Volume 5, Issue 1, January 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Hybrid Expert System for Tea Insect Pest Identification

Indrajit Ghosh
Agro-Computing Research Laboratory, Dept. of Computer Science,
Ananda Chandra College, Jalpaiguri, West Bengal, India

Abstract - Tea is one of the major plantation crops of India, providing the bread and butter to a large number of people along with the inflow of foreign capital. Insect pests lead to the crop loss upto 40% and are considered as one of the productivity barriers. Undoubtedly, insect pest management can be identified as one of the most important issues to take constant attention throughout the year. Normally human experts are employed for this purpose. But they are very scarce commodity for such a widespread industry. So Expert System is a graceful alternative. Some Expert Systems have been reported but they are all rule based and suffers from serious shortcomings. As a remedy, A Case Based hybrid methodology for designing Expert Systems has been proposed by Ghosh and Samanta. This is an attempt to develop a hybrid expert system for insect pest identification in tea by using the methodology proposed by Ghosh and Samanta.

Keywords: Tea insect pest, Case Based Reasoning, hybrid expert systems.

I. INTRODUCTION

Tea (*Camelio Sinensis* (L) O. Kurtze) is one of the major plantation cropsin India and is spread over a vast area, from North-East to South providing the bread and butter to a large number of people. It also earns a huge amount of foreign capital. The demand of tea is gradually increasing but the land under cultivation is not increasing significantly to keep pace with the raising demand of productivity. So it is important to find out the productivity barriers and to adopt some appropriate and better management practices.

One of the major issues of tea crop protection is insect pest management. As it is under varying agro-climatic conditions, it provides a favorable breeding ground for a variety of insect pests. Several species of the insect pests attack tea plants in different seasons. Loss of crop varies according to the severity of attack(s) and it could be up to 40% in devastating attacks by defoliators [1].Undoubtedly, before taking any action of control, proper identification of insect pest is very vital and demands constant attention throughout the year for better protection of crops.

Designing Case Based Reasoning (CBR)based Expert Systems(ES) and some CBR based hybrid systems is a prominent technique for developing diagnostic type of systems in different domains [2-14]andoffers greater advantages [15], where the domains are less understood by nature and formalization of knowledge is very difficult. But there is no universal CBR method suitable for every application domain.

In general, CBR based ESs have two major parts, the case retriever and the case reasoner[15]. In such systems, these two tasks are done by selecting a set of potentially optimal feature subsets and designing a robust classifier.

Specifically, for developing an ES using CBR approach, selection of appropriate subset of significant features and fixation of feature weights are important.In tea insect pest domain, most of the features (field observations) are presented in linguistic terms, so it is more appropriate to select the subset of significant features on the basis of their relevance (feature weight). A lot of works have been reported on selection of feature subsets[16-24].But, they suffer from any of the shortcomings: computational complexity, classifier dependency, large storage capacity and time consuming similarity measures. Most of them are concerned with the retrieval of previous cases which is a complex task.

On the other hand, for designing a classifier, several approaches have been proposed which includes Rough Set theory, Nearest Neighbour approach, Beysian classifier, ANN, RBF networks and its variant Probabilistic Neural Networks. But the selection of feature subsets which are relevant to a specific classifier is one of the central problems [24]. The computational complexity of training algorithm is also a vital issue [25].

In this regard, a hybrid model proposed by Ghosh and Samanta(GS model)[26,27] can be considered as promising one. Here the classifier is designed as a simple single layer neural network. The learning is done by a CBR method from field data and the classifier accepts the prior trained data as inputs. This model was successfully used in resuscitation management [28] and for identification of tea diseases[29]. The aim of this work is to develop a hybrid expert system for identification of insect pest in tea by using the GS model.

The next section presents an overview of GS model (mathematical foundation and system architecture). Section IIIcontains a case example to present how this model can be used for insect pest identification in tea, for better understanding. The conclusion is provided in section IV.

II. AN OVERVIEW OF GS MODEL

A. Case representation

Since Exemplar representations are far more flexible than prototype representations [30], so the case-base of the system is composed of rules embodying domain specific knowledge with category-exemplar type of case representation where a set or sub-set of features plans for all possible goal(s) (cases) which is/are likely to be achieved [31]. Exemplars are the possible cases and categories are the case tables which contain the collection of cases of the same class. To incorporate the dynamic nature of the problem domain, it is argued that the concept description cannot be represented only with category- exemplars, but also with a set of specific weights associated with features and as well as exemplars(cases). The weights associated with features must be adaptive to the changing environment in order to represent the acceptability of the exemplars. The training data is not assigned arbitrarily, but obtained from the region specific real field cases of the problem domain. The system becomes rich with definite known cases and sorts out the relevant features for all possible categories (case tables) and fixes the primary weights (Relative Feature Weights) of all the features through an evaluation process as described in the following subsections.

B. Feature Vector (F)

Feature Vector (F) is a set of elements transformed from the feature set of all possible features in a domain. If N is the total number possible features in the feature set of a domain then the F can be represented as:

$$[F] = F_1, F_2, F_3, \ldots\ldots\ldots F_N \quad (1)$$

where each of the elements F_1, F_2, ...F_N possess a value either 1 or 0 depending upon the set of observed features. If for a particular observation, the i-th feature is observed, then $F_i = 1$ otherwise it is 0.
As an example, if a domain consists of total 6 features and for a particular case, features F_1, F_3 and F_6 are observed, then the feature set of this 6 features are transformed to a Feature Vector as: $[F] = 1, 0, 1, 0, 0, 1$.

C. Relative Feature Weight (RFW)

Relative Feature Weight is the weight assigned to each feature and is a measure of relevance of a feature with respect to a category. When a new case appears, the initial weight of an element of feature set is not known. So an unbiased weight should be considered. This unbiased weight can be estimated by 1/N, where N is the number of elements in the set of features or sub-set of features constructing the new case. The feature weight for m-th case should be the product of average feature weight (FWav) and the number of occurrences (I) of the feature up to m-th retained case in a case table.
The Feature Weight (FWm) for m-th case :

$$FWm = FW(av)m \,.\, I \quad (2)$$

Where $FW(av)m = \sum_{j=1}^{M} FWj /(N.M)$ and M is the total number of retained cases.

Case Weight (CW) :
It is the sum of the weights of all the possible features constructing the case. For m-th case, the Case Weight is

$$CWm = \sum FWmi \quad ; \quad 1 \leq i \leq N \quad (3)$$

where i represents the number of features contributing the m-th case.
Relative Feature Weight (RFW) :
It is the ratio of the feature weight to the case weight. For m-th case , the relative feature weight is

$$RFWm = FWm / CWm \quad (4)$$

D. Bias vector (Φ)

Bias is a critical barrier of the weighted sum above which a case is considered to be fired and the fixation of the bias is a vital task to improve the performance of the system. The bias for cases under a category has been fixed on the basis of Relative Feature Weights (RFW). We have classified the features of each category in 3 sub-sets; (i) major (Mj), (ii) moderate (Md) and (iii) minor (Mn). The features having RFW greater than the arithmetic mean (M_a) are labeled as 'major' features. The features having RFW in the range between $M_a/2$to M_a is the 'moderate' and features those having RFW less than $M_a/2$are the 'minor' features respectively.
The rationale for estimation of bias value is based on the theme that at least one member of each group should have the contribution and the features with highest RFW within each sub-set should have the highest priority to be considered in fixing the bias. The bias for i-th category is

$$\Phi_i = [Max(RFW_{Mj}) + Max(RFW_{Md}) + Max(RFW_{Mn})]_i \quad (5)$$

The Bias Vector is a set where the elements represent the biases (case firing thresholds) of all the possible categories in the domain. If M is the total number of categories in a domain, then the Bias Vector (Φ) can be represented as:

$$[\Phi] = \Phi_1, \Phi_2, \Phi_3, \ldots\ldots\ldots \Phi_M$$

E. Synapse Matrix (ω)

Synapse Matrix (ω) is a matrix whose elements are the Relative Feature Weights of all the features constructing the possible categories in a domain. The Relative Feature Weights of all the features corresponding to a category constitutes the row of this Synapse Matrix. An element of the row will be considered as 0 (zero) if the corresponding feature has no contribution (non-significant feature, that is, RFW = 0) to that particular category. So this ω is a M x N matrix where M is the number of categories and N is the total number of features in a domain. The ω can be represented as:

$$[\omega] = \begin{bmatrix} \omega_{11}\omega_{12}\omega_{13} \ldots \omega_{1N} \\ \omega_{21}\omega_{22}\omega_{23} \ldots \omega_{2N} \\ \ldots\ldots\ldots \end{bmatrix} \quad (6)$$

Where ω_{11} is the Relative Feature Weight (synapse) of feature-1 of the category-1, ω_{12} is that of feature-2 of the category-1 and so on.

F. Saturation

It is the measure of whether a new case when classified is to be retained into case library or not for primary training. A threshold value δ has been considered for measuring saturation. The relative feature weights are tested for saturation with interval of 5 cases. Saturation is attained when

$$\left| RFW_{\mathbf{m\text{-}th}} - RFW_{\mathbf{(m+5)\text{-}th}} \right| \le \delta \quad (7)$$

for all contributing features of the category. This interval may be accepted on the argument that the contributions from less than 5 cases might not be significant for saturation testing and on the worst case this may increase case memory by 5 cases only.

G. Classifier

The classifier is designed as a single layer neural network (McCulloch-Pitts model)[25], where the features are considered as input values and the Relative Feature Weights as the synapse values respectively. The bias values are also obtained from Relative Feature Weights. When a set/sub-set of observed features is received, it is then transformed to a binary feature vector with elements 1 or 0. The input values thus processed, reach to the classifier. The classifier fires only when the weighted sum of all the potential values exceeds a certain limiting value (bias) of that particular case. The successful (fired) casesupdates the Relative Feature Weights (synapses) of the concerned features if retained in the case library. Thus the learning makes sense within the context of the problem domain.

If for a particular domain the number of possible categories is M, then the classifier contains M neurons (nodes) each of which is responsible to classify a particular category. Each neuron accepts Feature Vector and Bias Vector as inputs. The synapse values are accepted from the Synapse Matrix. Elements of each row of the Synapse Matrix provide the synapses to the corresponding neuron. The neurons then generate the Activation Values and give output by using output function. Depending upon the output values, the case(s) is/are considered to be classified. The block diagram of the classifier is presented in Fig.1.

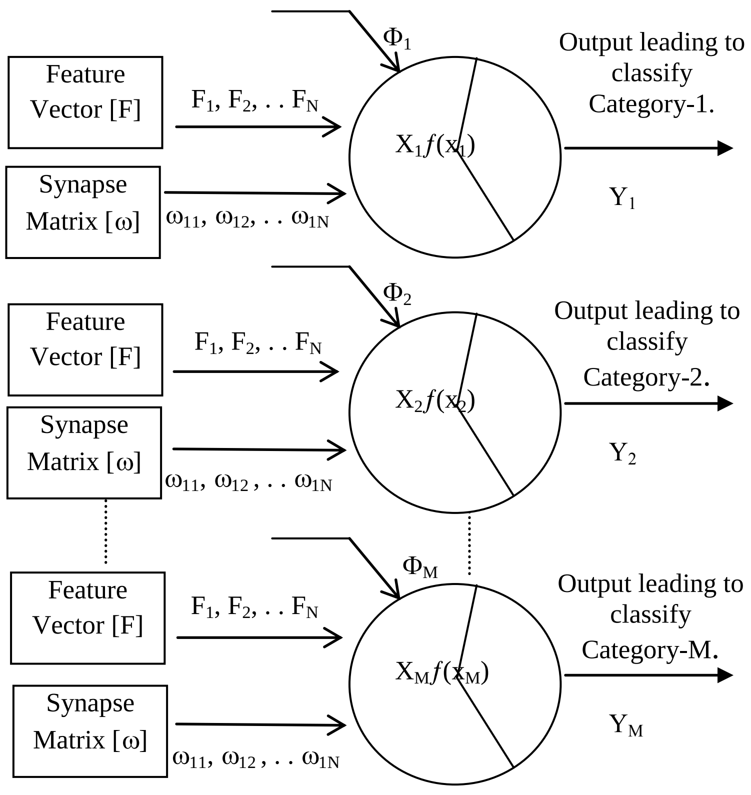

Fig.1. Block diagram of the classifier for M categories.

H. Activation Value (X) and the System Output (Y)

The general model of the classifier consists of a summing part followed by an output part. The Activation Value (X) is given by a weighted sum of input values and a bias term. The output signal is typically a nonlinear transfer function of the activation value. For present problem, if F_1, F_2, F_3. . . .F_N be the elements of the Feature Vector (F) and ω_{i1}, ω_{i2}, ω_{i3}, …..ω_{iN} are the synapse values for i-the node, then the Activation Value (X_i) for i-th node can be represented as:

$$X_i = \sum_{j=1}^{N} \omega_{ij} . F_j - \Phi_i \quad (8)$$

where N is the total number of features in the domain.

Then the output value of the i-th node can be defined by the output function

$$Y_i = f(x_i). \quad (9)$$

The transfer function, used here is a binary function, where

$$f(x_i) = 1, \quad \text{if } x > 0$$
$$= 0, \quad \text{if } x \le 0$$

If $f(x_i) = 1$, the i-th category is considered to be fired (classified); otherwise not considered.

I. System architecture

Each case is represented with a set of features accepted as case descriptors and the collection of cases of the same type form a category (case table). In the learning phase, when the system gets trained with definite observed cases, the RFW of each feature gradually approaches to saturation and the all significant features are obtained when $|RFW_{m\text{-}th} - RFW_{(m+5)\text{-}th}| \le \delta$. At saturation, the RFWs of all the features of each category are known. The system is now considered to be primarily trained and the primary case library is formed.

Now if a new case appears, the system receives the set of observed features and the Feature Vector Generator transforms observed feature set to Feature Vector (F). The system activates the Synapse Matrix Generator. The Synapse Matrix Generator then scans the case library and feature database to pick up the Relative Feature Weights of all the features for all categories and generates the Synapse Matrix (ω) according to equation 6. The Bias Vector Generator calculates the bias of each category and constructs the Bias Vector (Φ).

The classifier calculates the activation values (X) of the new case against each category and generates the outputs (Y) by using the output function in equation (9). The category for which the output is 1, is the classified category, that is, the new case is of this category. The block diagram of the system architecture is presented in Fig.2.

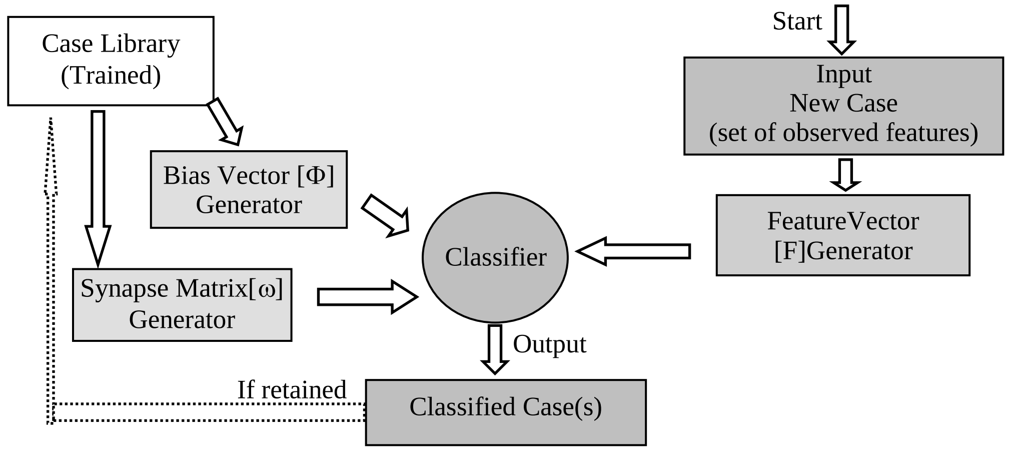

Fig.2. Block diagram of the system architecture.

III. CASE ILLUSTRATION

In tea gardens, the bushes are often subject to the attack of various insect pests throughout the year. When a bush is attacked by a particular type of insect pest, some set of signs and symptoms should be observed on a specific part of the bush. These set of sign and symptoms (features) lead to proper identification of the insect pests. For all cases, all the features for a particular insect pest may not be observed at a time.

There are 8 major insect pests of tea (***Red Spider, Helopeltis, Scarlet Mite, Thrips, Aphid, Jussid, Purple Mite and Pink Mite***) which have been considered in this system [32]. So the case library has been designed to contain 8 categories. Each category is allotted to contain the observed set of features of a particular insect pest. 31possible features leading to cover these 8insect pests are identified by using domain knowledge from various literaturesand domain experts and are grouped under 11 rubrics as presented in Table I. This general feature set contains the all possible sub sets of features due to the attack of these 8 insect pests.

TABLE I

Sl. No.	Code and Description		Sl. No.	Code and Description
1. Site of damage			***5. Leaf spot***	
1	A1- Young leaf		20.	E1 – Brown ring
2	A2 – Matured leaf		21.	E2 – Reddish brown patch
3	A3 – Bud & young leaf		22.	E3 – Sand papery lines
4	A4 – Tender stem		***6. Mid Rib colour***	
2. Leaf surface			23.	F1 – Brownish
5	B1 - Upper		24.	F2 – Upper side brownish
6	B2 - Lower		***7. Edge colour***	
3. Leaf appearance			25.	G1- Brownish
7.	C1 – Dry up		26.	G2 - Pinkish
8.	C2 - Leathery		***8. Tip colour***	
9.	C3 – Dusted white		27.	H1- Brownish
10.	C4 – Curved downwards		***9. Vein colour***	
11.	C5 – Curved upwards		28.	I1 - Brownish
12.	C6 – Crinkled & curled		***10. Finger Tip test***	
13.	C7 - Deformed		29.	J1 – Red smear
14.	C8 – Margin recurved & brown		***11. Bush appearance***	
4. Leaf colour			30.	K1 – New flush stunted
15.	D1 – Copper bronze		31.	K2 - Defoliation
16.	D2 - Yellow			
17.	D3 - Pale			
18.	D4 – Purple bronze			
19.	D5 – Brown			

Observed set of features leading to the definite known cases of attack of these 8 insect pests have been collected from the tea gardens of northern part of West Bengal State in India and have been supplied to the system as first cum first serve basis to train the system. After each entry, the iterative evaluation process generates the values of RFW of each feature against each category and the system tests for saturation with $\delta \sim 10^{-3}$. A positive result of saturation testing indicates the end of primary training. The features with RFWs = 0 are the non-significant or noisy features for that particular insect pest and have no contribution.

The significant features and the Relative Feature Weights (which constitute the non-zero elements of Synapse Matrix) are obtained against each of these 8 categories. As an example, The RFWs of 5 significant features (RFW # 0) of Red spider attack are shown in Table II for better understanding.

TABLE II

Feature codes	Significant features	Non-zero elements of Synapse Matrix
A1	Site of damage IS Matured leaf	$\omega_{1,02} = 0.209$
D5	Leaf colour IS Brown	$\omega_{1,19} = 0.160$
E2	Leaf spot IS Reddish brown patch	$\omega_{1,21} = 0.385$
J1	Finger tip test IS Red smears	$\omega_{1,29} = 0.195$
K1	Bush appearance IS New flush stunted	$\omega_{1,30} = 0.051$

Now let us proceed to calculate the bias values (elements of Bias Vector) for each category. As a primary task to calculate the bias, the type (i.e. major, moderate or minor) of the features are identified. From the equation 5, the bias Φ_1 for the Category-1 is

$$\Phi_1 = 0.051 + 0.160 + 0.385 = 0.596$$

Similarly, the bias values for other 7 categories are calculated. The bias values for 8 categories are presented in Table III.

TABLE III

Insect pests	Bias (Threshold) values
1. Red spider	$\Phi_1 = 0.596$
2. Helopeltis	$\Phi_2 = 0.532$
3. Scarlet Mite	$\Phi_3 = 0.412$
4. Thrips	$\Phi_4 = 0.685$

5. Aphid	$\Phi_5 = 0.572$
6. Jussid	$\Phi_6 = 0.411$
7. Purple Mite	$\Phi_7 = 0.349$
8. Pink Mite	$\Phi_8 = 0.505$

Now the system is ready to classify the new cases.
To test the performance of the system, more than 60 definite and real field cases of the problem domain were supplied to the system for classification. The system classified more than 89% cases very accurately. Due to space constraints, only one example is being presented here.

Case example 1.
The observed set of features and feature codes for Case example 1 are:
1. Site of damage IS Young leaf (A1)
2. Leaf appearance IS Leathery (C2)
3. Leaf appearance IS Crinkled & curled (C6)
4. Leaf colour IS Pale (D3)
5. Vein colour IS Brownish (I1)

On the basis of this feature set, the classifier generates the activation values of the above case in respect to all categories and uses the transfer function to classify the case. The activation values and the classification of case by the system are shown in Table IV. The actual field observation is done by an expert and is presented in the last column of the same table for comparison.

TABLE IV

Insect pests	$\sum\omega.$ F	Φ	X	Y	Classification by the system	Actual field observation
1. Red spider	0.000	0.596	-0.596	0		Pink Mite
2. Helopeltis	0.000	0.532	-0.532	0		
3. Scarlet Mite	0.091	0.412	-0.321	0		
4. Thrips	0.269	0.685	-0.416	0		
5. Aphid	0.436	0.572	-0.136	0		
6. Jussid	0.255	0.411	-0.156	0		
7. Purple Mite	0.032	0.349	-0.317	0		
8. Pink Mite	0.591	0.505	+0.086	1	Pink Mite	

IV. CONCLUSION

Being inspired by the performance of the GS model in resuscitation management and plant disease identification, this tea insect pest identification system has been developed by using Microsoft Visual FoxPro, based ON GS model. It has been installed in some selected tea gardens where agro-climatic conditions are different from each other. The feedbacks obtained are very satisfactory(89%) and the domain experts do concur with the accuracy level. The system rules out non-relevant features from the real field cases at the time of training. So the uncertainty is reduced to a great extent. The uncertainty concerned is related to the field observation which can be minimized by using other methods. The next aim is to modify the system in this direction.

ACKNOWLEDGEMENT

This work is funded by the University Grant Commission of India, Eastern Regional Office, LB-8, Sector III, Salt Lake, Kolkata, WB, India under Minor Research Project, No. F. PSW-076 /12-13(ERO). The contribution of Prof. R. K. Samanta, University of North Bengal is highly acknowledged.

REFERENCES

[1] B. C. Phukan,"Towards Attaining Sustainable Productivity",Workshop Report,Tea Research Association, Jorhat, India,pp. 51-65, 1996.
[2] M. M. Yusof and C. D. Buckingham, "Medical case-based reasoning: A review of retrieving, matching and adaptation process in recent systems",in *Proceedings of the IASTED International Conference*,2009,p. 639-53.
[3] W. M. Roddis, J. Bocox,"Case-Based approach for steel bridge fabrication errors",*Journal of Comp. in Civil Engineering, ASCE*,pp. 84-91, 1997.
[4] I. Watson and F. Marir, "Case-based reasoning: A review",*The Knowledge EngineeringReview*,vol. 9(4),pp. 327-354, 1994.

[5] A. K. Goel, J. Kolodoner, L. M. Pearce,R.Billington and C. Zimring,"Towards a case-based tool for aiding conceptual design problem solving",in Proc. the DARPA Case-Based Reasoning Workshop, San Francisco, Morgan Kaufmann, 1991.

[6] M. Corchado Jun and B. Lees, "A hybrid case-based model for forecasting",*Applied Artificial Intelligence*, vol. 15, pp.105-127, 2001.

[7] Xu and D. Li,"Developing a case-based knowledge system for AIDS prevention",*Expert Systems*, vol. 11(4), pp. 237-244, 1994.

[8] A. J. Frangou, Wan Yan,J. Antony and M. Kaye,"ESAS: A case-based approach to business planning and monitoring", *Expert Systems,* vol. 15(3), pp. 82-196, 1998.

[9] T. Felix andS. Chan, "Application of a hybrid case-based reasoning approach in electroplating industry", *Expert Systems with Applications*, vol. 29(1), pp. 121-130, July, 2005.

[10] M. L. Huang, Y. H. Huang, W. M. Lee, R. K. Li andT. H. Wang,"Using a Case-Based Reasoning, Neural Network and Adaptive Neuro-Fuzzy Inference System Classification Techniques in Breast Cancer Dataset Classification Diagnosis",*Journal of medical systems*, vol. 36(2), pp. 407-414, 2012.

[11] X. Blanco, S. Rodriguez, J. M. Corchado and C. Zato, "Case-Based Reasoning Applied to Medical Diagnosis and Treatment". *Distributed Computing and Artificial Intelligence*, Springer International Publishing, pp. 137-146, 2013.

[12] Chang Pei-Chann, Huang Chinng-Hua and Tsai Chi-Yang,"A Hybrid System by the Integration of Case-Based Reasoning, with Support Vector Machine for Prediction of Financial Crisis",*International Journal of Innovative Computing Information and Control*,vol. 9(6), pp.2245-2260, June, 2013.

[13] A. Dina, Sharaf-el Deena, Moawad F. Ibrahim and M. E. Khalifa, "A Breast Cancer Diagnosis System using Hybrid Case-based approach",*International Journal of Computer Application*, vol. 72(23), pp. 14-19, 2013.

[14] S. K. Biswas, N. Sinha, B. Purakayastha and L. Marbaniang, "Hybrid Expert System using Case based reasoning and neural network for classification",*Biologically Inspired Cognitive Architectures*, vol. 9, pp. 57-70, July, 2014.

[15] S. K. Pal and C. K. Shiu Simon, *Foundations of Soft Case-Based Reasoning*, Wiley-Interscience Johnons Inc., pp. 4-11, 2003.

[16] H. Almuallin and T. G. Dietterich, "Learning with many irrelevant features",in *Proc. Ninth National Conf. on Artificial Intelligence*, 1991, MIT Press, pp. 547-552.

[17] K. Kira and L. A.Rendell, "The feature selection problem : traditional methods and a new algorithm", *in Proc. of the Tenth Nat. Conf. on Artificial Intelligence*, 1992, San Jose, CA, AAAI Press / The MIT Press, Menlo Park, CA, pp. 129-134.

[18] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning",*Artificial Intelligence*, vol. 97, pp. 245-271, 1997.

[19] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection", in*Proc. of the 18th Int. Conf. on Machine Learning*,2001, Williamstown, MA, pp. 74-81.

[20] M. Dash and H. Liu, "Feature selection for clustering", in*Proc. of the Fourth Pacific Asia Conf. on Knowledge Discovery and Data Mining, (PAKKD-2000),*2000, Kyoto, Japan, Springer, Berlin, pp. 110-121.

[21] R. Kohavi and G. H. John, "Wrappers for feature subset selection",*Artificial Intelligence*, vol. 97(12), pp. 273-324, 1997.

[22] H. Liu, H. Motoda and L. Yu,"Feature selection with sampling",in *Proc. of the Nineteenth Int. Conf. on Machine Learning*,2002, Sidney, Australia, pp. 395-402.

[23] H. Liu, H. Motoda and L. Yu, "A selective sampling approach to active feature selection",*Artificial Intelligence*, vol. 159, pp. 49-74, 2004.

[24] T. R. Payne and P. Edwards, "Implicit Feature Selection with the Value Difference Metric",in *Proc. 13thEuropean Conference on Artificial Intelligence(ECAI-98)*, 1998, Ed. Henri Prade, John Wiley & Sons. Ltd.

[25] M. Minsky and S. Papert, "*Perceptrons*", The MIT Press, Cambridge, MA, 1998.

[26] I. Ghosh and R. K. Samanta, "A Model for Case-Based Learning", *The Journal of the Computer Society of India*, vol. 30(4), pp. 9-14, 2000.

[27] I. Ghosh and R. K. Samanta, "Design and Implementation of a Context-Sensitive Case-Based Classifier Approach", *The Journal of the Computer Society of India*, vol. 33(1), pp. 12-21, 2003.

[28] I. Ghosh, S. K. Datta and R. K. Samanta, "Feature Selection for Neonatal Resuscitation Management using Case-Based Reasoning", *Modeling, Measurements and Control C,* AMSE, France, vol. 68(3), pp. 67-75, 2007.

[29] I. Ghosh, "A CBR-Classifier Hybrid System for Plant Disease Diagnosis", *IFRSA's International Journal of Computing*,vol. 1(4), pp. 647-654, 2011.

[30] D. W. Aha and R. L. Goldstone, "Concept learning and Flexible Weighting", in *Proc. of the Fourteenth Annual Conf. of the Cognitive Science Society*, 1992, Bloomington, IN, Laurence Erlbaum, pp. 534-539.

[31] A. Aamodtand Enric Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches", *AICom–Artificial Intelligence Communications*, vol. 7(1), pp. 39-59, 1994.

[32] Workshop Report on Pest and Disease Management. Tea Research Association, Nagrakata Branch, West Bengal, India, 23rd to 25th March, 2000.

