Online Pattern Classification With Multiple Neural Network Systems: An Experimental Study

Chee Peng Lim and Robert F. Harrison

***Abstract*—In this paper, an empirical study of the development and application of a committee of neural networks on online pattern classification tasks is presented. A multiple classifier framework is designed by adopting an Adaptive Resonance Theory-based (ART) autonomously learning neural network as the building block. A number of algorithms for combining outputs from multiple neural classifiers are considered, and two benchmark data sets have been used to evaluate the applicability of the proposed system. Different learning strategies coupling offline and online learning approaches, as well as different input pattern representation schemes, including the "ensemble" and "modular" methods, have been examined experimentally. Benefits and shortcomings of each approach are systematically analyzed and discussed. The results are comparable, and in some cases superior, with those from other classification algorithms. The experiments demonstrate the potentials of the proposed multiple neural network systems in offering an alternative to handle online pattern classification tasks in possibly nonstationary environments.**

***Index Terms*—Adaptive Resonance Theory, benchmark studies, decision combination algorithms, multiple neural network systems, online learning.**

I. Introduction

IN PATTERN classification, the idea of using a committee of classifiers in solving a particular problem is not a new one. As early as in the eighteenth century, the Condorcet Jury model was designed to study the conditions under which a democracy model as a whole is more effective than any of its constituent members [1]. In general, members of a committee of classifiers can be statistical-based, syntactical-based, neural-network-based, or hybrid classifiers, or even a mixture of these classifiers. The primary objective of combining outputs from more than one classifier is to achieve better generalization than would be achieved by any of the constituent classifiers and, hence, to obtain better performance. The use of a single classifier system hinges on the assumption that the system is able to capture and to process all the input features satisfactorily regardless of what the features might be. In cases where the above assumption fails to hold true, e.g., the input features might consist of a variety of syntactic primitives, linguistic variables, continuous, discrete, or nominal attributes, presenting all these features to one classifier for it to make a decision is often difficult and can result in poor performance. Furthermore, concatenating all the features into a high-dimensional input vector will unduly induce the problem known as the "curse-of-dimensionality" [2]. Hence, many researchers have proposed the application of multiple classifier systems and the combination of results using some information fusion algorithm to reach an integrated consensus. In general, a committee of classifiers can be used in two ways: 1) select the output from the "best" (e.g., lowest error rate, highest posterior probability) of the constituent classifiers for each input; 2) combine the outputs from all the constituent classifiers. In this paper, we are concerned with the latter approach when using a committee of neural-network-based classifiers.

Methods for combining multiple networks can largely be categorized into two, i.e., the ensemble and modular approaches [3], [4]. In the ensemble approach, each network is trained using the same inputs such that each network provides a solution to the same task. Outputs from these redundant networks are combined to reach an integrated result. On the contrary, in the modular approach, a task is first decomposed into several subtasks and a specialist network is then trained using the inputs pertaining to the corresponding subtask. Subsolution outputs from each of the specialist networks are combined so that the complete solution to the task is obtained. Given a particular problem, both the ensemble and modular multiple network systems can coexist at different levels of the problem solution. A detailed review on multiple neural network systems can found in [3].

There are a number of situations where an ensemble of neural networks is applicable. For instance, an ensemble of networks can be used for selecting a subset of input features that best represents a particular problem [5]. In general, the main motivation for using an ensemble of neural networks is to mitigate the limitations of each constituent network. Since each network is prone to making errors from one realization to another, their outputs can be combined in such a way that the effect of these errors, in terms of ensemble bias and variance [6], is minimized. It has been shown that ensemble averaging is effective in reducing the variance, rather than bias, of errors from the networks [3].

Bagging and boosting [7]–[9] are recent methods for improving the predictive power of an ensemble of learning systems, particularly in pattern classification problems. Bagging generates an ensemble of classifiers, each learned from a training set formed by resampling (with replacement) the original data samples. On the other hand, boosting uses all data samples in each repetition, and maintains a weight for

Manuscript received February 2, 2001; revised November 25, 2002. This work was supported in part by Ministry of Science, Technology, and the Environment, Malaysia, under Grants 06-02-05-8002 and 04-02-05-0010, and in part by the University of Science, Malaysia. This paper was recommended by Associate Editor M. Embrechts.

C. P. Lim is with the School of Electrical and Electronic Engineering, University of Science Malaysia, Engineering Campus, 14300, Nibong Tebal, Penang, Malaysia (e-mail: cplim@usm.my).

R. F. Harrison is with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, U.K. (e-mail: r.f.harrison@sheffield.ac.uk).

Digital Object Identifier 10.1109/TSMCC.2003.813150

each sample in the training set that reflects its importance. By adjusting these weights, boosting forces a classifier to focus on different samples and thus leads to different results. Both bagging and boosting apply voting to aggregate predictions. Bagging assigns the same vote to each constituent classifier, while boosting gives different voting strengths to different classifiers according to their accuracy rates. While bagging primarily focuses on variance reduction of the ensemble classifiers, boosting acts to reduce both bias and variance. An empirical comparison [10] has shown that boosting is generally better than bagging, but not uniformly better for all the data sets used.

In the combination of modular networks, the concept of "divide and conquer" is usually adopted whereby a complicated task is decomposed into a number of subtasks with reduced complexity, and a specialist network is assigned to handle each subtask. The constituent networks, which can be composed by different learning systems, can exploit their specialist capabilities, and achieve results that would not be possible in a single network. For example, modular systems coupling neural networks with hidden Markov models have been proposed for speech recognition [11]. The mixture-of experts [12] and hierarchical mixtures-of-experts [13] have been used to partition the input space into several subspaces, and simple models are used to learn information contained in each subspace. It is argued that such data partitioning is more effective than training on the whole input data space. A gating network is used to output a set of scalar coefficients that serve to weight the contribution of the various experts [12], [13]. Different methods for "hybridization and specialization" of neural networks is presented in [14]. The idea is to use modularity to extend the capabilities of single networks for forming a system that is easier to train and to understand. For example, a combination of feedforward and recurrent networks is proposed such that the relative strengths of both networks for handling spatial and temporal tasks, respectively, are exploited.

The organization of this paper is as follows. In Section II, motivations for developing online learning systems along with a review on some incremental learning networks are presented. The building block of the proposed multiple-neural-network system, i.e., an autonomously learning model based on the supervised Adaptive Resonance Theory (ART) [15] network, is then described. Rationales and derivations of three algorithms used to combine predictions from multiple networks are presented in Section III. In Section IV, two benchmark pattern classification problems are employed to assess applicability of the proposed system. Different input pattern representation schemes (ensemble and modular) and learning strategies (offline and online) are studied systematically. The results are analyzed and compared with other existing algorithms. Implications of the results are also discussed, and conclusions are drawn in Section V.

II. Online Learning Neural Network Systems

Although many characteristics of neural networks have been studied, one domain that receives little attention, and yet is important to "intelligent" learning systems, is the ability to absorb knowledge continuously and autonomously without corrupting or forgetting previously acquired knowledge. This ability is necessary, for instance, in handling online learning problems in nonstationary environments. In certain situations, offline, batch learning is sometimes not a viable option. This may be because the data sets are too large or, more importantly, the data may evolve with time. It is this very reason that triggers the needs of autonomous, online learning systems. Specifically, the issue of stability-plasticity dilemma [15] is addressed, i.e., how a learning system is able to protect useful historical data from corruption (stability) while simultaneously learning new data (plasticity). This dilemma has also been addressed as the sequential learning problem [16], [17]. If a multilayer feedforward network with standard back-propagation is used to learn the training data sequentially, a phenomenon known as catastrophic forgetting may occur in which new knowledge will overwrite existing information stored in the network [17]. As a result, the learning system is neither stable nor usefully plastic.

To overcome the stability-plasticity dilemma, researchers have proposed a number of incremental learning algorithms with adaptive network structures. Instead of fixing the network size *a priori*, the idea is to recruit an appropriate number of nodes incrementally as data arrive so that a parsimonious network structure can be formulated. On the other hand, one can first use a large, oversized network. As learning proceeds, the network structure is adjusted by removing redundant nodes using some information-theoretic analysis [18]. According to [19, Ch. 15] online learning networks can largely be divided into three categories:

1) growing networks, which start with no nodes and gradually add new nodes when input samples are presented;
2) pruning networks, which start with a large number of nodes and then delete them subsequently;
3) growing and pruning networks, which add and delete nodes simultaneously to reach an appropriate network size.

Some incremental learning networks that are related to our work are discussed in the following section, and a detailed review of structurally adaptive networks can be found in [19]. Theoretical analysis, especially within the Bayesian framework, of online learning methods in neural networks can be found in [20].

On the issue of catastrophic forgetting, French [21] argues that forgetting is a direct consequence of distributed representation of information in a standard feedforward back-propagation network. One way to maintain generalization while reducing catastrophic forgetting is to use "semi-distributed" representations. An algorithm is proposed in [21] in which the network is allowed to develop semi-distributed representations by using a factor to compute the correlation between weight vectors encoded by the hidden nodes. However, this approach might result in a loss of information, and might affect generalization of the resulting network. Instead, adaptive versions of back-propagation that are suitable for on-learning have been introduced [22], [23]. Methods within the statistical physics framework have been adopted to analyze dynamics of online learning multilayer neural networks. Training samples are repetitively sampled from a fixed data set, and the correlations between network

parameters and training samples are examined. The proposed approaches provide useful insights to monitor the evolution of the error rate and to design optimal learning parameters.

The cascade-correlation Algorithm [24] is another learning algorithm for the construction of architecturally dynamic multilayer feedforward networks. The learning procedure starts with a minimal network and incrementally builds a suitable cascaded structure with as many layers as the number of added hidden nodes. Many researchers have investigated and modified the Cascade-Correlation Algorithm to suit various application domains [25]–[27]. In addition, researchers have proposed algorithms for growing or shrinking the radial basis function (RBF) networks. A resource allocating network (RAN) for function interpolation is introduced in [28]. The system is essentially a growing Gaussian RBF network which processes information sequentially and adapts the network weights using the least mean squares algorithm. If novelty is detected, i.e., the network prediction is unacceptably inaccurate with regard to certain criteria [28], then a new basis function located on the input is added. In [29], an enhanced learning algorithm based on Kalman filter is proposed for RAN, and work on the enhanced RAN algorithm has been described in [30] and [31].

In an attempt specifically to overcome the stability-plasticity dilemma, a family of competitive learning networks known as ART has been proposed [15], [32]–[37]. The ART models have revealed promising characteristics for building autonomous learning systems. In general, the growth criterion of many neural networks relies upon a similarity measure (for instance, a distance metric) between the input pattern and learned exemplars to select the best-matched prototype. An arbitrary threshold is then applied to decide whether or not to add a new node. ART has a similar growing methodology. However, one distinct difference is that an ART network has a two-stage hypothesis selection and test process. On presentation of a new input pattern, a feedforward pass is initiated to select the most similar prototype according to a competitive selection process. The winning prototype, nonetheless, has to undergo a feedback pass to perform a test against a vigilance threshold. If the vigilance criterion is not satisfied, then a new cycle of search (selection and test) ensues until the criterion is satisfied by an existing prototype, or the creation of a new node to code the input pattern. It is the inclusion of this feedback mechanism that assists in forming a stable, and yet plastic knowledge structure in ART networks. This characteristic, thus, differentiates ART from other incremental neural network models.

A. Fuzzy ARTMAP (FAM)

A supervised ART network known as FAM [35] which realizes a synthesis of ART and fuzzy logic has been introduced. Fig. 1 depicts a schematic diagram of the FAM network. It consists of two fuzzy ART [33] modules, ART_a and ART_b, linked by a map field, F^{ab}. The ART_a (ART_b) module has two main layers of nodes: F_1^a (F_1^b) is the input layer; and F_2^a (F_2^b) is a dynamic layer where each node encodes a prototype pattern of a cluster of input patterns, and the number of nodes can be increased when necessary. F_0^a (F_0^b) is a normalization layer which performs complement-coding [33]–[35] of the input patterns to avoid the category proliferation problem.

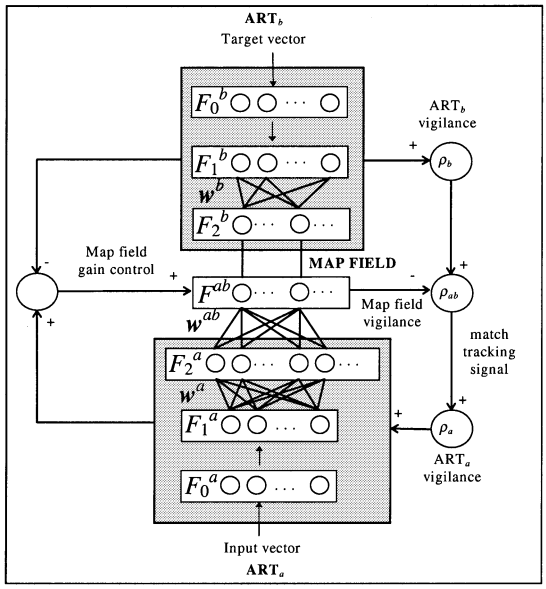

Fig. 1. FAM network.

During supervised learning, ART_a receives a stream of input pattern vectors, $\{\boldsymbol{A}\}$, whereas ART_b receives the corresponding target-class vectors, $\{\boldsymbol{B}\}$. In general, ART_b consists of an independent Fuzzy ART module to self-organize the target vectors. However, in one-from-N classification (i.e., each input pattern belongs to only one of the N possible output classes), ART_b can be replaced by a single layer containing N nodes. Then, the N-bit teaching stimulus can be coded to have unit value corresponding to the target category and zero for all others.

The learning algorithm of FAM is similar to the sequential leader clustering algorithm [38]. However, FAM does not directly associate input patterns at ART_a to target patterns at ART_b. Rather, input patterns are first classified into prototypical category clusters before being linked with their target outputs via a map field. At each input pattern presentation, this map field establishes a permanent link between the winning F_2^a category prototype and the target output in F_2^b. This association is used, during test, to recall a prediction when an input pattern is presented to ART_a.

B. Probabilistic Neural Network (PNN)

The PNN [39] is a neural network model that implements the Bayes' theorem in its learning methodology. It learns instantaneously in one-pass through the data samples and is able to form complex decision boundaries that approximate asymptotically the Bayes optimal limits. In addition, the decision boundaries can be modified online when new data is available without having to retrain the network. The key feature of the PNN is its ability to estimate the probability density functions (pdfs) based on data samples by using the Parzen-window technique [40]. Fig. 2 depicts a schematic diagram of the PNN for binary classification tasks (class A or B). The PNN consists of four layers of nodes: the input layer, pattern layer, summation layer, and output layer. Nodes in the pattern layer are organized

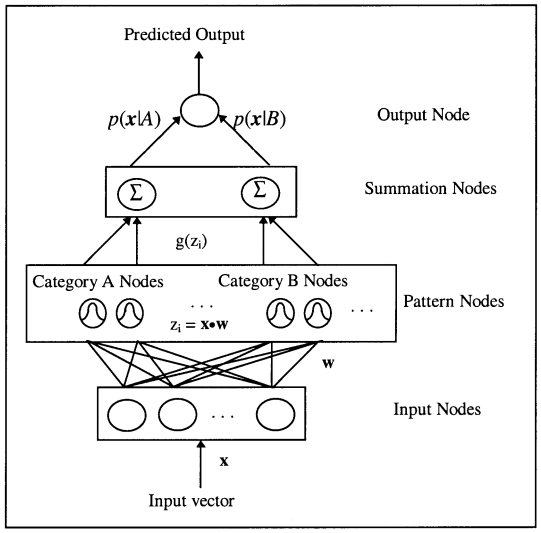

Fig. 2. Probabilistic neural network.

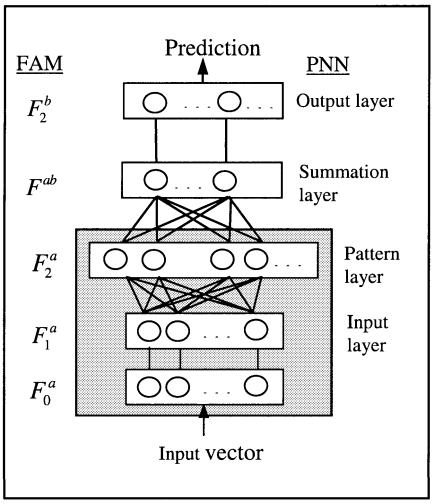

Fig. 3. Structure of the PFAM network.

in groups corresponding to different target classes. The pattern nodes belonging to the same output are then linked to a summation node dedicated to that particular target class. During operation, the input pattern, $\boldsymbol{x}$, is first fanned-out to the pattern layer where each pattern node computes a distance measure between the input and the weight pattern represented by that node. The distance measure (e.g., dot-product) is then shunt through a Parzen kernel. The summation nodes sum outputs from the Parzen kernels. These outputs correspond to estimates of the pdfs of the input pattern with respect to each target class, i.e., $p(\boldsymbol{x}|A)$, $p(\boldsymbol{x}|B)$.

C. Probabilistic FAM (PFAM)

One disadvantage of the PNN is that it encodes every input pattern as a new node in the network, thus, increases the network complexity and computational cost if large or unbounded data sets are used. Nevertheless, this problem can be alleviated by using a clustering technique such as FAM. Our studies have found that there is a close similarity in the network topology between FAM and the PNN, as shown in Fig. 3. Notice that the F_1^a and F_2^a layers correspond to the input and pattern layers whereas the map field layer (F^{ab}) corresponds to the summation layer. In one-from-N classification, each node in F_2^a is permanently associated with only one node in F^{ab}, which is then linked to the target output in F_2^b. Thus, the map field nodes can be used to sum outputs from all the F_2^a nodes corresponding to a particular target class, taking the role of the PNN summation nodes. In view of the suitability of the incremental learning property and the similarity of the network topology between FAM and the PNN, a novel hybrid network, based on the integration of FAM and the PNN, has been proposed for online classification and probability estimation tasks, and is called PFAM [41].

The online PFAM algorithm is divided into two phases. First, the FAM clustering procedure is used for classifying the input patterns into different categories (learning phase). Subsequently, the PNN probability estimation procedure is used to predict a target output (prediction phase). The advantage of this integration is twofold: 1) a probabilistic interpretation of output classes is established which enables the application of Bayes, risk-weighted, classification in FAM; 2) the number of pattern nodes in the PNN is reduced by the clustering procedure of FAM. The above description provides a conceptual framework for incorporating FAM and the PNN into a unified, hybrid system, and the rationale behind their integration. In practice, several modifications are necessary to allow effective combination of both the networks, and to increase generalization ability of the resulting system. These include procedures to estimate kernel centers and widths. Explanation on all these procedures can be found in [41].

III. Decision Combination in Multiple Classifier Systems (MCSs)

In addition to the motivations for combining multiple networks stated in Section I, there is another reason for the use of multiple ART-based classifiers (and most incremental learning systems), i.e., to minimize the effect of data ordering in online learning. In incremental learning systems such as ART, the formation of prototypes is affected by the sequence of input sample presentations. This could lead to different predictions of target classes, and thus different accuracy scores for each network realization. Thus, good long-term performance of the network depends on a good initial formation of cluster prototypes. The data ordering effect is further exacerbated if the prototypes are to be established autonomously, online, because in this case the input samples are presented only once, and in a fixed order. One way to mitigate this problem is to train a pool of networks offline, each with a different ordering of input samples. During the prediction phase, the results from several networks can be combined to give an overall prediction. It is, therefore, worthwhile to investigate how to integrate decisions from multiple PFAM networks so that a robust and high performance classification system could be formed.

Here, three decision combination algorithms are discussed to form a PFAM-based MCS, as shown in Fig. 4. The first one is a simple majority-voting scheme [42] where the target class that receives the highest number of votes is selected as the final prediction. The second method is based on the Bayesian theorem

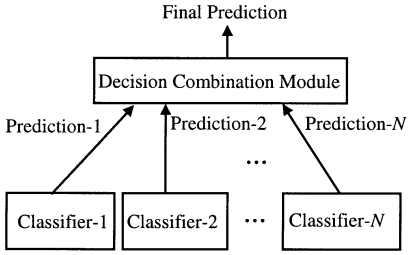

Fig. 4. Schematic diagram of an MCS with N independent classifier modules. Outputs from all classifiers are combined using a suitable decision combination algorithm to give an overall prediction.

[42], [43] that is commonly employed in evidence gathering and uncertainty reasoning. The third one is the behavior-knowledge space (BKS) approach proposed in [44].

A. Majority Voting

Suppose there are M target classes where each class is represented by $C_i, \forall i \in \Lambda = \{1, 2, \ldots, M\}$, the task of a classifier is to assign the input sample, $\boldsymbol{x}$, to one of the $(M+1)$ classes, with the $(M+1)$th class denoting that the classifier rejects $\boldsymbol{x}$. The most common method to combine the outputs is by majority voting. If there are K classifiers denoted by $e_1, \ldots, e_K$, the problem is to produce a combined result, $E(\boldsymbol{x}) = j$, $j \in \{1, 2, \ldots, M, M+1\}$, from all K predictions, $e_k(\boldsymbol{x}) = j_k$, $k = 1, \ldots, K$. A binary function [42] can be used to represent the number of votes, i.e.,

$$V_k(\boldsymbol{x} \in C_i) = \begin{cases} 1, & \text{if } e_k(\boldsymbol{x}) = i, i \in \Lambda \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Then, sum the votes from all K classifiers for each C_i

$$V_E(\boldsymbol{x} \in C_i) = \sum_{k=1}^{K} V_k(x \in C_i), \qquad i = 1, \ldots, M \tag{2}$$

and the combined result, $E(\boldsymbol{x})$, can be determined by

$$E(\boldsymbol{x}) = \begin{cases} j, & \text{if } V_E(\boldsymbol{x} \in C_j) = \max_{i \in \Lambda} V_E(\boldsymbol{x} \in C_i) \\ & \quad \text{and } \dfrac{V_E(\boldsymbol{x} \in C_j)}{K} \geq \lambda \\ M+1, & \text{otherwise} \end{cases} \tag{3}$$

where $0 \leq \lambda \leq 1$ is a user-defined threshold that controls the confidence in the final decision [42].

B. Bayesian Approach

The voting strategy is solely based on the predicted outcomes produced from all classifiers, where each classifier is treated equally without considering its errors. A more reasonable approach is to take into account the predictive accuracy of each classifier. The predictions from highly accurate classifiers should be given more weight than those from the less accurate ones. This is the rationale behind the use of the Bayesian approach to combine decisions from multiple classifiers. The Bayesian algorithm presented below has been proposed in [42], but the basic idea has been used previously for evidence propagation and uncertainty reasoning in intelligent systems [43].

Given a data set containing N samples, all predictions (correct and incorrect ones) of the kth classifier, e_k, is recorded in its confusion matrix constructed as follows:

$$CM^k = \begin{pmatrix} n_{11}^k, & n_{12}^k & \cdots & n_{1(M+1)}^k \\ n_{21}^k, & n_{22}^k & \cdots & n_{2(M+1)}^k \\ \vdots & \vdots & \ddots & \vdots \\ n_{M1}^k, & n_{M2}^k & \cdots & n_{M(M+1)}^k \end{pmatrix} \tag{4}$$

where n_{ij}^k, $i = 1, \ldots M$, $j = 1, \ldots, M+1$ indicates the number of samples belonging to C_i, but assigned to class j by e_k. The total number of samples encountered by e_k is

$$N = \sum_{i=1}^{M} \sum_{j=1}^{M+1} n_{ij}^k \tag{5}$$

and the number of samples belonging to C_i is

$$n_{i\bullet}^k = \sum_{j=1}^{M+1} n_{ij}^k \tag{6}$$

i.e., summation through row i. The number of samples that is assigned to class j by e_k is

$$n_{\bullet j}^k = \sum_{i=1}^{M} n_{ij}^k \tag{7}$$

i.e., summation through column j. This confusion matrix provides information regarding a classifier's ability to classify accurately samples from a particular target class. In the event $e_k(\boldsymbol{x}) = j$ (classifier e_k predicts that $\boldsymbol{x}$ belongs to C_j), the truth that $\boldsymbol{x}$ really comes from class C_j is associated with a factor of uncertainty. By utilizing information stored in the confusion matrix, the uncertainty of proposition $\boldsymbol{x} \in C_i$ given $e_k(\boldsymbol{x}) = j$ can be computed according to

$$P(\boldsymbol{x} \in C_i | e_k(\boldsymbol{x}) = j) = \frac{n_{ij}^k}{n_{\bullet j}^k} = \frac{n_{ij}^k}{\sum_{i=1}^{M} n_{ij}^k}, \qquad i = 1, \ldots, M. \tag{8}$$

From the viewpoint of uncertainty reasoning, the confusion matrix of a classifier can be regarded as a collection of evidence supporting different target classes, and (8) can be interpreted as a set of belief functions, $bel(\bullet)$, on M propositions that $\boldsymbol{x} \in C_i$ [42]. The higher the belief function of a proposition is, the more likely that it is true. With K classifiers, there will be K confusion matrices and K events, $e_1(\boldsymbol{x}) = j_1, \ldots, e_K(\boldsymbol{x}) = j_K$. Each classifier $\boldsymbol{x} \in C_i$ expresses its belief functions [42] as

$$bel(\boldsymbol{x} \in C_i | e_k(\boldsymbol{x}) = j_k, EN) = P(\boldsymbol{x} \in C_i | e_k(\boldsymbol{x}) = j_k) \qquad i = 1, \ldots, M \tag{9}$$

where EN denotes the common classification environment that consists of all K events. The problem now is how to integrate the belief functions from K classifiers into a combined set of belief functions.

Bayesian formalism has been adopted to propagate and update the belief functions in the Bayesian network [43]. The same

idea has been used in [42] to combine K sets of belief function. Using Bayes' theorem, (9) is expanded to

$$\begin{aligned} &bel(i) \\ &= bel(\boldsymbol{x} \in C_i | e_1(\boldsymbol{x}) = j_1, \ldots, e_K(\boldsymbol{x}) = j_K, EN) \\ &= P(\boldsymbol{x} \in C_i | e_1(\boldsymbol{x}) = j_1, \ldots, e_K(\boldsymbol{x}) = j_K, EN) \\ &= \frac{P(e_1(\boldsymbol{x}) = j_1, \ldots, e_K(\boldsymbol{x}) = j_K | \boldsymbol{x} \in C_i, EN) P(\boldsymbol{x} \in C_i | EN)}{P(e_1(\boldsymbol{x}) = j_1, \ldots, e_K(\boldsymbol{x}) = j_K | EN)}. \end{aligned} \tag{10}$$

To simplify the combination of the belief functions, it is assumed that the environment EN consists of K-independent events with M mutually exclusive sets of target output. Thus, the joint probability is reduced to

$$\begin{aligned} &\frac{P(e_1(\boldsymbol{x}) = j_1, \ldots, e_K(\boldsymbol{x}) = j_K | \boldsymbol{x} \in C_i, EN)}{P(e_1(\boldsymbol{x}) = j_1, \ldots, e_K(\boldsymbol{x}) = j_K | EN)} \\ &\qquad = \frac{\prod_{k=1}^{K} P(e_k(\boldsymbol{x}) = j_k | \boldsymbol{x} \in C_i, EN)}{\prod_{k=1}^{K} P(e_k(\boldsymbol{x}) = j_k | EN)}. \end{aligned} \tag{11}$$

Using the Bayes' rule

$$\frac{P(e_k(\boldsymbol{x}) = j_k | \boldsymbol{x} \in C_i, EN)}{P(e_k(\boldsymbol{x}) = j_k | EN)} = \frac{P(\boldsymbol{x} \in C_i | e_k(\boldsymbol{x}) = j_k, EN)}{P(\boldsymbol{x} \in C_i | EN)} \tag{12}$$

(11) becomes

$$\begin{aligned} &\frac{\prod_{k=1}^{K} P(e_k(\boldsymbol{x}) = j_k | \boldsymbol{x} \in C_i, EN)}{\prod_{k=1}^{K} P(e_k(\boldsymbol{x}) = j_k | EN)} \\ &\qquad = \frac{\prod_{k=1}^{K} P(\boldsymbol{x} \in C_i | e_k(\boldsymbol{x}) = j_k, EN)}{\prod_{k=1}^{K} P(\boldsymbol{x} \in C_i | EN)}. \end{aligned} \tag{13}$$

By using (11) and (13), (10) becomes

$$bel(i) = \frac{\prod_{k=1}^{K} P(\boldsymbol{x} \in C_i | e_k(\boldsymbol{x}) = j_k, EN)}{\prod_{k=1}^{K} P(\boldsymbol{x} \in C_i | EN)} P(\boldsymbol{x} \in C_i | EN). \tag{14}$$

To further simplify calculation of the combined belief function, the following estimate can be used [42]:

$$bel(i) = \frac{\prod_{k=1}^{K} P(\boldsymbol{x} \in C_i | e_k(\boldsymbol{x}) = j_k)}{\sum_{i=1}^{M} \prod_{k=1}^{K} P(\boldsymbol{x} \in C_i | e_k(\boldsymbol{x}) = j_k)} \tag{15}$$

e_2 \ e_1	1	2	$\cdots$	$M+1$
1	U_{11}	U_{12}	$\cdots$	$U_{1(M+1)}$
2	U_{21}	U_{22}	$\cdots$	$U_{2(M+1)}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$M+1$	$U_{(M+1)1}$	$U_{(M+1)2}$	$\cdots$	$U_{(M+1)(M+1)}$

Fig. 5. Two–dimensional BKS.

where each $P(\boldsymbol{x} \in C_i | e_k(\boldsymbol{x}) = j_k)$ can be computed from the confusion matrix using (8) by replacing j with j_k. Based on the combined belief functions, the one with the highest estimate is selected as the final outcome, i.e.,

$$E(\boldsymbol{x}) = \begin{cases} j, & \text{if } bel(j) = \max_{i \in \Lambda} bel(i) \\ & \quad \text{and } bel(j) \geq \lambda \\ M+1, & \text{otherwise.} \end{cases} \tag{16}$$

Again, $0 \leq \lambda \leq 1$ is a user-defined threshold to regulate confidence associated with the final decision.

C. BKS Approach

One of the criticisms of the Bayesian approach is the assumption that all classifiers must operate independently in order to tackle the computation of the joint probabilities. This assumption is unlikely to hold in many applications. To avoid using the assumption, the BKS approach that concurrently records the decisions of all classifiers on each input sample is proposed in [44].

A BKS is a K-dimensional space in which each dimension corresponds to the decision of one classifier. The intersection of the decision from multiple classifiers occupies one unit of the BKS, e.g., $\text{BKS}(e_1(\boldsymbol{x}) = j_1, \ldots, e_K(\boldsymbol{x}) = j_K)$ denotes a BKS unit where each e_k produces a prediction j_k, $k = 1, \ldots, K$. In each BKS unit, there are M partitions (cells) for accumulating the number of samples actually belonging to C_i. As an example, suppose two classifiers are used to categorize the input samples into M target classes. Then a two-dimensional (2–D) BKS can be formed, as shown in Fig. 5.

Each BKS unit, U_{ij}, is further divided into M cells, $n_1^H, \ldots, n_M^H$, where H denotes the overall prediction $e_1(\boldsymbol{x}) = j_1, \ldots, e_K(\boldsymbol{x}) = j_K$. The frequency of the number of samples belonging to n_i^H, $i = 1, \ldots, M$ is recorded in each n_i^H, $i = 1, \ldots, M$. When an input sample, $\boldsymbol{x}$, is presented, one of the BKS units will become active (known as the focal unit) after the decisions from all K classifiers have been received. In the above example, U_{34} will be selected as the focal unit if $e_1(\boldsymbol{x}) = 3$ and $e_2(\boldsymbol{x}) = 4$. Then, the total number of samples in the focal unit is computed

$$T(H) = \sum_{i=1}^{M} n_i^H \tag{17}$$

and the best representative class (i.e., the one that contains the highest number of samples) is identified

$$R(H) = j \quad \text{where} \quad n_j^H = \max_{i \in \Lambda}(n_i^H). \tag{18}$$

The decision rule for determining the final outcome is

$$E(\boldsymbol{x}) = \begin{cases} R(H), & \text{if } T(H) > 0 \text{ and } \dfrac{n^H_{R(H)}}{T(H)} \geq \lambda \\ M+1, & \text{otherwise} \end{cases} \quad (19)$$

where $0 \leq \lambda \leq 1$ is a user-defined confidence threshold.

The BKS appears to be similar to the confusion matrix used in the Bayesian approach. However, the Bayesian method needs to compute the multiplication of evidence taken from the confusion matrices to approximate the joint probability of K events when evaluating the combined belief function. This step is avoided in the BKS approach, in which a final decision is made by assigning the input sample directly to the class that has accumulated the largest number of samples. The simplicity of this BKS approach may not be detrimental; instead it has provided a fast and effective way of combining multiple predictions, as demonstrated in [44] for the classification of unconstrained handwritten numerals.

IV. Offline and Online Pattern Classification

In the following experiments, two benchmark data sets were used and a few practical operating strategies were envisaged to allow the system to learn incrementally and to classify patterns autonomously. These two benchmark problems have been studied in [37] using single-channel FAM and multichannel Fusion ARTMAP. Note that Fusion ARTMAP is an extension of FAM that introduces a modularized technique for data fusion and classification. Hence, the results in [37] can be compared with those in the following experiments. In order to have a fair comparison, the procedures in [37] have been followed as closely as possible, e.g., network parameters, performance indicators, number of training and test samples, number of experimental runs. Unless otherwise stated, the PFAM network parameters used throughout all experiments were: baseline vigilance parameter, $\overline{\rho}_a = 0.0$; learning rate, $\beta_a = 1.0$ (fast learning); $\alpha_a \approx 0.0$ (conservative mode) [35]; overlapping parameter, $r = 1.0$ [41]. Note that the above parameters were set to their "default" values, and no additional effort was spent in fine-tuning these parameters. Indeed, it is our belief that a good system should require as few user-specified parameters as possible. Nevertheless, a system with some adjustable parameters may allow users to regulate the system complexity, e.g., $\overline{\rho}_a$ offers a means to govern the coarseness or fineness of the clusters to be formed in ARTMAP networks [34], [35]. Specifically, two different ways of input pattern representation were experimented using the PFAM-based MCSs.

1) *Ensemble Approach*: where the input samples were used in their original, concatenated form. In this approach, a number of PFAM classifiers were first trained using different orderings of the training set, and then tested on the test set. Predictions from all classifiers were combined with the voting, Bayesian and BKS methods to give an overall decision.

2) *Modular Approach*: where each input sample was divided into groups of related attributes. During training, each group of attributes extracted from a training sample was assigned to a PFAM classifier, and during test, the predictions from all classifiers were combined using the three combination methods to give a final decision.

A. Quadruped Mammals Data Set

This data set is an artificial domain first used to evaluate CLASSIT, an unsupervised machine learning algorithm [45]. There are four types of mammals, namely cats, dogs, horses, and giraffes. Each input sample is described by a set of eight components: head, tail, neck, torso, and four legs; and each component is further described by nine attributes: texture, height, radius, three locations, and three axes. Hence, there are 72 attributes per sample in total. The program for generating the data samples can be obtained from [46].

1) Offline Learning With the Ensemble Approach: In [37], fusion ARTMAP applied each of the eight components (head, tail, neck, torso, and four legs—each with nine attributes) to a different unsupervised ART network (modular approach), and the results were concatenated to a global network to make a final prediction. Two training sets comprising 100 and 1000 samples were applied. The trained network was tested on 1000 samples. The performance, averaged over three runs, was compared to an ensemble approach by concatenating all 72 attributes to a single FAM network.

Here, two experiments were conducted using the same training and test set sizes as in [37]. In addition to averaging the results of three runs, MCSs were formed to combine the outcomes from these three runs. Table I lists the experimental results together with the reported FAM results in [37].

A few observations can be made from Table I. The performance of PFAM is inferior to that of FAM, which shows perfect results in both experiments. The failure of PFAM to achieve the same performance as FAM might be due to the small number of prototype patterns being created in the system. The average number of prototypes created in PFAM was, respectively, 5.7 for 100 training samples and 7.8 for 1000 training samples. Notice that PFAM (so as PNN) uses Parzen windows to estimate the pdfs. Accuracy of the pdfs depends on the number of prototypes, i.e., the larger the number of prototypes, the more accurate the estimated pdfs become. Theoretically, the pdfs will converge asymptotically to the actual underlying functions in the limit as the number of prototypes increases [39], [40]. As can be seen in Table I, performance of PFAM improved as the training samples increased from 100 to 1000, in which case more prototypes are created. In comparing performance between single and multiple classifiers, MCSs were able to improve the results of individual classifiers. This is true for all three decision combination methods. From Table I, the Bayesian and BKS methods show a better performance than the simple majority-voting rule.

2) Offline Learning With the Modular Approach: Three MCSs were formed, each with eight modules of PFAM classifiers. Each classifier was dedicated to handle one group of the data components, i.e., head, tail, neck, torso, and four legs. All the experiments were repeated three times. The average results of individual classifiers are listed in Table II. The results of Fusion ARTMAP in [37] and MCSs are shown in Table III.

Again, a committee of classifiers proves to be useful in improving the performance of single classifiers. Perfect results (100% accuracy) were obtained with the Bayesian and BKS

TABLE I
RESULTS OF INDIVIDUAL AND MULTIPLE CLASSIFIERS FROM THE ENSEMBLE APPROACH USING THE QUADRUPED MAMMALS DATA SET (THE FAM RESULTS ARE ADAPTED FROM [37])

Training Set Size	FAM	PFAM	Multiple Classifier System		
			Voting	Bayesian	BKS
100	100%	97.7%	99.6%	100%	100%
1000	100%	99.6%	100%	100%	100%

TABLE II
RESULTS OF INDIVIDUAL CLASSIFIERS FROM THE MODULAR APPROACH USING THE QUADRUPED MAMMALS DATA SET

Classifier	Accuracy (%)	
	100 Samples	1000 Samples
Head	96.3	98.7
Neck	98.9	99.0
Torso	95.8	97.2
Tail	96.5	97.0
Leg 1	98.1	98.9
Leg 2	96.6	98.1
Leg 3	96.8	99.2
Leg 4	96.5	97.3

TABLE III
RESULTS OF MULTIPLE CLASSIFIERS FROM THE MODULAR APPROACH USING THE QUADRUPED MAMMALS DATA SET

Training Set Size	Fusion ARTMAP	Multiple Classifier Systems		
		Voting	Bayesian	BKS
100	96%	99.9%	100%	100%
1000	100%	100%	100%	100%

approaches for three runs on both cases of 100 and 1000 training samples. Notice that the MCS results listed in Table III are equivalent to, if not better than, those of fusion ARTMAP. Nevertheless, by increasing the training samples to 1000, all classifiers were able to perform with perfect accuracy. When individual classifiers are concerned, the ensemble approach (Table I—FAM and PFAM) achieves better performance than the modular approach (Table II). By reducing the input dimension from 72 to 9, fewer prototypes were formed using the modular approach (average numbers of prototypes were 4.2 and 6.1) during the experiments. As a result, the estimated pdfs were less accurate compared to those formulated using the ensemble approach, which in turn caused a lower classification accuracy. This drawback, however, can be overcome by combining the decisions from multiple classifiers (Table III).

3) Online Learning: In online learning, the system imitates the condition of a human operating in a natural environment. Each incoming datum is used as a training sample as well as a test sample. The online learning cycle proceeds as follows: an input pattern is presented to PFAM, and a prediction is made. The prediction is compared with the target class to determine its correctness or otherwise. This outcome constitutes the classification accuracy. Learning then ensues to associate the input with its target class.

The modular approach with eight PFAM classifiers (each with nine attributes) was compared with the ensemble approach of a single PFAM classifier (72 attributes). Note that in online learning, the voting, Bayesian and BKS procedures are not applicable to combine the results from the ensemble approach. This is because there is no differentiation between training and test sets—all data samples were first tested and then trained in a fixed order. Hence, knowledge established in all classifiers would be the same since they are trained on the same sequence of samples. In the modular approach, however, each classifier is trained on only a group of attributes of the sample. Therefore, MCSs can be formed to combine the predictions from classifiers based on disparate attribute groups.

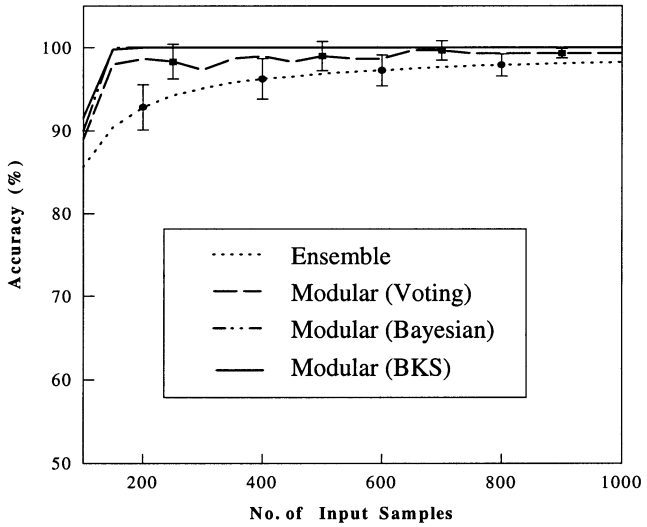

Fig. 6. A comparison of the online results between the individual classifiers with the ensemble approach and the MCS with the modular approach. The error bars indicate the standard deviations of the average results from three runs.

In this experiment, 1000 data samples were generated. To calculate the online accuracy, a 100-sample window was applied, e.g., accuracy at sample 200 was the percentage of correct predictions from trials 101–200. All the online results were averaged across three runs. Fig. 6 depicts a comparison of the online results between the ensemble and modular approaches. The standard deviations of three runs are plotted as error bars to indicate the spread of individual results across the averages. As can be seen, the modular approach attained a performance superior to that of the ensemble approach. Perfect results (100% accuracy) were achieved by the Bayesian and BKS methods after encountering fewer than 200 input samples. This perfect performance was maintained until the end of experiments in all three attempts. Although voting exhibited inferior results to the Bayesian and BKS methods, it still outperformed the ensemble approach.

B. Landsat Satellite Images

This database comprises a small subsection (82×100 pixels) of a scene from the original satellite images. Each pixel covers an area of approximately 80×80 meters on the ground. One frame of the Landsat satellite imagery comprises intensities of four spectral bands of the same scene. Two of the spectra are in the visible region (corresponding approximately to green and red regions), and two are in the (near) infrared region. The Landsat satellite images database is also obtainable from [46]. The database has been divided into a training set of 4435 samples, and a test set of 2000 samples. Each sample has 36 attributes (nine attributes for each of the four spectral bands), and there are altogether six target classes.

TABLE IV
CLASSIFICATION ACCURACY RATES, AS REPORTED IN [47], FROM VARIOUS ALGORITHMS FOR THE LANDSAT SATELLITE DATA SET

Algorithm	Accuracy (%)	Algorithm	Accuracy (%)
k-NN	90.6	Cal5	84.9
LVQ	89.5	Quadisc	84.5
DIPOL92	88.9	AC^2	84.3
RBF	87.9	SMART	84.1
ALLOC80	86.8	Logdisc, Cascade	83.7
CART	86.2	Discrim	82.9
IndCART	86.2	Kohonen	82.1
Back-prop	86.1	CASTLE	80.6
Baytree	85.3	NaiveBay	71.3
NewID, CN2, C4.5	85.0	Default	23.1

TABLE V
RESULTS OF SINGLE AND MULTIPLE CLASSIFIERS FROM THE ENSEMBLE APPROACH USING THE LANDSAT SATELLITE DATA SET (THE FAM RESULTS ARE ADAPTED FROM [37])

Algorithm		$\overline{\rho}_a = 0.0$	$\overline{\rho}_a = 0.9$
FAM	Accuracy (%)	83.0	89.0
	No. of Prototypes	89	704
PFAM	Accuracy (%)	81.4	89.0
	No. of Prototypes	87	518
Accuracy of MCS (%)	Voting	86.1	90.8
	Bayesian	87.0	91.6
	BKS	91.7	94.5

This Landsat database serves as a more challenging benchmark problem compared with the quadruped mammal database as it comprises real and noise-corrupted satellite images. Furthermore, many classification algorithms have been evaluated using this data as part of the Statlog Project [47]. There are a variety of indicators presented in [47] for performance comparison among various algorithms. Here, we only extracted the accuracy rates in [47], as listed in Table IV, for comparison purposes with those from FAM and Fusion ARTMAP in [37]. Other indicators such as memory storage, computational time, have been excluded. Note that the "Default" accuracy rate in Table IV is calculated by categorizing all the test samples as belonging to the class that has the highest number of samples, i.e., the maximum *a priori* classification rule.

With the ensemble approach, each input sample to PFAM consisted of a 36 dimensional vector which had been normalized between zero and one. With the modular approach, the same input sample was divided into four spectral bands, each comprising a nine-dimensional vector (again, normalized between zero and one) corresponding to a different spectral band. All training data were randomized to produce five differently ordered training sets, and the results were averaged across five runs.

1) Offline Learning With the Ensemble Approach: Table V shows the results of PFAM and the MCSs from our experiments as well as those of FAM in [37]. In accordance with the experiments reported in [37], two baseline vigilance values were tested, i.e., a low value of $\overline{\rho}_a = 0.0$ to create coarse clusters of input samples, and a high value of $\overline{\rho}_a = 0.9$ to create fine clusters. Other network parameters were the same as those used in the quadruped mammal experiments. A considerable improvement in classification accuracy was achieved with $\overline{\rho}_a = 0.9$ with the tradeoff being the increased numbers of prototypes created by FAM and PFAM, which were about eightfold and sixfold, respectively, compared to those created by $\overline{\rho}_a = 0.0$. Note that the k-NN approach, which achieved the best accuracy of 90.6% among all algorithms used in the Statlog Project, has to store all 4435 training samples as prototypes [37]. Thus, in comparison with k-NN, FAM and PFAM (both with $\overline{\rho}_a = 0.9$) could achieve a slightly lower accuracy (89% versus 90.6%) but with a higher degree of code compression.

TABLE VI
RESULTS OF INDIVIDUAL CLASSIFIERS FROM THE MODULAR APPROACH USING THE LANDSAT SATELLIET DATA SET

Classifier		$\overline{\rho}_a = 0.0$	$\overline{\rho}_a = 0.9$
Band 1	Accuracy (%)	52.7	56.2
	No. of Prototypes	891	991
Band 2	Accuracy (%)	51.2	54.0
	No. of Prototypes	628	1021
Band 3	Accuracy (%)	37.3	42.2
	No. of Prototypes	1063	1375
Band 4	Accuracy (%)	50.9	54.9
	No. of Prototypes	766	1099

TABLE VII
RESULTS OF MULTIPLE CLASSIFIERS FROM THE MODULAR APPROACH USING THE LANDSAT SATELLIET DATA SET

MCS	$\overline{\rho}_a = 0.0$	$\overline{\rho}_a = 0.9$
Voting	59.4%	65.6%
Bayesian	69.2%	74.8%
BKS	83.3%	86.1%

As might be expected, a committee of classifiers was able to improve on the results of individual classifiers. In terms of performance comparison, the BKS-based MCS achieved the best results, i.e., 91.7% accuracy for $\overline{\rho}_a = 0.0$ and 94.5% accuracy for $\overline{\rho}_a = 0.9$. The voting and Bayesian MCSs also outperformed FAM, PFAM, and many other algorithms in Table IV.

With $\overline{\rho}_a = 0.0$, the performance of PFAM was relatively poor compared with other methods. This may be accounted for by the same phenomenon observed in the experiments using the quadruped mammal data set, i.e., accuracy of the estimated pdfs was directly affected by the number of prototypes. That is why a substantial improvement could be achieved by PFAM with $\overline{\rho}_a = 0.9$, in which more than 500 prototypes were created.

2) Offline Learning With the Modular Approach: In addition to the concatenated input samples, a modular approach of dividing the input sample attributes into the four corresponding spectral bands has been tested. Table VI summarizes the average results and the number of prototypes of five runs. The modular approach proved to be a failure with this Landsat data set. As can be seen from Table VI, not only were the classification results exceptionally poor, but there was a proliferation of prototypes. Even by raising $\overline{\rho}_a$ to 0.9, the results were still significantly inferior to those from the ensemble approach. Among the four spectral bands, accuracy of the band 3 classifier was the worst with the highest number of prototypes.

Table VII shows the results of the MCSs by combining the outcomes from the four individual (band) classifiers. Generally, classification accuracy improved with the voting, Bayesian or BKS methods. In particular, the BKS approach produced an increase of more than 30% in accuracy for both $\overline{\rho}_a = 0.0$ and

$\overline{\rho}_a = 0.9$ compared to individual classifiers. The BKS results are also comparable with those in Table IV.

Deficiency of the modular approach is also experienced with the Fusion ARTMAP network. According to [37], the best performance of Fusion ARTMAP was about 70%, and the same prototype proliferation problem was observed. Failure of the modular approach on this data set might be due to two factors: interspectral dependency and the presence of noise in the satellite images. By segmenting the attributes of different spectral bands to different classifiers, interspectral information is lost. Without the benefits of information from other spectra, noise in the data is aggregated. For example, the "red" sensor is insensitive to certain "green" frequencies, and it will produce very irregular images associated with different output classes when the region under scrutiny is mostly green [37]. This pitfall is avoided in FAM when all sensor data are concatenated into one input sample so that interspectral information can be utilized by the classifier to form input features which are more consistent with the target class. Thus, FAM is able to achieve good results even with a reduced number of prototypes.

3) Offline Learning With Variable Confidence Threshold: In binary classification problems, it is useful to apply the receiver operating characteristics (ROC) curve to assess the tradeoff between false positive/false negative of a classifier, especially in medical domain [48]. Indeed we have reported the use of ROC and PFAM in medical applications [49]. In multiple-class problems, the ROC curve becomes less practical. Here, each decision combination algorithm includes a confidence threshold to regulate the confidence associated with the predictions from multiple classifiers. This threshold, $0 \leq \lambda \leq 1$, can be manipulated either to accept a classifier's prediction if the combined confidence level is higher than λ, or to reject the prediction otherwise. Hence, the classifier's reliability can be adjusted accordingly. The following performance measures, as recommended in [42], can be used to evaluate the effects of λ on classification performance:

Recognition Rate: Ratio of the number of correct classifications to the total number of samples (i.e., the accuracy index in previous experiments).

Substitution Rate: Ratio of the number of incorrect classifications to the total number of samples.

Rejection Rate: Ratio of the number of rejected classifications to the total number of samples.

$$\text{Reliability Rate: Reliability} = \frac{\text{Recognition}}{1 - \text{Rejection}}.$$

The experimental results of the MCSs with the modular approach were re-evaluated using different threshold values. As might be expected, trade-off between recognition and rejection took place as λ was increased from zero to unity. In other words, by manipulating the confidence threshold, the system designer could adapt a classifier system to perform with high or low reliability to suit the problem under investigation. Furthermore, manipulation of the confidence threshold provides an alternative means of comparing the performances of the three decision combination schemes. Fig. 7 depicts a plot of the percentages of the substitution rate against recognition rate, parameterized by the confidence threshold. The plot illustrates that by increasing λ from zero to one, more and more input samples are being rejected as it becomes more difficult for the predicted output to satisfy the confidence level. Both the recognition and substitution rates gradually decrease to 0%. Notice that even in the situation of low substitution rates, the BKS approach is able to maintain a relatively high recognition rate (and high reliability) compared with the Bayesian and voting methods. These observations are true for both experiments using $\overline{\rho}_a = 0.0$ and $\overline{\rho}_a = 0.9$.

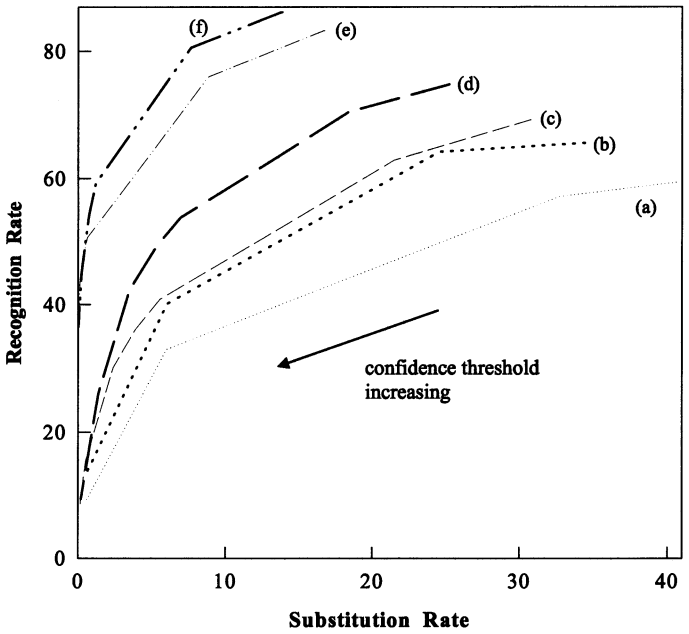

Fig. 7. Percentages of the substitution rate against recognition rate of the MCSs, parameterized by varying confidence threshold. (a) Voting ($\overline{\rho}_a = 0.0$). (b) Voting ($\overline{\rho}_a = 0.9$). (c) Bayesian ($\overline{\rho}_a = 0.0$). (d) Bayesian ($\overline{\rho}_a = 0.9$). (e) BKS ($\overline{\rho}_a = 0.0$). (f) BKS ($\overline{\rho}_a = 0.9$).

4) Dual-Mode Learning: In cases where there is a high correlation and dependency between attributes of the input samples, segmenting these attributes into different classifiers offers no benefits at all. As a result, online learning experiments with modularized inputs, such as those conducted for the quadruped mammal database, do not seem to be appropriate for the Landsat database.

With the ensemble approach, online combination of decisions across a committee of classifiers is unrealizable because data arrive in a fixed order. It seems that MCSs are not applicable for online learning with concatenated inputs. In practice, however, there is no reason why such MCSs cannot be employed online, after an initial period of training. As a result, a strategy is devised where an offline learning process is first conducted to equip each individual classifier with a different "knowledge" base before online learning is initiated. During offline learning, different classifiers will establish different category prototypes, thus predictions will be different when they are switched to online learning even though the classifiers are now receiving incoming samples in the same order. The decision combination schemes, once again, can be implemented to combine outcomes from a variety of differently trained classifiers using concatenated samples. We call this strategy *dual-mode learning*.

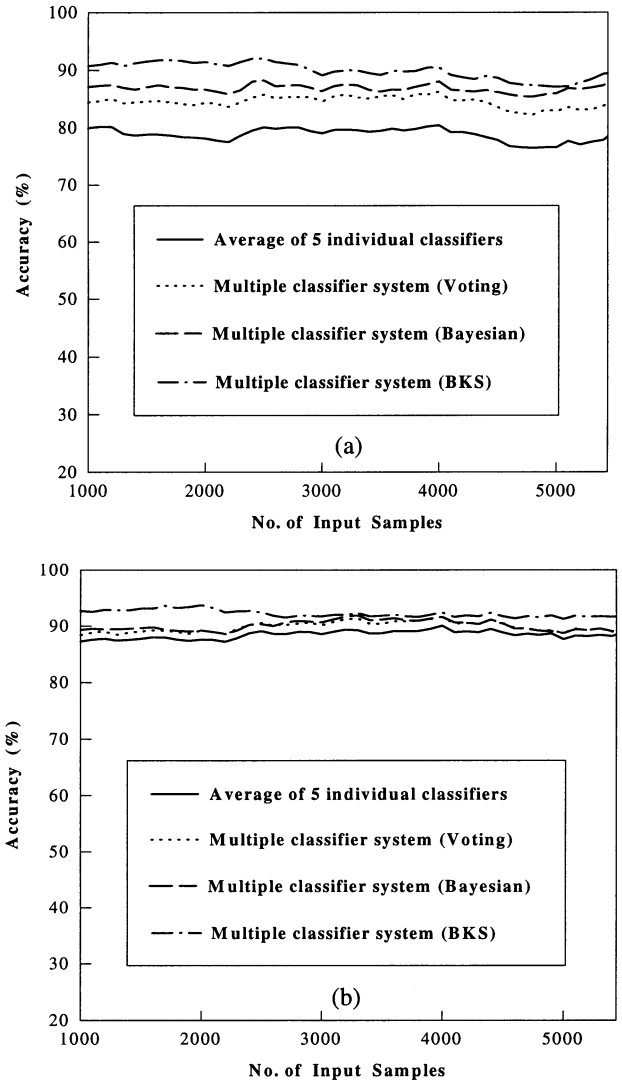

Fig. 8. Overall classification accuracy rates against increasing number of input samples from the individual classifiers as well as the voting, Bayesian and BKS multiple classifier systems. (a) $\overline{\rho}_a = 0.0$, (b) $\overline{\rho}_a = 0.9$.

Two sets of dual-mode learning experiments were conducted with $\overline{\rho}_a = 0.0$ and $\overline{\rho}_a = 0.9$. Five classifiers were initially trained using 1000 samples, each with a different ordering of data, and then tested using the remaining samples. The classification accuracy was calculated with a 1000-sample moving window as in the online learning experiments. Fig. 8 depicts the overall accuracy rates against increasing number of input samples from the individual classifiers, as well as the MCSs. A few observations can be made from the following results:

1) performance improves with a high vigilance value;
2) all three MCSs perform better than the individual classifiers;
3) BKS approach performs the best, followed by the Bayesian approach and then the voting method.

Nevertheless, with $\overline{\rho}_a = 0.9$, both the Bayesian and voting methods achieve virtually the same performance.

This dual-mode learning strategy also achieved results which are comparable with the offline results (Table V). However, one additional advantage here is that the system is able to learn online and the learning process is on-going. The problems faced in offline learning, such as a predefined network size and retraining, are avoided. Decisions from multiple classifiers can also be combined to produce a classification system with high accuracy.

V. Conclusion

A committee of neural network classifiers has been studied to tackle online pattern classification problems. The classifier used, PFAM, is a hybrid system of FAM and the PNN. It is an incremental adaptive system capable of online, supervised learning and probability estimation. In the manifestation of MCSs presented in this paper, an independent classifier module is dedicated to handle a set of attributes (either concatenated or modularized attributes), and outputs from these classifiers are then combined using some decision combination procedure to give an overall prediction. Three algorithms, namely the majority voting, Bayesian, and BKS methods, have been implemented to integrate the results of multiple PFAM classifiers. The efficacy of these algorithms has been empirically studied using two benchmark data sets obtained from a public-domain repository.

From the experiments, it is obvious that multiple classifiers are able to enhance the performance of individual classifiers. Among the three decision combination algorithms, the BKS approach demonstrated the best performance. Nevertheless, it should be noted that approximations, according to [42], have been made in the Bayesian approach, and this, in turn, might compromise the performance. A comprehensive Bayesian formalism would be able to take dependency between classifiers into account when combining predictions (good discussion on Bayesian methods for neural network can be found in [50]).

Apart from investigation into the performance of different decision combination algorithms, applicability of various input pattern representation methods and learning strategies have also been examined in the two benchmark problems. In contrast to the usual ensemble approach that codes an input pattern into a single vector and assigns to one classifier, the modular approach segregates the input pattern into groups of related attributes and feeds each group to an independent classifier. A decision is then made by combining the predictions from all group classifiers. However, the use of this modular approach is strongly dependent on the correlation between the input attributes. If a strong correlation exists, the modular approach will not only diminish the overall performance, but will also induce unnecessary complexity in the resulting system.

In view of the incremental learning property of PFAM, experiments have been conducted to study the applicability of MCSs in offline as well as online environments. One practical strategy is to employ a dual-mode learning approach where each PFAM classifier is first trained, offline, with a set of input samples with different orderings. This approach helps establish a knowledge base in the classifier before online, incremental learning is engaged. Very encouraging results have been achieved which are

comparable with, if not superior to, many reported results. By combining the predictions from a committee of PFAM classifiers, an autonomously learning system with improved accuracy can be realized to undertake online pattern-classification problems.

REFERENCES

[1] B. Grofman and G. Owen, Eds., *Information, Pooling and Group Decision Making*. Greenwich, CT: JAI, 1986.

[2] R. O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[3] A. J. C. Sharkey, "Multi-net systems," in *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, A. J. C. Sharkey, Ed. New York: Springer-Verlag, 1999, pp. 1–30.

[4] ——, "On combining artificial neural nets," *Connect. Sci.*, vol. 8, no. 3/4, pp. 299–314, 1996.

[5] P. Van de Laar and T. Heskes, "Input selection based on an ensemble," *Neurocomputing*, vol. 34, pp. 227–239, 2000.

[6] L. Breiman, "Arcing classifiers," Statist. Dept., Univ. California, Berkeley, Tech. Rep. 460, 1996.

[7] ——, "Bagging predictors," *Mach. Learn.*, vol. 26, no. 2, pp. 123–140, 1996.

[8] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[9] J. R. Quinlan, "Bagging, boosting, and C4.5," in *Proc. 13th Nat. Conf. Artificial Intelligence*, 1996, pp. 725–730.

[10] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Mach. Learn.*, vol. 36, no. 1/2, pp. 105–139, 1999.

[11] Y. Bennani and P. Gallinari, "Task decomposition through a modular connectionist architecture: A talker identification system," in *Proc. 3rd Int. Conf. Artificial Neural Networks*, I. Aleksander and J. Taylor, Eds. Amsterdam, The Netherlands: North-Holland, 1992, vol. 1, pp. 783–786.

[12] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79–97, 1991.

[13] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Comput.*, vol. 6, no. 2, pp. 181–214, 1994.

[14] T. Catfolis and K. Meert, "Hybridization and specialization of real-time recurrent learning-based neural networks," *Connec. Sci.*, vol. 9, no. 1, pp. 51–70, 1997.

[15] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vis., Graph. Image Process.*, vol. 37, pp. 54–115, 1987.

[16] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *The Psychology of Learning and Motivation*, G. H. Bower, Ed. New York: Academic, 1989, pp. 109–165.

[17] N. E. Sharkey and A. J. C. Sharkey, "An analysis of catastrophic interference," *Connect. Sci.*, vol. 7, pp. 301–329, 1995.

[18] A. P. Engelbrecht, "A new pruning heuristic based on variance analysis of sensitivity information," *IEEE Trans. Neural Networks*, vol. 12, pp. 1386–1399, Nov. 2001.

[19] C. T. Lin and C. S. George Lee, *Neural Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[20] D. Saad, Ed., *On-Line Learning on Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[21] R. M. French, "Semi-distributed representations and catastrophic forgetting in connectionist networks," *Connect. Sci.s*, vol. 4, pp. 365–377, 1997.

[22] A. H. L. West and D. Saad, "Online learning with adaptive back-propagation in two-layer networks," *Phys. Rev. E*, vol. 56, pp. 3426–3445, 1997.

[23] A. C. C. Coolen, D. Saad, and Y. S. Xiong, "Online learning from restricted training sets in multilayer neural networks," *Europhys. Lett.*, vol. 51, pp. 698–704, 2000.

[24] S. E. Fahlman and C. Lebiere, "The cascade-correlation architecture," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1990, vol. 2, pp. 524–532.

[25] J. Yang and V. Honavar, "Experiments with the Cascade-Correlation algorithm," *Microcomput. Appl.*, vol. 7, pp. 40–46, 1998.

[26] S. Sjogaard, "Generalization in cascade-correlation networks," in *Proc. Neural Networks Signal Processing*, vol. II, 1992, pp. 59–68.

[27] M. Hoehfeld and S. E. Fahlman, "Learning with limited numerical precision using the cascade-correlation algorithm," *IEEE Trans. Neural Networks*, vol. 3, pp. 602–611, July 1992.

[28] C. J. Platt, "A resource allocating network for function approximation," *Neural Comput.*, vol. 3, pp. 213–225, 1991.

[29] V. Kadirkamanathan and M. Niranjan, "A function estimation approach to sequential learning with neural networks," *Neural Comput.*, vol. 5, pp. 954–975, 1993.

[30] D. Lowe and A. McLachlan, "Modeling of nonstationary processes using radial basis function networks," in *Proc. IEE 4th Int. Conf. Artificial Neural Networks*, 1995, pp. 300–305.

[31] I. T. Nabney, A. McLachlan, and D. Lowe, "Practical methods of tracking of nonstationary time series applied to real world data," in *Proc. SPIE Conf. Applications Science Neural Networks*, vol. 2760, 1996, pp. 152–163.

[32] G. A. Carpenter and S. Grossberg, "ART 2: Stable self-organization of pattern recognition codes for analogue input patterns," *Appl. Opt.*, vol. 26, pp. 4919–4930, 1987.

[33] G. A. Carpenter, S. Grossber, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759–771, 1991.

[34] G. A. Carpenter, S. Grossberg, and J. H. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, pp. 565–588, 1991.

[35] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Networks*, vol. 3, pp. 698–712, Sept. 1992.

[36] R. Y. Asfour, G. A. Carpenter, S. Grossberg, and G. W. Lesher, "Fusion ARTMAP: A neural network architecture for multi-channel data fusion and classification," in *Proc. World Congress Neural Networks*, vol. II, 1993, pp. 210–215.

[37] R. Y. Asfour, "Fusion ARTMAP: Neural networks for multi-sensor fusion and classification," Ph.D. dissertation, Boston Univ., Boston, MA, 1995.

[38] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Acoustic, Speech, Signal Processing Mag.*, pp. 4–22, Apr. 1987.

[39] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, pp. 109–118, 1990.

[40] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, pp. 1065–1076, 1962.

[41] C. P. Lim and R. F. Harrison, "An incremental adaptive network for on-line supervised learning and probability estimation," *Neural Networks*, vol. 10, pp. 925–939, 1997.

[42] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 418–435, May/June 1992.

[43] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[44] Y. S. Huang and C. Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 90–94, Jan. 1995.

[45] J. H. Genneri, P. Langley, and D. Fisher, "Models of incremental concept formation," *Artif. Intell.*, vol. 40, pp. 11–61, 1989.

[46] C. L. Blake and C. J. Merz. (1998) UCI repository of machine learning databases. Univ. California, Dept. Information Computer Science, Irvine, CA. [Online]. Available: http://www.ics.uci.edu/~mlearn/ML-Repository.html.

[47] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, *Machine Learning, Neural and Statistical Classification*. New York: Oxford Univ. Press, 1994.

[48] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, pp. 29–36, 1982.

[49] C. P. Lim, R. F. Harrison, and R. L. Kennedy, "Application of autonomous neural network systems to medical pattern classification tasks," *Artif. Intell. Med.*, vol. 11, no. 3, pp. 215–239, 1997.

[50] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon, 1995.

Chee Peng Lim received the B.Eng. (Elect.) degree with first class honors from the University of Technology Malaysia in 1992, the M.Sc. (Eng) and Ph.D. degrees from the University of Sheffield, Sheffield, U.K. in 1993 and 1997.

He is currently Associate Professor, School of Electrical & Electronic Engineering, University of Science Malaysia. His research interests include soft computing, pattern classification, medical diagnosis, fault diagnosis, and condition monitoring. He has published more than 60 technical papers, and received three best paper awards at national and international conferences.

Dr. Lim is recipient of the Japan Society for the Promotion of Science (JSPS), Fulbright, and Commonwealth Fellowships, as well as The Outstanding Young Malaysians Award (Scientific & Technological Development), 2001.

Robert F. Harrison was born in 1956 and received the B.Sc. (Hons.) degree in acoustical engineering and Ph.D. degree from the University of Southampton,Southhampton, U.K. in 1979 and 1983, respectively.

He and has undertaken research for the UK Atomic Energy Authority, British Aerospace, and Rolls Royce. He has held research posts at the University of Southampton, U.K., Oxford University, U.K., and Princeton University, Princeton, NJ and is currently Reader in Systems and Control at the University of Sheffield. His research interests encompass machine learning, function approximation, and decision theory, and their applications in technological and medical domains.

Dr. Harrison is a Fellow of the IEE