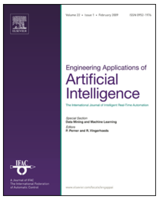
Engineering Applications of Artificial Intelligence 37 (2015) 34–42

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Proposing a classifier ensemble framework based on classifier selection and decision tree

Hamid Parvin [a], Miresmaeil MirnabiBaboli [b], Hamid Alinejad-Rokny [c,d,e,*]

[a] School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
[b] Institute of Informatics and Automation Problem, NAS RA, Armenia
[c] Faculty of Medicine, The University of New South Wales, Sydney, Australia
[d] School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia
[e] Faculty of Computer Engineering, University of Mazandaran, Mazandaran, Iran

ARTICLE INFO

Article history:
Received 4 January 2013
Received in revised form
7 July 2014
Accepted 8 August 2014
Available online 19 September 2014

Keywords:
Decision tree
Classifier ensembles
Clustering
Bagging
Learning
Ada Boosting

ABSTRACT

One of the most important tasks in pattern, machine learning, and data mining is classification problem. Introducing a general classifier is a challenge for pattern recognition communities, which enables one to learn each problem's dataset. Many classifiers have been proposed to learn any problem thus far. However, many of them have their own positive and negative aspects. So they are good only for specific problems. But there is no strong solution to recognize which classifier is better or good for a specific problem. Fortunately, ensemble learning provides a good way to have a near-optimal classifying system for any problem. One of the most challenging problems in classifier ensemble is introducing a suitable ensemble of base classifiers. Every ensemble needs diversity. It means that if a group of classifiers is to be a successful ensemble, they must be diverse enough to cover their errors. Therefore, during ensemble creation, a mechanism is needed to ensure that the ensemble classifiers are diverse. Sometimes this mechanism can select/remove a subset of base classifiers with respect to maintaining the diversity of the ensemble. This paper proposes a novel method, named the Classifier Selection Based on Clustering (CSBS), for ensemble creation. To insure diversity in ensemble classifiers, this method uses the clustering of classifiers technique. Bagging is used to produce base classifiers. During ensemble creation, every type of base classifier is the same as a decision tree classier or a multilayer perceptron classifier. After producing a number of base classifiers, CSBC partitions them by using a clustering algorithm. Then CSBC produces a final ensemble by selecting one classifier from each cluster. Weighted majority vote method is used as an aggregator function. In this paper we investigate the influence of cluster number on the performance of the CSBC method; we also probe how we can select a good approximate value for cluster number in any dataset. We base our study on a large number of real datasets of UCI repository to reach a definite result.

1. Introduction

Classification is the most important task in pattern recognition institute. Therefore, since the beginning of the pattern recognition science, one of the most challenging problems in this field was introducing a general classifier that can learn properly every dataset of any given problem. Many classifiers have been proposed to learn problems thus far. However, all of them have their own positive and negative aspects. So they are good only for specific problems. But there is no strong solution to recognize which classifier is a better or good classifier for a specific problem. Since finding the best classifier is an impractical problem, we must use another approach. Thus might be using many inaccurate classifiers, where each of them is assigned to a subspace of dataset as an ensemble (i.e. use their gathering vote as the decision of ensemble). Ensemble learning is a strong approach to produce a near-to-optimal classifier for any problem. This method reinforces the ensemble in error-prone subspaces, and hence can lead to better performance of classification. In general the following sentence can be true to say that the result of combination of diverse classifiers, is better classification. Diversity is an important factor for each ensemble to be successful. The existence of diversity in an ensemble ensures that those classifiers are independent of each other. It means that misclassifications do not occur simultaneously. Kuncheva showed that increasing the number of diverse classifiers

[*] Corresponding author.
E-mail addresses: H.Alinejad@ieee.org, Hamid.AlinejadRokny@UoN.edu.au (H. Alinejad-Rokny).

http://dx.doi.org/10.1016/j.engappai.2014.08.005
0952-1976/© 2014 Elsevier Ltd. All rights reserved.

can lead to better performance (even perfect accuracy) (Kuncheva, 2005; Minaei-Bidgoli et al., 2004; Alizadeh et al., 2011). Also, ensemble philosophy is applicable to Bayesian Networks (Peña, 2011). The main challenge in the creation of classifier ensemble is to provide a general approach to ensure diversity, which is an important factor for an ensemble. It means that if an ensemble of classifiers has to be a successful ensemble, they should be diverse enough to cover their errors. Creating some suitable diverse classifiers that can participate in an ensemble is a challenging problem. There are a number of ways to obtain a desired diversity in an ensemble. Kuncheva proposed some approaches based on the metrics that indicates the amount of similarities or differences among classifiers' outputs (Kuncheva and Whitaker, 2003; Parvin et al., 2013a, 2013b, 2013c, 2011c, 2011d; Rezaei et al., 2011).

Clustering is a process of assigning a group of objects into clusters. So objects in the same cluster are more similar to each other than the objects in other clusters. This is used a lot in some applications of data mining, especially for information retrieval, text categorization and text ranking (Yang, 2006; Dasgupta and Ng, 2010; Amigó et al., 2011; Kurland and Krikon, 2011).

Giacinto and Roli, by producing a large number of artificial neural network classifiers by different initializations of their parameters and then selecting a subset of them based on their distances in output space, proposed an approach to hosting the classifiers with a high degree of diversity.

Hamid et al. were inspired from the clustering and selection method and proposed a new clustering and selection method that enables one to reduce the drawbacks of the simple ensemble methods in creating diversity (Parvin et al., 2001, 2013c, 2013e). They consider how the base classifiers are created. They also investigated the usage of Boosting and Bagging method as a source of diversity generation on Giacinto and Roli's method. At first, they trained a large number of classifiers using the Boosting and Bagging method, and then partitioned them based on the output over the training set. Finally they chose a classifier randomly and then inserted it into the ensemble. The weighted majority voting mechanism was used as the consensus function of the ensemble.

In this paper a novel method, named the Classifier Selection Based on Clustering (CSBC), has been proposed. This method can provide the necessary diversity between ensemble classifiers by using the clustering technique. And it uses the Bagging method as a generator of base classifiers. Base classifiers are still fixed in the decision tree classifier or the multilayer perceptron classifier during the creation of an ensemble. Then the clustering algorithm partitions the classifiers. Weighted majority vote mechanism was used as the consensus function of ensemble. We investigated how the number of clusters can affect the performance of the CSBC method. We based our study on a large number of real datasets of the UCI repository to reach a definite result. In this paper we investigate the better selection of the classifier from each cluster, how to select a good parameter according to the dataset and the effect of the training set ratio of each base classifier on CSBC's performance.

Machine learning is an important paradigm in artificial intelligence, and Artificial Neural Network (ANN) is a common approach to learning. Unlike the traditional approach, ANN has some properties such as self-adaptivity, ability to generalize, and so on. But the source of these properties is not explained well. They are almost explained by the comparison between ANNs and real neural networks. From bionic point of view, these formal and biological neurons do not have anything in common.

An ANN is a model that enables one to obtain any input and produce the desired set of outputs. An ANN includes two base elements: neurons and connections. An ANN is a set of neural network with connections between them. From another perspective an ANN includes two distinct views: topology and learning. Topology is related to the existence or nonexistence of a connection. Learning in an ANN indicates the power of topology connections. Multi-Layer Perceptron (MLP) is one of the most representative of ANNs. There are different methods to set the power of connections in MLP. One method is setting the weights using a priori knowledge. Another method is to train the MLP, and then using the teaching patterns and finally changing the weights based on some learning rules. In this paper we use MLP as the base classifier.

However, there is no common theory for ANN learning; some training algorithms are proposed for any given ANN architecture. However, these algorithms are all over classical and external to ANNs. Neural networks, which include coded learning algorithm within astrocytic nets, are rather interesting, because different learning rules can be made internal for them. But the source and structure of these rules remain unclear. Therefore, it seems that there is a paradox between the self-learning capabilities of ANNs and their distinctions from the traditional algorithms. Outwardly, ANNs cannot solve the machine learning problems. Particularly, over-learning is one difficult subject and there is no good explanation in the ANN theory for it. Indeed limiting the training time prevents over-learning. Despite all that, the popularity of ANN is not by chance. However, their benefits must be considered, in order to improve them in the future. Decision tree (DT) is one of the versatile classifiers in the machine learning field. DT is an unstable classifier that can introduce different outputs in successive trainings on the same condition. DT uses a tree-like graph or model for decision. The type of presentation helps experts understand the classifiers (Yang, 2006; Parvin et al., 2011a, 2011b). The natural instability of this method can be a source of diversity for classifier ensemble. An ensemble of a number of DTs is similar to a Random Forest (RF) algorithm, which is one of the powerful ensemble algorithms. This algorithm was developed by Breiman (1996). In this paper, DT is used as the base classifier.

The rest of this paper is organized as follows. Section 2 is related works. In Section 3, the proposed method is explained. Section 4 demonstrates the results of our proposed method against traditional methods. Finally, the conclusion is presented in Section 5.

2. Related work

In general, there are two challenging approaches to mix a number of classifiers that use different training sets: Bagging and Boosting. Both of them are two sources for diversity generation and they are the best ensemble methods.

We suppose that a training set is represented by TS. And the ith data item in TS is represented as O_i and m is the number of data items in TS. Training phase of CSBC is shown in Fig. 1, which uses the Bagging method as the base classifier generation.

Breiman is one of the first researchers to have used Bagging for Bootstrap AGGregatING.

The idea of Bagging is simple and attractive: the classifiers of the ensemble are made by bootstrap copies of the training set. Using different training sets can ensure the necessary diversity of ensembles. It is noticeable that Bagging cannot ensure the necessary diversity.

Another kind of Bagging named “Random Forest” has been proposed by Breiman. RF is a method for ensemble creation that uses a decision tree as the base classifier generator. In “Random Forest”, an ensemble of decision trees must be built by creating independent identically distributed random vectors and each vector is used to grow a decision tree. Random Forest also cannot ensure the necessary diversity of ensembles. In this paper

```
Input:
   TS: Training Set
   L:  Labels of Training Set
n:  Ensemble Size
b:  Ration of Subsamplings from TS
Output:
   E:  Ensemble of Classifiers
   P:  Accuracies of Classifiers
   O:  Outputs of Classifiers
m=length (TS)
For i = 1: n
   Bag = subsample (TS, b)
   Train (Classifier, Bag)
   For j = 1: m
O(i, j) = Test (Classifier, TS(j))
   End
P(i) = Accuracy of Classifier$_i$ //Sum(L==O(i,:));
E(i) = Classifier
End
```

Fig. 1. Pseudo-code of algorithm for training phase of CSBC by the modified Bagging method as the generator of base classifiers.

“Random Forest” algorithm is implemented as a version of Bagging classifier and it is compared with the proposed method. It must be noticed that we modified “Random Forest” before usage.

The Boosting method is inspired by the Hedge(β) algorithm which is an online learning algorithm. This algorithm assigns weights to strategies that are used to predict the outcome of a certain event. At this point we want to relate the Hedge(β) to classifier combination problem. In Freund and Schapire (1997) Boosting is defined as “general problem of creating a very accurate prediction rule by combining rough and moderately inaccurate rules of thumb”. The main idea of Boosting is creating a team of classifiers incrementally (adding one classifier into the team at a time). The classifier that joins the ensemble at step k is trained on a selected dataset from the trained dataset Z. The sampling distribution starts uniformly and progresses toward increasing likelihood of “difficult” data points. Therefore distribution is updated at each step, and the likelihood of each misclassified object from previous step increases. In this section, correspondence with Hedge(β) is transposed. Classifiers in D are events and the data points in Z are strategies in which their probability distribution is updated at each step. This algorithm is inspired by ADAptiveBoosting, and is called AdaBoost. Another version of it is the arc-x4 algorithm, which is the same as the latest version of ADAboost (Kuncheva, 2005).

Giacinto and Roli (2001) proposed clustering and selection method. First they produced a large number of MLP classifiers with different initialization and then partitioned their outputs by a clustering method and chose one classifier from each cluster. Finally, the selected classifiers were considered as an ensemble and majority voting was their aggregation function.

Hamid et al. proposed a framework to develop the combined classifiers. In this framework, a number of trained data-bags are bootstrapped from the trained dataset at first and then a set of weak base classifiers is created; each classifier is trained on a distinct data-bag. After that classifiers are partitioned using the clustering algorithm in order to release similar classifiers in the ensemble and choose a diverse subset of classifiers. In the partitioning phase, the outputs of classifiers on the training dataset are considered as a new feature space. One classifier is selected from each cluster randomly to create the final ensemble. Then different votes are gathered to form an ensemble to produce consensus vote. And then the weighted majority voting mechanism is applied as their aggregation function. The weights are specified based on the accuracies of the base classifiers on training dataset (Parvin et al., 2001).

3. Classifier selection by clustering

The main idea of classifier selection using the clustering method is to use the diverse classifiers obtained from the modified Bagging and Boosting mechanism. A number of classifiers are trained by Bagging or Boosting. The training phase of CSBC is shown in Fig. 2 in which the modified Bagging method is used as the base classifier generator. Then a random classifier is selected from each cluster. This method selects one classifier from each cluster and considers these classifiers as a diverse ensemble. So it performs the traditional Bagging and Boosting (which use all of the classifiers as an ensemble). It is also likely that it selects one of the nearest classifiers to the head of each cluster. This method can produce more diversity ensemble than others that select the classifiers randomly. Pseudo-code of training phase of CSBC, which uses the modified Bagging and modified Boosting methods as the base classifier generator, is shown in Fig. 2

With due attention to Fig. 2, n subset containing b% of the training dataset are bootstrapped at first. The ith dataset bootstrapped with b% is denoted as DB_i which is the ith data-bag. Cardinality of DB_i is equal to $m \times b/100$. After that a classifier is trained into each DB_i. Suppose that C_i is a classifier that was trained into DB_i. Then C_i is tested over the whole training dataset and its accuracy is calculated. The output of the ith classifier over the ith data item TS is a vector which is marked O_{ij}. O_{ijk} is the confidence of the ith classifier that O_j belongs to class k. The output of the ith classifier over the whole of the training dataset is equal to O_i and its accuracy is presented by P_i. The only difference between the proposed algorithms and the Bagging method is related to the b value. In the proposed method b is in [30–100] and in the Bagging method, $b=100$. The training phase of the CSBC method, which uses modified Boosting as a base classifier generator, is shown in Fig. 3. Similar to that in Fig. 2, a subset containing b% of the training dataset is selected. Then the first classifier is trained on this subset. After that the first classifier is tested in the whole training dataset, O1 and O2 are the results. Next the subset containing b% of the training dataset is obtained using the O1; this mechanism is continued until the O_i obtained by O_{i-1}. The difference between the proposed method and the Boosting method is related to the b value. In Boosting, $b=100$ and in the proposed approach b is in [30–100]. Further information on Boosting can be found in Kuncheva (2005). Pseudo-code of the classifier selection-based clustering framework and its illustration are depicted in Figs. 4 and 5 respectively. In the CSBC a dataset of classifiers named DC is created at first. The ith data item of DC is identified as X_i having f features. The pth feature of the ith data item in DC is identified as X_{ip} which is obtained from Eq. (1):

$$X_{ip} = O_{ij}^{k} \tag{1}$$

where j and k are obtained by Eqs. (2) and (3), respectively.

$$j = \lceil p/c \rceil \tag{2}$$

where c is the number of classes.

$$k = p - j \times c \tag{3}$$

Features of the DC dataset are obtained from different opinions over real data items of the under-learning dataset. A new dataset contains n data item where any of them stands for a classifier and N feature where $N=m \times c$. n is a predefined parameter which indicates the number of produced classifiers from Bagging and Boosting. After producing the DC dataset it is partitioned by using the clustering algorithm and its result is some clusters of classifiers. The number of clusters is denoted by r. The outputs of classifiers in a cluster are the same. It means that these classifiers have low diversity. So it is better to use one of them in the final ensemble instead of using all of them. For escaping from outlier

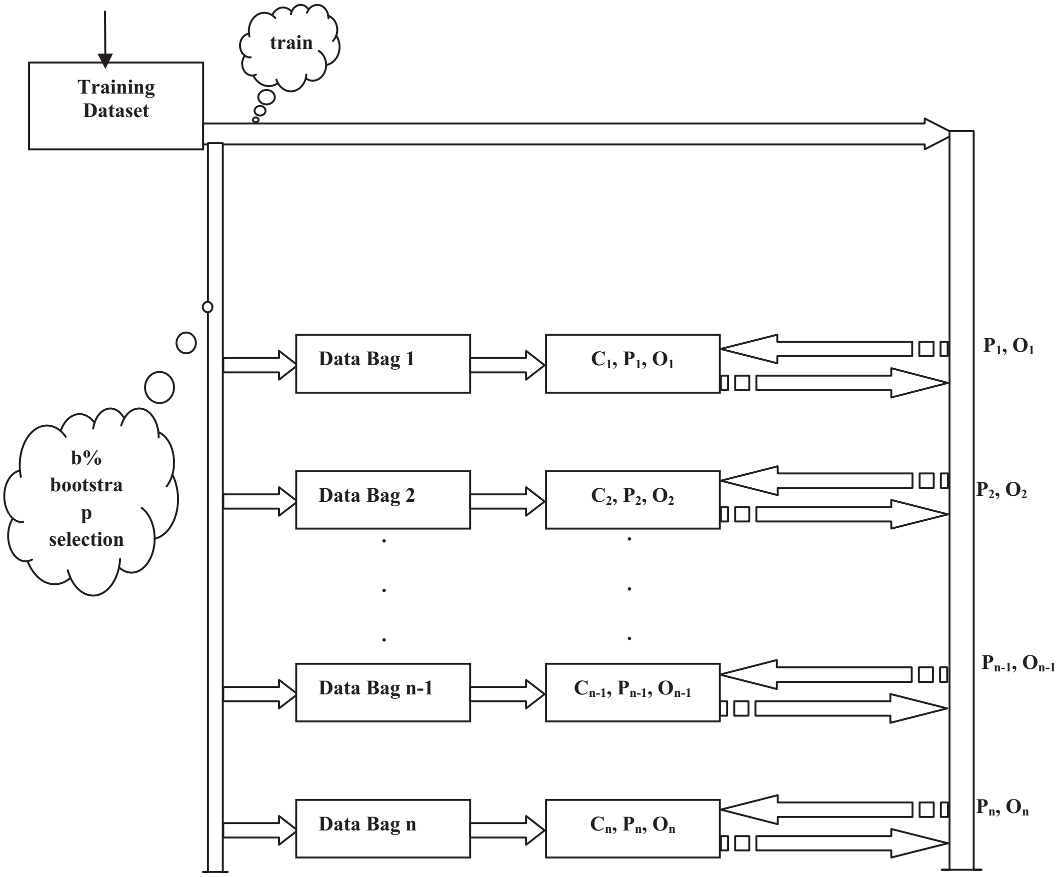


Fig. 2. Training phase of CSBC by the modified Bagging method as the generator of base classifiers.

classifiers, we ignore the clusters that have a number of classifiers under the predefined threshold.

In this paper we assumed the ensemble of n classifiers {C1, C2, C3, …, Cn} is denoted E, and there are c classes. And applying the ensemble over data sample O_i produces the following result:

$$D^j = \begin{bmatrix} O_{1j}^1 & O_{2j}^1 & . & O_{nj}^1 \\ . & . & . & . \\ O_{1j}^{c-1} & O_{2j}^{c-1} & . & O_{nj}^{c-1} \\ O_{1j}^c & O_{2j}^c & . & O_{nj}^c \end{bmatrix} \tag{4}$$

Now the ensemble decides the data sample O_j to belong to class q according to Eq. (5):

$$q = \arg\max_{i=1}^{c} \left| \sum_{k=1}^{n} w_k \times D^j_{i \quad k} \right| \tag{5}$$

where w_j is the effect weight of classifier j, which is obtained optimally (Kuncheva, 2005) according to Eq. (6):

$$w_j = \log \frac{p_j}{1-p_j} \tag{6}$$

P_j indicates the accuracy of classifier j in TS. a tie breaks randomly in Eq. (5). We consider vector L_j for data item O_j. L_{jq} is 1 if O_j belongs to class q and zero otherwise. Now we can calculate the accuracy of classifier CK over TS using Eq. (7):

$$p_k = \frac{\sum_{j=1}^{m} \sum_{i=1}^{c} \left| L_i^j - O_{kj}^i \right|}{c \times m}. \tag{7}$$

4. Experimental study and discussion

Evaluation metric is based on the classifier which is reported in Section 4.1. The next section describes the used datasets and then the settings of experimentations and finally the experimental results are presented.

4.1. Evaluation metric

In this paper the accuracy is taken as the evaluation metric. Are the experiments have been carried out by using a 4-fold cross-validation. The results of this 4-fold cross-validation are repeated in 10 independent runs. It means that to investigate the accuracy of methods on a dataset, e.g. *Iris*, its accuracy is calculated by 4-fold cross-validation and its result is represented by acc_1. This scenario is repeated until reaching acc_{10}. So acc_i, $i \epsilon \{1, 2, \ldots, 10\}$. Accuracy of the method over the *Iris* dataset is equal to the average accuracies acc_i in 10 independent runs.

4.2. Datasets

The proposed approach is examined on 13 different standard datasets and one artificial dataset. We tried to maintain the diversity of the dataset and also their number of classes, features and samples to be true. Usage of large number of diverse datasets can improve the validation of the results and lead to definite results. The information about used datasets is shown in Table 1. Half-Ring datasets are described in Minaei-Bidgoli et al. (2014).

Some of the datasets are marked with star (⁎) in Table 1. They are normalized. In these datasets, all experiments are performed over the normalized features in the starred dataset. It means that

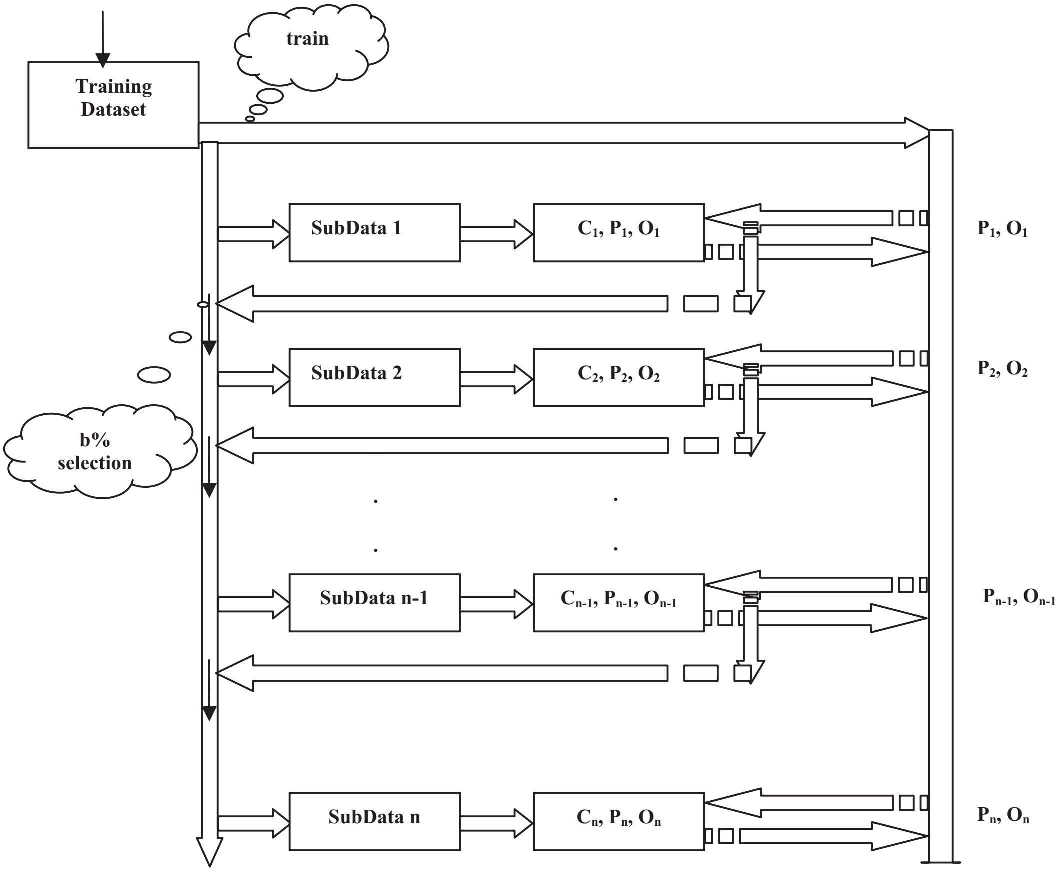
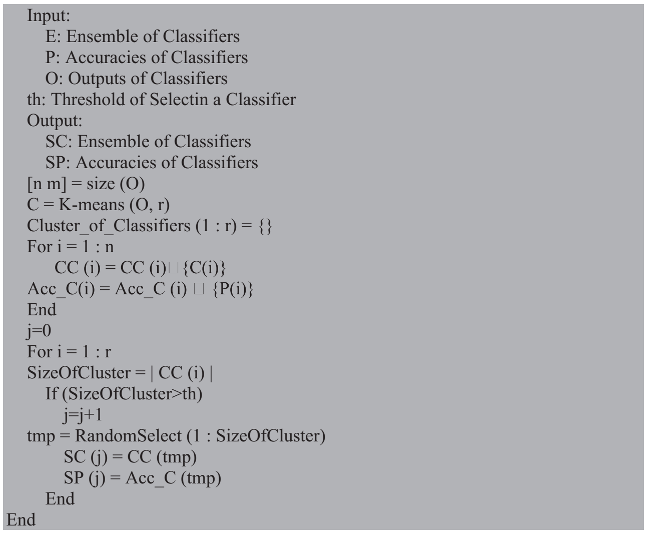


Fig. 3. Training phase of CSBC by the modified Boosting method as the generator of base classifiers.



```
Input:
   E: Ensemble of Classifiers
   P: Accuracies of Classifiers
   O: Outputs of Classifiers
th: Threshold of Selectin a Classifier
Output:
   SC: Ensemble of Classifiers
   SP: Accuracies of Classifiers
[n m] = size (O)
C = K-means (O, r)
Cluster_of_Classifiers (1 : r) = {}
For i = 1 : n
     CC (i) = CC (i)□{C(i)}
Acc_C(i) = Acc_C (i) □ {P(i)}
End
j=0
For i = 1 : r
SizeOfCluster = | CC (i) |
   If (SizeOfCluster>th)
      j=j+1
tmp = RandomSelect (1 : SizeOfCluster)
      SC (j) = CC (tmp)
      SP (j) = Acc_C (tmp)
   End
End
```

Fig. 4. Pseudo-code of the classifier selection-based clustering framework.

each feature is normalized with a mean of 0 and variance of 1, *N* (0, 1). The artificial Half-Ring dataset is depicted in Fig. 6.

4.3. Experimental setting

The measure of decision for each decision tree is based on the Gini measure. The threshold of pruning is set to 2. The classifiers' parameters are fixed during their usages.

All of the used MLPs in our experiments had two hidden layers including 10 and 5 neurons, respectively, in the first and second hidden layers. All of them are trained in 100 epochs.

In experiments the value of parameters *n*, *b* and threshold for accepting a cluster are set to 151, 30, and 2. (i.e. only the clusters with one classifier is dropped down). We use the 4-fold cross-validation in our experiments. We use the *k*-means clustering algorithm with different *k* parameters.

4.4. Experimental results

To see the effect of parameter *r* on the performance of classification over CSBS methods (by Bagging, Boosting and Gianito) with two base classifiers (MLP, DT) see Figs. 7–12.

These figures show the accuracy of different methods by 4-fold cross-validation on some benchmarks. With due attention to these figures, increasing the cluster number parameter r does not always lead to performance improvement. $R=15$ is the best choice for all of the datasets. It means that if $n=151$ then $r=15$ is the best choice for cluster number parameter. In other words, using the 10% of base classifiers in the final ensemble can be a good option. And then a classifier which contains about 10 classifiers is selected from each cluster. So it enables the method to select the classifier that covers other classifiers.

The performance of CSBC by Boosting over some datasets with $n=151$ and different *r* are shown in Figs. 7 and 8, respectively, while MLP and DT are used as the base classifier. The same results are depicted in Figs. 9 and 10 for CSBC using Gianito's method, respectively, while the same classifiers are used as the base classifier. Figs. 11 and 12 finally represent the performances of CSBC by the Bagging method, respectively, while the same classifiers are used as the base classifier. Figs. 13 and 14 depict

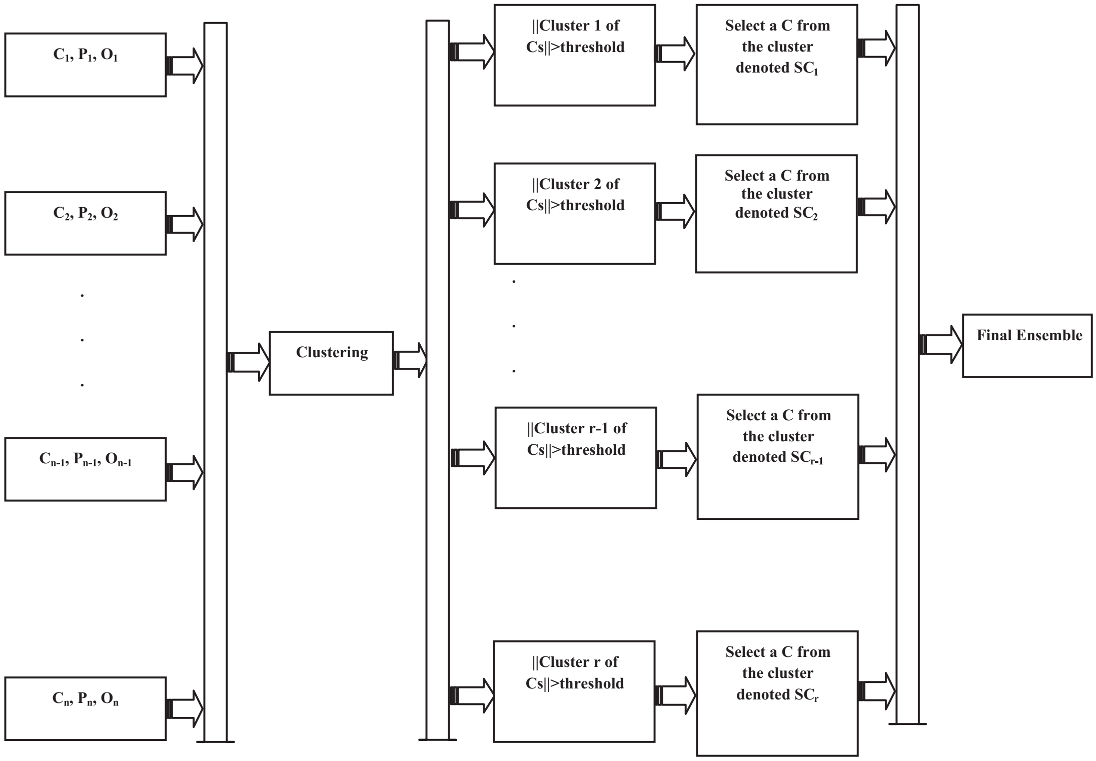

C_1, P_1, O_1
C_2, P_2, O_2
C_{n-1}, P_{n-1}, O_{n-1}
C_n, P_n, O_n
Clustering
||Cluster 1 of Cs||>threshold
||Cluster 2 of Cs||>threshold
||Cluster r-1 of Cs||>threshold
||Cluster r of Cs||>threshold
Select a C from the cluster denoted SC_1
Select a C from the cluster denoted SC_2
Select a C from the cluster denoted SC_{r-1}
Select a C from the cluster denoted SC_r
Final Ensemble

Fig. 5. The classifier selection-based clustering framework.

Table 1
Details of used dataset.

Dataset name	# of Data items	# of Features	# of Classes	Data distribution per classes
Breast Cancer*	683	9	2	444-239
Balance Scale*	625	4	3	288-49-288
Bupa*	345	6	2	145-200
Glass*	214	9	6	70–76–17–13–9–29
Galaxy*	323	4	7	51-28-46-38-80-45-35
Half-Ring*	400	2	2	300-100
SAHeart*	462	9	2	160-302
Ionosphere*	351	34	2	126-225
Iris*	150	4	3	50-50-50
test Monk1	412	6	2	216-216
test Monk2	412	6	2	216-216
test Monk3	412	6	2	216-216
train Monk1	124	6	2	62-62
train Monk2	169	6	2	105-64
train Monk3	122	6	2	62-60
Wine*	178	13	3	59-71-48
Yeast*	1484	8	10	463-5-35-44-51-163-244-429-20-30

the averaged accuracies over all 14 different datasets. Fig. 15 shows the effect of sampling rate over the performance of two different proposed methods.

The results of CSBC by Gianoto's method and the same base classifiers are shown in Figs. 9 and 10.

The performance of the CSBC method by Bagging and the same base classifiers is represented in Figs. 11 and 12.

Figs. 13 and 14 depict the averaged accuracies over all 14 different datasets. Fig. 13 represents the performances of the proposed framework by using MLP as the base classifier and Fig. 14 represents the performances of the proposed framework by using DT as the base classifier.

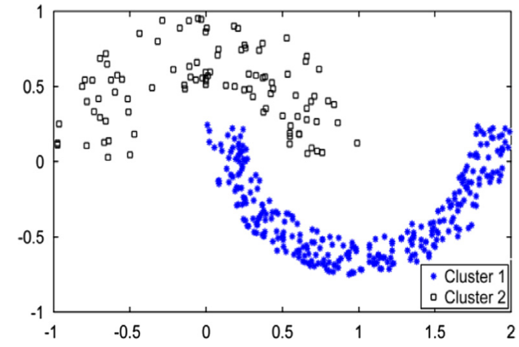


Fig. 6. Half-Ring dataset.

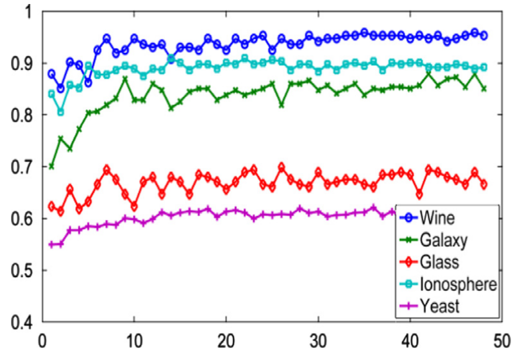


Fig. 7. The performance of CSBC by the modified Boosting ensemble over some datasets with $n=151$ and different r and MLP as base classifier.

As represented in Figs. 13 and 14, using Bagging is better than Boosting and Giacinto and Roli's ensemble to generate base classifiers. Also it is better to use $r=33$ for all 14 ensembles.

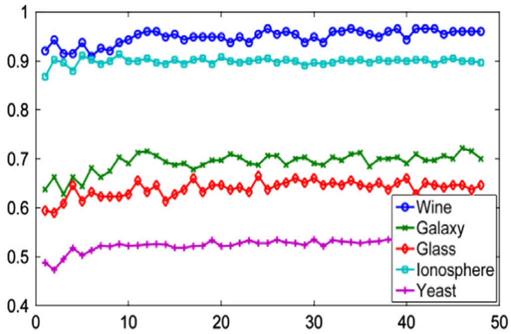


Fig. 8. The performance of CSBC by the modified Boosting ensemble over some datasets with $n=151$ and different r and DT as base classifier.

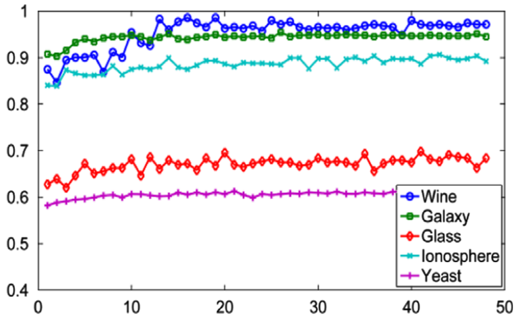


Fig. 9. The performance of Gianito's ensemble method over some datasets with $n=151$ and different r and MLP as base classifier.

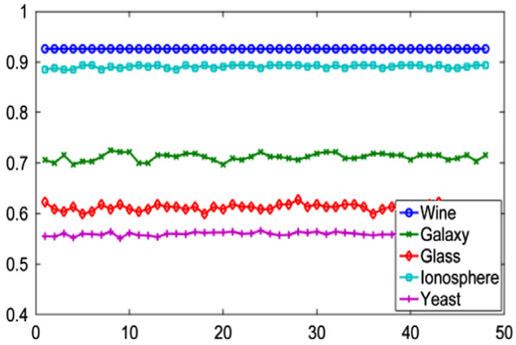


Fig. 10. The performance of Gianito's ensemble method over some datasets with $n=151$ and different r and DT as base classifier.

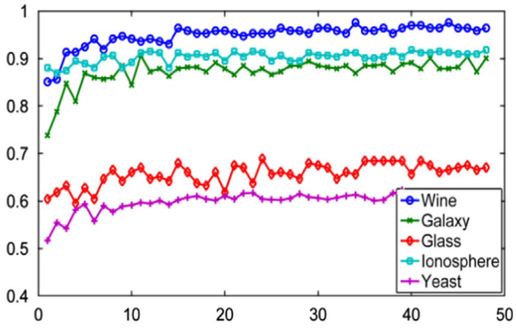


Fig. 11. The performance of CSBC by the modified Bagging ensemble over some datasets with $n=151$ and different r and MLP as base classifier.

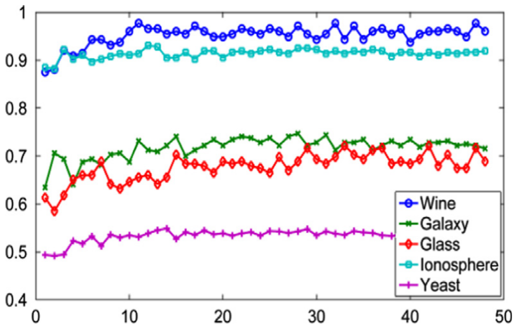


Fig. 12. The performance of CSBC by the modified Bagging ensemble over some datasets with $n=151$ and different r and DT as base classifier.

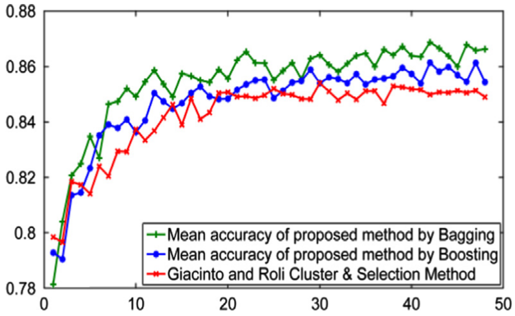


Fig. 13. The performance of CSBC methods averaged over 14 datasets of Table 1 with $n=151$ and different r and MLP as base classifier.

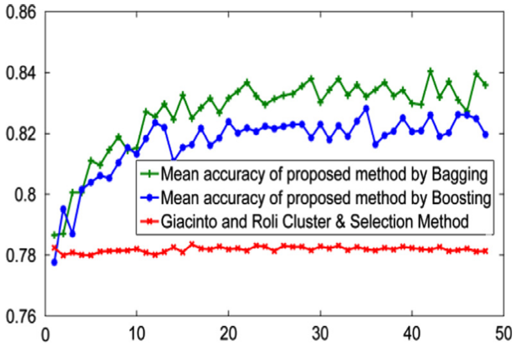


Fig. 14. The performance of CSBC methods averaged over 14 datasets of Table 1 with $n=151$ and different r and DT as base classifier.

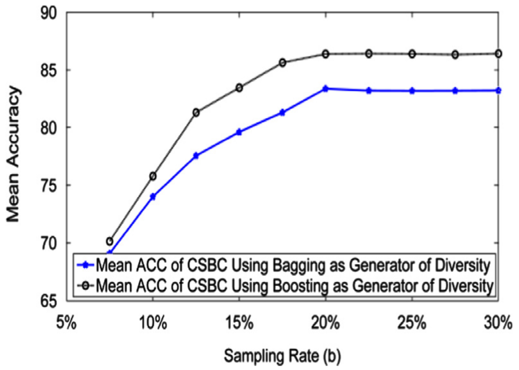


Fig. 15. The performance of the proposed CSBC methods averaged over 14 datasets of Table 1 with $n=151$ and different $r=33$ and different b.

In other words, it is better to use the 22% of base classifiers in the final ensemble.

Comparison between Figs. 13 and 14 shows that using a decision tree as a base classifier can lead to increasing the gap between three approaches in generating an ensemble of base classifiers. Because a decision tree is sensitive to its training set, so the use of a decision tree as the base classifier is consistent with the Bagging mechanism.

You can see the effect of sampling rate over the two proposed methods. In Fig. 15 to reach these results, we use the decision tree

Table 2
Comparison between the results obtained by applying different ensemble methods considering DT as the base classifiers.

	AdaBoost	Arc-X4	Random Forest	Classifier selection By RF	Classifier selection By Arc-X4	Cluster and selection
Breast Cancer*	96.19	95.74	96.32	**96.47**	95.05	93.68
Balance Scale*	91.52	94.44	93.60	**94.72**	94.24	94.44
Bupa*	66.96	70.64	72.09	**72.97**	66.28	64.53
Glass*	70.09	65.04	**70.28**	**70.28**	62.26	60.85
Galaxy*	71.83	70.59	**73.07**	72.45	70.28	70.94
Half-Ring*	**97.25**	**97.25**	95.75	**97.25**	95.75	95.75
SAHeart*	67.32	70.00	71.30	**72.61**	69.70	68.04
Ionosphere*	91.17	90.31	**92.31**	91.45	89.74	87.64
Iris*	94.67	**96.62**	95.27	**96.62**	95.95	94.59
Monk problem1**	98.03	98.11	97.49	**98.76**	97.37	98.34
Monk problem2**	91.65	97.01	86.64	**97.62**	86.73	97.14
Monk problem3**	95.51	87.29	96.92	**96.97**	96.34	87.31
Wine*	96.63	96.07	**97.19**	**97.19**	95.51	92.61
Yeast*	**54.78**	53.17	53.98	53.98	52.09	54.51
Average	84.54	84.45	85.16	**86.38**	83.38	82.88

* Indicates that the dataset is normalized. 4-Fold cross-validation is used for performance evaluation.
** Indicates that the trained and test sets are predefined and averaged over 10 independent runs is reported.

Table 3
Comparison between the results obtained by applying different ensemble methods considering MLP as the base classifiers.

	AdaBoost	Arc-X4	Bagging	Classifier selection by Bagging	Classifier selection by Arc-X4	Cluster and selection
Breast Cancer*	96.49	**97.06**	96.91	96.91	96.47	96.19
Balance Scale*	93.12	93.27	91.99	91.35	92.95	**95.75**
Bupa*	68.41	70.06	71.22	**72.09**	68.02	71.98
Glass*	66.36	66.04	66.98	**67.45**	66.04	67.05
Galaxy*	85.14	**87.00**	85.62	85.62	84.52	**87.00**
Half-Ring*	**97.25**	**97.25**	95.75	**97.25**	**97.25**	**97.25**
SAHeart*	69.48	**73.04**	72.39	71.52	71.09	70.18
Ionosphere*	89.17	90.03	88.51	**90.31**	87.64	88.51
Iris*	94.67	96.62	96.62	**97.97**	97.33	93.33
Monk problem1**	**98.99**	98.06	92.23	98.43	97.87	98.34
Monk problem2**	87.48	87.35	85.68	**87.41**	87.23	87.21
Monk problem3**	96.84	97.09	95.87	**97.33**	96.99	96.77
Wine*	94.38	96.59	96.06	**97.19**	95.51	95.23
Yeast*	59.50	60.85	**61.19**	**61.19**	60.85	60.56
Average	85.52	86.45	85.50	**86.57**	85.70	86.10

* Indicates that the dataset is normalized. 4-Fold cross-validation is taken for performance evaluation.
** Indicates that the trained and test sets are predefined and averaged over 10 independent runs is reported.

as the base classifier. As is obvious, if b is a very low value, the performance is very weak, and a very high value of b cannot improve the performance. It even causes to decrease the performance for values after 40%.

Table 2 shows the averaged accuracies obtained from different ensembles by DT as the base classifier. Table 3 shows the accuracies obtained from the same ensemble methods presented in Table 1 by MLP as the base classifier. The parameter $r=2$ is used in both tables.

If we select only at most 22% of the base classifiers, the accuracy of the ensemble can be better than the full ensemble. Also, it is better than the Boosting method and the proposed method based on Boosting.

Because the selected classifiers in this manner have different outputs, they are more suitable than ensemble of all them. It is noticeable that the Boosting method is diverse enough to be a complete ensemble. And in a Boosting ensemble, each member can cover the errors of the previous members. In the CSBC method, a classifier is selected from a cluster randomly. It means that a random classifier is selected from each part. Table 4 shows the performance of the three methods for selecting a classifier from a partition. These methods are RS, TAMS and TNTCCS. RS stands for "random selection", TAMS stands for "the most accurate selection". In TAMS, the most accurate classifier is selected from a cluster as the index classifier of that cluster. In order to guarantee the accuracy of the classifier, we can test that classifier over the whole of the training dataset. TNTCCS stands for "the nearest-to-cluster-center selection". In this method the nearest classifier to cluster center is selected. The measure of distance is the same measure used in the partitioning algorithm.

As mentioned in Table 4, the TNTCCS method is the best one, followed by RS. Although we expected TMAS to be the best method, the experimental results show it is the worst one.

5. Conclusion and future works

In this paper a new approach is proposed to improve the performance of classification. In the proposed approach we used the modified version of Bagging as the base classifier selection. We used the k-means method to partition the base classifiers and a random classifier is selected from each partition. A set contains all of the selected classifiers considered as an ensemble. Because each cluster is selected based on the classifiers' output, it is likely that we choose one classifier from each cluster. And then a diverse ensemble is created from them. This method is better than the traditional Bagging and Boosting methods, which use all of the

Table 4
Effect of usage of the new methods based on which a classifier is selected from a cluster.

	CSBC By Bagging and RS	CSBC by Bagging and TMAS	CSBC by Bagging and TNTCCS
Breast Cancer*	96.91	95.73	**97.36**
Balance Scale*	91.35	91.51	**91.98**
Bupa*	**72.09**	71.87	71.21
Glass*	67.45	65.98	**68.67**
Galaxy*	**85.62**	85.34	84.51
Half-Ring*	97.25	**97.72**	97.18
SAHeart*	71.52	71.49	**72.13**
Ionosphere*	90.31	**91.07**	90.01
Iris*	97.97	96.06	**98.21**
Monk problem1**	98.43	**100.00**	**100.00**
Monk problem2**	87.41	86.83	**87.89**
Monk problem3**	97.33	98.10	**98.18**
Wine*	97.19	98.21	**98.55**
Yeast*	61.19	60.91	**61.62**
Average	86.57	86.49	**86.96**

RS stands for “random selection”; TMAS stands for “the most accurate selection”; TNTCCS stands for “the nearest-to-cluster-center selection”.

classifiers as an ensemble. Also, the TNTCCS is a better method than RS. It can produce an ensemble with high diversity.

Using the decision tree as a base classifier can increase the gap between the tree methods in generating an ensemble of base classifiers. Because the decision tree is sensitive to its trained set, we use it as a base classifier, which is consistent with the Bagging method.

When we use the 22% of the base classifier, the accuracy of their ensemble is better than their full ensemble and also it is better than the Boosting method. As a result, using the 22% of the base classifier can be a good option.

The parameters and the aims of this paper are as follows: (1) Clustering of classifiers is better than ensembles created by the Bagging and Boosting methods. (2) Clustering of classifiers work well by Bagging; however, it does not work by Boosting as well as Bagging.

Our future work is to further study the variance of the method, since it is said that Bagging can reduce variance and Boosting can simultaneously reduce variance and error rate.

References

Alizadeh, H., Parvin, H., Alinejad-Rokny, H., Mozayani, N., 2011. Surface matching degree. Aust. J. Basic Appl. Sci. 5 (9), 653–660.

Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F., 2011. Combining evaluation metrics via the unanimous improvement ratio and its application to clustering tasks. J. Artif. Intell. Res. 42, 689–718.

Breiman, L., 1996. Bagging predictors. J. Mach. Learn. 24 (2), 123–140.

Dasgupta, S., Ng, V., 2010. Which clustering do you want? Inducing your ideal clustering with minimal feedback. J. Artif. Intell. Res. 39, 581–632.

Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to Boosting. J. Comput. Syst. Sci. 55 (1), 119–139.

Giacinto, G., Roli, F., 2001. An approach to the automatic design of multiple classifier systems. Pattern Recognit. Lett. 22, 25–33.

Kuncheva, L.I., 2005. Combining Pattern Classifiers. Methods and Algorithms. Wiley, New York.

Kuncheva, L.I., Whitaker, C., 2003. Measures of diversity in classifier ensembles and their relationship with ensemble accuracy. Mach. Learn. 14 (2), 181–207.

Kurland, O., Krikon, E., 2011. The opposite of smoothing: a Language model approach to ranking query-specific document clusters. J. Artif. Intell. Res. 41, 367–395.

Minaei-Bidgoli, B., Topchy, A.P., Punch, W.F., 2004. Ensembles of partitions via data resampling. In: ITCC, pp. 188–192.

Minaei-Bidgoli, B., Parvin, H., Alinejad-Rokny, H., Alizadeh, H., Punch, W.F., 2014. Effects of resampling method and adaptation on clustering ensemble efficacy. Artif. Intell. Rev. 41 (1), 27–48.

Parvin, H., Minaei-Bidgoli, B., Shahpar, H., 2001. Classifier selection by clustering. In: Proceedings of the 3rd Mexican Conference on Pattern Recognition (MCPR2011). LNCS, Springer, Heidelberg, ISSN: 0302-9743.

Parvin, H., Minaei-Bidgoli, B., Alinejad-Rokny, H., Ghatei, S., 2011a. An innovative combination of particle swarm optimization, learning automaton and great deluge algorithms for dynamic environments. Int. J. Phys. Sci. 6 (22), 5121–5127.

Parvin, H., Helmi, H., Minaie-Bidgoli, B., Alinejad-Rokny, H., Shirgahi, H., 2011b. Linkage learning based on differences in local optimums of building blocks with one optima. Int. J. Phys. Sci. 6 (14), 3419–3425.

Parvin, H., Alinejad-Rokny, H., Parvin, S., 2011c. Divide and conquer classification. Aust. J. Basic Appl. Sci. 5 (12), 2446–2452.

Parvin, H., Alinejad-Rokny, H., Parvin, S., 2013a. A classifier ensemble of binary classifier ensembles. Int. J. Learn. Manag. Syst. 1 (2), 37–47.

Parvin, H., Minaei-Bidgoli, B., Alinejad-Rokny, H., Punch, W.F., 2013b. Data weighing mechanisms for clustering ensembles. Comput. Electr. Eng. 39 (5), 1433–1450.

Parvin, H., Alinejad-Rokny, H., Minaei-Bidgoli, B., Parvin, S., 2013c. A new classifier ensemble methodology based on subspace learning. J. Exp. Theor. Artif. Intell. 25 (2), 227–250.

Parvin, H., Minaei-Bidgoli, B., Alinejad-Rokny, H., 2013d. A new imbalanced learning and dictions tree method for breast cancer diagnosis. J. Bionanosci. 7 (6), 673–678.

Parvin, H., Alinejad-Rokny, H., Seyedaghaee, N., Parvin, S., 2013e. A heuristic scalable classifier ensemble of binary classifier ensembles. J. Bioinform. Intell. Control 1 (2), 163–170.

Peña, J.M., 2011. Finding consensus Bayesian network structures. J. Artif. Intell. Res. 42, 661–687.

Rezaei, Z., Alizadeh, H., Parvin, S., Alinejad-Rokny, H., 2011. An extended MKNN modified *K*-nearest neighbor. J. Netw. Technol. 4 (2), 162–168.

Yang, T., 2006. Computational verb decision trees. Int. J. Comput. Cogn. 4 (4), 34–46.