Mahalanobis Quality Threshold ARTMAP for Pattern Prediction and Classification

Shahrul Nizam Yaakob
School of Computer and Communication Engineering
University Malaysia Perlis,
Arau, Perlis, Malaysia
ysnizam@gmail.com

Lakhmi Jain
School of Engineering, Mawson Lakes Campus
University of South Australia
South Australia, Australia
Lakhmi.Jain@unisa.edu.au

***Abstract*—This paper introduced an enhancement version of Quality Threshold ARTMAP using the Mahalanobis function. It's known as Mahalanobis Quality Threshold ARTMAP (QTAM-*m*) that increase its capability for pattern classification and prediction purposes. In addition this enhancement also does not consist any initial parameters setting that will affect the final classification outcomes. Thus it's fulfilling the requirement of on-line learning scheme. All parameters involved are also updated using an equation that produced the exact value and not just based on estimation. In general the results had indicates that this improvement increase the classification result based on several testing of benchmark dataset. The proposed technique is also compared with several other techniques within it class.**

Keywords—Fuzzy ARTMAP, Quality Threshold ARTMAP, Pattern Classification, Neural Network.

I. INTRODUCTION

Ever since the Fuzzy ARTMAP neural network (FAM) network introduced back in 1992, there are many improvements have been done. One of the essential objectives of these enhancements is to improve the FAM prediction accuracy. One way to accomplish it is by adopting the probability function to the FAM algorithm. Carpenter, G.A. [1] had produced the FAM network that is capable of being used as nonparametric probability estimator. They had achieved this by modifying the learning algorithm of FAM map field. This modified version of FAM was capable to be applied in a noisy input environment.

Marriott, S. and Harrison, R.F. [2] had introduced other version of FAM called Probability ARTMAP (ProAM). This network also was constructed based on the FAM map field modification. ProAM retains all the FAM network desirable properties but requires fewer categories especially for function approximation problem. Further improvement of ProAM was taken by Srinivasa, N. [3] where the winner-take-all prediction strategy was replaced with distributed prediction mode.

Lim, C.P. and Harrison, R.F. [4,5] had further improved the FAM network by combining it with Probability Neural Network (PNN). The combination is based on two different algorithms. First, FAM algorithm is utilized as an underlying clustering method for learning phase. This is important because FAM can classify the input patterns into different categories for a given class. Secondly, the PNN paradigm is employed during the prediction phase for non-parametric probability estimation procedure. Jervis, B.W., [6] too used the same method by combining both PNN and FAM network. This technique is known as Integrated Probabilistic Simplified Fuzzy ARTMAP (IPSFAM). The main different between this method and Lim work is that Jervis utilized this hybrid method is for both learning and testing phase. The IPSFAM is actually an extended version of Probabilistic Simplified Fuzzy ARTMAP (PSFAM) [7] that consists of simplified FAM and associated program for calculating the estimated Bayes posterior probabilities. This probabilities function is used to find the possible classes for test input pattern.

In order to reduce the effect of noise data environment and inefficiency of fuzzy categories, Williamson, J.R. [8,9] had introduced the Gaussian ARTMAP (GAM) network. This method adopted the concept of separable Gaussian distribution for learning and testing phase of GAM. Unlike FAM, this technique used the maximum probability estimation function during the prediction phase. Granger, E. [10] had demonstrated that GAM with its prediction scheme can achieve better classification results compare to FAM network. This is especially true when the incremental learning process is desired.

Verzi, J.S., [11] also had developed a modified version of FAM known as Boosted ARTMAP (BoAM) that applied boosted learning in probabilistic setting. BoAM network had better performance result in classification task compare to FAM. This technique also produced less categories and it was demonstrated that BoAM does not over-fit the training data.

Dagher, I., [12] used Mahalanobis distance in order to increase the prediction performance of FAM. This improvement method is called Fuzzy ARTMAP-Var (FAM-V). One of the advantage features of FAM-V is that its algorithm does not depend on any parameters tuning. Vuskovic, M. and Du, S. [13] also adopted Mahalanobis distance to reduced the network size and increased the efficiency in training and classification. This network is known as Mahalanobis ARTMAP (MaAM). However, they had found that the computational of covariance matrix and its inversion has introduced some computational overhead to the MaAM. Xu, H. and Vuskovic, M. [14] had extended the MaAM network by providing a better technique to update the covariance value of Mahalanobis distance function. This function also was adapted by Vigdor, B. and Lerner, B[15] to produce a new technique called Bayesian ARTMAP (BaAM). This technique is developed based on Bayes' decision theory for learning and inference with combination of Gaussian distribution for category representation. Other benefit of using BaAM is that its main algorithm allowed category in recognition layer to grow and shrink.

Nevertheless, the next section will discussed on the Quality Threshold ARTMAP algorithm then followed with the explanation on the proposed enhancement of QTAM. Then, a brief presentation is given on the experiment setup. Section V will provide the discussion of the result. Finally the summary is given in Section VI.

II. Quality Threshold ARTMAP (QTAM)

The QTAM was introduced in [18] and it has the same architecture as FAM neural network. This technique also had been used in image classification as shown in [19]. Basically QTAM is designed based on the combination of FAM and Quality Threshold (QT) [16] clustering technique. Generally the algorithm consists of a Euclidean Quality Threshold ARTa (QTAa) module as ARTa to process a stream of input patterns and a Fuzzy ART module as ARTb to process a stream of the corresponding output classes. Similar to FAM, the map field is used to connect both ARTa and ARTb modules together. Each QTA category (node j) encodes three important parameters: n_j which represents the number of training samples it has coded; μ_j which represents the M-dimensional mean vector; and σ_j which represents the standard deviation. Each category j also has its own QT parameter, εj. A number of modifications are introduced to the original QT clustering algorithm in order to make it more efficient for supervised learning. First, the user-defined QT parameter, ε, is determined automatically during the learning phase of QTA. The threshold value is set according to data characteristics. Second, there is no minimum cluster size n_T for QTA. So, a node created in QTA can be constructed by only one input pattern. Third, instead of the Standard Correlation Coefficient QTA utilizes the Euclidean distance as its main similarity measure for the input patterns.

A. QTAM Training Phase

The details of the training procedure of QTAM are as follows:

1) Given a set of training data (M input patterns/vectors, ***I***, and K output classes), find the threshold parameter for each input pattern using Eq. (1). The threshold of input pattern j (Eq. (2)) refers to the nearest Euclidean distance between input pattern j and another input pattern k of a different class (label),

$$\delta_{jk} = \sqrt{\Sigma} \tag{1}$$

$$\varepsilon_j \tag{2}$$

2) Start recruiting nodes for QTA. An input pattern is randomly selected from the input pattern list. Then, find the best candidate node from every available input pattern
   i) To find the candidate node x of input pattern j, first set (initial value)

$$\tag{3}$$

$$\mu_{xi} = I_{ii} \quad i \tag{4}$$

   ii) Compute the Euclidean distance between input pattern j and another input pattern k using Eq. (1). Find the smallest value of δ_{xk}, and select the associated input pattern as the winner.
   iii) Input pattern k is to be accepted as part of the candidate node x if both input patterns j and k belong to the same class and fulfill the following conditions.

$$\tag{5}$$

$$\tag{6}$$

   If either one of the conditions is not satisfied, go to step 3.
   iv) Update the information associated with candidate node x, as follows:

$$\tag{7}$$

$$\mu_{xi}^{new} = \left(1 - \frac{1}{n^{new}}\right)\mu_{xi}^{old} + \left(\frac{1}{n^{ne}}\right. \tag{8}$$

   v) Based on the new μxi, search for another winner using Eq. (1). The winning input pattern should again satisfy all conditions in step (iii). The process continues until no further input patterns can be added to candidate node j without surpassing the conditions in step (iii). The process iterates through all input patterns and forms a candidate node with reference to each available input pattern. In other words, during the early phase of training, the number of candidate nodes x is the same as the number of input patterns.
   vi) Then, select the candidate node with the highest value of n_x for each class (label). If there are two or more candidate nodes that have the same number of n_x, the candidate node with the lowest total standard deviation, S_x as in Eq.(9), is selected as the winner. The standard deviation of node x can be found using Eq. (10).

$$\tag{9}$$

$$(\mu_{xi}^{new})^2 = \left(1 - \frac{1}{n^{new}}\right)A + \left(\frac{}{n}\right. \tag{10}$$

Where

vii) Retain the selected winning node in the recognition layer of QTA for the rest of the training phase. The input patterns that have been used to produce this winning node are now removed. All the remaining input patterns are used again to find another winning node in the QTA training phase.

3) Repeat step 2 until all input patterns have been used to create nodes in the recognition layer of QTA.

B. QTAM Test Phase

The testing phase of QTAM is designed based on the fact that each node can be constructed using one or many input patterns. The following steps are required during the testing phase in order to find the appropriate predicted class for a given input pattern, *k*.

1) Compute the Similarity Degree of Match (SDM) for each candidate node *z* using

$$\mathrm{SDM}_z = \sqrt{\sum_i^M (I_{\cdot\cdot} - \mu \ldots)^2} \quad (11)$$

2) The predicted class is the one with the lowest value of SDM. *N* is the total number of category/node available.

$$Z = \arg\min_z(\mathrm{SDM} \cdot z = 1, 2, \ldots N) \quad (12)$$

III. MAHALANOBIS QUALITY THRESHOLD ARTMAP (MQTAM)

The proposed enhancement for QTAM prediction is based on Mahalanobis distance function. Therefore it is called QTAM with Mahalanobis prediction (QTAM-*m*). Basically this method has the same training and testing phase of QTAM except that Eq. (13) is used to replace Eq. (1) in training and Eq. (11) in testing phase. The variable C_j represents the covariance matrix of the *j*th category.

$$P(I|j) = \frac{1}{(2\pi)^{\frac{m}{2}} \prod_{i=1}^{m} \sigma_{ii}} \times \exp\left(-0.5(I - \mu_j)^T C_j^{-1}(I\right. \quad (13)$$

A. Updating Variance

One of the issues is on how to update the variance value. Even though, some of the past research introduced an equation to update this variable, but the formula constructed is just estimation only. Therefore, a new equation is used to perform this task as shown in Equation (21).

Proof: Given the standard computational method for variance in Eq.(14). Where *n* is equal to the number of elements that produced σ. This equation can be written back as Eq. (15) given that $(x-\mu)^2$ and

$$\sigma^2 \quad (14)$$

$$\sigma^2 = \frac{1}{n}[(x_1^2 + x_2^2 + .. + x_{n-1}^2 + x \quad (15)$$

Replace with n with n_{new} and μ with μ_{new}

$$\sigma_{new}^2 = \frac{1}{n_{new}}[(x_1^2 + x_2^2 + .. + x_{n-1}^2 + x_n^2) - n_{new}\mu \quad (16)$$

$$\sigma_{new}^2 = \frac{1}{n_{new}}[(Z + x_n^2) \quad (17)$$

Then, *Z* can be derived from Eq. (16) to produce Eq. (17) where *Z* can be shown in Eq. (18). Given Eq. (18) and $n_{old} = n_{new} - 1$, Eq. (20) is constructed based on Eq. (19)

$$Z = \quad (18)$$

$$\sigma_{old}^2 = \frac{1}{n_{old}}[(x_1^2 + x_2^2 + .. + x_{n-1}^2 + x_n^2) - n_{ol} \quad (19)$$

$$Z = (n_{new} - \quad (20)$$

Insert Eq. (20) back to (17) and change Input to . Finally the below equation is generated

$$\sigma_{new}^2 = \frac{n_{new}-1}{n_{new}}(\sigma_{old}^2 + \mu_{old}^2) - \frac{n_n}{} \quad (21)$$

B. Updating Covariance

The computational of covariance matrix and its updates is one of the issues discussed when adopting Mahalanobis distance [13]. The process of updating covariance is based on each individual value rather than using matrix operation [15]. Thus a new equation is adopted to perform this task and it is shown in Equation (30). While the covariance matrix of *j*th category is demonstrates in Eq. (22)

$$C_j = | \quad (22)$$

Proof: Given the standard computational method for variance in Eq. (23). Where *n* (for QTAM *n* can be referred as the number of input pattern coded in j^{th} category) is equal to the number of elements that produced cv_{xy} . This equation can be written back as Eq. (24).

$$cv_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \quad (23)$$

$$cv_{xy} = \frac{1}{n}(x_1y_1 + x_2y_2 + .. + x_ny_n - \quad (24)$$

Replace with , with and with

$$cv_{xy} = (x_1y_1 + x_2y_2 + .. + x_ny_n)/n^{new} - \mu_x^{new} \quad (25)$$

$$cv_{xy} = (Z + x_ny_n)/n^{new} \quad (26)$$

Then, *A* can be derived from Eq. (25) to produce Eq. (4.33) where *Z* can be shown in Eq. (27). Given Eq. (27) and $n_{old} = n_{new} - 1$, Eq. (29) is constructed based on Eq. (28)

$$Z = x_1y_1 + x_2y_2 \cdot \quad (27)$$

$$cv_{xy}^{old} = (x_1y_1 + x_2y_2 + .. + x_ny_n)/n^{old} - \mu_x^{ol} \quad (28)$$

$$Z = (n^{new} - 1)(cv_{xy}^{old} \quad (29)$$

Insert Eq. (29) back to (26) and change Input x_n to I_i and yn to I_{i+1}. Finally Eq. (21) is generated and can be utilized to update the covariance value.

$$cv_{xy}^{old} = \frac{\left((n^{new}-1)(cv_{xy}^{old}+\mu_x^{old}\mu_y^{old})+I_i I_{i+1}\right)}{n^{new}} - \mu_x^{old}\mu \qquad (30)$$

IV. EXPERIMENTAL SETUP

Overall 10 different datasets are used to analyze the performance of proposed technique. Where three artificial data sets [23] is used which are the Gaussian 2-D, Concentric and Clouds. Four of the data sets are taken from [20], i.e., Pima Indian Diabetes (PIMA), Wisconsin Breast Cancer (WBC), Heart Disease (HD) and Dermatology, while the Kala-azar Disease (KZAR) and Hepatobiliary Disorders (HEPATO) data sets are based on [22] and [21] respectively. The Acute Coronary Syndrome (ACS) data set is collected from real patient records from a hospital in Malaysia. Generally the accuracy measurement of all the techniques used is prepared based on the 10-folds cross validation process. This work also implemented the Gaussian Quality Threshold ARTMAP (QTAM-*g*) used in [19] for further comparison with the proposed technique. Details of the data sets are summarized in Table 1.

Table 1: Details of the data sets used

	Number of Classes	Number of Data Samples
Gaussian 2D	2	1000
Concentric	2	1000
Clouds	2	1000
PIMA	2	768
WBC	2	699
HD	2	270
KZAR	2	68
ACS	2	118
DERMATOLOGY	6	366
HEPATO	4	536

V. RESULT AND DISCUSSION

To further understand the performance of those proposed techniques, BaAM also been utilized. Table 2 shows the classification results produced by each neural network used in this work.

Table 2: Classification results for prediction analysis

	QTAM	QTAM-*g*	QTAM-*m*	FAM	BaAM
Gaussian 2D	62.54	66.17	66.24	62.74	**66.8**
Concentric	97.41	98.31	**99.03**	95.12	98.76
Clouds	83.63	85.16	**86.48**	79.71	84.52
PIMA	71.18	73.83	**74.2**	63.03	74.14
WBC	96.91	98.15	**99.1**	94.71	98.55
HD	75.93	78.85	79.32	63.33	**79.98**
KZAR	69.12	**73.54**	73.48	70.88	73.02
ACS	80.91	81.97	82.15	80.82	**82.57**
DERMA-TOLOGY	94.44	95.78	**96.13**	88.61	95.21
HEPATO	77.74	78.12	**79.59**	66.79	78.63

From the classification results, it is found that QTAM-*g* network demonstrated better prediction performance compared to the original QTAM for every data set. This is because, QTAM-*g* used the probability function with Gaussian distribution which can measure the similarity between patterns based on mean, μ_{ij} and variance, σ_{ij} . Unlike the original QTAM which utilized Euclidean distance as the primary measurement tool to predict the unknown class of input pattern. The QTAM-*g* also produced similar classification results with QTAM-*m* for KZAR and Gaussian data sets. However, QTAM-*g* performed slightly better in prediction phase using the KZAR data set. Like other networks used in this work, QTAM-*g* only achieved around 66% accuracy in predicting the Gaussian data set. It is proved that the noise data imitated by this data set is not so easy to classify. But, the prediction accuracy demonstrated promising characteristic of QTAM-*g* to classify the densely overlapped data distribution. QTAM-*g* demonstrated a good result in predicting the unknown input pattern for Concentric data set which consists of non-overlapping classes.

Meanwhile, from Table 2 it is shown that QTAM-*m* produced significantly increased in the classification performance compared to the original QTAM network and QTAM- *g*. QTAM-*m* generated the highest accuracy for 6 out of 10 data sets used in this experiment. In addition, 2 of them which are the Concentric and WBC data sets demonstrated that QTAM-*m* performance achieved almost 100% in the prediction accuracy. This proved that the probability function with Mahalanobis distance introduced in the prediction phase is exceptionally accurate and reliable. This shown that Mahalanobis function is better in measuring the similarity between given input patterns compared of using the Euclidean distance. It is because QTAM-*m* can determine the similarity between patterns not only based on mean, μ_{ij} and variance, σ^2 but also using the covariance, cv_{ji} measurement. Similar to QTAM-*g*, this technique also achieved around 66% accuracy in predicting the Gaussian data set which proved that the noise data imitated by this data set is tough to classify. But, the prediction accuracy produced by QTAM-m show that its capability to classify the densely overlapped data distribution.

This work also investigated the performance of BaAM in prediction accuracy. It is perceived that this network produced the highest classification result for three different data sets which are Gaussian, HD and ACS. This indicates that BaAM with its category hyper volume limitation scheme capable of predicting the given unknown input patterns. Furthermore, all of its classification results are better than the original FAM network. However it is hard to set the initial value for mean, μ_{ij} and covariance matrix, cv_{ij} for BaAM. The proposed solution to this problem is by using a small number as the initial value of both parameters. This initial will then updated to the actual value as the training progress. However this enhancement should be investigated in the future work.

As overall, it is notice that all the networks adopted in this experimental work generated the lowest classification result in Gaussian data set. In contrast, the Concentric and WBC data sets produced high accuracy rates (more than 90%) for all the techniques. This signified that data with overlapping classes is most likely to become very challenging task to classify. It is because the characteristic of the data is similar to the noise data where there are no feasible boundaries between classes.

Table 3: The number of nodes produced of each technique

	QTAM	QTAM-*g*	QTAM-*m*	FAM	BaAM
Gaussian 2D	513	510	**505**	730	530
Concentric	295	292	292	467	**289**
Clouds	450	445	**436**	693	476
PIMA	438	430	**428**	534	443
WBC	392	390	**385**	501	399
HD	103	101	99	130	**97**
KZAR	45	44	39	47	**32**
ACS	45	42	**40**	60	49
DERMA-TOLOGY	140	137	128	188	**121**
HEPATO	254	243	240	324	**237**

Table 3 shows the number of nodes generated by all techniques used in this work. In general the highest number of nodes produced by all methods is in Gaussian 2D data set. This happen because it is found that the huge number of overlapping pattern between two classes occurred in this data set. Meanwhile the lowest numbers of nodes (compared to the number of data) created by all the techniques are in Clouds data set. This is due to the low number of overlapping between data which also the opposite side of Gaussian data set characteristic.

It is not surprised to find that FAM generated the highest number of node since all other techniques are the improvement version of it. While QTAM-*m* produced similar number of nodes compared to BaAM network. This is also true when compare with the original QTAM and QTAM-*m*. In general the difference between these techniques is considered small.

VI. CONCLUSION

The Mahalanobis function is introduced in order to enhance the prediction accuracy of QTAM. The proposed prediction technique is based on the Mahalanobis distance function. The improvements are taken place during the testing phase of QTAM network. An innovative approach had been developed to update the variance and covariance values for both Mahalanobis function respectively. It is important to update both parameters so that the proposed prediction techniques can generate high classification accuracy. From the experimental study it is found that QTAM-*m* can achieved better prediction compared to BaAM and other networks.

One of the advantage features of QTAM is that its algorithm is designed such that there are no initial parameters value setting required. Furthermore these all parameters involved in QTAM-*m* are updated automatically. This characteristic is important to make the technique suitable for on-line learning system development. This is not true with BaAM [17] technique where the users have to decide the initial value for several parameters. In addition, the value of variance σ_{ij} and cv_{ij} are updated based on estimation only. However in QTAM-*m*, these two variables are updated precisely using both Eq.(21) and Eq.(30).

REFERENCES

[1] Carpenter, G.A., Grossberg, S. and Reynolds, J.H. (1995) A fuzzy ARTMAP nonparametric probability estimator for nonstationary pattern recognition problems. IEEE Transactions on Neural Networks, Volume 6, Issue 6, Page(s): 1330–1336.

[2] Marriot, S. and Harrison R.F. (1995) A modified fuzzy ARTMAP architecture for the approximation of noisy mappings. *Neural networks*. Volume 8, Issue 4, Pages: 619-641

[3] Srinivasa, N. (1997) Learning and generalization of noisy mappings using a modified PROBART neural network, IEEE Transactions on Signal Processing, Volume 45, Issue 10, Page(s):2533–2550

[4] Lim, C.P. and Harrison, R.F. (1995). Probabilistic Fuzzy ARTMAP: an autonomous neural network architecture for Bayesian probability estimation. Fourth International Conference on Artificial Neural Networks. Page(s): 148–153.

[5] Lim, C.P. and Harrison, R.F. (1996) Estimation of Bayesian a posteriori probabilities with an autonomously learning neural network. UKACC International Conference on Control, (Conf. Publ. No. 427). Volume 1, Page(s): 199–204.

[6] Jervis, B.W., Djebali, S. & Smaglo, L. (2004) Integrated probabilistic simplified fuzzy ARTMAP. IEE Proceedings on Science, Measurement and Technology, Volume: 151, Issue: 3, Pages: 218–228

[7] Jervis, B.W., Garcia, T. and Giahnakis, E.P. (1999) Probabilistic simplified fuzzy ARTMAP (PSFAM). IEE Proceedings - Science, Measurement and Technology, Volume: 146, Issue: 4, Pages: 165–169

[8] Wiliamson, J.R. (1996) Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, vol. 9, no.5, pp. 881-897

[9] Williamson, J.R. (1997) A constructive, incremental-learning network for mixture modeling classification. Journal of Neural Computation, Vol. 9, Issue 7, Pp. 1517-1543

[10] Granger, E., Connolly, J.F. and Sabourin, R. (2008) A comparison of fuzzy ARTMAP and Gaussian ARTMAP neural networks for incremental learning. IEEE International Joint Conference on Neural Networks. IJCNN. IEEE World Congress on Computational Intelligence. Page(s): 3305–3312.

[11] Verzi, S.J., Heileman, G.L., Georgiopoulos, M. and Healy, M.J. (1998) Boosted ARTMAP. The IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence, Volume: 1, Pages: 396–401

[12] Dagher, I., Georgiopoulos, M., Heileman, G.L. and Bebis, G. (1998) Fuzzy ARTVar: an improved fuzzy ARTMAP algorithm. The IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence, Volume: 3, Pages: 1688–1693

[13] Vuskovic, M. and Sijiang D.,(2002) Classification of prehensile EMG patterns with simplified fuzzy ARTMAP networks. Proceedings of the International Joint Conference on Neural Networks. Volume 3, Page(s): 2539-2544

[14] Xu, H. and Vuskovic, M. (2004) Mahalanobis distance-based ARTMAP network. IEEE International Joint Conference on Neural Networks. Volume 3, Page(s): 2353-2359

[15] Vigdor, B. and Lerner, B. (2007). The Bayesian ARTMAP. IEEE transaction on Neural Networks. Volume 18, Issue 6, Page(s): 1628–1644

[16] Heyer L.J., Kruglyak, S. and Yooseph, S. (1999) Exploring Expression Data: Identification and Analysis of Coexpressed Genes, Genome Research, Vol. 9 Pages: 1106-1115

[17] Vigdor, B. and Lerner, B. (2007). The Bayesian ARTMAP. IEEE transaction on Neural Networks. Volume 18, Issue 6, Page(s): 1628–1644.

[18] Shahrul N.Y., C.P. Lim and Lakhmi Jain (2010). A novel Euclidean quality threshold ARTMAP network and its application to pattern classification. Journal of Neural Computing & Applications (ISSN 0941-0643). Volume: 19 Issue 2, Pages: 227-236

[19] Shahrul N.Y. and Lakhmi Jain (2011). An Insect Classification Analysis Based on Shape Features Using Quality Threshold ARTMAP and Moment Invariant. Jour. of Applied Intelligence. Page 1-19

[20] Asuncion A and Newman DJ (2007) UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. http://www.ics.uci.edu/~mlearn/MLRepository .html. Accessed 30th October 2011

[21] Mitra, S. (1994) Fuzzy MLP based expert system for medical diagnosis, Fuzzy Sets and Systems, Vol. 65, Pages: 285-296

[22] Mitra, S. and Pal, S. K. (1995) Fuzzy multi-layer Perceptron, inferencing and rule generation, IEEE Transactions on Neural Networks, Vol. 6, Pages: 51-63

[23] Verleysen M, Bodt ED and Wertz V (2008) Université catholique de L-ouvain (UCL) Neural Network Group, http://www.dice.ucl.ac.be/neura l- nets/Research/Project- s/ELENA/elena.htm, Accessed 15 Nov 2010.