.10}$

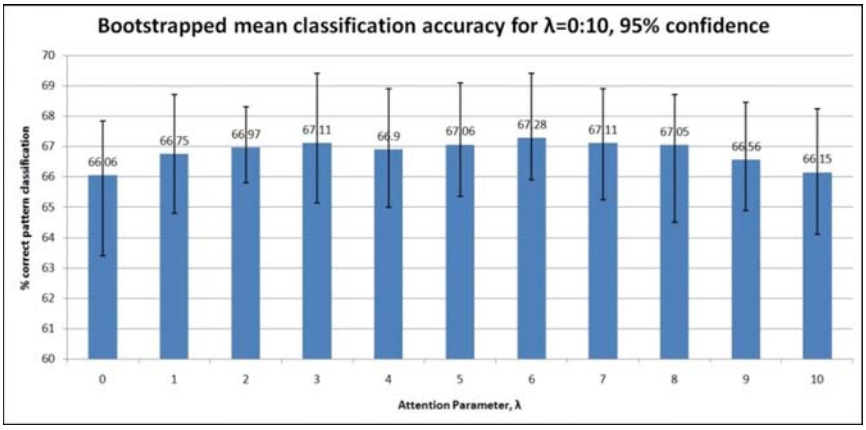

FIGURE 10. Bootstrapped mean classification accuracy for each biasing parameter

TABLE 5. Classification accuracy of fuzzy ARTMAP $\lambda = 0$ with probabilistic ensemble voting and reliability threshold

Voters	Reliability $R = 0$	$R = 0.5$	$R = 0.9$	$R = 0.99$
1	76.00 (0)	76.00 (0)	NaN (100)	NaN (100)
2	73.00 (0)	80.00 (20)	80.00 (20)	NaN (100)
3	76.00 (0)	76.76 (1)	85.13 (26)	85.13 (26)
5	70.00 (0)	70.40 (2)	77.64 (15)	78.31 (17)
7	71.00 (0)	71.42 (2)	76.13 (12)	78.57 (16)
10	73.00 (0)	74.22 (3)	73.95 (4)	73.33 (10)

TABLE 6. Classification accuracy of Biased ARTMAP $\lambda = 0 : 10$ with probabilistic ensemble voting and reliability threshold

Voters	Reliability $R = 0$	$R = 0.5$	$R = 0.9$	$R = 0.99$
1	78.00 (0)	78.00 (0)	NaN (100)	NaN (100)
2	78.00 (0)	82.95 (12)	82.95 (12)	NaN (100)
3	79.00 (0)	79.00 (0)	87.17 (22)	87.17 (22)
5	79.00 (0)	79.38 (3)	84.44 (10)	84.88 (14)
7	75.00 (0)	74.48 (2)	77.77 (10)	78.65 (11)
10	79.00 (0)	79.78 (6)	80.64 (7)	80.89 (11)

A probabilistic ensemble voting system was applied, in which **N** voters were individually trained by **N** of the best training sequences from the combined population of 220 chromosomes. Testing was performed on the voting system based on probabilistic majority rules to determine the final class prediction of the test data. Testing was repeated using a reliability metric to evaluate each class prediction. Class predictions which did not meet the reliability threshold were removed from the final accuracy calculation.

The number in brackets represents the percentage of class predictions which were rejected due to low reliability. In particular, predictions from a voting system with few voting members are considered less reliable due to lack of information compared with systems with more voting members. However, this experiment also indicates that while predictive accuracy increased when a more stringent reliability threshold was applied, increasing the number of voters did not elicit an improvement. This may be explained by

Table 7. Comparison of different classification methods

Classification method	Predictive accuracy (%)
Linear discriminant analysis	66.00
k^{th} nearest neighbour	$72.00^{-1.80}_{+1.50}$
Multilayer perceptron	83.00
Genetic Ensemble Fuzzy ARTMAP	85.13 (3-voter, 90% reliability)
Genetic Ensemble Biased ARTMAP	87.17 (3-voter, 90% reliability)

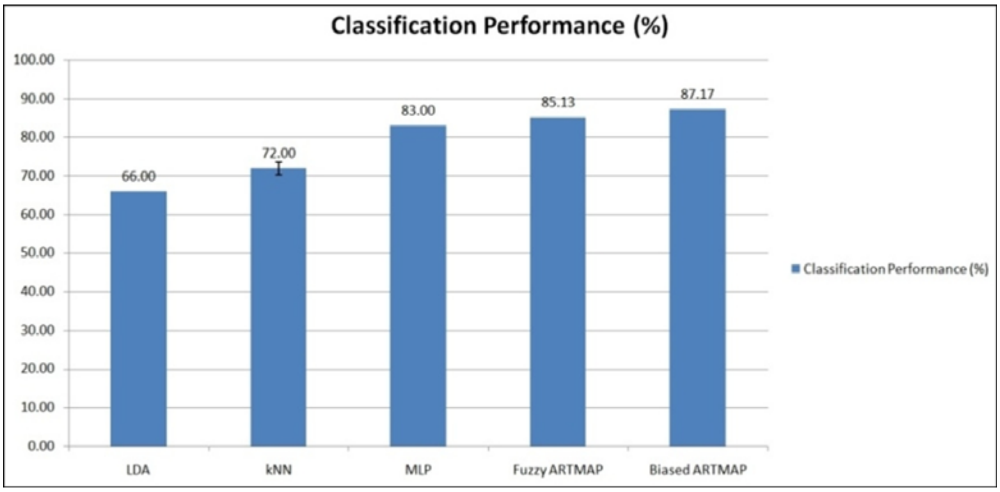

Figure 11. Classifier predictive accuracy comparison

the method in which each voter was trained. Each additional voter besides the first was trained using training sequences which were increasingly less accurate, effectively affecting the systems predictive accuracy by adding an increasing amount of noisy data. Even so, each additional voter served to contribute additional information into the ensemble classifier by improving recognition rates of reliable training data.

The above results were then compared against similar pattern classification methods: linear discriminant analysis (LDA), k-nearest neighbor (kNN), and multilayer perceptron (MLP). For kNN, a series of training and testing was performed for a range of values for k, in the range [1, 10]. A bootstrapped mean was generated from the results. For MLP, the main initial network parameters are the number of hidden layers (set to 9), the rate of learning (set to 1), and the number of training iterations (set to 100). For Fuzzy ARTMAP and Biased ARTMAP, the results were using the classification performance from the best combination of voter ensemble, reliability threshold and training sequence.

Both ARTMAPs show comparable classification performance with the multilayer perceptron (MLP). However, ARTMAPs have several distinct advantages over the MLP classification method, including the ability for incremental learning to evolve the classification system over time, and a faster convergence during training and testing.

The resultant genetically-trained Biased ARTMAP voting system can be viewed as a prototype emotion recognition system that translates input features extracted from four biosignal channels (ECG, EMG, RSP and SC) into one of four emotion class predictions (Anger, Joy, Pleasure, Sadness) with approximately 85% accuracy as shown with the above results. Predictions with a high degree of reliability may be flagged and used to further train the Biased ARTMAP to increase its predictive capability.

3.2. **EEG-based affect recognition.** The complete Genetic Ensemble Biased ARTM AP system was trained and tested offline using a different database consisting of electroencephalogram (EEG) data. The database consists of alpha1, alpha2, beta1, and beta2 channel measurements from two subjects. Measurements were obtained using a Neurosky headset at a 1Hz sampling rate. Subjects were shown a series of pictures selected from the International Affective Picture System (IAPS) [17] to elicit one of four affect conditions: Positive Valence, Negative Valence, High Arousal, and Low Arousal. A total of 12 pictures were selected for each affect condition, and were displayed one after another with duration of 14 seconds per picture. The subject was given a 5-minute interval before proceeding to the next series of pictures for affect elicitation. EEG measurements were obtained for the entire duration when the pictures were displayed. Feature extraction was performed on the EEG signal measurements to reduce dimensionality of the data.

The database was divided into two sets, Positive-Negative Valence and High-Low Arousal, effectively creating two sets of binary classification problems. Each database consists of 48 data samples (12 pictures × 4 channels) with 16 features each. Genetic ordering was performed for each data set, using Biased ARTMAP for fitness-testing. Mating and mutation operators were used to select training sequences for each successive generation of chromosomes. The best training sequences were selected to train an ensemble Biased ARTMAP probabilistic voting system. Training and testing was performed using leave-one-out.

Table 8. Performance classification of genetic ensemble biased ARTMAP in distinguishing between high v. low arousal EEG, and positive v. negative valence EEG

High vs Low	Reliability=0	R=0.5	R=0.9	R=0.99
1-voter	86.67 (0)	86.67 (0)	NaN (100)	NaN (100)
5-voters	77.78 (0)	77.78 (0)	86.67 (0)	86.67 (0)
10-voters	75.55 (0)	75.55 (0)	80.00 (0)	80.00 (0)

Positive vs Negative	Reliability=0	R=0.5	R=0.9	R=0.99
1-voter	75.55 (0)	75.55 (0)	NaN (100)	NaN (100)
5-voters	68.89 (0)	68.89 (0)	100.00 (20)	100.00 (20)
10-voters	66.67 (0)	66.67 (0)	71.11 (0)	71.11 (0)

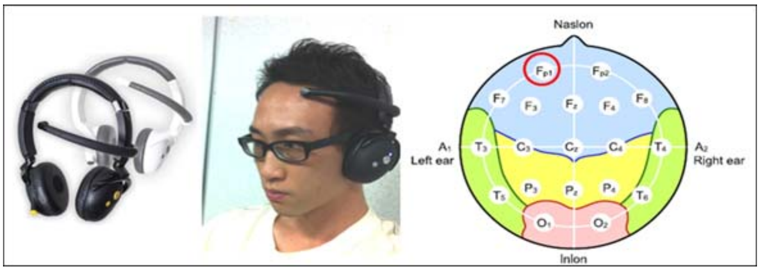

Figure 12. Positioning of the Neurosky headset to measure EEG in the highlighted area

As with the previous data set, imposing a high reliability threshold on systems with fewer voters was counter-productive. For the 5-voter Positive-Negative data set, the 100% classification accuracy occurred when paired with a high data rejection rate (20% of the training samples), compared with 71.11% predictive accuracy with no data rejection when using 10 voters. This demonstrates a trade-off between classification accuracy and data reliability, and is entirely subjective as to which method is more practical. This also demonstrates that increasing the number of voters also increased the reliability of each prediction as more information was introduced into the voting system.

A potential application is to combine the EEG affect classification system with the ECG emotion classification system to create a single affect recognition system. By assigning a number of classifiers to evaluate EEG, and another group of classifiers to evaluate ECG, the combined classifier ensemble will possess a higher diversity than both of the classification systems used singularly.

4. Conclusions and Future Work.

The experiment results showed no significant benefits of using Biased ARTMAP over Fuzzy ARTMAP, with or without the use of genetic optimization. The presented results also showed degradation in the classification accuracy when more than one voter was introduced into the probabilistic ensemble voting system. The genetic optimization method produced significant improvement in classification accuracy. The probabilistic ensemble voting system was able to evaluate individual class predictions using a single reliability metric. Setting different levels of reliability filtering allows classifier accuracy to be improved by rejecting class predictions with low reliability.

This experiment raised several new suggestions where the system could be improved. A different method could be employed to select training sequences for optimal voting results, as selecting the best individual fitness results to be combined into a voting strategy does not guarantee improvement in the classification performance. This is most likely due to the lack of diversity in the generated population caused by the genetic permutation method. Further analysis of ensemble voting effectiveness should include diversity measurements to select a wider selection of training sequences. In addition, this experiment did not adequately present the effectiveness of using Biased ARTMAP over Fuzzy ARTMAP under supervised training conditions as well as under online learning conditions.

REFERENCES

[1] *Moore G Cramming More Components onto Integrated Circuits*, ftp://download.intel.com/museum /Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf/, 2011.

[2] S. E. A. Ahmad, J. Yan and M. Tayara, The robustness of Google CAPTCHAs, *Proc. of the 3rd European Workshop on System Security*, 2010.

[3] G. A. Carpenter and S. Grossberg, Adaptive resonance theory, *The Handbook of Brain Theory and Neural Networks*, vol.2, pp.87-90, 2003.

[4] G. A. Carpenter and S. C. Gaddam, Biased ART: A neural architecture that shifts attention toward previously disregarded features following an incorrect prediction, *Neural Networks*, vol.23, no.3, pp.435-451, 2010.

[5] R. Palaniappan and C. Eswaran, Using genetic algorithm to select the presentation order of training patterns that improves simplified fuzzy ARTMAP classification performance, *Applied Soft Computing*, vol.9, no.1, pp.100-106, 2009.

[6] C. K. Loo and M. V. C. Rao, Accurate and reliable diagnosis and classification using probabilistic ensemble simplified fuzzy ARTMAP, *IEEE Transactions on Knowledge and Data Engineering*, vol.17, no.11, pp.1589-1593, 2005.

[7] M. M. Kuan, C. P. Lim and R. F. Harrison, On operating strategies of the fuzzy ARTMAP neural network: A comparative study, *International Journal of Computational Intelligence and Applications*, vol.3, no.1, pp.23-43, 2003.

[8] R. L. Haupt and S. E. Haupt, Matlab code: Continuous genetic algorithm, in *Practical Genetic Algorithms*, 2nd Edition, John Wiley and Sons, 2004.

[9] G. A. Carpenter, S. Grossberg and J. H. Reynolds, ARTMAP: A self-organizing neural network architecture for fast supervised learning and pattern recognition, *Neural Networks*, vol.4, no.5, pp.565-588, 1991.

[10] G. A. Carpenter, S. Grossberg, N. Marzukon, J. H. Reynolds and D. B. Rosen, Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Transactions on Neural Networks*, vol.3, pp.698-713, 1992.

[11] Carpenter GA default ARTMAP, *Proc. of the International Joint Conference on Neural Networks*, pp.1396-1401, 2003.

[12] X. Lin, S. Yacoub, J. Burns and S. Simske, Performance analysis of pattern classifier combination by plurality voting, *Pattern Recognition Letters*, vol.24, pp.1959-1969, 2003.

[13] J. Wagner, J. Kim and E. Andre, From physiological signals to emotion: Implementing and comparing selected methods for feature extraction and classification, *IEEE International Conference on Multimedia and Expo 2005*, 940-943, 2005.

[14] J. Wagner, *The Augsburg Biosignal Toolbox*, http://www.informatik.uniaugsburg.de/en/chairs/hcm/projects/aubt/, 2011.

[15] Y. Zhong, H. Wang, K. H. Ju, K. M. Jan and K. H. Chon, Nonlinear analysis of the separate contributions of autonomics nervous systems to heart rate variability using principal dynamic modes, *IEEE Transactions on Biomedical Engineering*, vol.51, no.2, pp.255-262, 2004.

[16] J. Choi and R. Gutierrez-Osuna, Using heart rate monitors to detect mental stress, *The 6th International Workshop on Wearable and Implantable Body Sensor Networks*, pp.219-223, 2009.

[17] P. J. Lang, M. M. Bradley and B. N. Cuthbert, International affective picture system (IAPS): Affective ratings of pictures and instruction manual, *Technical Report A-8*, University of Florida, Gainesville, FL, 2008.

[18] J. C. Hsieh, P. C. Chang and S. H. Chen, Integration of genetic algorithm and neural network for financial early warning system: An example of Taiwanese banking industry, *The 1st International Conference on Innovative Computing, Information and Control*, Beijing, China, pp.562-565, 2006.

[19] S. Hengpraprohm and P. Chongstitvatana, A genetic programming ensemble approach to cancer microarray data classification, *The 3rd International Conference on Innovative Computing, Information and Control*, Dalian, China, pp.340-340, 2008.