CDEBMTE: Creation of Diverse Ensemble Based on Manipulation of Training Examples

Hamid Parvin, Sajad Parvin, Zahra Rezaei Nourabad Mamasani Branch Islamic Azad University Nourabad Mamasani Iran hamidparvin@mamasaniiau.ac.ir, {s.parvin, rezaei}@iust.ac.ir



ABSTRACT: Ensemble methods like Bagging and Boosting which combine the decisions of multiple hypotheses are among the strongest existing machine learning methods. The diversity of the members of an ensemble is known to be an important factor in determining its generalization error. We present a new method for generating ensembles, named CDEBMTE (Creation of Diverse Ensemble Based on Manipulation of Training Examples), that directly constructs diverse hypotheses using manipulation of training examples in three ways: (1) sub-sampling training examples, (2) decreasing/increasing error-prone training examples and (3) decreasing/increasing neighbor samples of error-prone training examples.

The technique is a simple, general meta-learner that can use any strong learner as a base classifier to build diverse committees. Experimental results using two well-known classifiers (1) decision-tree induction and (2) multilayer perceptron as two base learners demonstrate that this approach consistently achieves higher predictive accuracy than both the base classifier, Adaboost and Bagging. CDEBMTE also outperforms Adaboost more prominent when training data size is becomes larger.

We propose to show that CDEBMTE can be effectively used to achieve higher accuracy and to obtain better class membership probability estimates.

Keywords: Classifier Ensemble, Diversity, Training Examples Manipulation

Received: 17 September 2011, Revised 23 October 2011, Accepted 29 October 2011

© 2012 DLINE. All rights reserved

1. Introduction

There are many inherently different classifiers in the pattern recognition. It may be worthy to mention that usage of the broad versatility in kind of classifiers is not a trivial matter to be ignored. It means although a classification algorithm may obtain a good performance for a specific problem, it has not enough robustness for another problem. It has been always an ideal for pattern recognition communities to present an approach of how to select the best (or at least approximately the best) classifier for a specific problem. It has been shown that this desire is far from the reach; at least by employing simple classifiers as the main learner.

Fortunately there is another way to cope with the problem. It is to use many learners instead of one learner and then to employ a democracy-wise method to firm the final decision of the committee (classifier). One of the major advances in inductive learning in the past decade was the development of ensemble or committee approaches that learn and retain multiple hypotheses and combine their decisions during classification [3]. For example, Boosting [4]-[5] is an ensemble method that learns a series of

"weak" classifiers each one focusing on correcting the errors made by the previous one; and it is currently one of the best generic inductive classification methods [8]. Therefore, recent researches are directed to the combinational methods which have more power, robustness, resistance, accuracy and generality than simple method. Combination of Multiple Classifiers (CMC) can be considered as a general solution method for pattern recognition problems. Inputs of CMC are results of separate classifiers and output of CMC is their consensus decision [23]-[25].

So pattern recognition communities turn from single-based classification systems to ensemble-based classification systems due to same reasons that humanity societies converge from dictatorial sovereignties to democratic sovereignties. By the way, the four mentioned preconditions that are necessary to guarantee a democratic sovereignty to succeed and to emerge wisdom of crowd must be consequently dealt with. So researchers in pattern recognition communities turn to construct an ensemble in which each individual (classifier) is (a) as different as possible (satisfying preconditions one and two), (b) while still maintaining consistency with the training data having authority to learn from a manipulation of them in such a reasonable way (satisfying precondition three) [10]. Also they use (c) a suitable and consistent to democracy concept to aggregate their opinions (satisfying precondition four). Although all successful ensemble methods encourage diversity to some extent, few have focused directly on the goal of maximizing diversity. Existing methods that focus on achieving diversity [21] and [29] are fairly complex and are not general meta-learners like Bagging [1] and Boosting which can be applied to any base learner to produce an effective committee [32].

Melville [18] presents a new meta-learner DECORATE (Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples), that uses an existing "strong" learner (one that provides high accuracy on the training data) to build an effective diverse committee in a simple, straightforward manner. This is accomplished by adding different randomly constructed examples to the training set when building new committee members. These artificially constructed examples are given category labels that disagree with the current decision of the committee, thereby easily and directly increasing diversity when a new classifier is trained on the augmented data and added to the committee.

Different categorizations of combinational classifier systems are represented in [13], [22], [26] and [28]. Valentini and Masouli divide methods of combining classifiers into two major categories: generative methods, non-generative methods. In generative methods, a set of base classifiers are created by a set of base algorithms or by manipulating dataset. This is done in order to reinforce diversity of base classifiers. Generally, all methods which aggregate the primary results of the fixed independent classifiers are non-generative. They are also named fusion methods [9] and [30].

In generative methods, diversity is usually made using two groups of methods. One group of these methods obtains diverse individuals by training classifiers on different training set, such as bagging [1], boosting [4], cross validation [10] and using artificial training examples [17].

Another group of methods for creating diversity employs different structures, different initial weighing, different parameters and different base classifiers to obtain ensemble individuals. For example, Rosen [29] adapted the training algorithm of the network by introducing a penalty term to encourage individual networks to be decorrelated. Liu and Yao [16] used negative correlation learning to generate negatively correlated individual network.

The third group is named selective approach group where the diverse components are selected from a number of trained accurate networks. For example, Opitz and Shavlik [21] proposed a generic algorithm to search for a highly diverse set of accurate networks. Lazarevic and Obradovic [14] proposed a pruning algorithm to eliminate redundant classifiers. Navone et al. [19] proposed another selective algorithm based on bias/variance decomposition. GASEN proposed by Zhou et al. [33] and PSO based approach proposed by Fu et al. [6] also were introduced to select the ensemble components.

There have been many methods developed to construct an ensemble. Some of these methods, such as Bagging and Boosting are *meta-learners* i.e. they can be applied to *any* base classifier. Other methods are specific to a particular base classifier. For example, Negative Correlation Learning [15] is used specifically to build committees of Neural Networks. We only deal with construction of general ensemble, i.e. *meta-learners*. Although a special base classifier may be better suited for a particular domain than others, a *general* ensemble approach that is independent of the particular base classifier is preferred in an unknown domain. So the aim of our research is to only propose a *meta-learner*.

In an ensemble, the output of several classifiers is useful to be participated in an ensemble only when they disagree on some

inputs [7] and [31]. The measure of disagreement is called as the *diversity* of the ensemble. For regression problems, *mean* squared error is generally used to measure accuracy, and variance is used to measure diversity.

There have been defined several measures of diversity for an ensemble in the literature. In a recent study, Kuncheva and Whitaker [11] compared ten different measures of diversity. They found that most of these measures are highly correlated. However, to the best of our knowledge, there has not been a conclusive study showing which measure of diversity is the best to use during constructing and evaluating an ensemble.

Melville and Mooney [17] have introduced a new meta-learner DECORATE that uses an existing learner to build an effective diverse committee in a simple, straightforward manner. This is accomplished by adding different randomly constructed examples to the training set when building new committee members. These artificially constructed examples are given category labels that *disagree* with the current decision of the committee, thereby easily and directly increasing diversity when a new classifier is trained on the augmented data and added to the committee.

In this paper, a new method to obtain diverse classifiers is proposed which uses manipulation of dataset structures. In this new method it is presented to make diversity in base classifiers (of type MultiLayer Perceptron (MLP) or Decision Tree (DT)). The effect of existing or non-existing of boundary instances is evaluated. Firstly, using a simple classification, the error-prone instances and their neighbors are detected. Using these subsets, different datasets of our main dataset are created. Then, several classifiers are trained on these datasets. They may be selected into or eliminated from the ensemble based on a diversity metric. This method that we have named "CDEBMTE: Creation of Diverse Ensemble Based on Manipulation of Training Examples" is described in section 3, accurately. The used decision-tree is J48 that is a Java implementation of C4.5 [27] introduced in [32]. Cross-validated learning curves support the hypothesis that CDEBMTE generally result in greater classification accuracy.

2. CDEBMTE

In CDEBMTE, using a simple classification, the error-prone samples and their neighbors which are called respectively EPS and NS are first detected. Using EPS and NS and also the original training set, different training sets are created from the original one. Then, several classifiers are trained on these datasets. Based on a diversity metric a subset of them is selected into an ensemble. Finally, a consensus function is employed to aggregate the results of these diverse classifiers.

TS: Training Set
P: Error-Proneness of TS
<i>EPS</i> : Error Prone Set, <i>EPS</i> ={};
<i>CEPS</i> : Complement of Error Prone Set, <i>CEPS</i> ={};
NS: Neighbor Set, NS={ };
K: Folds
N= TS
Function FindEPSandNS()
For $i=1$ to K
Produce an E_i of <i>H</i> classifiers by RF(α) on <i>TS</i> \ <i>TS</i> _{<i>i</i>}
Simulate E_i over $x_j \in TS_i$, then tag x_j by P_j
If $P_i > 0$ Then add x_i to EPS Else add x_i to CEPS
End.
For $j=1$ to N
$c = \text{real tag of } x_i$
set nx_i as the nearest neighbor of x_i in CEPS
that belongs class c
End.
Return P. EPS. NS

Figure 1. Pseudo-code of algorithm for finding EPS and NS

3.1 Preparing Different Subsets of Training Dataset

To extract different subsets out of training set, we divide the primary Training Set (TS) in K random partition. Let us denote *i*-th partition by TS_i . Presume that cardinality of TS, |TS|, is denoted by N. By training a fixed number (denoted by H) of simple base classifiers on $TS \setminus TS_i$, we reach an ensemble that is expert for TS_i . Please note that these H classifiers are produced by a Bagging mechanism named Random Forest (RF). Breiman [2] proposes a variant of bagging which he calls RF. RF is a general class of ensemble building methods that uses decision tree as its base classifier. It has a parameter a that shows the ratio of subsampling. To be labeled a "random forest" an ensemble of decision trees should be built by generating independent identically distributed random vectors and use each vector to grow a decision tree. In this paper RF which is one of the well known versions of bagging classifier [12] is considered as Bagging method. It means when Bagging is used it refers to RF in the paper.

Let us denote the ensemble that is expert for TS_i by E_i . We test each data point in TS_i to find out how the data point has potential to be wrongly classified. We assign a number P_j to *j*-th data point in TS_i indicating how many classifiers out of *H* ones in the ensemble E_i , wrongly classify the *j*-th data point. The bigger P_i , the more error-prone the *i*-th data point. It is also worthy to mention that P_i is an integer number between 0 to *H*.

This procedure is repeated *K* times, choosing a different part for testing, each time. When N = K, the method is called the leaveone-out or U-method [12].

The proposed algorithm to discover Error-Prone Subset (EPS) and Neighbor Subset (NS) is shown in Figure 1.

Figure 2 illustrates the EPS and NS in Iris dataset projected in features 3 and 4. In Figure 2 all samples of Iris dataset is considered as training set.



Figure 2. EPS and NS in Iris dataset projected into features 3 and 4

3.2 Creating an Ensemble of Diverse Classifiers

Using more diverse classifiers is one of the most important factors controlling the performance of combinational classifiers. Based on three sets of data, TS, EPS and NS, in this step, a number of classifiers are heuristically retrained. As shown in Table 1, a number of combinations are used to create diversity in the ensemble. Also, Table 1 expresses the reasons of their usage. It is worthy to be mentioned that κ , λ and μ are three parameters that are to be set. The domains of these parameters are {0.2, 0.4, 0.6, 0.8, 1}. For example, κ^*TS means that a sub-sampling from *TS* that contains a κ ration of *TS*; and $\kappa^*TS \cup \lambda^*EPS \cup \mu^*NS$ means that a union of three sub-samplings: (1) κ ration from *TS*, (1) λ ration from *EPS* and (1) μ ration from *NS*.

Retraining of classifiers, according to combinations of Table 1, results the classifiers so that each of them concentrates on

special aspect of data. This can result in very good diversity in the ensemble. In other hand, although the accuracy of each classifier may not be significantly better than a simple classifier, it can yield to satisfactory diverse base classifiers.

In this study, six classes of producing a subset are defined. In first class of producing subsets, by sliding parameter κ in all possible values in its domain, we obtain 5 training subsets. Equivalently in second class, by sliding parameters κ and μ in all possible values in their domains, we obtain 5*5 training subsets. In 3rd, 4th, 5th and 6th classes, we respectively reach 125 (5*5*5), 25, 25 and 5 training subsets by sliding their parameters.

Classes of Producing Training Subset	Feature of Resultant Subsets		
1. κ* TS	To create the base classifiers		
2. κ *TS $\cup \mu$ *NS	Classification by high complex boundaries with more concentration on crucial points		
3. $\kappa^*TS \cup \lambda^*EPS \cup \mu^*NS$	Classification by complex boundaries with more concentration on error prone and crucial points		
4. $\kappa^*(TS \setminus EPS) \cup \mu^*NS$	Classification by high simple boundaries with more concentration on crucial points		
5. $\kappa^*TS \cup \lambda^*EPS$	Classification by high complex boundaries with more concentration on error prone data samples		
6. κ* (TS\EPS)	Classification by very simple boundaries		

Table 1. Different data combinations and reasons of its usages

3.3 Diversity of an Ensemble

To define a meaningful diversity metric first we define a between-classifier diversity metric over one example. As equation (1) shows, the between-classifier diversity metric over one example is denoted by $d(C_i, C_i, x_k)$

$$d(C_{i}, C_{j}, x_{k}) = \begin{cases} 0 & C_{i}(x_{k}) = C_{j}(x_{k}) \\ 1 & C_{i}(x_{k}) \neq C_{j}(x_{k}) \end{cases}$$
(1)

In the equation (1), C_i stands for *i*-th classifier from the ensemble and $C_i(x_k)$ stands for its output over an exemplary sample x_k . Based on the equation (1), we define a diversity metric for an ensemble according to equation (2).

$$D(E) = \frac{1}{|E|^2 * |TS|} \sum_{x_k \in TS} \sum_{C_i \in E} \sum_{C_j \in E} d(C_i, C_j, x_k)$$
(2)

where E stands for an ensemble of classifiers. D(E) is the diversity of ensemble .

3.4 Performance of an Ensemble

Let us assume that *E* is the ensemble of *r* classifiers $\{C_1, C_2, C_3, ..., C_r\}$. Also assume that problem has *q* classes. Next, assume applying the ensemble over data sample *x* is resulting in a binary *D* matrix like equation (3).

Journal of E- Technology Volume 3 Number 1 February 2012

$$D = \begin{bmatrix} d_{1 \ 1} & d_{1 \ 2} & . & d_{1 \ n} \\ . & . & . & . \\ d_{m-1 \ 1} & d_{m-1 \ 2} & . & d_{m-1 \ n} \\ d_{m \ 1} & d_{m \ 2} & . & d_{m \ n} \end{bmatrix}$$
(3)

where $d_{i,j}$ is one if classifier *j* votes that data sample *x* belongs to class *i*. Otherwise it is equal to zero. Now the ensemble decides the data sample to belong class *tc* according to equation (4).

$$tc = \arg\max_{i=1}^{m} \left| \sum_{j=1}^{n} d_{i,j} \right|$$
(4)

```
TS: Training Set

FinalEnsemble={}

Function ProduceFinalEnsemble (E)

pre_acc=0;

pre_div=0;

For i =1 to |E|

tempEnsemble= Ci \cup FinalEnsemble;

div=D(tempEnsemble);

acc=TEST(tempEnsemble, TS);

If (acce \geq pre) & (dive \geq pre_div) Then

FinalEnsemble=tempEnsemble;

pre_acc=acc;

pre_div=div;

End.

Return FinalEnsemble
```

Figure 3. Pseudo-code of CDEBMTE algorithm

We define the number of correct assignments by an ensemble as its accuracy.

3.5 CDEBMTE

In CDEBMTE, a rough ensemble is first generated and then pruning it into the final ensemble. A classifier is trained on each subset defined on the training data. We train each classifier on each subset defined on the training data, thereby forcing them to differ from the each other. Therefore these classifiers as an ensemble can be considered a diverse one. While enforcing the diversity into the rough initial ensemble, we still want to extract the most diverse subset of classifiers out of the rough initial ensemble, we still want to extract the most diverse subset of classifiers out of the rough initial ensemble, we still want to extract the most diverse subset of classifiers out of the rough initial ensemble. We do this by eliminating a new classifier if adding it to the existing ensemble decreases its accuracy. This process is repeated until we exceed the maximum number of iterations. The pseudo code of the CDEBMTE algorithm is presented in Fig 3. As you can understand from Figure 3, a greedy algorithm is employed to find the most diverse subset of the classifier to the pool of current ensemble. The CDEBMTE algorithm incrementally examines possibility of adding the next classifier to the pool of current ensemble by considering two factors. First it temporarily adds the next classifier to the current ensemble. If both accuracy and diversity of the current ensemble increases over TS, the search algorithm accepts the current ensemble; else the temporarily new added classifier is removed from current ensemble.

4. Experimental Results

In this chapter we present experiments comparing CDEBMTE with the leading ensemble methods like Bagging and AdaBoost. We also discuss some additional experiments that we run to better understand CDEBMTE's performance.

4.1 Benchmarks

To evaluate the performance of CDEBMTE we ran experiments on 18 representative datasets from the UCI repository [20]. The datasets are summarized in Table 2. Note that the datasets vary in the numbers of training examples, classes, numeric and nominal attributes; thus providing a diverse testbed.

4.2 Experimental Setting

The performance of each learning algorithm is evaluated using 10 complete runs of 10-fold cross-validation (except 3 monk problems). In each 10-fold cross-validation, each dataset is randomly split into 10 equal-size segments and results are averaged over 10 trials. For each trial, one segment is set aside for testing, while the remaining data is available for training.

The size ensemble in each algorithm is considered same with CDEBMTE algorithm. It means after running of CDEBMTE algorithm, the size of final ensemble is extracted and is feed as the ensemble size to other algorithms.

Dataset Name	Cases	Classes	Attributes
anneal	898	6	39
audio	226	6	69
autos	205	6	25
breast-w	699	2	9
credit-a	690	2	15
glass	214	6	9
heart-c	303	2	13
hepatitis	155	2	19
colic	368	2	22
iris	150	3	4
labor	57	2	16
lymph	148	4	18
monk	432	2	6
soybean	683	19	35
splice	3190	3	62
wine	178	3	13

Table 2. Summary of datasets

• The training and test sets in all three Monk's problems are predetermined. The training sets in Monk 1, 2 and 3 are 124, 169 and 122, respectively. More info about this problem can be found in: (The Monk's Problems, Donor: Sebastian Thrun School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213, USA, 1992).

4.3 Experimental Results

Figure 4 depicts the performance of CDEBMTE comparing to the performances of DECORATE, AdaBoost and Bagging over Iris dataset. In Figure 4 the accuracy of different ensemble methods in terms of different ratios of TS is presented.

Figure 5 depicts the performance of CDEBMTE comparing to the performances of DECORATE, AdaBoost and Bagging averaged over all 18 datasets. In Figure 5 the averaged accuracy of different ensemble methods in terms of different ratios of TS is presented. As it is depicted, DECORATE outperforms other methods considerably when the size of *TS* is small. It is due to adding artificial data samples to train set. It means while the number of training set is not enough to properly the other methods learn the models of the classes, DECORATE increases the size of training set to learn the models of the classes better.

For TS sizes above 50 percent, DECORATE falls gradually in comparing with AdaBoost. It is finally slightly placed in rank three

after AdaBoost method. While CDEBMTE sliding improves as *TS* size becomes greater. It is finally placed in rank one above all methods.



Figure 5. Averaged accuracy over all 18 datasets of different ensemble methods in terms of different ratios of TS

For *TS* sizes above 50 percent, DECORATE falls gradually in comparing with AdaBoost. It is finally slightly placed in rank three after AdaBoost method. While CDEBMTE sliding improves as *TS* size becomes greater. It is finally placed in rank one above all methods.

5. Conclusion

In this paper, a new method to improve performance of combinational classifier systems, CDEBMTE, is proposed. CDEBMTE is

based on increasing the diversity of ensemble. First different datasets are extracted from training dataset. Then it trains a number of classifiers over them. Finally it selects a subset of the trained classifiers.

It shows that emphasizing on crucial data can cause improvement in diversity. Also we showed that usage of different datasets causes to quite diverse classifiers.

References

[1] Breiman, L. (1996). Bagging predictors. Machine Learning, 24, p.123-140.

[2] Breiman, L. (2001). Random forests. *Machine Learning* 45, p.5–32.

[3] Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40 (2) 139–157.

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

[5] Freund, Y., Schapire, R. E.(1996). Experiments with a new boosting algorithm. In Saitta, L. (Ed.), Proceedings of the Thirteenth International Conference on Machine Learning (ICML-96). Morgan Kaufmann.

[6] Fu, Q., Hu, S. X., Zhao, S. Y. (2004). A PSO-based approach for neural network ensemble. *Journal of Zhejiang University* (Engineering Science) 38, 1596-1600, (in Chinese).

[7] Hansen, L. K., Salamon, P. (1990). Neural network ensembles, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12, 993-100.

[8] Hastie, T., Tibshirani, R., Friedman, J. (2001). The Elements of Statistical Learning. Springer Verlag, New York.

[9] Kohonen, T. (1990). Improved versions of learning vector quantization. International Joint Conference on Neural Networks, p. I545–I550.

[10] Krogh, A., Vedelsdy, J. (1995). *In:* Advances in Neural Information Processing Systems: Neural Network Ensembles Cross Validation, and Active Learning., eds. G. Tesauro, D. Touretzky and T. Leen (MIT Press, Cambridge, MA), p.231-238, .

[11] Kuncheva, L. K. Whitaker, C. (2003). Measures of diversity in classifier ensembles and their relationship with ensemble accuracy, *Machine Learning*, p.181–207.

[12] Kuncheva, L. I. (2005). Combining Pattern Classifiers, Methods and Algorithms, Wiley, New York.

[13] Lam, L. (2000). *In:* Multiple Classifier Systems: Classifier combinations: implementations and theoretical issues, eds J. Kittler and F. Roli, p. 78–86.

[14] Lazarevic, A., Obradovic, Z. (2001): Effective pruning of neural network classifier ensembles, *In:* Proc. *Int. Joint Conference on Neural Networks*, V. II, p. 796-801.

[15] Liu, Y., Yao, X. (1999). Ensemble learning via negative correlation. Neural Networks, V. 12.

[16] Liu, Y., Yao, X. (2000). Evolutionary ensembles with negative correlation learning, *IEEE Trans. Evolutionary Computation*, V.4, p. 380-387.

[17] Melville, P., Mooney, R. (2003). Constructing Diverse Classifier Ensembles Using Artificial Training Examples, *In:* Proc. of the IJCAI, Vol. I, p.505-510.

[18] Melville P.(2006). Creating Diverse Ensemble Classifiers.

[19] Navone, H. D., Verdes, P. F., Granitto, P. M., Ceccatto, H. A. (2000). Selecting Diverse Members of Neural Network Ensembles. Brazilian Symposium on Neural Networks, Vol. I, p.255-260.

[20] Newman C. B. D. J., Hettich, S., Merz, C. (1998). UCI repository of machine learning databases, http://www.ics.uci.edu/ ~mlearn/MLSummary.html.

[21] Opitz, D., Shavlik, J. (1996). Actively searching for an effective neural network ensemble. *Connection Science*, 8 (3-4) 337-353.

[22] Roli, F., Kittler, J. (2001). Editors. Proc. 2nd International Workshop on Multiple Classifier Systems (MCS 2001), V. 2096 of Lecture Notes in Computer Science LNCS Springer-Verlag, Cambridge, UK.

Journal of E- Technology Volume 3 Number 1 February 2012

[23] Parvin, H., Minaei-Bidgoli, B., Alizadeh, H., Beigi, A. (2011a). Classifier Ensemble Based Class Weightening. International Symposium on Neural Networks, p. 144-150.

[24] Parvin, H., Minaei-Bidgoli, B., Alizadeh, H. (2011b). A New Adaptive Framework for Classifier Ensemble in Multiclass Large Data, *In:* International Conference on Computational Science and Its Applications, p. 526-536.

[25] Parvin, H., Minaei-Bidgoli, B., Karshenas, H., Beigi, A. (2011c). A New N-gram Feature Selection Method for Malicious Code. *In:* International Conference on Adaptive and Natural Computing Algorithms, p. 98-107.

[26] Puuronen, S., Tsymbal, A., Skrypnyk, I. (2001). Correlation-based and contextual meritbased ensemble feature selection, *In:* International Conference on Advances in Intelligent Data Analysis, p. 135-144.

[27] Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA.

[28] Roli, F., Kittler, J. (20020. Editors. Proc. 3rd Int. Workshop on Multiple Classifier Systems (MCS 2002), Vol. 2364 of Lecture Notes in Computer Science LNCS Springer Verlag, Cagliari, Italy.

[29] Rosen, B. E. (1996). Ensemble learning using decorrelated neural network. Connection Science, 8 (3-4)373-384.

[30] Skalak, D. B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms, *In*: Int. Conf. on Machine Learning, New Brunswick, V. I, p. 293-301.

[31] Tumer, K., Ghosh, J. (1996). Error correlation and error reduction in ensemble classifiers. Connection Science, 8 (3-4) 385–403.

[32] Witten, I. H., Frank, E. (1999). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco.

[33] Zhou, Z. H., Wu, J. X., Jiang, Y., Chen, S. F. (2001). Genetic algorithm based selective neural network ensemble, *International Joint Conference on Artificial Intelligence*, V-II, p.797-802.