Accepted Manuscript

A neuro-fuzzy classification technique using dynamic clustering and GSS rule generation

Heisnam Rohen Singh, Saroj 0snKr. Biswas, Biswajit Purkayastha

PII: S0377-0427(16)30196-0
DOI: http://dx.doi.org/10.1016/j.cam.2016.04.023
Reference: CAM 10616

To appear in: *Journal of Computational and Applied Mathematics*

Received date: 19 November 2015
Revised date: 9 April 2016

Please cite this article as: H.R. Singh, S.0. Biswas, B. Purkayastha, A neuro-fuzzy classification technique using dynamic clustering and GSS rule generation, *Journal of Computational and Applied Mathematics* (2016), http://dx.doi.org/10.1016/j.cam.2016.04.023

Title page

Name of Authors: Heisnam Rohen Singh[1], Dr. Saroj Kr. Biswas[2], Dr. Biswajit Purkayastha[3]

Title: A Neuro-fuzzy Classification Technique using Dynamic Clustering and GSS Rule Generation

Affiliations and addresses of the authors: NIT, Silchar-788010, Assam, India[1,2,3]

E-mail of corresponding author: rohenheisnam87@gmail.com

Telephone number of corresponding author: +91 9402612120

Abstract: An efficient feature subset selection for predictive and accurate classification is highly desirable in many application domains like medical diagnosis, target marketing etc. Many neuro-fuzzy models were proposed for feature selection and efficient classification. One of such existing neuro-fuzzy models is Enhance Neuro-Fuzzy (ENF) system for classification using dynamic clustering. The major problem of ENF is, huge number of linguistic variables generated for each feature, which results poor interpretation of the rules generated for classification. Therefore, this paper proposes a neuro-fuzzy model which is an extension of ENF. The novelty of the proposed model lies in determining less number of linguistic variables for each feature and also in generating significant linguistic variables in the rules for classification with better interpretation and accuracy. Six datasets are used to test the performance of the proposed model. 10-fold cross validation is used to compare the performance of the proposed model with others. It is observed from the experimental results that the performance of the proposed model is superior to others.

Keywords: Feature selection, neuro-fuzzy, dynamic clustering, linguistic variables, Golden section search method.

A Neuro-Fuzzy Classification Technique using Dynamic Clustering and GSS Rule Generation

Abstract: An efficient feature subset selection for predictive and accurate classification is highly desirable in many application domains like medical diagnosis, target marketing etc. Many neuro-fuzzy models were proposed for feature selection and efficient classification. One of such existing neuro-fuzzy models is Enhance Neuro-Fuzzy (ENF) system for classification using dynamic clustering. The major problem of ENF is, huge number of linguistic variables generated for each feature, which results poor interpretation of the rules generated for classification. Therefore, this paper proposes a neuro-fuzzy model which is an extension of ENF. The novelty of the proposed model lies in determining less number of linguistic variables for each feature and also in generating significant linguistic variables in the rules for classification with better interpretation and accuracy. Six datasets are used to test the performance of the proposed model. 10-fold cross validation is used to compare the performance of the proposed model with others. It is observed from the experimental results that the performance of the proposed model is superior to others.

Keywords: Feature selection, neuro-fuzzy, dynamic clustering, linguistic variables, Golden section search method.

1. Introduction

Data-driven rule extractions are widely used in machine learning and data mining algorithms for classification, prediction and clustering. These algorithms operate on a huge amount of data with multiple dimensions to extract knowledge. Moreover, most of these data are insignificant to the specific domain. An important concept that helps in classification, clustering and a better understanding of the domain is feature selection [1]. Feature selection is a process of selecting a subset of features from a set of features without losing the characteristics and identity of the original object. There are two factors that affect feature selection: irrelevant features and redundant features. Irrelevant features are those which provide no useful information in a context

and redundant features are those which provide no more information than the currently selected features.

Feature selection has been proved an inevitable part of a classifier through numerous researches. In the real-world scenario, to better represent the domain, many candidate features are introduced, which result the existence of irrelevant/redundant features to the target concept [2]. In many classification problems, due to the huge size of data, it is difficult to learn good classifiers before removing these unwanted features. Reducing the number of irrelevant/redundant features can drastically abate the running time of the learning algorithms and yields a more general classifier. Feature selection provides us with the advantages of facilitating data visualization and data understanding, reducing training and utilization times, reducing the measurement and storage requirements and defying the curse of dimensionality; which aids in the elevation of classification performance ([1], [3]). Hua et al. [4] have reported that feature selection is a part of the classification rule.

Feature selection can be done using various techniques like mutual information ([5], [6]), genetic algorithm ([6-8]), bayesian network [9], artificial neural network (ANN) [10] etc. All these techniques have certain limitations. In mutual information technique, it is hard to calculate mutual information between the features that have continuous values, as it is often difficult to compute the integral in the continuous space based on a limited number of samples. In the bayesian network, the number of structures super-exponentially increases as number of features increases and in this more focus is made on the dependency of the features rather than the importance of features. In the genetic algorithm, some kind of randomness is involved and is very hard to assign importance to more significant features. Among these techniques, ANN is mostly used for feature selection and classification. It is well-known massively parallel computing model that exhibits excellent behavior in input-output mapping and in resolving complex artificial intelligence problems in classification tasks. However, ANN is a black box in nature that doesn't give any description of how the classification or the operation is done. Moreover, due to the presence of imprecise information, ambiguity or vagueness in input data, overlapping boundaries among classes and indefiniteness in defining features some uncertainties can arise at any stage of data classification task. The fuzzy logic ([11–13]) is very flexible in handling different aspects of uncertainties or incompleteness about real life situations. Both ANN and fuzzy logic are very adaptable in estimating the input–output relationships, in which

ANN deals with numeric and quantitative data while fuzzy logic handles symbolic and qualitative data. Neuro-fuzzy hybridization leads to a crossbreed intelligent system widely known as Neuro-Fuzzy System (NFS) ([14-17]) that exploits the best qualities of these two approaches efficiently. NFS combines the advantages of both ANN and fuzzy logic, which covers up each other's disadvantages. In such system, the knowledge gained by the network from the linguistic interpretation of data can be used to generate rules that are used for feature selection as well as classification.

The linguistic rules generated by the neuro-fuzzy system are more helpful for understanding and analysis of the features. To generate these rules, the input features need to be labeled with some symbolic representation called linguistic variables. The knowledge extracted from the data is combined with the linguistic variables for rule-based classification.

The neuro-fuzzy schemes proposed for linguistic feature selection and rule-based classification in ([24], [25]) are complicated as the structure of the network keeps on changing during the training phase. The neuro-fuzzy schemes [26] and [27] use fixed number of linguistic variables i.e. 3, {SMALL, MEDIUM, LARGE} for each feature. However fixing the equal number of linguistic variables for each feature is not a correct way of interpreting features. Therefore, the neuro-fuzzy model [28] has determined significant linguistic variables for each feature by dynamic clustering instead of fixing the number of linguistic variables. The model has established a criterion for dynamic clustering in such a way that generates a huge number of clusters which results the huge number of linguistic variables. But excessive clusters need unnecessary computational effort and provide poor interpretation of the rules. This problem of a huge number of linguistic variables is considered and lightly resolved in [30] by the fuzzy union and Golden section search (GSS), however, the accuracy of classification tasks drops using the fuzzy union. The proposed model resolves the same problem with a different approach. The proposed model uses a modified equation of the threshold in the dynamic clustering algorithm, that reduces the number of linguistic variables and the model also uses GSS to determine fixed significant number of linguistic variables rather than a flexible (different) number of linguistic variables in classification rules.

2. Related Studies

Technique for classification tasks using neuro-fuzzy has been continually evolving to ensure efficient classification. There are many neuro-fuzzy techniques for feature selection and classification. **Li et al.** [18] have selected the important features and calculated the degree with which input pattern match the memory vector using maximum fuzzy entropy interpretation. **Kulkarni et al.** [19] have computed feature wise membership of each pattern to its class which is useful in classification when the classes are overlapping and ill-defined. **Basak et al.** [20] have described a neuro-fuzzy methodology which involves connectionist minimization of a fuzzy feature evaluation index with unsupervised training. **Yang et al.** [21] have used one triaxial accelerometer to acquire subjects' acceleration data and train the neuro-fuzzy classifier to distinguish different activities/movements. **Chen et al.** [22] have proposed Quantum Neuro-Fuzzy Classifier (QNFC) model which combines the compensatory-based fuzzy reasoning method with the traditional Takagi–Sugeno–Kang (TSK) fuzzy model. The compensatory-based fuzzy reasoning method uses adaptive fuzzy operations of neuro-fuzzy systems that can make the fuzzy logic system more adaptive and effective. **Azar et al.** [23] have presented linguistic hedges neuro-fuzzy classifier with selected features (LHNFCSF) for dimensionality reduction, feature selection and classification. **Chakraborty et al. (**[24], [25]) have integrated feature analysis and system identification which enables online feature selection and also builds a fuzzy rule-based classifier. **Eiamkanitchat et al.** ([26], [27]) have developed a good classification model using less complicated rule for that. **Wongchomphu et al.** [28] have proposed a neuro-fuzzy system for classification using dynamic clustering, which is an extension of ([26], [27]). **Napook et al.** [30] have further extended [28] using adaptive dynamic clustering algorithm.

3. Proposed Neuro-fuzzy Model

The proposed neuro-fuzzy model is mainly divided into five main phases- preprocessing, transition, learning, linguistic selection and rule generation phases. The control flow diagram of the proposed model is given in fig. 1. The details of each phase are given below.

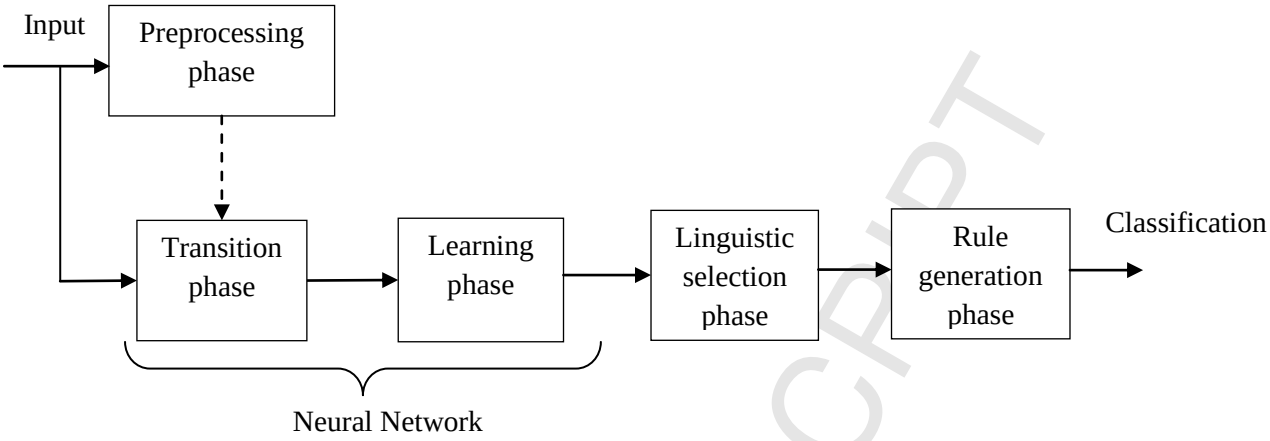

Fig 1. Proposed Neuro-fuzzy Model

3.1 Preprocessing Phase

This is the first phase of the proposed neuro-fuzzy model in which the number of linguistic variables of each input is determined. The significant linguistic variables are essential for proper interpretation of each input feature. The proposed Dynamic Clustering algorithm (DCA) is used for this purpose.

The detailed algorithm of DCA is as follows:

Suppose x_i represents the i^{th} input feature of all the data in the dataset. Each value in x_i is termed as a point.

Step 1

a) Begin by sorting the data points of feature x_i in ascending order.
b) The first point is assigned as cluster 1.
c) The consequence point is compared with the previous point
    i. If they belong to same class, the new sample is assigned to the cluster of the previous point.
    ii. If not, a new cluster is created in which the consequence point is assigned.
    iii. The step, c) is repeated for all the other remaining consequence points

Step 2

a) The centroid of each cluster is calculated.
b) Threshold (T_{th}) is calculated for each feature which is used to determine the existence of cluster.

$$T_{th} = \frac{\sum_{j=1}^{N} M_j}{N} + C_n \quad (1)$$

where, M_j: Number of members in cluster j

N : Number of clusters

C_n: Number of classes

Step 3

a) All the clusters which have points less than T_{th} are eliminated.

b) The points in the eliminated cluster are merged into cluster having the smallest distance from the eliminated cluster. The distance between clusters is the distance between centroid of clusters and is calculated as

$$d(c_j, c_k) = |c_j - c_k| \quad (2)$$

where, c_j: Centroid of the eliminated cluster.

c_k : Centroid of the other clusters

k ranges from 1 to the number of clusters generated in Step 1 except for k=j.

c) The above steps are repeated for the clusters having the points less than T_{th}.

Step 4

a) Mean (c) and variance (σ) of each cluster are calculated.

b) The Steps 1- 4 are repeated for all the other remaining features.

3.2 Transition Phase

This phase consists of fuzzification (Gaussian membership) and binary transformation process, which are carried out in Gaussian membership layer and binary transformation layer of the network structure respectively as shown in fig. 2. In the neural network the input layer acts as buffer for the input features. The input features are fed into the fuzzification (membership) layer to assign linguistic variables to each feature. The number of membership functions of each feature depends on the number of clusters generated for each feature in the preprocessing phase. The Gaussian membership function is used to assign membership value of each linguistic feature. The mean and standard deviation of each cluster are used in the Gaussian membership function. The membership of j^{th} linguistic feature of input feature x_i is given by equation (3).

$$\mu_{ij} = \begin{cases} 0, & \text{if } \sigma_{ij} = 0 \text{ and } x_i \neq c_{ij} \\ e^{-\left(\frac{1}{2}\frac{(x_i - c_{ij})^2}{{\sigma_{ij}}^2}\right)}, & \text{if } \sigma_{ij} \neq 0 \\ 1, & \text{if } \sigma_{ij} = 0 \text{ and } x_i = c_{ij} \end{cases} \quad (3)$$

where x_i is the input feature and σ_{ij} and c_{ij} are the standard deviation and mean respectively of the j^{th}cluster of the i^{th} feature. The weights of the links between the input layer and fuzzification layer are always unity.

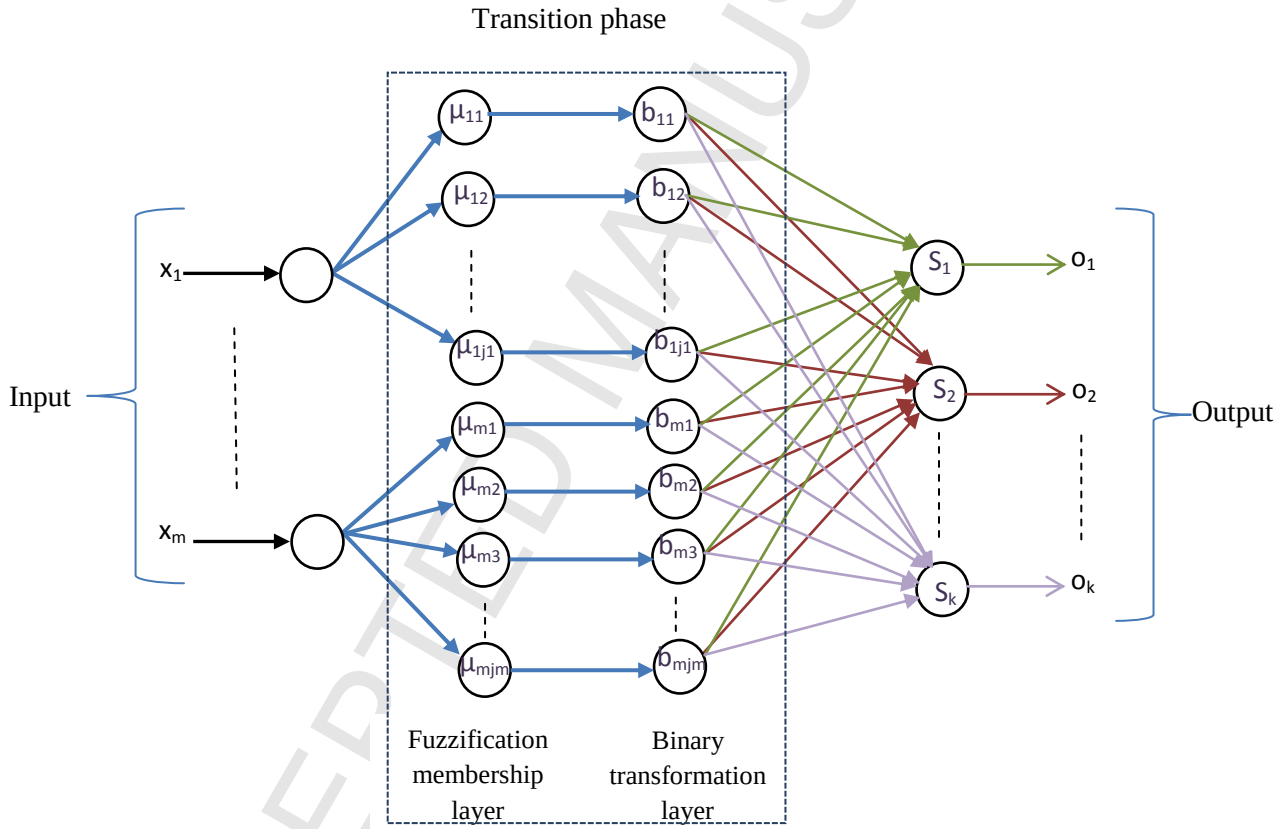

Fig 2. The network structure

The next layer in the transition phase is the binary transformation layer in which membership values are transformed into binary values. The equation (4) is used for binary transformation.

$$b_{ij} = \begin{cases} 1, & if\ \mu_{ij} \text{ is maximum among linguistic variables of the } i^{th} \text{feature} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where μ_{ij} is the membership of j^{th} linguistic variable of i^{th} input feature. The connection links between the Gaussian and binary transformation layers are also unity. The number of nodes in the binary transformation layer is also equal to the number of clusters generated for each feature. The results of binary layer are fed into the output layer where classification is done.

3.3 Learning Phase

The binary transformation layer of the transition process acts as input in this phase. The sigmoid function is used as activation function in the output layer which is given in equation (5). The number of nodes in the output layer is equal to the number of classes in a dataset.

$$O_k = \frac{1}{(1 + e^{-S_k})}, \text{ where } S_k = \sum_{i=1}^{m} \sum_{j=1}^{N_i} w_{kji} b_{ij} \tag{5}$$

In the above equation, m is the number of features, N_i is the number of clusters formed for each feature i, w_{kji} is the weight between the node k of the output layer and node j of the binary transformation layer which is linked with feature i. The error of the output node k at iteration z is given by equation (6).

$$e_k(z) = D_k(z) - O_k(z) \tag{6}$$

where $e_k(z)$ is the error of output node, $D_k(z)$ is the desired output and $O_k(z)$ is actual output. The delta value of the output node is given by equation (7).

$$\delta_k(z) = e_k(z)(1 - O_k)O_k \tag{7}$$

The weights are updated by equation (8).

$$w_{kji}(z + 1) = w_{kji}(z) + \eta \delta_k(z) b_{ij} \tag{8}$$

where, η is the learning rate.

3.4 Linguistic Number in Rule

The number of linguistic variables in the rules plays important roles in the easy interpretation of the neuro-fuzzy system and in achieving better accuracy. So, to generate the fixed and significant

number of linguistic variables in the rules Golden section search (GSS) is used. The algorithm of GSS is given below.

Step 1. Firstly the boundary x_l (lower) is initialized as zero and x_u(upper) is initialized as a summation of all weights.

Step 2. Then two intermediate points x_1 and x_2 are determine using equation (9) and (10).

$$x_1 = x_l + d \tag{9}$$

$$x_2 = x_u - d \tag{10}$$

$$\text{where,} \qquad d = \frac{\sqrt{5}-1}{2}(x_u - x_l)$$

Step 3. The weights at positions x_1 and x_2 are rounded to calculate $f(x_1)$ and $f(x_2)$ given by equation (11).

$$f(x) = \sum_{j=1}^{c_n} A_j, \qquad A = \begin{cases} 1, & \text{if } b_{nk} = 1 \\ 0, & \text{othewise} \end{cases} \tag{11}$$

Step 4. The Steps 2 and 3 are repeated till $f(x_1)$= $f(x_2)$. Otherwise on the basis of $f(x)$ new positions are calculated. If $f(x_1) > f(x_2)$, then set $x_u = x_1$, else if $f(x_1) < f(x_2)$, then set $x_l = x_2$,. The final boundary of the search is $[x_1, x_2]$. The number of linguistic features in the classification rule is given by number of weights.

3.5 Rules Generation

The IF-THEN rule is generated for classification tasks. The nodes in the binary transformation layer represent the premise part and the nodes in the output layer represent the consequence part of the rules. The efficiency of the rules not only depends upon the important linguistic features but also depends upon an exact number of linguistic features in the rules. The weights of the links between the binary transformation layer and output layer give the importance of linguistic features for rule-based classification. The binary transformation layer also helps in selecting the linguistic variables, as the highest linguistic variable is assigned '1' and all the others remaining linguistic variables are assigned '0' for an input feature. After fixed and significant linguistic

features are selected the rules are generated. In most of the data-driven rule-based systems, noises are present because of that the inconsistent rules may exist. Inconsistent rules are those which have same premise but different consequences. The inconsistent rules are pruned from the rule-based system and then final rule-based system is used for classification tasks.

3.6 Description of the Proposed Model

The transition and classification phases of the proposed model are same as [28] and [30]. The modifications are done in the preprocessing phase and in number of linguistic feature selection for classification rules using GSS. One major problem in [28] is the number of clusters formed in the preprocessing phase. This phase determines the linguistic variables of each feature and the number of linguistic variables is directly dependent on the number of clusters generated by DCA. As the number of clusters generated is very large, a huge number of linguistic variables are generated for each feature. As the number of linguistic variables increases, the interpretability of the system reduces i.e. it becomes hard for human beings to understand the system [29]. Hence in [30] the number of linguistic variables is reduced significantly by the fuzzy union, however, this operation highly depends upon the nature of input feature values. Even though the number of clusters depends on the nature of input features, it is also governed by the number of points in each cluster i.e. the threshold of the cluster. This paper modifies the equation of threshold in [28] by adding the term C_n that represents the number of classes. The modified equation is given in equation (1).

The following scenario can also be considered related to [28], here the number of data points of i^{th} feature is 5 and number of classes is 2. The features are sorted in ascending order as shown in table 1.

Table 1. An example of data sample

Feature i	Class
A1 (5)	C1
A2 (7)	C2
A3 (9)	C1
A4 (10)	C2
A5 (13)	C1

So table 2 is obtained which gives the clusters of feature i.

Table 2. Cluster of data sample

Feature i	Class	Cluster
A1 (5)	C1	1
A2 (7)	C2	2
A3 (9)	C1	3
A4 (10)	C2	4
A5 (13)	C1	5

Then the threshold is calculated according to the equation given in [28] that is.

$$F_{th} = \frac{\sum_{j=1}^{N} M_j}{N} \tag{12}$$

The value of threshold is set to 1, so the element in each cluster remains unchanged and the number of clusters is equal to the number of features. In some other scenario the following can be obtained.

Table 3. Example of other data sample for clustering

Feature i	Class	Cluster
A1 A2	C1 C2	1
A3 A4	C1 C2	2
A5 A6	C1 C2	3

Here, each cluster contains elements from different classes. A cluster should be a collection of similar objects but here dissimilar objects are forming the cluster, not even a single similar object is present in each cluster. Therefore, it is not a good cluster. To improve the nature of cluster equation, equation (12) is modified as equation (1).

After the network is trained, the rules for the classification are extracted from the knowledge within the network. In the previous models, different numbers of linguistic variables have been experimented in the classification rules. In the proposed model, instead of using different numbers of linguistic variables, fixed significant number of it is generated for classification rules. This significant number of linguistic variables is determined by the GSS and it helps to improve the generalization and interpretability of the model.

3.7 An Illustrative Example

Liver disorder dataset taken from the University of California, Irvine (UCI) repository is used as an example to demonstrate feature selection and classification using the proposed model. The dataset consists of 6 features, 350 patterns and 2 classes. To measure the performance of the model 10-fold cross validation is used. Here 315 patterns (90% of data) are used for training and 35 patterns (10% of data) are used for testing in each fold.

A hidden layer neural network is trained using back-propagation algorithm. The algorithm uses learning rate 0.1 and runs for 300 epochs. The network architecture consists of 6 nodes in input layer; 207 nodes in each hidden layer i.e. Gaussian membership layer and binary transformation layer and 2 nodes in the output layer. The number of input nodes is same as the number of features. The number of nodes in each hidden layers is same and is determined by the dynamic clustering process. The number of nodes in the output layer is equal to the number of classes. In dynamic clustering process, number of linguistic variables used for each feature is shown in table 4.

Table 4. Number of linguistic variables of liver dataset

Feature	No. of linguistic variables
1	17
2	29
3	31
4	56
5	51
6	23

For one of the folds, after training the network the number of linguistic variables determined by GSS is 8, which are important linguistic variables for classification and are shown in table 5.

Table 5. Important linguistic variables

Class	8 Significant Linguistic variables							
1	3	5	2	1	149	179	127	192
2	6	4	149	201	66	36	77	8

Then the following rules are generated by the model.

If **f1** is **lv3 or f1** is **lv5 or f1** is **lv2 or f5** is **lv149 or f5** is **lv127 or f6** is **lv179 or f6** is **lv192** then **Class 1**

Else if **f1** is **lv6 or f1** is **lv4 or f5** is **lv149 or f6** is **lv201 or f3** is **lv66 or f2 is lv33 or f3 is lv77 or f1 is lv8** then **Class 2**

Else **no class**

Here, **f6** represents **6^{th} feature** of the dataset and **lv199** represents **199^{th} linguistic variables**.

In these rules there is inconsistency as premise, f5 is lv149, is same for both the consequent classes, after removing the inconsistent rule, the classification rules are as follows.

If **f1** is **lv3 or f1** is **lv5 or f1** is **lv2 or f5** is **lv127 or f6** is **lv179 or f6** is **lv192** then **Class 1**

Else if **f1** is **lv6 or f1** is **lv4 or f6** is **lv201 or f3** is **lv66 or f2 is lv33 or f3 is lv77 or f1 is lv8** then **Class 2**

Else **no class**

The accuracy for this fold of the liver disorder dataset is 60%.

4. Experiment and Results

The neuro-fuzzy classification models namely Enhance Neuro-Fuzzy (ENF) system [28], Adaptive Neuro-Fuzzy (ADCNF) system [30] and proposed neuro-fuzzy model are trained and tested on six datasets: breast cancer, diabetes, heart, liver disorder, and sonar taken from the UCI

machine learning repository and one dataset, swine flu taken from local hospitals and internet. All the parameters such as learning rate, epoch of the comparing models are kept same. 90% of each dataset is used as training set and remaining 10% as test set. 10-fold cross validation is used to find the simple overall accuracy of the models. And an arbitrary fold is taken from these 10 folds and accuracy, false positive rate, precision, recall, f-measure and Matthews correlation coefficient (MCC) are measured for that fold to draw performance comparisons amongst the neuro-fuzzy classifiers. All the experiments are done using MATLAB software (version R2012a).

4.1. Performance Measures

The test dataset is applied to the classifiers for performance evaluation. The performances of these models are then estimated based on different performance measures described below.

4.1.1. Confusion Matrix

The confusion matrix contains information about actual and predicted classifications done by a classifier. The performance of such classifier is commonly evaluated using the data in the matrix. table 6 presents the confusion matrix for a two class classifier with the following data entries: (a) True Positive (TP) is the number of 'positive' instances categorized as 'positive'. (b) False Positive (FP) is the number of 'negative' instances categorized as 'positive'. (c) False Negative (FN) is the number of 'positive' instances categorized as 'negative'. (d) True Negative (TN) is the number of 'negative' instances categorized as 'negative'.

Table 6: Confusion matrix

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Several standard terms have been defined for the 2 class matrix.

The *accuracy* is the proportion of the total number of predictions that are correct. It is determined using the equation (13).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

The *recall* or *true positive rat*e is the proportion of positive cases that are correctly identified, as calculated using the equation (14).

$$recall = \frac{TP}{TP + FN} \tag{14}$$

The *false positive rate* (*FP-rate*) is the proportion of negative cases that are incorrectly classified as positive, as calculated using the equation (15).

$$false\ positive\ rate = \frac{FP}{FP + TN} \tag{15}$$

Finally, *precision* (*P*) is the proportion of the predicted positive cases that are correct, as calculated using the equation (16).

$$precision = \frac{TP}{TP + FP} \tag{16}$$

In some scenarios high precision may be more important, while in other scenarios high recall may be more significant. However, in most types, we try to improve both values. The combined form of these values is called the f-measure, and usually expressed as the harmonic mean of both these values:

$$f - measure = \frac{2 * precision * recall}{precision + recall} \tag{17}$$

The Matthew's correlation coefficient (MCC) is used in machine learning as a measure of the quality of binary (two-class) classifications. The MCC is, in essence, a correlation coefficient between the observed and predicted binary classifications; it returns a value between −1 and +1. A coefficient of +1 represents a perfect prediction, 0 represents no better than random prediction and −1 indicates total disagreement between prediction and observation.

The MCC can be calculated directly from the confusion matrix using the equation (18).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{18}$$

4.2 Result and Analysis

The papers [28] and [30] have proven a highly accurate generalization among various algorithms. Therefore, the performance of the proposed model is only compared with ENF [28] and ADCNF [30] using 6 datasets. The details of the datasets are described in table 7.

Table 7: Datasets used in experiments

Dataset	Size	Feature	Feature type	No. of class
Breast cancer	680	9	Real	2
Heart	270	13	Categorical, Real	2
Diabetes	768	8	Integer, Real	2
Liver disorder	345	6	Categorical, Integer, real	2
Swine flu	250	11	Integer	2
Sonar	208	50	Real	2

The accuracy of the classification tasks using 10-fold cross validation for ENF and ADCNF without GSS is shown in table 8 with different numbers of linguistic variables i.e. 15,10,5. Table 9 shows the comparison of the accuracy of the proposed model using 10-fold cross validation with that of ADCNF with GSS and also with the best accuracy among different linguistic variables of ENF and ADCNF without GSS given in table 8.

Table 8. Accuracy of ENF and ADCNF without GSS with different linguistic variables

Dataset	Model	15 linguistic variables	10 linguistic variables	5 linguistic variables
Breast cancer	ADCNF [30]	88.9	90.1	90.1
	ENF [28]	92.3	90.7	88.1
Heart	ADCNF [30]	59.2	53.3	48.5
	ENF [28]	59.2	56.7	53.7
Diabetes	ADCNF [30]	71.1	69.5	66.4
	ENF [28]	70.5	67.1	63.3
Liver disorder	ADCNF [30]	53.8	53.3	50.5
	ENF [28]	53.8	50.3	47.6
Swine flu	ADCNF [30]	72.4	78.4	77.6

	ENF [28]	88.4	85.6	76.4
Sonar	ADCNF [30]	62.5	59.3	59.6
	ENF [28]	63.2	56.8	52.1

Table 9. Accuracy comparison of the proposed model with other models

Dataset	**ENF [28]**	**ADCNF without GSS [30]**	**ADCNF with GSS [30]**	**Proposed**
Breast cancer	92.3	90.1	85.6	90.2
Heart	59.2	59.2	56.3	67
Diabetes	70.5	71.1	67	74.1
Liver disorder	53.8	53.8	53.3	54.3
Swine flu	88.4	78.4	79	90
Sonar	63.2	62.5	63.3	67.1

It is observed from table 9 that the proposed model gives better accuracy than ENF, ADCNF without GSS and ADCNF with GSS for heart, swine flu, sonar, diabetes and liver disorder datasets. For the breast cancer dataset the accuracy of the proposed model is better than ADCNF with GSS and slightly better than ADCNF without GSS but is slightly less than ENF. It can be concluded that the proposed model is better than ENF, ADCNF without GSS and ADCNF with GSS. Graphical representation is also shown in fig 3 for more clarity.

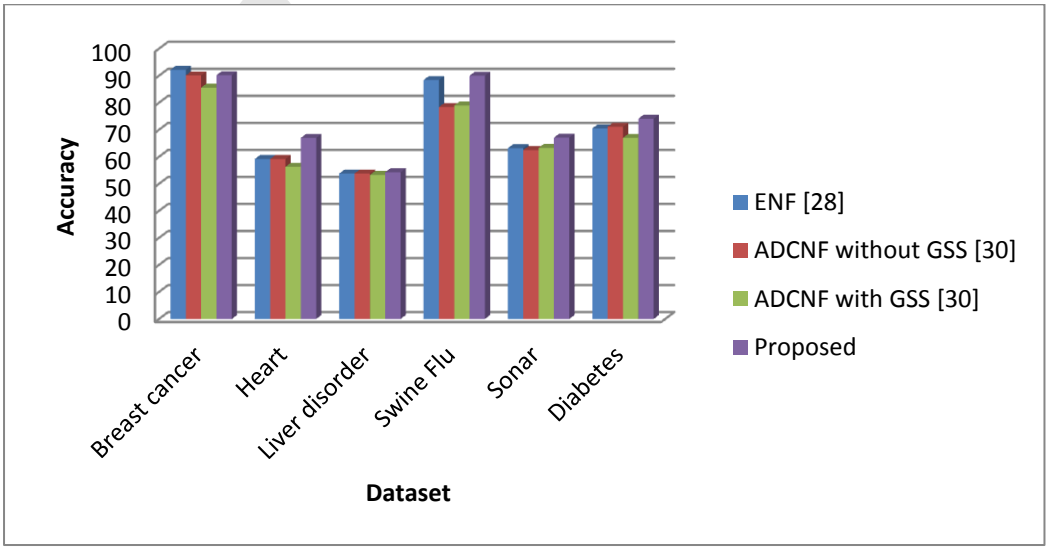

Fig. 3 Accuracy comparison using 10-folds for various models

The number of clusters generated by ENF [28], ADCNF [30] and the proposed models is also shown in fig. 4. The y-axis represents the total number of clusters formed for each dataset. The number of clusters generated in the proposed model is highly reduced with respect to ENF in all the datasets. However, the number of clusters generated in ADCNF is less than that of proposed model for breast cancer, heart and swine flu datasets. But the accuracy of the proposed model is better than ADCNF in all the datasets.

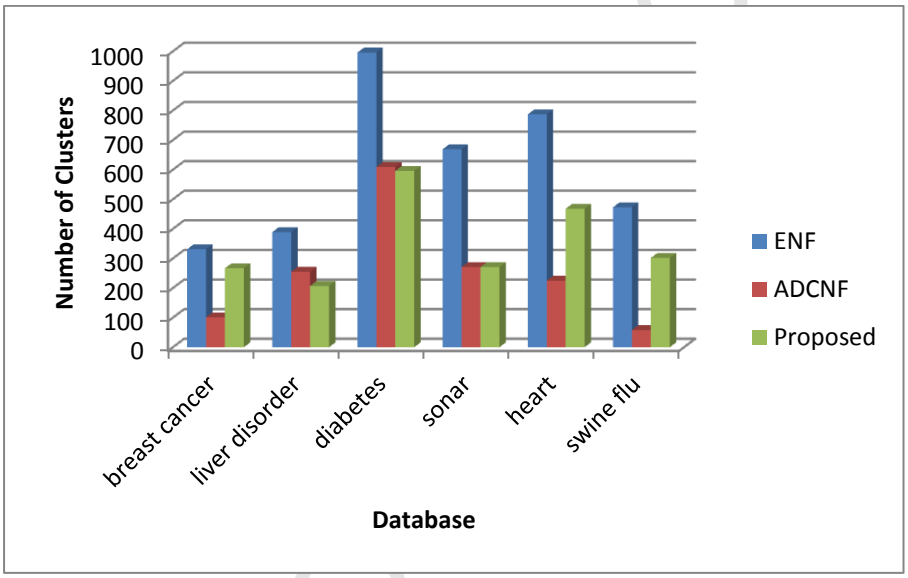

Fig. 4 Comparison of number of clusters generated

4.3 Performance Analysis

Results of one fold are analyzed with different performance measures, where 90% of dataset is used as training set and 10% used as test set for evaluating the performance of the proposed, ENF and ADCNF with and without GSS models. To measure the performance of a classifier, higher value for accuracy, precision, recall, f-measure and MCC is expected and lower value for false positive rate.

Breast cancer dataset consists of 680 samples, 9 features and 2 classes. The confusion matrix of breast cancer dataset is shown in table 10 for the classifiers.

Table 10: Confusion matrix of breast Cancer

Proposed model		
Actual	Observed	
	Class 1 (positive)	Class 2 (negative)
Class 1 (positive)	32	3
Class 2 (negative)	2	31
ENF [28]		
Actual	Observed	
	Class 1 (positive)	Class 2 (negative)
Class 1	29	2
Class 2	5	32
ADCNF without GSS[30]		
Actual	Observed	
	Class 1 (positive)	Class 2 (negative)
Class 1 (positive)	21	1
Class 2 (negative)	14	20
ADCNF with GSS[30]		
Actual	Observed	
	Class 1 (positive)	Class 1 (positive)
Class 1 (positive)	21	1
Class 2 (negative)	16	33

The accuracy, precision, FP-rate, recall, f-measure and MCC of various models for the breast cancer dataset are shown in table 11. The accuracy, MCC, precision and FP-rate of the proposed model are better than that of ENF, ADCNF with and without GSS. The recall and f-measure of the proposed model are much higher than that of ENF and ADCNF with GSS however, these performance measures are slightly less than that of ADCNF without GSS. Hence, it is clear that the proposed model is a better classifier than other existing models for breast cancer dataset.

Table 11: Performances on breast cancer dataset

	Accuracy	Precision	FP-rate	Recall	f-measure	MCC
Proposed	92%	94%	6%	50%	65%	.853
ENF [28]	65%	33%	41%	25%	28%	.386
ADCNF without GSS[30]	91%	94%	6%	51%	67%	.825
ADCNF with GSS[30]	76%	56%	32%	39%	46%	.581

The accuracy, precision, FP-rate, recall, f-measure and MCC of various models for the liver disorder dataset are shown in table 12. Liver disorder dataset consists of 345 samples, 6 features and 2 classes. In all the performance measures, the proposed model is better than all the existing models i.e. ENF, ADCNF with and without GSS. Hence, it is clear that the proposed model is a better classifier than other models for liver disorder dataset.

Table 12: Performances on liver disorder dataset

	Accuracy	Precision	FP-rate	Recall	f-measure	MCC
Proposed	88%	80%	3%	13%	22%	.610
ENF [28]	85%	60%	7%	10%	18%	.461
ADCNF without GSS[30]	82%	60%	7%	10%	18%	.404
ADCNF with GSS[30]	82%	60%	7%	10%	18%	.404

The accuracy, precision, FP-rate, recall, f-measure and MCC of various models for the heart dataset are shown in table 13. Heart dataset consists of 270 samples, 13 features and 2 classes. The accuracy, FP-rate and MCC of the proposed model are better than all other existing models. The precision, recall and f-measure of the proposed model are also better than ADCNF with GSS however these performance measures are slightly less than that of ENF and ADCNF without GSS. It is clear that the proposed model is a better classifier than other models for heart dataset.

Table 13: Performances on heart dataset

	Accuracy	Precision	FP-rate	Recall	f-measure	MCC
Proposed	77%	53%	30%	33%	41%	.614
ENF[28]	54%	62%	36%	47%	53%	.114
ADCNF without GSS[30]	54%	62%	36%	47%	53%	.114
ADCNF with GSS[30]	40%	8%	52%	9%	8%	-.299

The accuracy, precision, FP-rate, recall, f-measure and MCC of the three models for the diabetes dataset are shown in table 14. Diabetes dataset consists of 768 samples, 8 features and 2 classes. The accuracy, FP-rate and MCC of the proposed model are better than other models. Precision of the proposed model is also better than ADCNF with and without GSS but is same with that of ENF. The recall and f-measure of the proposed model are better than ADCNF with GSS, however slightly less than ENF and ADCNF without GSS. It is clear that the proposed model is a better classifier than other models for diabetes dataset.

Table 14: Performances on diabetes dataset

	Accuracy	Precision	FP-rate	Recall	f-measure	MCC
Proposed	73.8%	59%	23%	30%	40%	.404
ENF[28]	59.5%	59%	30%	38%	46%	.184
ADCNF without GSS[30]	59%	56%	30%	36%	43%	.174
ADCNF with GSS[30]	54%	34%	37%	24%	28%	.017

The accuracy, precision, FP-rate, recall, f-measure and MCC of various models for the swine flu dataset are shown in table 15. Swine flu dataset consists of 250 samples, 11 features and 2 classes. The accuracy, precision, FP-rate, and MCC of the proposed model are better than other existing models; however the recall and f-measure are slightly less than ADCNF with GSS but better than ENF and ADCNF without GSS. By considering all the performance measures it is clear that the proposed model is a better classifier than other models for swine flu dataset.

Table 15: Performances on swine flu dataset

	Accuracy	Precision	FP-rate	Recall	f-measure	MCC
Proposed	84%	84%	15%	47%	60%	.679
ENF [28]	76%	50%	31%	31%	38%	.584
ADCNF without GSS[30]	76%	50%	31%	31%	38%	.584
ADCNF with GSS[30]	80%	83%	17%	50%	63%	.602

The accuracy, precision, FP-rate, recall, f-measure and MCC of the three models for the sonar dataset are shown in table 16. Sonar dataset consists of 205 samples, 50 features and 2 classes. All the performance measures of the proposed model are same with that of ADCNF without GSS. The accuracy and MCC of the proposed model are better than ENF and ADCNF with GSS. The precision of the proposed model is less than ENF but better than ADCNF with GSS. The FP-rate of the proposed model is same with that of ENF and ADCNF without GSS however is less than ADCNF with GSS. The recall and f-measure of the proposed model are slightly less than ENF and ADCNF with GSS. By considering all the performance measures it is clear that the proposed model for sonar dataset is moderate.

Table 16: Performances on sonar dataset

	Accuracy	Precision	FP-rate	Recall	f-measure	MCC
Proposed	71%	70%	50%	70%	70%	.411
ENF [28]	68%	82%	50%	82%	82%	.217
ADCNF without GSS[30]	71%	70%	50%	70%	70%	.411
ADCNF with GSS	55%	59%	88%	91%	71%	-0.057

5. Conclusion

An improved neuro-fuzzy model for feature selection and classification is proposed which resembles [28] and [30]. The novelty of the proposed model lies in determining the number of linguistic variables for each feature using dynamic clustering process and also in determining a significant number of linguistic variables in rules with Golden section search. The proposed

model reduces the number of linguistic variables drastically and also generates fixed significant number of linguistic variables in the rules, which results better performance than [28] and [30]. The superiority of the proposed model is demonstrated with 6 datasets. The accuracy calculated by 10-fold cross validation finds that the proposed model has better performance than already proven neuro-fuzzy systems for classification. The different performance parameters are also used to draw comparison between the proposed model, and [28] and [30] neuro-fuzzy models. The performance analysis has proved that the proposed model is superior to the existing models. However, the accuracy of the proposed model and other neuro-fuzzy models ([26-28], [30]) depends on the order in which rules are applied for classification tasks. In future this can be overcome by determining significant generalized order of rules for classification tasks. The proposed model can be used in any binary classification tasks with great understandability and accuracy.

References

[1] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[2] M. Dash and H. Liu, "Feature selection for classification", Intelligent data analysis, vol. 1, no. 3, pp. 131–156, 1997.

[3] K. Anand, G.V. Karthik, S. Alexandra, S. Li, S. Krishnan, and R. Arifur, "A Novel Cloud Intrusion Detection System Using Feature Selection and Classification", International journal of intelligent information Technologies, vol. 11, no. 4, pp. 1-15, 2015.

[4] J. Hua, W. D. Tembe and E. R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data", Pattern Recognition, Elsevier, vol. 42, no. 3, pp. 409–424, 2009.

[5] R. Battiti, "Using mutual information for selecting features in supervised neural net learning", IEEE Trans Neural Networks, vol. 5, no. 4, pp. 537-550, 1994.

[6] G. Chandrashekar and F. Sahin, "A survey on feature selection methods", Computers and Electrical Engineering, vol. 40, no. 1, pp.16–28, 2014.

[7] Y. Sun, C. Babbs and E. Delp, "A comparison of feature selection methods for the detection of breast cancers in mammograms: adaptive sequential floating search vs. genetic algorithm", Conf proc IEEE eng med biol soc, vol. 6, pp. 6532 – 6535, 2006.

[8] W. Puch, E. Goodman, M. Pei, L. Chia-Shun, P. Hovland and R. Enbody, "Further research on feature selection and classification using genetic algorithm", In International conference on genetic algorithm, pp. 557–564, 1993.

[9] I. Inza, P. larranaga and B. Sierra, "Feature selection by Bayesian networks: a comparison with genetic and sequential algorithm", In International Journal of Approximate Reasoning, vol. 27, no. 2, pp. 143-164, 2001.

[10] S. Ledesma, G. Cerda, G. Avina, D. Hernandez, and M. Torres, "Feature Selection Using Artificial Neural Networks", In MICAI 2008: Advances in Artificial Intelligence, vol. 5317, pp. 351-359, 2008.

[11] L. A. Zadeh, "Fuzzy sets", Inf Control, vol. 8, no. 3, pp. 338–353, 1965.

[12] B. Liu, "Uncertainty theory: an introduction to its axiomatic foundations", Berlin: Springer Verlag, vol. 154, pp. 191–346, 2004.

[13] D. Dubois and H. Prade, "Fuzzy sets and systems: theory and applications", New York: Academic Press, vol. 144, pp. 255–348, 1980.

[14] S. Ghosh, S. Biswas, D. Sarkar and P. P. Sarkar, "A novel Neuro-fuzzy classification technique for data mining", Egyptian Informatics Journal, vol. 15, no. 3, pp. 129–147, 2014.

[15] A. T. Azar and A. E. Hassanien, "Dimensionality reduction of medical big data using neural-fuzzy classifier", Soft computing vol. 19, no. 4, pp. 1115-1127, 2015.

[16] D. Chakraborty and N. R. Pal, "Integrated Feature Analysis and Fuzzy Rule-Based System Identification in a Neuro-Fuzzy Paradigm", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol.31, no. 3, pp. 391-400, 2001.

[17] Z.G. Vranesic, "Multiple-Valued Logic: An Introduction and Overview," IEEE Transactions on Computers, vol. C-26, no. 12, pp. 1181-1182, 1977.

[18] R. P. Li, M. Mukaidono and I. B. Turksen, "A fuzzy neural network for pattern classification and feature selection", Fuzzy Sets and Systems, vol. 130, no. 1, pp. 101–108, 2002.

[19] U. V. Kulkarni and S. V. Shinde "Hybrid Fuzzy Classifier Based on Feature-wise Membership Given by Artificial Neural Network", In fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1-7, 2013.

[20] J. Basak, R. K. De and S. K. Pal, "Unsupervised feature selection using a neuro-fuzzy approach", Pattern Recognition Letters, vol. 19, no. 11, pp. 997-1006, 1998.

[21] J. Y. Yang, Y. P. Chen, G. Y. Lee, S. N. Liou and J. S. Wang, "Activity Recognition Using One Triaxial Accelerometer: A Neuro-fuzzy Classifier with Feature Reduction", Entertainment Computing – ICEC vol. 4740, pp. 395-400, 2007.

[22] C. H. Chen, C. J. Lin and C. T. Lin, "An efficient quantum neuro-fuzzy classifier based on fuzzy entropy and compensatory operation", Soft Computing, vol. 12, no. 6, pp. 567-583, 2008.

[23] A. T. Azar and A. E. Hassanien, "Dimensionality reduction of medical big data using neural-fuzzy classifier", Soft computing vol. 19, no. 4, pp. 1115-1127, 2015.

[24] D. Chakraborty and N. R. Pal, "Integrated Feature Analysis and Fuzzy Rule-Based System Identification in a Neuro-Fuzzy Paradigm" Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol.31, no. 3, pp. 391-400, 2001.

[25] D. Chakraborty and N. R. Pal, "A Neuro-Fuzzy Scheme for Simultaneous Feature Selection and Fuzzy Rule-Based Classification," IEEE Transactions on Neural Networks, vol. 15, no. 1, pp. 110-123, 2004.

[26] N. Eiamkanitchat, T. Umpon and S. Auephanwiriyakul, "A Novel Neuro-Fuzzy Method for Linguistic Feature Selection and Rule-Based Classification" in 2nd International Conference on Computer and Automation Engineering (ICCAE), pp. 247-252, 2010.

[27] N. Eiamkanitchat and N. Theera-Umpon, "Colon tumor Microarray classification Using Neural Network with Feature selection and rule based Classification," Advances in Neural Network Research and Applications, vol. 67, pp. 363-372, 2010.

[28] W. Poonari and N. Eiamkanitchat, "Enhance Neuro-Fuzzy System for Classification Using Dynamic Clustering," 4th Joint International Conference on Information and Communication Technology, Electronic and Electrical Engineering (JICTEE), pp. 1-6, 2014.

[29] Y. Jin, "Fuzzy Modeling of High-Dimensional Systems: Complexity Reduction and Interpretability Improvement," IEEE Transactions on Fuzzy Systems, vol. 8, no. 2, pp. 212- 221, 2000.

[30] P. Napook and N. Eiamkanitchat, "The adaptive dynamic clustering neuro-fuzzy system for classification," Information Science and Applications, Springer, vol. 339, pp. 721-728, 2015.