A Binary Greedy Algorithm for Feature Selection and Classification

Vimal Kumar Dubey[1], Abhishek Dubey[2], Amit Kumar Saxena[3]

[1]*Department of Computer Science and Information Technology, Guru Ghasidas Vishwavidaylaya, Bilsapur, India, 495009*

[2]*Information Technology Department, Salalah College of Technology, Salalah, Sultanate of Oman*

[3]*Department of Computer Science and Information Technology, Guru Ghasidas Vishwavidaylaya, Bilsapur, India, 495009*

Abstract: In this paper, a novel Wrapper based feature selection algorithm is proposed. A set of various combinations of features denoted as 0 or 1 is taken in a population. The goodness of a feature set in this population is measured as its classification accuracy computed by some classifier. The best feature set is saved and removed from the original population. Now the remaining feature sets are forced to match and replace one of their features (starting from one end of the feature set) with the best feature set in the previous iteration, saved separately. The technique is repeated for a given number of iterations and every time the best feature set is saved and separated. The best feature subset at the end of the algorithm is taken as the reduced feature set and it is applied on benchmark datasets. Results obtained by the proposed method on nine benchmark datasets taken from UCI Repository show that proposed method performs better on six datasets out of nine datasets than those obtained by other reported methods.

Keywords: Data mining, Feature Selection, Evolutionary Computation, Greedy Approach

1. Introduction

In the recent decades, researchers have focused on feature selection techniques with more intensity. The reason is that the increasing sizes of databases have become a problem for scientists and researchers who are engaged in mining useful and interesting knowledge from these databases. Classification (or prediction) is an indispensable part of data mining [1], machine learning [2] or pattern recognition [3]. A good classifier is capable to predict the classes of the unknown patterns and thus produces good classification accuracy. Higher is the accuracy, better will be the prediction for classification models like Support Vector Machines [4], Naive Bayes [5], Artificial Neural Networks [6] and others. Classification accuracy can be increased if non-redundant, relevant and noise free dataset is used for learning. On the contrary, if irrelevant, redundant and noisy features are present in the dataset, it will decrease the classifier performance (accuracy commonly) and often it is termed as **Curse of Dimensionality** [7]. Removing irrelevant, redundant and noisy features is termed as Dimensionality Reduction [8, 9, 10]. Feature selection [8] and feature extraction [9] are two well-known methods applied for the dimensionality reduction problem. Processing the existing features of the dataset to obtain new features is termed as feature extraction while selection of a subset of features from existing set of features without a single extra effort is known as feature selection. In this paper, a novel method is proposed to achieve feature selection in databases.

This paper is organized into following sections. Section -2 is a literature review. Some preliminaries about the terms used in the paper at later stages viz SVM, KNN is given in section -3. The proposed method is explained via algorithms and model in Section -4. Section -5 lists and briefly explains datasets used in the experiment. The details of experiments are presented in Section -6. Section -7 contains results derived from proposed method on listed datasets with discussions. Section -8 concludes the paper with the future scope.

2. Literature Review

An unsupervised feature selection algorithm is developed using an improved version of a recently developed Differential Evolution (DE) technique called MoDE (Modified Differential Evolution). One of the parameters of MoDE decays very fast in original DE algorithm so that its effective range is very narrow at a later stage. This limitation is overcome in MoDE. It is applied for feature selection on various datasets available at UCI [12] repository [11]. Fuzzy–Rough Feature Selection model with Shuffled Frog Leaping (SFL) Algorithm is developed. The SFL Algorithm is a meta-heuristic algorithm and uses the concept of mems evolution. This is used with Multi-Tree GE for feature selection and classification on various datasets [13]. Saxena et al proposed an unsupervised feature selection algorithm using Sammon's stress function and evolutionary method [14]. Jiang et al [17] found the problem with correlation measure. They found that features can be continuous or discrete, so they present a novel correlation (similarity) measure called ECMBF approach to filtering the features. Sparse discriminative

feature selection algorithm is proposed by yan and yang [18].

1. Preliminaries

This section briefly explains some information.

Support Vector Machine [4] is a classifier which classifies the patterns into only two classes. A Support vector machine (SVM) classifies data by finding the best hyper plane that separates all data points of one class from those of the other class. The best hyper plane for an SVM means the one with the largest margin between the two classes. Margin means the maximal width of the slab parallel to the hyper plane that has no interior data points. The support vectors are the data points that are closest to the separating hyper plane; these points are on the boundary of the slab. RBF kernel based SVM, MLP kernel based SVM, Linear SVM and polynomial kernel base SVM are some approaches for classification of data patterns. **K- Nearest Neighbour [1]**, this algorithm (KNN) used for the classification of data items based on distance measure. This is widely used for real data classification. K-Nearest Neighbour (KNN) is a supervised learning algorithm and treats as a straight forward classifier. It is a type of instance-based learning method or lazy learning method. **Classification measures,** Performance of classification model is measured using some technique called classification measures. Some measures are Accuracy, sensitivity, specificity. Below is a way to calculate such measures.

Classification measures

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{..... (2)}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad \text{.... (3)}$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad \text{.... (4)}$$

where

TN:Denotes number of negative patterns classified as negative.

TP: Denotes number of positive patterns classified as positive.

FP: Denotes number of negative patterns declared positive.

FN:Denotes number of positive patterns declared negative.

2. Proposed approach

In this paper, we proposed Binary Greedy Algorithm for feature selection. It is also a wrapper-based feature selection technique. In this algorithm, a set of points first initialised for possible solution. Let call each solution point as chromosomes (popular term hence used). Representation of chromosome is shown in figure 2. Each chromosome will have *N* number of components. Here *N* i.e. the number of component is equal to the number of features. Each component can take value either zero (0) or one (1). Each component represents the respective feature of the dataset. If the component value is 1 then the respective feature is selected otherwise not selected. Each chromosomes goodness is measured using fitness function. Fitness function can be accuracy obtained by a classifier, error or something . After calculating goodness (fitness) value of each chromosome, a chromosome whose goodness is the best, will be taken as possible candidate solution. Then each other chromosomes mutate their component to become like candidate solution. Chromosome mutation is performed by move operation. This process iterates till a termination criteria not found.

Move operation: Movement of solution point's results into the convergence of the algorithm. To understand movement operation let there are four points p1, p2, p3, p4 in the solution space and each solution point is represented by 10 bits. Let there is a classifier, which calculates the goodness measure of each solution point. Suppose after calculation of goodness, point p1 is having the highest goodness value. Then all other points except p1 mutate their component (bits). It means other point's p2, p3 and p4 changes their bits to become like copy p1. Here each solution points copy a fixed number of bits from possible candidate solution. There is be fixed criteria that every time each solution point will mutate only *S* number of bits. Let S number of bits is mutated by solution points. S may depend upon a parameter R a small number say R (value between 0 and 1) and the number of features. The product of R and number of features *N* determines *S* i.e. $S=RXN$. Algorithm and model for the proposed method is shown by the figure1 and figure 2. Fitness function plays an important rule. A good fitness function results in better classification accuracy. As here we have to maximize classification accuracy, sensitivity, and specificity, we cannot choose classification accuracy as a fitness measure. So here fitness measure is maximum of Sensitivity*Specificity. Classification accuracy obtained by optimization of this fitness function would be more reliable as classifier model is able to recognize all the classes at max extent.

Greedy Algorithm for feature selection

Input : Dataset

Output : Highest Accuracy, Number of features, features id and other parameters

Get datasets
Divide the complete dataset in user defined folds
Initialize number of chromosomes (solution space), component and termination criteria
Initialize Number of bits to mutate i.e. *S*
Initialize chromosomes
Repeat following step till termination criteria not found
Calculate goodness measure of each chromosomes
using cross-validation and classifier.
Sort chromosomes in decreasing order of goodness measure.
For : Second chromosome to last chromosomes in sorted list
Mutate bits of chromosomes to become like copy of best
chromosomes. (It results into creation of new
chromosomes (solution point) whose S bit will be
Similar to best solution.)
End of for
// end
Outputs are Accuracy, Number of features, and others.

Figure 1: algorithm of the proposed method

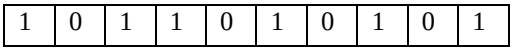

Figure 2: Binary representation of chromosomes

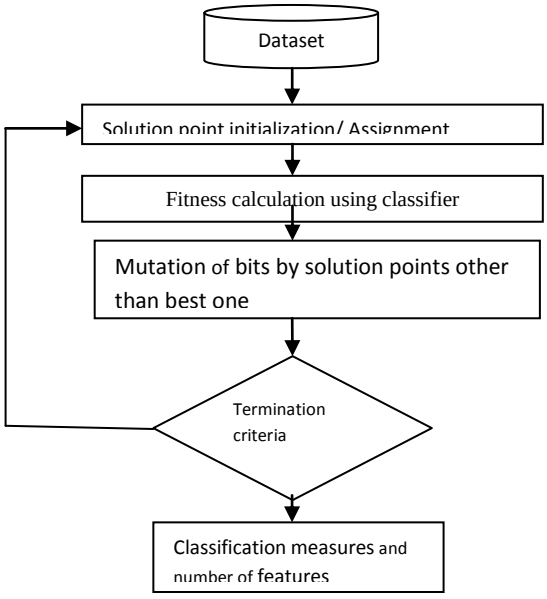

Figure 3: Shows model of the proposed method

1. Datasets

The datasets used in this experiment is listed in table1. All dataset is taken from UCI machine learning repository [12]. All dataset is of two class one class is represented by -1 and another one is represented by +1. Summary is given in table 1.

Table 1: Datasets Description

Dataset ID	Datasets	Classes	Features	Patterns
D1	Australian Credit	2	14	690
D2	German Credit	2	20	1000
D3	Heart STALOG	2	13	270
D4	Ionosphere	2	33	351
D5	Parkinson	2	22	195
D6	Pima Indian	2	8	768
D7	Sonar	2	60	208
D8	WBDC	2	30	569
D9	Tic-Tae-	2	09	958

2. Experiment

We tested our proposed method on the benchmark dataset given in table1. All experiment is performed on System having Core i-5 central processing unit, 4 GB RAM, and 500 GB Hard Disk and frequency of Central Processing Unit is 3.30 GHZ. We used MATLAB as a tool to develop code related to our proposed method. To find robust classification accuracy it is very necessary that each pattern must be tested one time and must take part during training. Due to this reason cross validation is applied during calculation of goodness of each chromosome. We have used two classifiers namely Support Vector Machine (SVM) and K- Nearest Neighbour (K-NN). Here RBF and MLP kernel based SVM is used and the comparison is made between them on the basis of obtained measures. K-Nearest Neighbour classifier is used with three value of K (1, 2, and 3). The comparison is also performed on other obtained measures, and it also clears that why value of K is also important in case of KNN. Here we have used one parameter R. This R is responsible to determine how many number of bits must be mutated for each dataset. A smaller number like 0.10 is more preferable over 0.9 etc as it gives much scope for search space.

1. Results and Discussion

The experiment is performed on the datasets listed in table 1 with two classifier Support vector machine and K-nearest Neighbour. Table R1 shows the classification accuracy obtained by methods. This table has 6 columns, the first column of this table R1 contains Dataset ID and other 5 columns contain classification

accuracy obtained by methods. The total number of rows in this table is 11 and last rows contain information about performance of model on datasets. According to this table it is very clear that GD-RBF-SVM's classification accuracy is highest with respect to other methods in case of six datasets. GD-kNN-1, GD-kNN3 and GD-kNN-5 each one performs better than other methods in case of the datasets D7, D9 and D6 respectively. Classification accuracy of dataset D1 (87.10), D2 (73.9), D3 (76.30), D4 (94.58), D5 (94.42), D6 (76.29), D7 (88.48), D8 (95.25) and D9 (87.05).

Figure R1 is the representation of accuracy by methods in form of bar. Blue line represents GD-RBF-SVM, red represents GD-MLP-SVM, green represents GD-KNN-1, navy blue represents GD-KNN-3 andGD-KNN-5 is represented by sky colour. In figure, blue bar is taller than others in case of the datasets D1, D2, D3, D4, D5, and D8. While in case of datasets D6 sky colour bar, D7 green colour bar and D9 navy blue colour bar is taller than others.

Table R1: Accuracy obtained by proposed method with Support vector machine (RBF and MLP kernel based) and K nearest Neighbour (with value of K=1, 3 or 5)

Dataset name	**GD-RBF-SVM**	**GD-MLP-SVM**	**GD-KNN-1**	**GD-KNN-3**	**GD-KNN-5**
D1	**87.10**	80.58	82.17	75.80	83.19
D2	**73.9**	64.6	72	73.3	73.1
D3	**76.30**	66.30	71.11	68.15	71.11
D4	**94.58**	65.83	91.18	90.59	86.87
D5	**94.42**	76.30	90.79	89.26	89.76
D6	75.14	68	69.53	72.91	**76.29**
D7	76.48	69.26	**88.48**	86.14	83.14
D8	**95.25**	90.52	93.50	93.15	94.19
D9	86.74	69.94	81.53	**87.05**	72.03
Best performance	**06**	**00**	**01**	**01**	**01**

*Best performance means number of times best accuracy found by respective approach

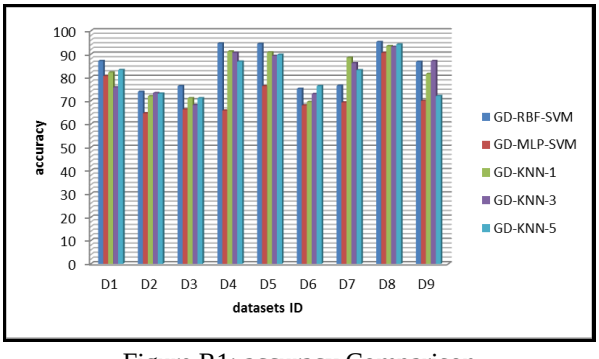

Figure R1: accuracy Comparison

Table R2: Sensitivity obtained by proposed method with Support vector machine (RBF and MLP kernel based) and K nearest Neighbour (with value of K=1, 3 or 5)

Dataset name	**GD-RBF-SVM**	**GD-MLP-SVM**	**GD-KNN-1**	**GD-KNN-3**	**GD-KNN-5**
D1	**85.07**	77.11	80.50	72.86	82.41
D2	**83.12**	81.83	78.58	77.96	77.07
D3	81.08	**81.78**	78.02	74.05	73.58
D4	89.01	52.92	97.40	**98.26**	96.18
D5	**94.40**	90.71	93.79	92.69	91.70
D6	61.40	53.32	56.67	61.88	**68.56**
D7	**98.75**	63.75	92.71	87.76	85.13
D8	92.54	85.00	93.72	92.02	**95.64**
D9	95.00	77.42	78.31	83.72	70.06

Table R2 contains information about sensitivity obtained by proposed methods. Good value (100) means model have 100% capability to identify class 1patterns. Classification accuracy obtained by GD-RBF-SVM is best on the dataset D1-D5 and D8 but sensitivity is good in the datasets D1, D2, D5, D7, and D9. It shows that model accuracy and sensitivity both is only good in case of datasets D1, D2, D5. Figure R2 is bar plot of sensitivity.

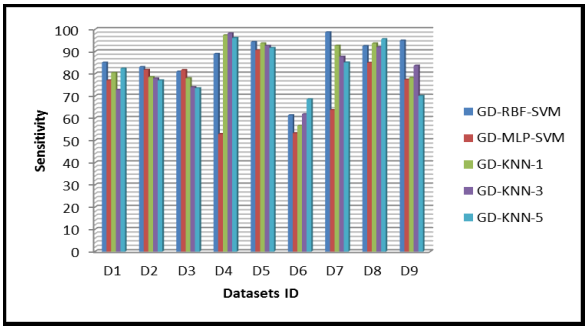

Figure R2: Sensitivity graph

Table R3: Specificity obtained by proposed method with Support vector machine (RBF and MLP kernel based) and K nearest Neighbour (with value of K=1, 3 or 5)

Dataset ID	GD-RBF-SVM	GD-MLP-SVM	GD-KNN-1	GD-KNN-3	GD-KNN-5
D1	89.28	84.03	**84.44**	78.77	83.95
D2	55.73	44.19	53.30	**58.10**	57.91
D3	**66.82**	51.19	56.85	54.54	60.36
D4	**98.59**	79.39	89.16	88.00	84.34
D5	**95.50**	56.75	85.81	84.48	87.33
D6	**86.27**	80.63	76.35	78.80	79.79
D7	70.81	77.57	**86.57**	86.52	83.11
D8	**97.23**	94.72	93.65	94.02	93.61
D9	78.46	56.49	98.70	99.05	**100**

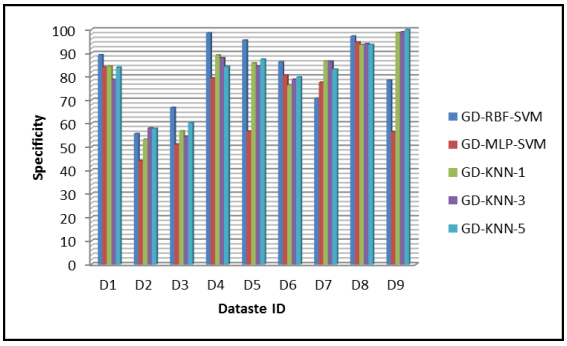

Figure R3: Specificity graph

Table R3 contains information about specificity. A good value of specificity means good recognition of class -1 patterns. From this table it is very clear that GD-RBF-SVM outperforms other in case of the datasets D3, D4, D5, D6 and D8. GD-KNN-1 gives better specificity value in datasets D1 and D7. In the case of Dataset D2 GD-KNN-3 gives best specificity value. Figure R 3 is also a representation of specificity in bar plot diagram. A taller bar to respective datasets shows goodness of method, means taller the bar greater the specificity value.

Table R7 shows the number of feature considered by respective model to give respective classification accuracy. In first row i.e. row corresponding to D1 shows that model GD-RBF-SVM takes 6 features to give best classification measures given in row first row of table R1. Correspondingly for each dataset and model is shown this table. Bold values in table R7 shows the number of features considered by respective model to give best classification accuracy.

Table R7: number of feature selected by different models

Dataset ID	GD-RBF-SVM	GD-MLP-SVM	GD-KNN-1	GD-KNN-3	GD-KNN-5
D1	**6**	5	6	6	7
D2	**13**	16	14	13	11
D3	**9**	5	6	4	6
D4	**15**	18	14	10	13
D5	**15**	8	11	11	9
D6	5	4	5	6	**4**
D7	21	36	**27**	30	36
D8	**14**	17	19	13	9
D9	7	1	5	**7**	4

Multiple classifier systems (MCS) [15, 16] are used for giving best classification accuracy in MCS system more than one classifier is used for evaluation of classification accuracy. Here we used two classifiers (with their variation) to obtain better accuracy. Table C1 contains the accuracy proposed by different methods and proposed accuracy. It is compared with some latest methods accuracy. Some latest methods are ECMBF [17], GP-FRFS [13], IVDE [11], SFS [18], LS [18] and JSDFS [18]. Table C1 has total 9 columns first one is for dataset Id and remaining eight columns represents

Table C1: Classification accuracy obtained by different methods (header contains method name and in column respective accuracy is given) Bold values in the table shows the highest accuracy obtained on the respective dataset. Here (-) shows that on particular datasets that model is not applied so no classification accuracy is available.

Dataset ID	**Proposed**	**ECMBF**	**GP-FRFS**	**SDFS**	**LS**	**JSDFS**	**IVDE**
D1	**87.10 (6)**	85.79 (2)	-	-	-	-	-
D2	**73.9 (13)**	72.18 (4)	-	-	-	-	-
D3	76.30 {9)	80.19 (4)	**81.85 (7)**	-	-	-	-
D4	**94.58 (15)**	89.75 (3)	91.19 (7)	85.91 (23)	87.29 (23)	89.00 (13)	93.68
D5	**94.42 (15)**	-	87.69 (6)	-	-	-	-
D6	**76.29 (4)**	72.60 (2)	75.42 (6)	-	-	-	-
D7	**88.48 (27)**	72.39 (6)	79.81 (6)	54.0 (13)	52.03 (1)	72.97 (13)	84.18
D8	95.25 (14)	-	-	-	-	-	**96.98**

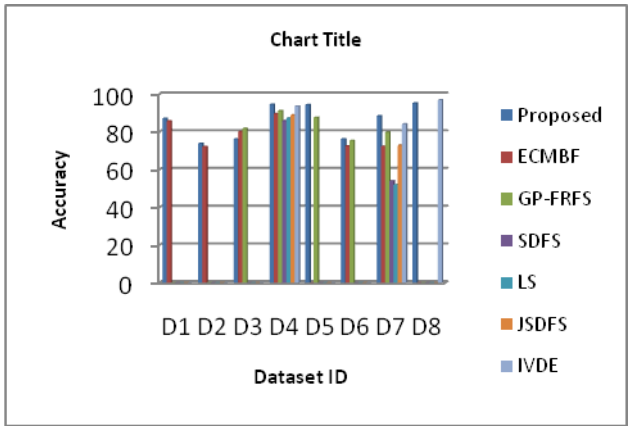

Figure C1: representation of classification accuracy comparison in bar form

methods. We made our comparison based on two priorities; first priority is given to classification accuracy. If model gives the best classification accuracy then number of features does not have much importance. If classification accuracy is equal for two methods then number of features matters. Out of 9 datasets we made comparison on 8 datasets. We found that proposed method outperforms on 6 out of 8 comparable datasets. As shown in table C1 Proposed accuracy is more than other methods in case of the datasets D1, D2, D4, D5, D6, D7. Figure C1 also shows that our method gives best classification accuracy. In the figure C1 proposed accuracy is represented by blue line and it is taller than other in case of D1, D2, D4, D5, D6, and D7 datasets shows high classification accuracy.

2. Conclusion

As Feature selection is one of the very important pre-processing steps, in this paper, a novel Wrapper based feature selection approach is proposed. This algorithm is based on the concept of incorporation of feature by weak solution from strong solution point (possible candidate solution) can also termed greedy approach. It starts with a random initialization of a number of solution points called solution spaces. Each solution point's (of the solution space) goodness is measured using a classifier. Solution point with maximum classification accuracy will be taken as a possible candidate solution. Other solution points choose some features from possible candidate solution point and incorporate into itself (means replace its components with components of best solution point), so a new solution point will be generated. This is done by all solution points other than possible candidate solution point and results into a new solution space. Then again above process is repeated till a termination criterion is not achieved. The Move operation is one of the important points in this paper. Fitness function plays an important role during classification hence we took Sensitivity*Specificity as fitness function. The result obtained by the application of proposed method on the some benchmark datasets shows that our method performs well on six datasets out of nine datasets when compared to other registered method's accuracy. Since this algorithm is based on greedy approach i.e. incorporation of features from best possible solution point into weak solution points there may be possibility that best solution point still has some redundant component.

References

[1] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Second Edition, 2006.

[2] T. Mitchell, Machine Learning, McGraw-Hill Science/Engineering/Math, 1997.

[3] R.O. Duda, P.E. Hart, D. G. Stork, Pattern classification (2nd edition), Wiley, New York, 2001.

[4] C. Cortes, V. Vapnik , Support Vector Networks, Machine Learning, Volume 20, Issue 3, pp. 273-297, 1995.

[5] G.H. John, P. Langley, Estimating continuous distributions in bayesian classifiers, the Eleventh Conference on Uncertainty in Artificial Intelligence, 1995.

[6] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1999.

[7] A. Jain, D. Zongker, Feature selection: evaluation, application, and small sample performance, IEEE Trans. Pattern Anal. Mach. Intell., 19(2), pp. 153-158, 1997.

[8] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Springer, 1998.

[9] I. Guyon, A. Elisseeff (Eds.), Feature Extraction: Foundations and Applications, vol. 207, Springer, 2006.

[10] A. Saxena, M. Kothari, N. Pandey, Evolutionary Approach to Dimensionality Reduction, Encyclopedia of Data Warehousing and Mining, Edition 2 , Vol. II, pp. 810-816, 2009.

[11] T Bhadra , S Bandyopadhyay, Unsupervised feature selection using an improved version of Differential Evolution, Expert Systems with Applications 42, pp. 4042–4053, 2015.

[12] http://archive.ics.uci.edu/ml/datasets.html

[13] J.H. Lee , J.R. Anaraki, C.W. Ahn , J. An, Efficient classification system based on Fuzzy–Rough Feature Selection and Multitree Genetic Programming for intension pattern recognition using brain signal, Expert Systems with Applications 42, pp. 1644–1651, 2015.

[14] A. Saxena, N. R. Pal, M. Vora, Evolutionary Methods for Unsupervised Feature Selection Using Sammon's Stress Function, Springer, Fuzzy Inf. Eng., pp. 229-247, 2010 .

[15] M. Wozniak, M. Grana, E. Corchado, A survey of multiple classifier systems as hybrid systems, Information Fusion 16 (2014) 3–17.

[16] R. Polikar, Ensemble based systems in decision making, IEEE Circuits and Systems Magazine 6 (3), pp. 21–45, 2006.

[17] Sheng-yiJianga, Lian-xi Wangb, Efficient feature selection based on correlation measure between continuous and discrete features, Information Processing Letters 116 (2016) 203–215.

[18] Hui Yan, Jian Yang , Sparse discriminative feature selection, PatternRecognition48(2015)1827–1835.