Journal of Computers Vol. 28, No. 2, 2017, pp. 29-42
doi:10.3966/199115592017042802003

Hybrid-Neuro-Fuzzy System and Adaboost-Classifier for Classifying Breast Calcification

Joseph Hang Leung[1], Yu-Liang Kuo[2, 3], Ting-Wei Weng[5], and Chiun-Li Chin[4*]

[1] Department of Radiology, Chia-Yi Christian Hospital,
Chiayi 600, Taiwan, ROC
01289@cych.org.tw

[2] Department of Medical Imaging, Chung Shan Medical University Hospital,
Taichung 402, Taiwan, ROC

[3] Department of Medical Imaging and Radiological Sciences, Chung Shan Medical University,
Taichung 402, Taiwan, ROC
yuliangkuo@csmu.edu.tw

[4] Department of Medical Informatics, Chung Shan Medical University,
Taichung 402, Taiwan, ROC
ernestli@csmu.edu.tw

[5] Institute of Biomedical Informatics, National Yang-Ming University, Taiwan
Taipei 112, Taiwan, ROC
yaotomedoris@gmail.com

Received 09 November 2015; Revised 12 February 2016; Accepted 11 April 2016

Abstract. One of the major developments in machine learning in the past decade is the Ensemble method, which finds a highly accurate classifier by combining many moderately accurate component classifiers. In this paper, we propose a classifier of integrated neuro-fuzzy system with Adaboost algorithm. It is called Hybrid-neuro-fuzzy system and Adaboost-classifier classifier. Herein, Adaboost creates a collection of component classifiers by maintaining a set of weights over training samples and adaptively adjusting these weights after each iteration, and it is main architecture. The weak learner in Adaboost we used is SONFIN which is a neuro-fuzzy system. And, there is on-line learning ability in SONFIN. Finally, to demonstrate the capability of our proposed classifier, training and testing in different datasets including IRIS datasets, WISCONSIN breast datasets, and CSMU breast datasets are done. The contributions of this paper include implementation of Hybrid-neuro-fuzzy system and Adaboost-classifier for the classification and a classification accuracy of over 98% when training and testing on the IRIS dataset, 99% when training and testing on the WISCONSIN dataset, and 98.8% when training and testing on the CSMU Dataset. This means that our proposed classifier is good for classification and can be applied to variety field in the real world.

Keywords: breast datasets, machine learning, neuro-fuzzy system, weak learner

* Corresponding Author

1 Introduction

Whatever research field such as classification or big data analysis, a good classifier plays an important role. Generally classification algorithms are either strong classifiers or weak classifiers. Strong classification algorithms use the techniques such as ANN, SVM etc. Weak classification algorithms use the techniques such as Decision trees, Bayesian Networks, Random forests etc..

First, while many data sampling techniques are designed specifically to address the class imbalance problem, boosting is a technique that can improve the performance of any weak classifier. The most common boosting algorithm is Adaboost [1-2], which iteratively builds an ensemble of models. The Adaboost algorithm produces a sequence of weak classifiers that collectively form a strong classifiers. The idea is that firstly a weak classifier is used. The data points that are poorly classified have their frequency increased and the new dataset is used to train a second weak classifier. This process iterates for a specified number of iterations. The final strong classifier is a linear combination of weak classifiers. There are many applications using Adaboost classifier. Yang [3] proposed an Adaboost-based face detection for embedded systems to the incorporation of a genetic algorithm in the Adaboost training to optimize the detection performance given the number of Haar features. Simulation results show the system can achieve about 75-80% detection rate for group portraits.

During the last years an extensive development has been produced in hybrid classifiers, that is, in classifiers that are implemented taking the ideas of various standard classifiers. This research has the potential to apply accurate composite classifiers to real world problems by intelligently combining known learning algorithms. Two of the commonly used techniques for constructing Ensemble classifiers are Adaboost [1-8] and SVM [9-13].

Subsequently, neuro-fuzzy refers to combinations of artificial neural networks and fuzzy logic. A neuro-fuzzy system is a fuzzy system that uses a learning algorithm derived from or inspired by neural network theory to determine its parameters (fuzzy sets and fuzzy rules) by processing data samples. It can hybridize the mapping and learning capabilities of neural networks, with the ability of fuzzy classifiers to handle imprecise data. Lin and Lee [14] proposed a reinforcement neural-network-based fuzzy logic control system (RNN-FLCS) for solving various reinforcement learning problems. The proposed system is constructed by integrating two neural-network-based fuzzy logic controllers, each of which is a connectionist model with a feed forward multilayered network developed for the realization of a fuzzy logic controller. Juang and Lin [15] proposed a self-constructing neural fuzzy inference network (SONFIN) with on-line learning ability. There are no rules initially in the SONFIN.

Moreover, the use of intelligent hybrid systems is growing rapidly with successful applications in many areas. The generated neuro-fuzzy system can learn its data base for specific applications. Owusu [16] proposed a neural-Adaboost based facial expression recognition system, it hopes to improve the recognition accuracy and execution time of facial expression recognition system. They use Gabor feature extraction techniques to extract thousands of facial features which represent various facial deformation patterns, and Adaboost-based hypothesis was used to select a few hundreds of the numerous extracted features to speed up classification. An average recognition rate of 96.83% and 92.22% are registered in JAFFE and Yale databases, respectively. The execution time for a 100×100 pixel size is 14.5 ms.

Stavrakoudis et al. [17] proposed a boosted genetic fuzzy classifier for land cover classification of remote sensing imagery, it integrates fuzzy theory, evolutionary algorithm and Adaboost classifier to perform the detection of remote sensing image. The proposed system is able to handle multi-dimensional feature spaces more efficiently, effectively exploiting information from different feature sources.

Merler et al. [18] proposed a Parallelizing Adaboost by weights dynamics. It is a novel scheme for the parallelization of Adaboost, which builds upon earlier results concerning the dynamics of Adaboost weights and yields approximations to the standard Adaboost models that can be easily and efficiently distributed over a network of computing nodes. Properties of P-Adaboost as a stochastic minimizer of the Adaboost cost functional are discussed. Gao [19] proposed an edited Adaboost by weighted kNN (Adaboost) is designed where Adaboost and kNN naturally complement each other. Adaboost is then used to enhance the classification accuracy and avoid overfitting by editing the data sets using the weighted kNN algorithm for improving the quality of training data. Cheng and Jhan [20] proposed a cascade classifier combining Adaboost and support vector machine, and applied this to pedestrian detection. This cascade-Adaboost-SVM classifier can adjust numbers of cascade classifiers adaptively, it can construct cascade classifiers effectively based on training set. Kumar [21] proposed a adaptive and hybrid neuro-fuzzy systems as subsystems of the ensemble. Here, an Adaptive Neuro-Fuzzy Inference System (ANFIS) has been chosen as a base classifier for our research as Mamdani type ANFIS is not suitable for real time due to its high computational complexity and non-adaptiveness to extract exact knowledge from the dataset. Single classifier makes error on different training samples. Therefore, by creating an ensemble of classifiers and combining their outputs, the total error can be reduced and the detection accuracy can be increased.

Mammography is a screening tool for breast abnormalities detection. Mammography allows identification of tumour before being palpable. Breast calcifications are deposits of calcium that can be seen on a mammogram of the breast. There are two types: Macro calcifications and Micro calcifications. However, certain features and presentations of micro calcifications are more likely to be associated with malignant breast cancer. Vijayalakshmi et al. [22] proposed an automatic detection of tumor of micro calcification clusters on mammograms. The feature extraction of this scheme adopted Local Binary Pattern method and classified using ANN. The proposed scheme provides high accuracy in classification of micro calcifications. Alain Tiedeu et al. [23] proposed that clustered micro calcifications are detected on mammograms based on texture analysis. Arun kumar and Sheshadri [24] proposed a methodology for the classification of micro calcification in mammograms. An improved classifier that introduces balanced learning for the accurate classification for the classification of microcalcification is proposed as one of the main steps in the methodology.

When considering classification tasks, neuro-fuzzy system exhibit a number of advantages, compared to conventional. First of all, neuro-fuzzy system (the more popular term is used henceforth) incorporates the human-like reasoning style of fuzzy systems through the use of fuzzy sets and a linguistic model consisting of a set of IF-THEN fuzzy rules. Therefore, a neuro-fuzzy system exhibits higher operational applicability than classical classifiers, which always infer a class decision for a given input. The contributions of this paper include implementation of Hybrid-neuro-fuzzy system and Adaboost-classifier classifier for the classification and a classification accuracy of over (1) 98% when training and testing on the IRIS dataset, (2) 99% when training and testing on the WISCONSIN dataset, and (3) 98.8% when training and testing on the CSMU Dataset. This means that our proposed classifier is good for classification and can be applied to variety field in the real world.

The rest of the paper is organized as follows. Section 2 is materials and methods, we review the different concepts needed to understand the new hybrid classifier. And, we introduces the new our proposed Hybrid-neuro-fuzzy system and Adaboost-classifier classifier approach. The results obtained are presented in section 3. Last section is dedicated to give conclusions and to point out the future work.

2 Materials And Methods

In this section, this paper will introduce our proposed algorithm. First, it is starting to introduce two important algorithms, which are used in our proposed architecture. We will explain method in detailed.

2.1 Backgrounds

Adaboost algorithm. Adaboost classifier [20] is an ensemble classifier composed of many weak classifiers (such as linear classifier); each classifier performs classification according to only one dimensionality of input vector, so it is also called weak classifier; the result of ensemble classifier can be expressed as:

$$H(x) = sign\{\sum_{t=1}^{T} \beta_t h_t(x)\} \quad \textbf{(1)}$$

where x represents input vector; $h_t(x)$, t = 1, …, T means that the number of classifiers is T; β_t, t = 1, …, T refers to weight of each weak classifier; Fig. 1 shows Adaboost algorithm; suppose a training sample set $\{x_i, y_i\}$, i = 1, …, m, where $x_i \in R^n$, $y_i \in \{1, -1\}$ is given and first initialize the initial distribution values of all training samples; to make the classification results maintain a higher detection rate, let the distribution value of a positive sample be equivalent to that of all negative values; if the training sample set includes p positive samples and q negative samples, that is, m = p + q, we set the distribution value of the positive sample and that of the negative sample as 1/(p+1) and 1/q(p+1) respectively and then perform the selection loop to T weak classifiers; each time the algorithm performs the loop, it first searches for weak classifiers $h_j(x)$, j = 1, …, n, with minimum error according to each dimension and then finds out the weak classifier $h_t(x)$ with minimum error from these weak classifiers; after the selection of weak classifiers at *t* the loop, readjust the probability value of the training sample so that samples of classification error in this loop can have the priority when the next loop is performed; then calculate the corresponding weight β_t of the weak classifier $h_t(x)$; finally, the strong classifier $H(x)$ can be obtained by calculating sum of products of weight of T weak classifiers and corresponding β values; determine the classification result of the input vector with result of ensemble classifier. Table 1 shows the Adaboost algorithm.

Table 1. Adaboost algorithm

Input: example $(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$, where $x_1 \in R^n, y_1 \in \{1, -1\}$, for negative and positive example respectively.
output: $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$
• Initialization distribution $\omega_i(1) = 1/(p+1)$ for $y_i = -1$ and $\omega_i(-1) = 1/q(p+1)$ for $y_i = 1$, respectively where p and q are the number of negatives and positives respectively.
• FOR $t = 1,...,T$
➢ Find classifier $h_t : x \to \{1, -1\}, h_t = \arg\min \varepsilon_j$, where $\varepsilon_j = \sum_{i=1}^{m} \omega_i(t)\left[y_i = h_j(x_i)\right]$
➢ Weight classifier: $\alpha_t = 0.5 \times \log\left(\dfrac{1-\varepsilon_t}{\varepsilon_t}\right)$
➢ Update distribution: is for normalization.
$\omega_i(t+1) = \omega_i(t)\exp-[-\alpha_t \times y_i \times h_t(x_i)]/z_t = \omega_i(t)\exp[-\alpha_t \times y_i \times h'_t(x_i)]$

SONFIN algorithm. SONFIN [14] is a neuro-fuzzy system, and it consists of nodes, each of which has some finite “fan-in” of connections represented by weight values from other nodes and “fan-out” of connections to other nodes. It consists of six layers. In Layer 1, each node in this layer, which corresponds to one input variable, only transmits input values to the next layer directly. In Layer 2, each node corresponds to one linguistic label (small, large, etc.) of one of the input variables in Layer 1. In other words, the membership value which specifies the degree to which an input value belongs a fuzzy set is calculated in Layer 2. In Layer 3, a node in this layer represents one fuzzy logic rule and performs precondition matching of a rule. In Layer 4, the number of nodes in this layer is equal to that in Layer 3 and the firing strength calculated in Layer 3 is normalized. Layer 5 is called the consequent layer. In layer 6, each node in this layer corresponds to one output variable. The node integrates all the actions recommended by Layer 5 and acts as a defuzzifier. Fig. 1 shows the structure of SONFIN.

2.2 Proposed Algorithm: Hybrid-Neuro-Fuzzy System and Adaboost-Classifier

In this paper, we present a new hybrid classifier based on two families of well- known methods; the first one is Adaboost algorithm, and the second one is a neuro-fuzzy system, which is called SONFIN. Fig. 2 shows the flowchart of our proposed classifier. And, it is entitled Hybrid-neuro-fuzzy system and Adaboost-classifier classifier.

First, To model the target system, a training dataset of input/output pair (x1, x2, x3, x4, x5, x6, …, y) is required, and the sampling weights are initialized. The instance weights in the dataset are assigned uniformly. Next, the feature processing is performed. Its goal is to reduce feature dimensionality and time complexity. It is divided into two parts. First, for feature dimensionality reduction, we rely on principal component analysis (PCA). It is a pattern recognition tool that uses an orthogonal transformation to project a set of possibly correlated variables into a group of linearly uncorrelated ones that are called principal components. These components attempt to maintain most of the variability of the data. Second, it

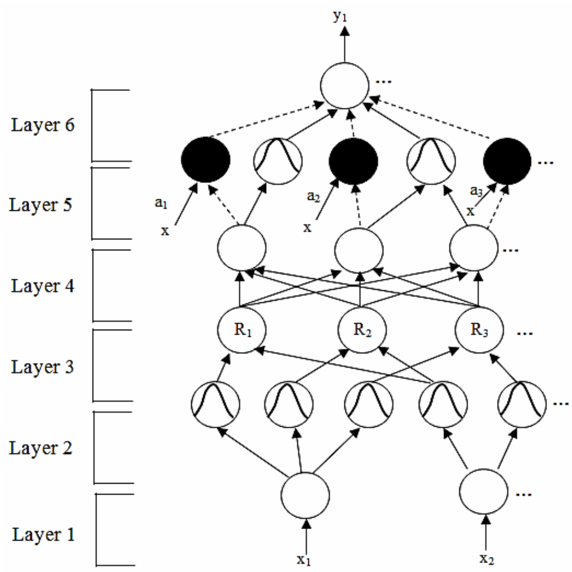

Fig 1. The structure of SONFIN

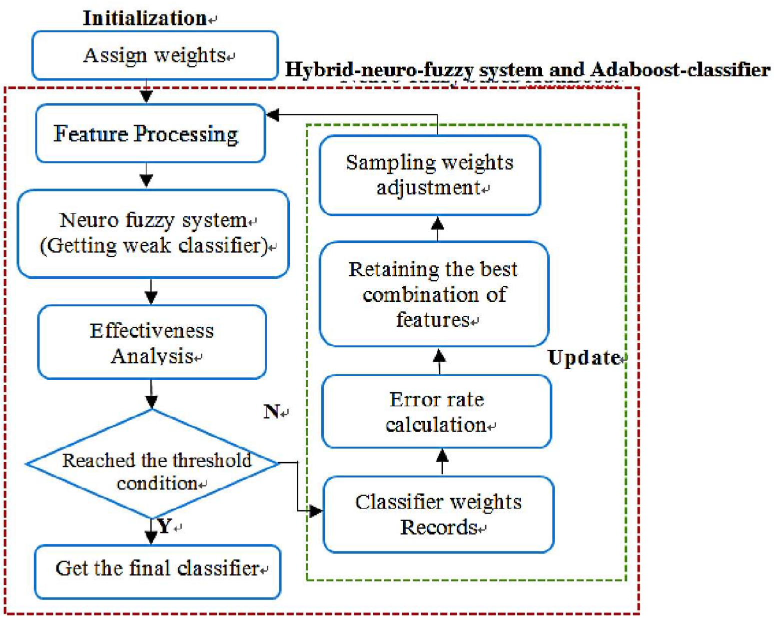

Fig. 2. The architecture of Hybrid-neuro-fuzzy system and Adaboost-classifier classifier

performs feature combination. Its goal is to produce the combination between different features for pro ducing training datasets. Subsequently, we use a neuro-fuzzy system to be a weak learner in Adaboost classifier. Neuro-fuzzy has been combined into ensembles to improve accuracy and robustness. Ensemble of classifiers is created when trained. The neuro-fuzzy system we used adopts SONFIN architecture. SONFIN is a method for tuning an existing rule base with a learning algorithm based on a collection of training data. Hence, the input/output are mapped adaptively by SONFIN through membership functions, rules, precise, and consequent parameters. SONFIN is trained to obtain hypothesis h_t. The structure of the neuro-fuzzy system is shown in Fig. 1. The SONFIN can be used for normal operation at any time during the learning process without repeated training on the input–output patterns when on-line operation is required. There are no rules in the SONFIN initially. The training process continues till the desired least mean square is achieved. Second, the error in Equation (2) is calculated by statistics of different label between the output of SONFIN and desired output.

$$\varepsilon_t = \sum_{i=1}^{n} \left[d_t(i) \times (h_t(x_i) \neq y_i) \right] \tag{2}$$

Obviously, the error must never become zero. Therefore, before calculating Equation (2), we constrain the error to be strictly positive. Herein, the $d_t(i)$ will be updated in step e in the algorithm of Fig. 3. Then, the representative weak learner is selected. The weak learner has minima error value calculated from above step. When getting the representative weak learner, the weighting α_t will be calculated using Equation (3). The formula is shown as follow. The final strong classifier will use the weighting value.

$$\alpha_t = 0.5 \times \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \tag{3}$$

The α_t can be interpreted as a weight showing the importance of the rule: unreliable, low performing rules have a reduced significance in the final classification decision, as compared to rules exhibiting a small error. The Equation (6) is the update formula of d_t. Because the instance weights in the dataset are assigned uniformly, the distribution of the samples as input to the supervised base classifier is given in Equation (2), where ‘n’ is the total number of instances in the data subset and d_t in the Eq. (4) is the distribution of the classifier ‘t’. It is divided into two parts. First, if the output of SONFIN is not equal to desired output (i.e. $h_t(x_i) \neq y_i$), the next d_t will be updated using Equation (5) (i.e. $d_{t+1}(i) = d_t(i) \times \alpha_t$). On the contrary, the d_t will not be updated. Subsequently, it will be normalized in the Equation (6).

$$d_t(i) = \frac{1}{n} \tag{4}$$

$$\begin{aligned} &if \quad h_t(x_i) \neq y_i, \quad i = 1 \ldots n \\ &\qquad d_{t+1}(i) = d_t(i) \times \alpha_t \\ &else \quad d_{t+1}(i) = d_t(i) \end{aligned} \tag{5}$$

$$d_{t+1}(i) = \frac{d_{t+1}(i)}{\sum_{i=1}^{n} d_{t+1}(i)} \tag{6}$$

The updated distribution samples are given as input for the next iteration. Similarly, the process is repeated for T iterations. The Adaboost algorithm repeatedly invokes a neuro-fuzzy system (SONFIN) on various distributions of the training data and to aggregate the individual classifiers into a single overall classifier. The distribution of training instances is changed based on the error the current classifier exhibits on the training set after each iteration. Adaboost combines the individual classifiers (fuzzy rules) considering their errors at the time of their generation. When an unseen instance is input in the classifier, the classification decision is made following a weighted maximum scheme:

$$H = \arg\max\left(\sum_{t=1}^{T} \alpha_t \times h_t(x_i)\right) \tag{7}$$

The proposed Hybrid-neuro-fuzzy system and Adaboost-classifier classifier algorithm is as shown in Fig. 3. The training distribution is updated by including the cost of misclassification on successive iterations. The updated distribution samples are given as input for the next iteration. Similarly, the process is repeated for T iterations. The instances can be classified into multi-class, namely, not only two classes. If the instances are classified correctly, the weight of these instances is reduced aggressively such that these instances will not be considered during next iteration of training. The training process continues till the desired least mean square is achieved.

Input:

For A Dataset

- Training Data " TD_j " of size "M" with correct labels
- Adaptive Neural-Fuzzy based AdaBoost as Supervised algorithm base classifier
- Number of iterations or classifiers(T)

Feature Processing

- F_{es}=M //Number of feature
- F_{sc}=1…N…M //Number of feature combination
- $C_s = C_1^M + \ldots C_N^M + \ldots C_M^M$ //Total

Initialize:

- μ=0.5 //False Alarm Threshold
- L=[1…S] //Number of Classes
- $y_i \in Y = \{1, \ldots S\}$ // y_i is known classification results

Training:

for t=1……T

f_d=1…… C_s

a. Train Data f_d by SONFIN obtain result

b. Compute error of h_t:

$$\varepsilon_t = \sum_{i=1}^{n} [d_t(i) \times (h_t(x_i) \neq y_i)]$$

c. The best feature selection's model:

$$h_t(x_i)$$

d. Assign weight to the Classifier:

$$\alpha_t = 0.5 \times \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

e. Update Distribution of Instances:

$for\ i = 1 : n$

$if \quad h_t(x_i) \neq y_i$

$d_{t+1}(i) = d_t(i) \times \alpha_t$

$else \quad d_{t+1}(i) = d_t(i)$

end

$$d_{t+1}(i) = \frac{d_{t+1}(i)}{\sum_{i=1}^{n} d_{t+1}(i)}$$

End

f. Obtain Composite Hypothesis:

$$H = \arg\max\left(\sum_{t=1}^{T} \alpha_t \times h_t(x_i)\right)$$

End

Fig. 3. Hybrid-neuro-fuzzy system and Adaboost-classifier classifier

3 Results

All experiments were performed on a Windows 8 machine with a 2.8 GHz Intel Core i5-4200H CPU and 8 GB of RAM. The main programming language is MATLAB. The training time of each method depends on the datasets, the number of training samples, and the parameter setting.

This section will introduce the sample sets used in the experiment and the related experimental results; these sample sets include self-built samples and samples of IRIS and WISCONSIN breast database, which are publicly available datasets; first we will introduce these samples and then present the experiment numerical results.

3.1 Datasets

We performed experiments on two standard datasets. One is the IRIS dataset and breast cancer Wisconsin dataset downloaded from the UCI machine learning repository [22].

The dimensions of these data sets range from 2 to 20, the numbers of training samples range from 140 to 1000, and the numbers of test samples range from 75 to 1000. Each data set is partitioned into training and test subsets, usually in the ratio of 80-20%. On each partition, the compared algorithms are trained and tested, respectively. The final performance of each algorithm on a data set is the average of the results.

First, the Iris datasets has 150 instances. There are four attributes, i.e. Sepal Length, Sepal Width, Petal Length and Petal Width, and there are three classes, i.e. Setosa, Versicolor, and Virginica, where each class has 50 training instances. One class is linearly separable from the other two; the latter are not linearly separable from each other. Some samples of the data set are shown in Table 2.

Table 2. Iris datasets

Setosa	Versicolor	Virginica
(5.1, 3.5, 1.4, 0.2)	(7.0, 3.2, 4.7, 1.4)	(6.3, 3.3, 6.0, 2.5)
(4.9, 3.0, 1.4, 0.2)	(6.4, 3.2, 4.5, 1.5)	(5.8, 2.7, 5.1, 1.9)
(4.7, 3.2, 1.3, 0.2)	(6.9, 3.1, 4.9, 1.5)	(7.1, 3.0, 5.9, 2.1)

The second dataset used the dataset provided by researchers at the University of Wisconsin. The dataset was obtained from the University of California Irvine (UCI) Machine Learning Repository [27] consist of 699 data with 65.5% classified as benign and 34.5% as malignant The Wisconsin breast cancer diagnosis (WBCD) database is the effort made at the University of Wisconsin Hospital for accurately diagnosing breast masses based solely on an FNA (Fine Needle Aspirates) test. Nine visually assessed characteristics of an FNA sample considered relevant for diagnosis were identified, and assigned an integer value between 1 and 10. The Wisconsin Breast Cancer Diagnosis (WBCD) problem involves classifying a presented case as to whether it is benign or malignant. There are several studies based on this database. Among them, researchers having interpretability of the diagnostic as a prior objective have applied the method of extracting Boolean rules from neural network**.**

The third dataset is obtained by Chung Shan Medical University Hospital. There are 20 patients' mammography images. And, the image resolution is 256×256. In these patients, the 10 patient's breast calcifications are not normal. The rest are normal. We select 5 images from the set of breast image in each patient. The 20 areas are extracted in each image and its size is 25×25.Herein, different window sizes are tried. The statistical feature vectors obtained from three different window sizes (11×11, 25×25, and 33×33) are classified by several classification methods. As shown in Fig. 4, the window size 25×25 yields the highest accuracy. Based on this experience, we decide to use the 25×25window size in the rest of the study.

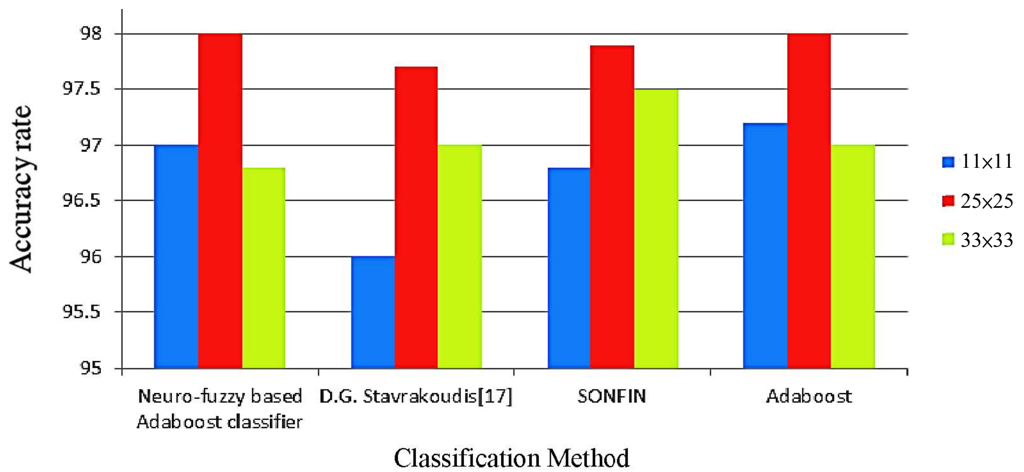

Fig. 4. Comparison of accuracy of different classification methods for 3 different window sizes as 11, 25, and 33

We adopt GLCM and laws mask method to extract three features including mean, standard deviation, and entropy. GLCM is a statistical method of examining texture that considers the spatial relationship of pixels and can extract second order statistical texture features. These features are generated by calculating the features for each one of the co-occurrence matrices obtained by using the directions 0°, 45°, 90°, and 135°. Laws developed a set of two-dimensional masks derived from some simple one-dimensional filters as level detection, edge detection, or spot detection and so on. These 1D filter are convolved with each other into 2D spatial filter. Herein, feature values were extracted based on the principle of texture energy, which encompasses five one-dimensional masks whose size is 1 by 5, namely Edge, Level, Spot, Ripple and Wave. Then 25 two-dimensional masks whose size is 5 by 5 can be generated by combining these five one-dimensional masks, and then 14 rotation invariant images may be generated from the 25 masks. These 14 masks are obtained by combining the 25 TEM descriptors. For example $TR_{E_5L_5}$ was obtained with Equation (8). The statistical properties such as mean, standard deviation, and entropy are computed as follows (Eq. (9) to Eq. (11)).

$$TR_{E_5L_5} = \frac{TEM_{E_5L_5} + TEM_{L_5E_5}}{2} \tag{8}$$

$$Mean = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[TR_{i,j}]}{M \times N} \tag{9}$$

$$SD = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(TR_{i,j} - Mean\right)^2}{M \times N}} \tag{10}$$

$$Entropy = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(TR_{i,j})^2}{M \times N} \tag{11}$$

where $TR_{i,j}$ is the rotation invariant image, and M and N are length and width of the block image. Hence, the total data number in each patient are 1200. The total data number of this datasets are 24000. Finally, the block schematic of CSMU datasets preprocessing is shown in Fig. 5. The preprocessing stage consists of feature extraction and normalization. The extracted features and its values are input to the normalization module. Normalization is a process of ensuring that each attribute value in a database is suitable for further querying and free from certain undesirable characteristics. Hence, each variable is normalized in the range [0, 1] to eliminate the effect of scale difference. The normalized method uses min-max normalization such that the training dataset contain values between [0, 1]. These values are used as inputs for machine learning algorithms.

The Adaboost and SONFIN are used to test the generalization ability and compare the performance of our proposed method with other method.

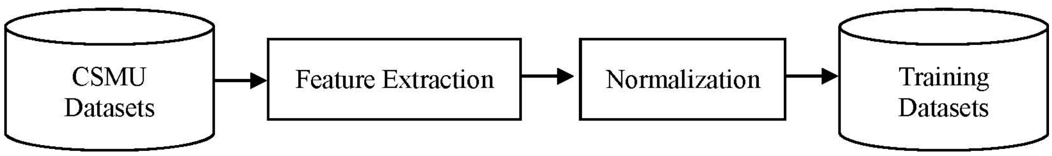

Fig. 5. CSMU datasets preprocessing steps

3.2 Numerical Result

To assess the accuracy of our methods, we report some standard error metrics. In this paper, the Specificity (SPC) rate, True positive rate (TPR), False positive rate (FPR) rate, and Accuracy (ACC) rate will be calculated using Eq. (12) to Eq. (15), respectively. Their formula is shown as the follow.

$$SPC=TN/N=TN/(FP+TN)=1-FPR \tag{12}$$

$$TPR=TP/P=TP/(TP+FN) \tag{13}$$

$$FPR=FP/N=FP/(FP+TN) \tag{14}$$

$$ACC=(TP+TN)/(P+N) \tag{15}$$

Where P and N denotes the number of positive (p) and negative (n) instances. If the outcome of a prediction is p and the actual value is also p, then it is called a true positive (TP); conversely if the actual value is n then it is said to be a false positive (FP). If it is classified as negative, it is counted as a false negative. If the instance is negative and it is classified as negative, it is counted as a true negative. Table 4 shows our comparison result among Hybrid-neuro-fuzzy system and Adaboost-classifier, D.G. Stavrakoudis [17], SONFIN, and Adaboost which is Adaboost.M1 in Matlab. Herein, the testing set is used for the comparison of different performance. The training set is used for training all algorithms and the successful rate is over 95% in the training stage of each algorithm. Hence, we can conclude that proposed Hybrid-neuro-fuzzy system and Adaboost-classifier classifier performs better than Adaboost and SONFIN in general on three data sets. In Table 3, we show the parameters of SONFIN used in the our proposed classifier. Experiments consisted of 10 runs of the system for every dataset and the results show average with standard deviation. The following datasets have been used for training/testing: (1) *iris*–4 attributes, 2 classes, 120 instances for training and 30 for testing; (2) breast cancer WISCONSIN (original)–9 attributes, 2 classes, 400 instances for training and 279 for testing. (3) CSMU -3 features, 2 classes, 100 instances for training and 100 for testing.

Table 3. Parameters used in the experiments

parameter	value
The learning rate of consequent part in a fuzzy rule	0.1
The learning rate of mean of Gaussian membership function	0.05
The learning rate of variance of Gaussian membership function	0.05
The initial value of variance of Gaussian membership function	0.4
The output number	1.0

classifier	PI / Datasets	Datasets: IRIS	WISCONSIN	CSMU
Hybrid-neuro-fuzzy system and Adaboost-classifier classifier	ACC	0.980	0.990	0.988
	SPC	0.976	0.985	0.985
	FPR	0.024	0.015	0.015
	TPR	0.970	0.985	0.980
D.G. Stavrakoudis [17]	ACC	0.978	0.983	0.985
	SPC	0.975	0.977	0.981
	FPR	0.025	0.023	0.019
	TPR	0.970	0.982	0.980
SONFIN	ACC	0.978	0.990	0.987
	SPC	0.980	0.976	0.980
	FPR	0.020	0.024	0.020
	TPR	0.968	0.986	0.978
Adaboost	ACC	0.960	0.965	0.970
	SPC	0.958	0.960	0.965
	FPR	0.042	0.040	0.035
	TPR	0.950	0.963	0.960

Furthermore, the Receiver Operating Characteristic (ROC) analysis has been done. We use the Area Under the ROC curve (AUC) to compare these three algorithms on three data sets. The AUC is defined as the area under an ROC curve. It is known that larger AUC values indicate generally better classifier performance. The AUC values listed in Table 5 illustrate that Hybrid-neuro-fuzzy system and Adaboost-classifier achieves the highest average AUC values in all the three data sets. Statistically, the higher AUC values obtained by Hybrid-neuro-fuzzy system and Adaboost-classifier means that it would favor classifying a positive instance with a higher probability than other algorithms and so it can better handle the variety of problems.

Table 5. AUC results on the three data sets

Classifier / Dataset	Hybrid-neuro-fuzzy system and Adaboost-classifier	D.G. Stavrakoudis [17]	SONFIN	Adaboost
IRIS	0.981	0.975	0.971	0.961
WISCONSIN	0.988	0.986	0.987	0.964
CSMU	0.983	0.981	0.979	0.969

The metric cost function was used to facilitate performance comparison of the existing ensemble methods with our proposed Hybrid-neuro-fuzzy system and Adaboost-classifier. Cost function is based on the number of samples that are misclassified. It was calculated using Equation (16). The cost function for Hybrid-neuro-fuzzy system and Adaboost-classifier, D.G. Stavrakoudis [17], SONFIN, and AdaBoost algorithms are calculated using Equation (16). The ensemble with least cost function emerges out as the best detection system.

$$\cos t = (1 - ACC) + 4 \times FPR \tag{16}$$

In the Equation (16), "4" is obtained by experiment. This value means cost difference between false alarm and miss. Subsequently, from Table 3, it can be seen that the accuracy rate is 98% with 2.4% false positives. Comparing the False positive rate of Hybrid-neuro-fuzzy system and Adaboost-classifier with D.G. Stavrakoudis [17], it can be seen that Hybrid-neuro-fuzzy system and Adaboost-classifier yields an improvement of 4%($(\frac{2.5-2.4}{2.5}) \times 100\%$)over D.G. Stavrakoudis [17]. Next, it can be seen that the accuracy rate is 99% with 1.95% false positives. Comparing the False positive rate of Hybrid-neuro-fuzzy system and Adaboost-classifier with SONFIN, it can be seen that neuro-fuzzy based Adaboost yields an improvement of 34.7% over D.G. Stavrakoudis [17]. And it also yields an improvement of 37.5% over D.G. SONFIN. Finally, it can be seen that the accuracy rate is 98.8% with 1.5% false positives. Comparing the False positive rate of Hybrid-neuro-fuzzy system and Adaboost-classifier with SONFIN, it can be seen that neuro-fuzzy based Adaboost yields an improvement of 21% over D.G. Stavrakoudis [17]. Moreover, it can be inferred from Fig. 6 that cost value is less for Hybrid-neuro-fuzzy system and Adaboost-classifier than the other methods besides the cost value obtained by using SONFIN to classify IRIS dataset is greater than cost values obtained by using Hybrid-neuro-fuzzy system and Adaboost-classifier.

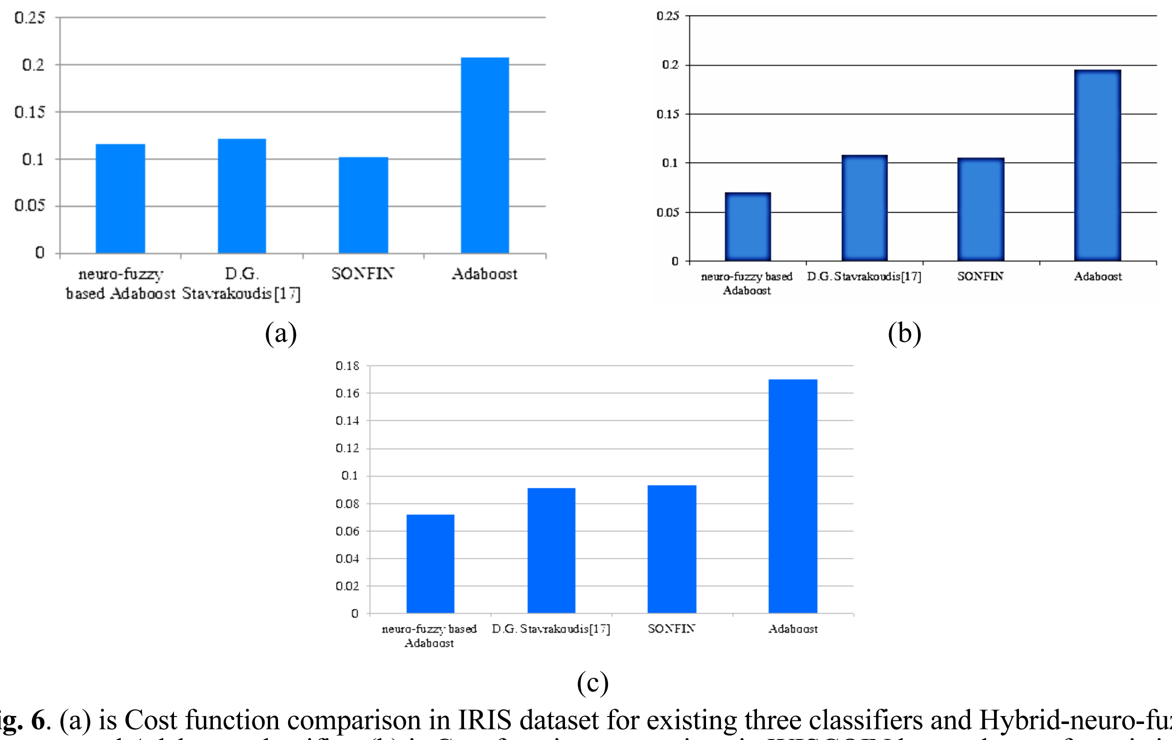

Fig. 6. (a) is Cost function comparison in IRIS dataset for existing three classifiers and Hybrid-neuro-fuzzy system and Adaboost-classifier, (b) is Cost function comparison in WISCOIN breast dataset for existing three classifiers and Hybrid-neuro-fuzzy system and Adaboost-classifier, and (c) is Cost function comparison in CSMU dataset for existing three classifiers and Hybrid-neuro-fuzzy system and Adaboost-classifier

Finally, Table 6 were measured on a 2.8 GHz Intel Core i5-4200H CPU and 8 GB of RAM. The main programming language is MATLAB. Here, we use CSMU dataset to test the performing time. The run time value is calculated on training phase. Form this table, the run time of our proposed method is less than D.G. Stavrakoudis [17] and Adaboost. But, it is greater than SONFIN. The reason for this result is that the other algorithms are hybrid method besides SONFIN. Hence, they need more run time to achieve task. The architecture of D.G. Stavrakoudis [17] proposed adopts ANFIS neuro-fuzzy system as a weak learner. The learning time is higher than SONFIN.

Table 6. Run time for the four algorithms for 20 patients.

Algorithm	Time (min)
Hybrid-neuro-fuzzy system and Adaboost-classifier	4
D.G. Stavrakoudis [17]	5
SONFIN	3
Adaboost	4

4 Conclusion

In this paper, we proposed a hybrid neuro-fuzzy system and Adaboost classifier, called Hybrid-neuro-fuzzy system and Adaboost-classifier classifier. The SONFIN is a general connectionist model of a fuzzy logic system, which can find its optimal structure and parameters automatically. Training and testing in different dataset including IRIS datasets, WISCONSIN breast datasets, and CSMU datasets have demonstrated the on-line learning capability of the Hybrid-neuro-fuzzy system and Adaboost-classifier classifier. The experiment shows that the proposed framework achieves higher classification accuracy. Finally, we can say that the proposed method is an important tool which can be integrated in a CAD (computer aided diagnosis) for assisting in diagnostic decision making, with providing an understandable explanation of the underlying reasoning. The contributions of this paper include implementation of Hybrid-neuro-fuzzy system and Adaboost-classifier classifier for the classification and a classification accuracy of over (1) 98% when training and testing on the IRIS dataset, (2) 99% when training and testing on the WISCONSIN dataset, and (3) 98.8% when training and testing on the CSMU Dataset.

Acknowledgement

This work was supported by research grants to Chung Shan Medical University Hospital and CHIA-YI CHRISTIAN HOSPITAL (CSMU-CYC-101-02) in Taiwan, ROC.

References

[1] R.E. Schapire, Y.Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning 37(1999) 297-336.

[2] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55(1)(1997) 119-139.

[3] M. Yang, J. Crenshaw, B. Augustine, R. Mareachen, Y. Wu, Adaboost-based face detection for embedded systems, Computer Vision and Image Understanding 114(11)(2010) 1116-1125.

[4] Y. Cao, Q.-G. Miao, J.-C. Liu, L. Gao, Advance and prospects of Adaboost algorithm, Acta Automatica Sinica 39(6)(2013) 45-758.

[5] Q. Nie, L. Jin, S. Fei, Probability estimation for multi-class classification using Adaboost, Pattern Recognition 47(2)(2014) 3931-3940.

[6] J. Lu, H. Hu, Y. Bai, Generalized radial basis function neural network based on an improved dynamic particle swarm opti-

mization and Adaboost algorithm, Neurocomputing 152(C)(2015) 305-315.

[7] Y. Gao, F. Gao, Edited Adaboost by weighted kNN, Neurocomputing 73(16-18)(2010) 3079-3088.

[8] C. Gao, N. Sanga, Q. Tang, On selection and combination of weak learners in Adaboost, Pattern Recognition Letters 31(9)(2010) 991-1001.

[9] R. Wang, Adaboost for feature selection, classification and its relation with SVM, a review, Physics Procedia 25(2012) 800-807.

[10] Z. Wang, Z. Zhao, S. Weng, C. Zhang, Solving one-class problem with outlier examples by SVM, Neurocomputing 149 (2015) 100-105.

[11] H. Yin, X. Jiao, Y. Chai, B. Fang, Scene classification based on single-layer SAE and SVM, Expert Systems with Applications 42(7)(2015) 3368-3380.

[12] H. Li, F.-L. Chung, S. Wang, A SVM based classification method for homogeneous data, Applied Soft Computing 36(2015) 228-235.

[13] G. Li, J. You, X. Liu, Support vector machine (SVM) based prestack AVO inversion and its applications, Journal of Applied Geophysics 120(2015) 60-68.

[14] C.T. Lin, C.S.G. Lee, Reinforcement structure/Parameter learning for neural-network-based fuzzy logic control systems, IEEE Trans. on Fuzzy Systems 2(1)(1995) 46-63.

[15] C.F. Juang, C.T. Lin, An on-line self-organizing neural fuzzy inference network and its applications, in: Proc. The Seventh International Fuzzy Systems Association World Congress, 1997.

[16] E. Owusu, Y.Z. Zhan, Q.R. Mao, A neural-Adaboost based facial expression recognition system, Expert Systems with Applications 41(7)(2014) 3383-3390.

[17] D.G. Stavrakoudis, J.B. Theocharis, G.C. Zalidis, A boosted genetic fuzzy classifier for land cover classification of remote sensing imagery, ISPRS Journal of Photogrammetry and Remote Sensing 66(4)(2011) 529-544.

[18] S. Merler, B. Caprile, C. Furlanello, Parallelizing Adaboost by weights dynamics, Computational Statistics & Data Analysis 51(5)(2007) 2487-2498.

[19] Y. Gao, F. Gao, Edited Adaboost by weighted kNN, Neurocomputing 73(16-18)(2010) 3079-3088.

[20] W.-C. Cheng, D.-M. Jhan, A self-constructing cascade classifier with Adaboost and SVM for pedestriandetection, Engineering Applications of Artificial Intelligence 26(3)(2013) 1016-1028.

[21] P.A.R. Kumar, S. Selvakumar, Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems, Computer Communications 36(3)(2013) 303-319.

[22] B. Vijayalakshmi, R. Bhanumathi, G.R. Suresh, Study of mammogram micro calcification to aid tumour detection using artificial neural network based classifier. <https://www.rroij.com/open-access/study-of-mammogram-micro-calcification-toaid-tumour-detection-using-artificial-neuralnetwork-based-classifier.php?aid=435012014>, 2014.

[23] A. Tiedeu, C. Daul, Texture-based analysis of clustered micro calcifications detected on mammograms, Elsevier Journal of Digital Signal Processing 22(1)(2012) 124-132.

[24] M.N. Arun kumar, H.S. Sheshadri, Building accurate classifier for the classification of microcalcification, International Journal of Computer Science and Information Technologies 3(6)(2012) 5346-5350.

[25] P. Thanathamathee, C. Lursinsap, Handling imbalanced data sets with synthetic boundary data generation using bootstrap re-sampling and Adaboost techniques, Pattern Recognition Letters 34(12)(2013) 1339-1347.

[26] X. Qian, Y.Y. Tang, Z. Yan, K. Hang, ISABoost: a weak classifier inner structure adjusting based Adaboost algorithm—

ISABoost based application in scene categorization, Neurocomputing 103(2013) 104-113.

[27] A. Asuncion, D. Newman, UCI machine learning repository. <http://www.ics.uci.edu/~mlearn/MLRepository.html>, 2007.