Expert Systems with Applications 157 (2020) 113482

Contents lists available at ScienceDirect

Expert Systems with Applications

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

HBoost: A heterogeneous ensemble classifier based on the Boosting method and entropy measurement

Hamid Reza Kadkhodaei [a], Amir Masoud Eftekhari Moghadam [a,*], Mehdi Dehghan [b]

[a] Faculty of Computer and Information Technology, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[b] Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 18 December 2019
Revised 7 March 2020
Accepted 24 April 2020
Available online 28 April 2020

Keywords:
Ensemble learning
Heterogeneous models
Boosting classifier
Ensemble pruning
Ensemble diversity

ABSTRACT

In recent years, ensemble classifiers have attracted a lot of attention in the field of machine learning. The main challenges with these classifiers are 1) to select the base classifiers and 2) to combine the outputs. The key point for an ensemble to be successful is the diversity and accuracy of the base classifiers. This paper proposes a heterogeneous Boosting-based ensemble classifier (HBoost) which is inspired by Boosting algorithm and aims at increasing the diversity by recruiting distinct learning algorithms. In this approach, for each learning algorithm, several classifiers are generated by using the Boosting method and a matrix of heterogeneous classifiers is formed. Since all generated classifiers may not be appropriate for generating the final output, the subset of those that make the most diverse and accurate classifiers is selected and the rest of the classifiers are eliminated. Finally, the classifiers in the subset are combined based on the weight assigned to each classifier. HBoost is a self-configured algorithm i.e., unlike existing methods, it does not require a human expert to specify the type of base classifiers manually. It can also automatically determine the optimum number of base classifiers which should be combined. For evaluating the HBoost, two evaluation metrics have been used: Accuracy and Geometric Mean which are the most popular metrics to measure the performance of a classifier. We compare the proposed approach in three different scenarios 1) all the base classifiers are in the ensemble, 2) four traditional ensemble methods including Bagging, Boosting, Stacking, and StackingC, 3) two state-of-the-art approaches in the literature including Random Forest and Classifier Subset Selection (CSS). HBoost is performed on 20 datasets from the UCI repository and experimental results reveal how adequate the HBoost is in comparison to other approaches.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Classification is a kind of supervised machine learning that addresses the problem of building a model, from correctly classified dataset, which is able to deal with new unclassified instances and assign them a predicted class (Mendialdua, Arruti, Jauregi, Lazkano, & Sierra, 2015).

While there are many algorithms for classification, experimental studies show that it is practically not feasible to construct a perfect classifier which is universally superior to others and have a good performance for any dataset (Polikar, 2006). In other words, there are lots of classification algorithms and each of them may outperform the others for different datasets. This is the so-called "No free lunch theorem" (Wolpert & Macready, 1997). One idea to improve the prediction performance of the classification task is to construct a set of classifiers and combine their outputs. In the literature, this approach has been named "Ensemble Classifier", "Classifier Fusion", "Combined classifier", "Multiple-classifier systems (MCS)", "Ensemble Learning", and "Hybrid classifiers" (Woźniak, Graña, & Corchado, 2014). The main idea behind an ensemble methodology is to combine a set of models in order to obtain a better composite global model, with more accurate and reliable outputs than can be obtained from using a single model (Rokach, 2009). In many applications, ensemble classifiers are the most accurate classifiers available (Palit & Reddy, 2012).

The individual classifiers contained in an ensemble are called base classifiers. If the base classifiers are all of the same algorithm (for example, all of the base classifiers are of the decision tree type), the ensemble is known as *homogeneous* (Bramer, 2013). In contrast, in *heterogeneous* models, the learning algorithm of the base classifiers are different (van Rijn, Holmes, Pfahringer, & Vanschoren, 2018).

There are two fundamental challenges in ensemble classifiers:

- How to choose base classifiers

* Corresponding author.
E-mail addresses: hr.kadkhodaei@qiau.ac.ir (H.R. Kadkhodaei), Eftekhari@qiau.ac.ir (A.M.E. Moghadam), dehghan@aut.ac.ir (M. Dehghan).

https://doi.org/10.1016/j.eswa.2020.113482
0957-4174/© 2020 Elsevier Ltd. All rights reserved.

The first phase for the construction of an ensemble classifier is the generation of the base classifiers. In homogeneous ensemble, since all base classifiers are from the same family, one classification algorithm should be specified. This is done by a human expert. It means that there should be an expert that knows the nature of the dataset and is aware of the classification algorithms and probably after some experiments, decides which type of the learning algorithm is suitable for classifying the dataset. On the other hand, although there are different learning algorithms for heterogeneous ensemble, the problem still exists, because despite the different types of classifiers, some of them may not train accurately for the given dataset (because the corresponding learning algorithm is not suitable for data). As a result, they can reduce the prediction performance of the ensemble.

- How to combine the output of base classifiers?

Once the base classifiers are trained with input instances and generate their prediction for an unlabeled instance, there should be a strategy to combine their outputs. There are different approaches in the literature for combining the prediction of base classifiers. Some of them are Majority voting, Plurality voting, Weighted voting, and using a meta classifier (Sultana & Islam, 2020).

The most effective criteria for increasing the quality of the ensemble classifiers are the independency of the base classifiers and the diversity between them (Cavalcanti, Oliveira, Moura, & Carvalho, 2016; Woźniak et al., 2014).

Two classifiers C1 and C2 are independent if the predictions of C1 give no information about the predictions of classifier C2 (Russell, 2010). Given a set of L independent classifiers, the error rate of the final model with simple voting is given by Eq. (1).

$$error\ rate = \sum_{i=\lfloor \frac{L}{2} \rfloor + 1}^{L} \binom{L}{i} \beta^i (1-\beta)^{L-i} \tag{1}$$

where β is the error rate of each base classifier (for example, given 15 classifiers each with an error rate equal to 0.35, the final error rate for the combination of them will be 0.00017 which is more than two thousand times better than each classifier). In Eq. (1), it is assumed that all of the base classifiers are independent. Unfortunately, this rarely happens for real datasets. That is, most of the time, instances in a dataset have correlation and this causes that base classifiers would have a dependency on each other.

In order for an ensemble classifier to be successful, it is required that base classifiers are both accurate (it is enough to be only slightly better than random guessing) and diverse (Golestani, Ahmadian, Amiri, & JahedMotlagh, 2007; Jankowski, 2013; van Rijn et al., 2018). Several methods are used to make diverse classifiers.

In homogeneous ensembles, the same learning algorithm is used for the base classifiers and diversity is achieved from training the base classifiers with different random subset of the input instances which is obtained by sampling, random selection of the features or both of them. When the sampling is performed with replacement, this method is called Bagging (short for Bootstrap Aggregation) which was presented in (Breiman, 1996). In Bagging, the final prediction is generated by combining the base classifiers' output either by simple voting or averaging. Another kind of homogeneous ensemble classifier is Boosting (Schapire, 1990) which carries the main idea of training the base classifiers sequentially, each trying to correct its predecessor (Géron, 2019). Unlike Bagging which the instances for training of the base classifiers are drawn independently, in Boosting, each data distribution depends directly on the previous steps (Jankowski, 2013). Indeed, the first classifier is trained with the input instances and the misclassified instances are more likely to be selected as the second classifier's input instances. This process continues until the specific criterion would satisfy. (for example, the number of iterations). The method for combining the output of the base classifiers in Boosting is weighted voting. Bagging is regarded as a variance reduction technique. In contrast, Boosting can reduce both classification bias and variance (Xing & Liu, 2020).

In heterogeneous ensembles, different learning algorithms are used to achieve a diverse set of classifiers. Stack Generalization (briefly Stacking) (David H. Wolpert, 1992) is the most popular heterogeneous ensemble classifier. In Stacking, different type of classifiers are considered as the base classifiers i.e., the base classifiers are not from the same family. An extra classifier, called Meta-Classifier, is used to combine the output of base classifiers. It means that the prediction of each base classifier is considered as a feature for the Meta-Classifier.

When employing a homogeneous ensemble, choosing an appropriate learning algorithm is an important decision. For heterogeneous ensembles this is even more true, as we have to choose a set of learning algorithms (van Rijn et al., 2018).

It is worthwhile to mention that although using many classifiers may seem more effective, selecting a subset of them can reduce the computational cost and improve the performance, assuming that the selected classifiers are diverse and independent between them (Mendialdua et al., 2015). In other words, if some inaccurate classifiers in an ensemble generate similar outputs, it is better to eliminate some of them to increase the diversity among the base classifiers.

In this paper, we propose a heterogeneous Boosting-based ensemble classifier, HBoost, which uses multiple classification algorithms for the base classifiers of the Boosting method. HBoost considers both accuracy and diversity for constructing the ensemble model and benefits the advantages of both Boosting method and heterogeneous classifiers. In other words, we use multiple learning algorithms to increase the diversity between the classifiers. Afterward, for each algorithm, we construct a model that is inspired by Boosting to reduce the bias. Finally, some heterogeneous classifiers are combined to generate the final output. The major contributions of the proposed method are:

1- A novel heterogeneous Boosting-based ensemble classifier which, unlike the Boosting, does not rely on a single learning algorithm and benefits the advantages of multiple learning algorithms.
2- A new approach for considering both accuracy and ensemble diversity to prune the base classifiers in order to increase the performance of the ensemble model
3- A self-configured ensemble classifier that does not require an expert to manually specify the type of the base classifiers and automatically selects the best set of classifiers which should be combined together to generate the final output.

For evaluating the proposed method, we have applied it on 20 standard datasets from the UCI repository and have compared the obtained results with the results of ten single classifiers, four traditional ensemble models and two state-of-the-art approaches in the literature. The results obtained on the datasets show the validity of the HBoost.

The rest of this paper is divided into five sections and is organized as follows: we describe some recent studies in ensemble classifiers in Section 2. The proposed method HBoost is elaborated in Section 3, while Section 4 demonstrates the experimental results and shows the comparisons with different approaches. Finally, Section 5 Concludes this study along with some future research directions.

2. Related work

There is a wide range of approaches to construct ensemble models. In this study, we focus on both homogeneous and hetero-

geneous ensembles as our approach applies heterogeneous model on the Boosting method which is itself a homogeneous ensemble classifier.

Two popular approaches for homogeneous ensemble classifiers are Bagging (Breiman, 1996) and Boosting (Schapire, 1990). A Vote-Boosting ensemble is presented in (Sabzevari, Martínez-Muñoz, & Suárez, 2018) which has proposed a new emphasis function for instances. Obregon et.al published a paper that investigates the simplification of production rules in the Boosting method for imbalanced datasets (Obregon, Kim, & Jung, 2019). They applied the decision tree as the base classifier. Another research on skew training data is presented in (Seiffert, Khoshgoftaar, Hulse, & Napolitano, 2010) that proposes a hybrid sampling/Boosting method, called RUSBoost, which is a faster alternative to SMOTE-Boost (Chawla, Lazarevic, Hall, & Bowyer, 2003). The application of multi-resolution Boosting for the ensemble model is investigated in (Reddy & Park, 2011). Reddy and Park have varied the complexity of the weak learners as the Boosting iteration progresses. Their algorithm starts with a simple learner (for example decision stump –a one-level decision tree-) and at each iteration, the complexity is increased (the number of splits in the tree). They have demonstrated the advantages of their framework in both classification and regression problems.

There are also several papers that consider more than one algorithm for Bagging Ensemble (Lertampaiporn, Thammarongtham, Nukoolkit, Kaewkamnerdpong, & Ruengjitchatchawalya, 2013; Zhao, Jiang, & Xu, 2010). Coelho et. al. (Coelho & Nascimento, 2010) propose a new method for recruiting heterogeneous algorithms in the Bagging method by applying the genetic algorithm. They have used five different classifiers for the base learners and mapped the presence of these learners to different chromosomes. They have reached better accuracy and Q-Statistics for 20 used datasets.

The authors of (de Souza & Matwin, 2012) argued that considering a single learning algorithm for all iterations of the Boosting method can reduce the performance of the model. They have proposed a dynamic method that changes the learning algorithm at each iteration of Boosting. They have searched the "best" classifier for the assigned data of the iteration and by proposing a new loss function have improved the performance of the Boosting.

A lot of research works have been conducted on dynamic ensemble selection (DES) techniques (Cruz, Sabourin, & Cavalcanti, 2017, 2018; Cruz, Sabourin, Cavalcanti, & Ing Ren, 2015; Dos Santos, Sabourin, & Maupin, 2008; Porwik, Doroz, & Wrobel, 2019). The main objective of DES is to select one or more competent classifiers for the classification of each new test sample (Cruz, Oliveira, Cavalcanti, & Sabourin, 2019). A dynamic approach for the pruning of base classifiers –called FIRE-DES- is suggested in (Oliveira, Cavalcanti, & Sabourin, 2017). The approach (1) detects the indecision regions for a test sample (2) pre-selects classifiers with decision boundaries within the region of competence of the test sample and (3) select the best classifier(s) for classifying the test sample.

There are other categories of articles in the field of selection of appropriate classifiers from a set of base classifiers. Selecting and pruning the classifiers is used to reduce the complexity of the final model as well as increase the output quality(Dai, Ye, & Liu, 2017). Pruning aims to build classifiers that err on different samples (Sabzevari et al., 2018).

The effectiveness of the diversity in selecting the base classifiers of Stack Generalization is investigated in (Mendialdua et al., 2015). The paper proposes an approach called classifier subset selection (CSS) which uses heterogeneous classifiers. The authors have argued that although using heterogeneous classifiers can increase the potential diversity, selecting all of them might decrease the overall quality of the model. They convert the problem of selecting the best base classifiers to the Feature Subset Selection (FSS) problem. Afterward, by applying an evolutionary algorithm called Estimation of Bayesian Network Algorithm –EBNA– have found the best combination of the base classifiers. An enhanced ensemble-based classifier with Boosting method is presented in (Volna & Kotyrba, 2017). The authors focused on some tasks to construct an enhanced classifier. They have, firstly, studied the influence of learning patterns "shape" on ability of neural network to adapt and have eliminated the irrelevant features from the training set using Hebbian network. Afterwards, they used neural network algorithm to construct some weak classifiers. They argued that for their proposed ensemble, the diversity of the weak classifiers is much more important in comparison with their accuracy. For the sake of increasing the diversity, they proposed three methods including doubling, shuffling, and filtering to enhance the diversity of the weak classifiers. Their method is performed on MINST dataset of handwritten digits (LeCun, Cortes, & Burges, 2014) and the obtained results reveal the superiority of their method in comparison to different approaches for the MNIST database. (Kadkhodaei & Moghadam, 2016) have applied a classifier pruning technique and find the most diverse combination of the individuals by applying the genetic algorithm for the Stack Generalization. Each combination of the classifiers has been mapped to a chromosome and then the chromosome with the highest fitness function (entropy) is explored. The output of each constituent classifier of the found chromosome is given to the meta classifier as a feature to train it and build the final model. Qun Dai et al. (Dai et al., 2017) propose three new pruning measures SDAcc, DFTwo and AccRein to employ diversity and accuracy simultaneously. They argued that two important properties of an ensemble are diversity and accuracy which are closely interrelated to each other and it is better to consider them concurrently while most of the pruning methods consider them separately.

3. HBoost: A heterogeneous Boosting ensemble

The main objective of this paper is to utilize heterogeneous classifiers as the base classifiers of the Boosting method. To this end, in this section, we first review the AdaBoost.M1 algorithm and then propose our approach HBoost.

3.1. AdaBoost.M1

As described in the introduction, Boosting is an ensemble classifier that constructs the final model iteratively. AdaBoost.M1 is the most popular implementation of the Boosting method proposed in (Freund & Schapire, 1997). Suppose $\mathscr{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ be the dataset where each member (x_i, y_i) contains $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \cdots, k\}$. Algorithm 1 describes the steps of AdaBoost.M1.

In this approach, once the learning algorithm of the base classifiers is manually selected by a human expert, the ensemble model grows iteratively and at each iteration, a new classifier is appended to the model. In the first step, all instances are given equal weights and a classifier with the selected learning algorithms is trained. The weighted error rate is thereupon calculated for the dataset (line 6) and then the classifier's weight is computed with the formula in (line 11). In the next step, the distribution of instances is updated based on the obtained weight for the classifier with formula in (line 12). Then a new classifier with the calculated distribution of instances is trained. This process continues until a criterion is satisfied (for example, the number of iterations). The final model is built from the weighted sum of the constructed classifiers (line 14). It is clear that the classifiers with higher weights will have more influence on the final output.

Boosting considers only one learning algorithm for the construction of the base classifiers, thereby selecting the type of base classifier requires a human expert to be aware of the nature of data. Moreover, since input instances for constructing the classifier at each step are dependent on each other, base classifiers will have a high correlation to each other and this reduces the diversity among them.

Algorithm 1 (*Adaboost.M1 for multi-class datasets (Freund & Schapire, 1997)*).

Input: Dataset $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ $y_i \in Y = \{1, \ldots, k\}$
Learning algorithm ***Learner***
Number of iterations T

Initialize the weight vector $w_i^1 = \frac{1}{N}, 1 \le i \le N$
$C = empty$
for $t = 1$ to T **do**
$\boldsymbol{p}^t = \frac{w^t}{\sum_{i=1}^{N} w_i^t}$
Call Learning algorithm ***Learner***; providing it with the distribution $\boldsymbol{p}^t$; get back $c^t: X \to Y$.
calculate error rate of $c^t: \epsilon^t = \sum_{i=1}^{N} p_i^t [\![c_t(x_i) \neq y_i]\!]$
if $(\epsilon^t > 0.5)$
T = t − 1
break;
end if
$\beta^t = \frac{\epsilon^t}{1-\epsilon^t}$, $C = C \cup c^t$
$w_i^{t+1} = w_i^t . \beta_t^{1-[\![c_t(x_i) \neq y_i]\!]}$
end for
Output: $C_f(x) = arg\max_{y \in Y} \sum_{t=1}^{T} \left(\log \frac{1}{\beta^t}\right) [\![c^t(x_i) = y_i]\!]$

For the proposed method (HBoost):
return $C = \{c^1, c^2, \ldots, c^T\}$ *and* $\{\beta^1, \beta^2, \ldots, \beta^T\}$

It is intuitively meaningful that at each step, a classifier with a different type from the previous classifier may be appropriate for the new input instances. Since Boosting is a homogeneous method, all classifiers are of the same type. To overcome these problems, in the following section, an approach is proposed that uses multiple learning algorithms for constructing a Boosting method and automatically configures the final model to preserve the appropriate classifiers and eliminate the others.

3.2. HBoost: A heterogeneous Boosting ensemble

Our main objective is to apply heterogeneous algorithms on the Boosting method and construct a model from dataset $\mathcal{D}$ so that the final model would classify unlabeled instances accurately. By recruiting multiple learning algorithms, we increase the diversity between the classifiers and by using the Boosting method, it is expected that the bias error of the final model is reduced. This allows the constructed model to cover more sections of training data. In other words, each part of the training data may be appropriate for a particular learning algorithm. Using multiple algorithms makes each section of the input data more likely to be classified accurately.

Fig. 1 depicts the HBoost method. It consists of three steps: 1) Constructing Boosting classifiers from each learning algorithm 2) Sorting the Classifiers and Creating the pruned matrix: this step is divided into two sub-steps: 2.A: Sorting the constructed classifiers and exporting them into a classifier matrix and 2.B: Pruning each column of the matrix to achieve the maximum diversity. However, for better understanding, we explain both of them in Section 3.2.2 together. 3) Combining the survived classifiers of the matrix to build the final model. In the following, we elaborate each step.

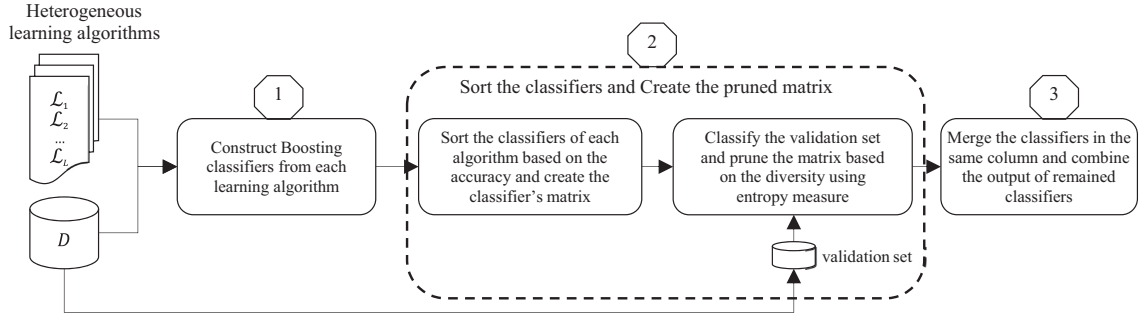


Fig. 1. Steps of the HBoost.

3.2.1. Constructing Boosting classifiers from each learning algorithm

Algorithm 2 shows the pseudo code of HBoost. It commences with considering a set of learning algorithms as the input parameter. Each learning algorithm is trained with input instances using the Adaboost.M1 method. For more clarification, if we consider three learning algorithms SVM, Naïve Bayes and Decision Tree, for each of these algorithms, a Boosting classifier is trained that uses the respective algorithm as its base learner's type. Assuming that there are L distinct learning algorithms and each algorithm generates T weak classifiers with the Boosting approach, the total number of generated classifiers is $L \times T$. Each classifier is given a weight β_t which was described in Section 3.1. The output of this step is a set of T classifiers for each learning algorithm (totally $L \times T$ classifiers) as well as their assigned weights.

Algorithm 2 (*The Proposed method, HBoost.*).

Input: Dataset $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$
$y_i \in Y = \{1, \ldots, k\}$
Set of Learning algorithms $\mathcal{L}$
Number of iterations T

1: $M_{L\times T} = \phi$
2: **for** $l = 1$ to L **do**
3: $[C_l, \beta_l] = ADABOOST.M1(\mathcal{D}, T, \mathcal{L}_l)$
4: $M_{Col_l} = Sort\ classifiers\ in\ C_l\ w.r.t\ to\ \beta_l$
6: **end for** /* constructing the classifiers and sorting */
7: **for** $t = 1$ **to** T **do**
8: $M^*_{Row_t} = Prune\ and\ get\ combination_that_maximize_the\ Entropy(M_{Row_t})$
9: $h^{(t)} = Vote(M^*_{Row_t})$
10: $\alpha^{(t)} = Average(\beta^{M^*_{Row_t}})$
11: **end for** /* pruning the classifiers matrix and combining the outputs */
12: $Renormalize(\alpha^{(t)}), 1 \le t \le T$
13: **Output:** $C(x) = arg\max_k \sum_{t=1}^{T} \left(\log\frac{1}{\alpha^{(t)}}\right).\mathbb{I}(h^{(t)}(x) = k)$

3.2.2. Sorting the classifiers and creating the pruned matrix

Once a pool of $L \times T$ classifiers was constructed, there should be a policy to combine their output for the generalization phase. We elaborate on the method for combining the outputs further. However, combining all of these classifiers is not a good idea; because some of them may be similar, and consequently reduce the diversity. In other words, if some inaccurate classifiers generate very similar outputs, it is better to eliminate some of them to increase the diversity in the ensemble model. Therefore, it is worthwhile to prune the pool and omit the classifiers which are dependent on each other. One of the most commonly used measurements for calculating the diversity is entropy. It generally reveals the randomness of different phenomena. In the context of classification, entropy means the difference in the prediction of the classifiers for the same data and calculated by Eq. (2) (Kuncheva & Whitaker, 2003):

$$E = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{(L - L/2)}\min\{\sigma(x_j), (L - \sigma(x_j))\} \quad (2)$$

In Eq. (2), N is the number of instances, L is the number of base classifiers and $\sigma(x_j) = \sum_{i=1}^{L}\mathbb{I}(c_i(x_i) = y_j)$ is a function that returns the number of base classifiers that classify the instance x_i correctly. The indicator function $\mathbb{I}$ returns 1 if its argument is true and 0 otherwise. For binary class datasets, the entropy reaches the maximum when one-half of the base classifiers vote to positive class and the rest vote to negative class. Zero value for the entropy indicates that all predictions of the classifiers are similar. In contrast, the maximum value of one reveals the maximum diversity between outputs.

Fig. 2 illustrates the steps of the HBoost. As can be seen, each learning algorithm generates T classifiers and assign a weight to each of them as described in Section 3.1. Then these classifiers are sorted in descending order with respect to their assigned weights. This is done for each learning algorithm i.e., the sorting is carried out L times. The motivation behind the sorting of the classifiers is to put the classifiers with similar accuracy into the same group. In other words, the most accurate classifier of each learning algorithms falls into the same group (row) and are compared with each other. The same is true for the next classifiers of each learning algorithm. If learning algorithm $\mathscr{L}_l$ generates T weak classifiers with Boosting method, then we have $C_l : \{c_l^1, \cdots, c_l^T\}$ (generated classifiers) and after sorting the classifiers in C_l with respect to their assigned weights, we have Eq. (3) as follows:

$$C_l = \{c_l^1, \cdots, c_l^T\} \overset{descsortw.r.tassignedweight}{\rightarrow} C_l^* = \left\{c_l^{*(1)}, \cdots, c_l^{*(T)}\right\} \quad (3)$$

Such that $weight\left(C_l^{*(t)}\right) \ge weight\left(C_l^{*(t+1)}\right), 1 \le t < T, 1 \le l \le L.$

Table 1 shows the classifiers matrix:

In the preceding matrix, in each column, the upper rows are more valuable than the lower rows (because classifiers are sorted in descending order) and in each row, there are classifiers with different learning algorithms that are closer to each other than the rest of classifiers in different rows (with sorting, the classifiers with closer accuracy, are more likely to be located on the same row). After completing this step, improper classifiers of each row (ones that are not diverse) should be pruned from the matrix. To decide which classifiers should be removed from each row, we use entropy as described before.

The presence or absence of a typical classifier in a row makes different individuals for the row. Since each row contains L classifiers, there are $(2^L - L - 1)$ different individuals that at least two classifiers are present.

The individual is more likely to survive which generates higher entropy than the other ones for the same row. For a better clarification, Table 2 shows the 11 individuals of four classifiers and their obtained entropy for an imaginary dataset which at least two classifiers are present at each row (values are unreal). Since individual

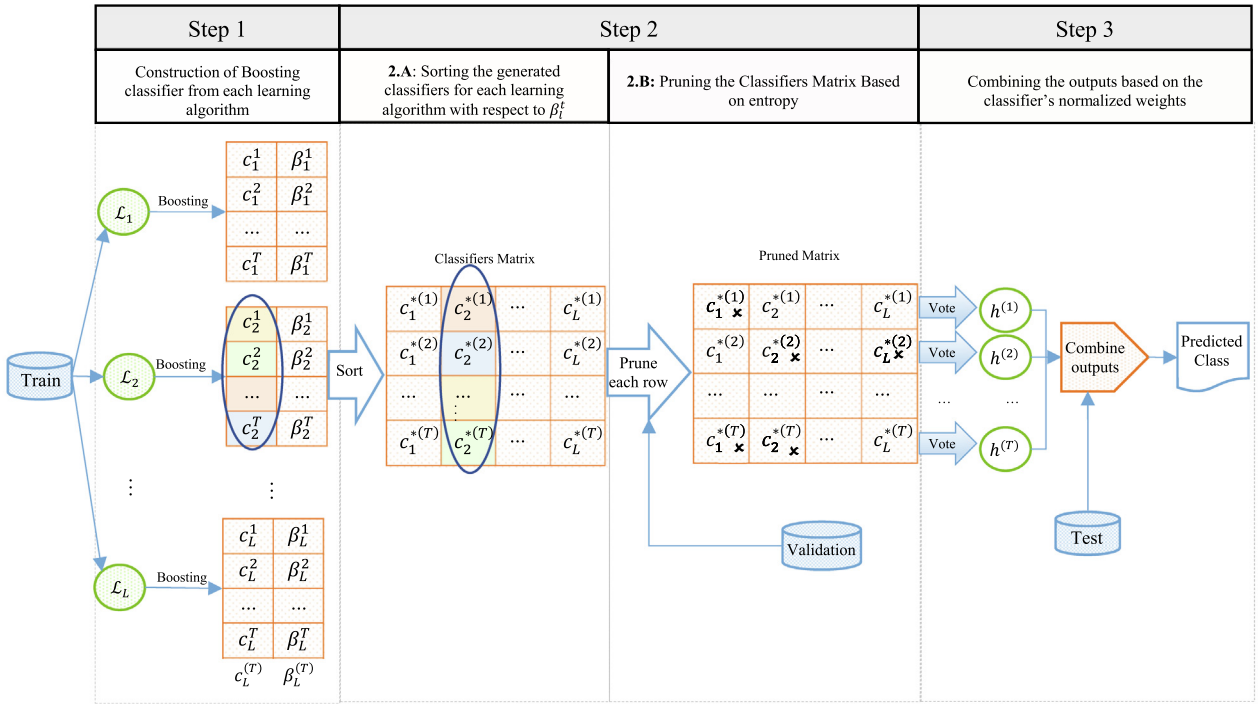


Fig. 2. Steps of the proposed method. Cells with "×" mark in step 2.B are those that have been eliminated after pruning of each row.

Table 1
The classifier's matrix after sorting.

$\mathscr{L}_1$(*Naivebayes*)	$\mathscr{L}_2$(*Decision tree*)	...	$\mathscr{L}_L$(*SVM*)
$c_1^{*(1)}$	$c_2^{*(1)}$	...	$c_L^{*(1)}$
$c_1^{*(2)}$	$c_2^{*(2)}$	...	$c_L^{*(2)}$
...	...	...	...
$c_1^{*(T)}$	$c_2^{*(T)}$	...	$c_L^{*(T)}$

8 generates higher entropy than the others, its state should be considered and consequently the KNN will be pruned from the set of classifiers. It means that combining the Naïve Bayes, Decision Tree, and SVM generates the highest entropy in comparison with other combinations. For calculating the entropy of each row, classifiers are required to generate their output for the same instances. This could be done through considering a "validation set" as it is depicted in step 2.B of the Fig. 2. Table 3 shows the pruned matrix after removing improper classifiers.

Cells with crossed-line are those that should be removed from each row. It should be taken into account that the number of remained classifiers in each row may vary. For example, in the first row, the combination of (L-1) classifiers may produce the maximum entropy, whereas in another row a different number of classifiers may be desirable.

3.2.3. Merging the classifiers and combining the outputs

Once the matrix was pruned, the remained classifiers in each row are simplified into one classifier using the simple voting method: $h^{(t)} = Vote\left(c_1^{*(t)}, \cdots, c_{L_t}^{*(t)}\right), 1 \le t \le T$ where L_t is the number of remained classifiers in the row t after pruning. The weight assigned to this classifier is equal to the average weight of its constituent classifiers: $\alpha^{(t)} = \frac{1}{L_t}\sum_{i=1}^{L_t} c_i^{*(t)}$. Since the classifiers had been sorted in descending order with respect to their weights before entering the matrix, it makes sense that the early rows in the matrix make a more significant impact on the final result than the last rows.

Since the number of contributor classifiers has been changed (because some of them have been pruned from the matrix), the weight of simplified classifiers should be normalized before combining the output of them such that it satisfies Eq. (4).

$$\sum_{i=1}^{T} h^{(i)} = 1 \tag{4}$$

After constructing the simplified classifiers, they can be combined using the weighted sum function in Eq. (5).

$$C(x) = arg\max_k \sum_{t=1}^{T} \left(\log\frac{1}{\alpha^{(t)}}\right).\mathbb{I}(h^{(t)}(x) = k) \tag{5}$$

where $h^{(t)}(x)$ is the weighted sum of the class k and $\alpha^{(t)}$ is the weight that is assigned to the classifier t in the matrix. To predicted the class label for instance x_i, the value of $C(x)$ is calculated for each k possible classes. The final output is the class that generates the highest value for $C(x)$.

The most time-consuming parts of the proposed method are steps one and three. The time complexity of the first step is dependent upon both the learning algorithm and the time for creating the Boosting classifier. In the third step, all classifiers should generate their prediction for a validation set and then the entropy would be calculated for all combinations of the classifiers to select the best individual that maximizes the entropy measure. However, once the final model was built, it can quickly generate its prediction for unlabeled instance. It should also be taken into account that at each step, the task for each learning algorithm can be done in parallel i.e., construction of Boosting classifiers, sorting the generated classifiers, pruning each row of the matrix, and simplifying the pruned rows.

4. Experimental results

In this section, a series of experiments and their obtained results are presented to evaluate the performance of the proposed method.

Table 2
Individuals of four learning algorithms with at least two classifiers present (1 means presence and 0 means absence of a classifier).

#Individual	Naïve Bayes	Decision Tree	KNN	SVM	Entropy
1	1	0	0	1	0.53
2	1	0	1	0	0.37
3	1	1	0	0	0.44
4	0	1	0	1	0.47
5	0	1	1	0	0.66
6	0	0	1	1	0.28
7	1	0	1	1	0.33
8	**1**	**1**	**0**	**1**	**0.74**
9	1	1	1	0	0.47
10	0	1	1	1	0.61
11	1	1	1	1	0.55

Table 3
Some classifiers are pruned from each row of the matrix. Cells with × mark are eliminated.

	$\mathscr{L}_1$	$\mathscr{L}_2$	...	$\mathscr{L}_L$
Group 1	$c_1^{*(1)}$ ×	$c_2^{*(1)}$	...	$c_L^{*(1)}$ ×
Group 2	$c_1^{*(2)}$	$c_2^{*(2)}$	...	$c_L^{*(2)}$ ×
...	...	...	...	...
Group T	$c_1^{*(T)}$	$c_2^{*(T)}$ ×	...	$c_L^{*(T)}$

4.1. Dataset

In order to evaluate the effectiveness of the proposed method as a new hybrid heterogeneous ensemble system, an empirical analysis is conducted. We have selected 20 classification datasets from the UCI repository (Dua & Graff, 2019) and consider both balanced and imbalanced datasets. Table 4 presents the characteristics of the used datasets, describing the number of instances, features, and classes for each data set. The last column shows the number of instances with majority classes to the number of instances with minority class.

4.2. Base classifiers

There are lots of learning algorithms in the Weka that can be used as the base classifiers of the proposed approach. However, we have selected the ten most popular ones such that they belong to different categories.

Table 5 shows the classifiers that have been used for the experiments. All these algorithms are imported from the Weka machine learning package with their default parameter setting. Furthermore, most of these algorithms are efficient and straightforward and have been widely used in ensemble classification problems. For more information about these algorithms, see (Witten, Frank, Hall, & Pal, 2016).

4.3. Experimental setup

There are different packages for Machine Learning tasks. One of the most famous is Weka which is based on Java language and has been developed by Waikato University. We use Weka 3.9.2 developer version on windows 10 operating system to perform our experiments. All datasets were partitioned into 10 folds using stratified cross-validation (one fold for test 9 folds for training) and this is done for each dataset 10 times, that is, each classifier has been trained 10 times and the mean and standard deviation has been reported. In order to reach a reliable result on the proposed approach, two performance indicator measures are implemented: 1) Accuracy and 2) Geometric mean. These are popular and can give a more detailed view of the model performance. The accuracy is equal to the proportion of the corrected classified instance to the total number of instances and is calculates by Eq. (6).

Table 4
Characteristics of the data sets from UCI.

Dataset	*#Instances*	*#Features*	*#Classes*	*Majority/Minority*
Balance-scale	625	4	3	288/49
Breast-cancer	286	9	2	201/85
Car	1728	6	4	1210/65
Cmc	1473	9	3	629/333
Colic	368	22	2	232/136
Diabetes	768	8	2	500/268
Ecoli	336	7	8	143/2
Glass	214	9	7	76/9
Hepatitis	155	19	2	123/32
Iris	150	4	3	50/50
Liver-disorders	345	6	2	200/145
Lymph	148	18	4	81/2
Solar-flare-1	323	12	6	88/8
Solar-flare-2	1066	12	6	331/43
Soybean	683	35	19	92/8
Vehicle	846	18	4	218/199
Vote	435	16	2	267/168
Vowel	990	13	11	90/90
Wine	178	13	3	71/48
Zoo	101	17	7	41/4

Table 5
Base Classifiers of HBoost.

Type	Learning Algorithm	Notation
Bayes	*Naïve Bayes*	*NB*
	Bayesian Network	*BN*
Lazy	*KStar*	*K**
	KNN	*KNN*
Tree	*J48*	*DT*
	RepTree	*RT*
	Decision Stump	*DS*
Rules	*OneR*	*1R*
	Decision Table	*Ds.Tab*
Functions	*SVM*	*SVM*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

However, accuracy itself does not reveal which class has been better classified. This is especially true for imbalanced datasets. As can be seen in Table 4, most of the datasets are imbalanced and accuracy cannot provide more information except for the overall performance of the typical classifier for these datasets.

The main goal of all classifiers is to improve the sensitivity, without sacrificing the specificity (Tharwat, 2018). Geometric Mean (G-Mean, GM) aggregates both sensitivity and specificity measures according to Eq. (7).

$$GM = \sqrt{TPR \times TNR} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \tag{7}$$

If a classifier has a low error rate for the majority class of an imbalanced dataset, it will have high accuracy, because most of the instances are classified correctly. Nevertheless, there is no further information about the performance of the classifier for the minority class. This is the case that the usage of GM makes sense. Since GM considers both TPR and TNR, if one of them declines, the value of GM decreases. A high value of GM metric for a typical classifier imposes that the classifier can classify both positive and negative instances with good performance. We use both accuracy

Table 6
Accuracy for single classifiers.

Dataset	1*R*	*NB*	*DS*	*KNN*	*DT*	*BN*	*K**	*Ds.Tab*	*RT*	*SVM*
Balance-Scale	58.80 ± 0.75*	**90.22 ± 0.43**	59.18 ± 0.40*	86.00 ± 0.56*	78.20 ± 0.80*	72.96 ± 0.98*	89.06 ± 0.19	74.46 ± 1.36*	77.84 ± 1.02*	87.56 ± 0.35
Breast-cancer	67.59 ± 1.21*	73.04 ± 0.55	70.10 ± 1.23*	72.80 ± 1.56	**74.23** ± 0.96	72.55 ± 0.50	73.74 ± 0.72	73.64 ± 1.14	68.88 ± 1.48*	69.13 ± 0.96*
Car	70.02 ± 0.00*	85.87 ± 0.14*	70.02 ± 0.00*	92.87 ± 0.33	92.17 ± 0.31	85.63 ± 0.22*	87.17 ± 0.35*	91.54 ± 0.32	88.61 ± 0.64*	**93.68 ± 0.48**
Cmc	47.82 ± 0.13*	50.43 ± 0.25	42.70 ± 0.00*	43.85 ± 0.61*	51.48 ± 0.93	50.36 ± 0.47	50.05 ± 0.58*	**53.17** ± 0.68	52.25 ± 1.03	48.59 ± 0.44*
Colic	81.52 ± 0.00*	78.61 ± 0.46*	81.52 ± 0.00*	79.24 ± 0.92	**85.38** ± 0.29	81.01 ± 0.35	75.41 ± 1.10*	83.40 ± 0.95	85.08 ± 0.52	83.10 ± 1.11
Diabetes	71.59 ± 0.48*	75.60 ± 0.37	71.29 ± 0.49*	70.42 ± 0.69*	74.30 ± 1.12	75.40 ± 0.68	70.13 ± 0.48*	74.73 ± 1.17*	74.73 ± 1.08	**76.86 ± 0.31**
Ecoli	67.32 ± 1.23*	**85.68** ± 0.41	64.58 ± 0.00*	80.80 ± 0.83*	83.18 ± 1.02	81.19 ± 0.80	81.82 ± 0.59	77.68 ± 0.69*	80.60 ± 1.28*	83.90 ± 0.54
Glass	56.13 ± 1.57*	47.20 ± 1.12*	44.57 ± 0.62*	69.80 ± 1.23*	68.11 ± 2.76*	71.61 ± 0.96	**74.59 ± 1.07**	65.65 ± 1.24*	65.36 ± 2.10*	55.84 ± 1.60*
Hepatitis	80.32 ± 1.20	83.94 ± 0.45	76.90 ± 1.07*	80.97 ± 1.01	80.00 ± 1.41	83.68 ± 0.58	80.32 ± 0.66	76.90 ± 1.03*	79.87 ± 2.08*	**85.10** ± 1.57
Iris	92.50 ± 0.73	96.00 ± 0.33	65.33 ± 0.00	95.25 ± 0.62	95.08 ± 0.94	93.42 ± 0.52	94.58 ± 0.22	92.50 ± 0.65	94.17 ± 0.99	**96.08** ± 0.40
Liver-disorders	56.06 ± 1.81*	55.39 ± 1.25*	60.96 ± 1.81*	62.64 ± 0.67	64.67 ± 1.66	57.83 ± 1.01*	**66.67** ± 1.39	59.25 ± 1.51*	65.83 ± 2.48	57.88 ± 0.23*
Lymph	74.59 ± 0.33*	83.45 ± 1.26	75.68 ± 0.00*	82.36 ± 1.58	77.09 ± 1.58*	85.14 ± 0.85	84.59 ± 1.41	73.18 ± 2.18	75.81 ± 1.62*	**87.36** ± 0.80*
Solar-flare-1	53.13 ± 1.27*	65.25 ± 0.80*	47.06 ± 0.00*	66.95 ± 1.09*	71.13 ± 1.22	66.80 ± 0.51*	68.34 ± 1.48	69.81 ± 0.99	70.67 ± 0.91	**71.13** ± 0.72
Solar-flare-2	61.44 ± 0.00*	74.36 ± 0.41	53.47 ± 0.00*	72.57 ± 0.45	74.45 ± 0.44	74.36 ± 0.23	73.89 ± 0.34	75.49 ± 0.19	**74.85** ± 0.50	74.62 ± 0.27
Soybean	39.79 ± 0.27*	92.95 ± 0.14	27.96 ± 0.00*	91.25 ± 0.37	91.75 ± 0.52	92.83 ± 0.23	87.74 ± 0.18*	83.53 ± 0.92*	85.40 ± 0.99*	**93.14** ± 0.48
Vehicle	52.39 ± 0.93*	45.31 ± 0.56*	39.55 ± 0.31*	69.57 ± 0.81*	72.09 ± 0.96	60.59 ± 0.59*	70.14 ± 0.56*	65.72 ± 1.08*	70.08 ± 1.20*	**73.83** ± 0.54
Vote	95.63 ± 0.00	90.05 ± 0.11*	95.63 ± 0.00	92.69 ± 0.35*	**96.64** ± 0.21	90.21 ± 0.21*	93.17 ± 0.40*	94.55 ± 0.63	95.47 ± 0.51	96.02 ± 0.37
Vowel	32.89 ± 0.82*	62.51 ± 0.54*	17.32 ± 0.20*	99.08 ± 0.23	79.70 ± 0.90*	58.04 ± 0.80*	**98.69** ± 0.26	57.83 ± 1.58*	65.42 ± 1.77*	70.64 ± 0.56*
Wine	78.30 ± 1.16*	97.61 ± 0.67	58.36 ± 1.10*	95.22 ± 0.40	91.78 ± 1.09*	**98.81 ± 0.19**	98.17 ± 0.37	89.26 ± 1.27*	90.24 ± 2.48*	98.53 ± 0.39
Zoo	42.57 ± 0.00*	95.17 ± 0.59	60.40 ± 0.00*	96.04 ± 0.86	92.45 ± 0.85*	94.80 ± 0.65	95.92 ± 0.77	87.25 ± 1.26*	40.59 ± 0.00*	**96.04 ± 0.00**

*Cells marked with an asterisk have a statistically significant difference with the best result (Boldface).

Table 7
G-Mean for single classifiers.

Dataset	1*R*	*NB*	*DS*	*KNN*	*DT*	*BN*	*K**	*Ds.Tab*	*RT*	*SVM*
Balance-Scale	41.84 ± 0.49*	**62.90** ± 0.29	42.08 ± 0.26*	61.11 ± 0.32	57.24 ± 2.98*	51.33 ± 0.66*	62.13 ± 0.13	54.34 ± 2.90*	56.00 ± 2.40*	61.12 ± 0.23
Breast-cancer	42.39 ± 2.98*	61.46 ± 0.79	**62.11** ± 2.24	55.44 ± 2.22*	48.34 ± 2.24*	61.01 ± 1.12	56.49 ± 1.66	49.54 ± 1.92*	38.11 ± 4.37*	50.97 ± 2.32*
Car	0.00 ± 0.00*	71.37 ± 1.00*	0.00 ± 0.00*	79.24 ± 1.29*	87.87 ± 0.72*	71.76 ± 0.71*	63.20 ± 1.10*	86.42 ± 0.92*	78.63 ± 2.11*	**93.67** ± 0.84
Cmc	44.04 ± 0.95*	**61.76** ± 0.20	0.00 ± 0.00*	54.50 ± 0.48*	59.74 ± 0.82	61.61 ± 0.36	59.60 ± 0.52	61.55 ± 0.44	60.78 ± 1.04	58.15 ± 0.34
Colic	81.39 ± 0.00	78.10 ± 0.48	81.39 ± 0.00	77.63 ± 0.97	**81.85** ± 0.34	79.97 ± 0.34	74.53 ± 0.92*	80.00 ± 1.21	81.75 ± 0.72	81.16 ± 0.95
Diabetes	61.33 ± 0.79*	71.11 ± 0.41	68.23 ± 1.23	65.21 ± 0.81*	69.42 ± 1.52	**71.81** ± 0.94	62.38 ± 0.76*	68.69 ± 1.20	68.89 ± 1.33	69.04 ± 0.34
Ecoli	35.12 ± 0.57*	**67.61** ± 0.42	20.91 ± 0.01*	65.18 ± 0.51	63.67 ± 1.50	63.25 ± 0.97	64.20 ± 0.57	48.99 ± 2.12*	53.49 ± 3.13*	63.85 ± 0.65
Glass	44.50 ± 0.92*	63.40 ± 1.47*	24.97 ± 0.06*	78.04 ± 0.78	77.53 ± 1.31	74.78 ± 2.91*	**78.70 ± 1.22**	67.08 ± 3.85*	71.62 ± 4.68*	42.73 ± 0.83*
Hepatitis	56.45 ± 3.37*	**78.28 ± 0.83**	21.91 ± 4.86*	63.15 ± 1.68*	58.94 ± 3.92*	77.86 ± 1.08	61.48 ± 1.50*	59.11 ± 5.12*	47.08 ± 8.87*	73.46 ± 3.05
Iris	94.32 ± 0.56*	96.99 ± 0.25	70.75 ± 0.00*	96.42 ± 0.47	96.30 ± 0.71	95.03 ± 0.39	95.92 ± 0.17	94.34 ± 0.49	95.60 ± 0.75	**97.04** ± 0.30
Liver-disorders	52.47 ± 2.34*	55.46 ± 1.34*	57.30 ± 3.35*	61.30 ± 0.71	61.09 ± 1.95	42.76 ± 4.91*	**65.76** ± 1.30	52.17 ± 4.82*	61.68 ± 2.08	1.66 ± 3.31*
Lymph	45.96 ± 8.62*	79.60 ± 5.68*	39.04 ± 0.00*	80.63 ± 1.33*	67.10 ± 11.67*	**92.25** ± 0.45	81.27 ± 0.69*	61.06 ± 7.67*	47.81 ± 9.05*	89.77 ± 1.38
Solar-flare-1	46.34 ± 0.82*	56.98 ± 2.18*	30.01 ± 0.00*	62.14 ± 1.00	61.81 ± 1.57	58.08 ± 2.39*	**62.29** ± 0.65	57.18 ± 1.46*	56.93 ± 1.50*	59.65 ± 1.74
Solar-flare-2	47.87 ± 0.00*	73.15 ± 0.48	27.21 ± 0.00*	66.91 ± 1.79*	67.87 ± 1.17*	**73.37** ± 0.46	69.99 ± 0.65*	65.71 ± 0.20*	66.43 ± 0.91*	69.62 ± 0.44*
Soybean	23.76 ± 1.42*	97.38 ± 0.13	12.36 ± 0.00*	95.66 ± 0.22	94.76 ± 0.29*	**97.81** ± 0.07	90.18 ± 0.12*	91.43 ± 0.90*	85.41 ± 1.04*	97.47 ± 0.17
Vehicle	65.97 ± 0.80*	57.10 ± 0.55*	44.20 ± 0.32*	78.58 ± 0.61	80.49 ± 0.70	71.16 ± 0.48*	78.71 ± 0.44	74.99 ± 0.99*	78.89 ± 0.92	**81.53** ± 0.41
Vote	95.88 ± 0.00*	90.34 ± 0.12*	95.88 ± 0.00	92.83 ± 0.36*	**96.39** ± 0.21	90.46 ± 0.20*	93.42 ± 0.39	94.33 ± 0.73	95.43 ± 0.57	96.08 ± 0.33
Vowel	53.33 ± 0.84*	77.26 ± 0.36*	21.32 ± 1.04*	**99.49 ± 0.13**	88.25 ± 0.55*	74.30 ± 0.54*	99.27 ± 0.14	73.69 ± 1.19*	79.20 ± 1.15*	82.40 ± 0.38*
Wine	83.62 ± 0.87*	98.31 ± 0.49	67.01 ± 1.25*	96.78 ± 0.28*	93.71 ± 0.79*	**99.21 ± 0.12**	98.74 ± 0.30	91.71 ± 1.01*	92.68 ± 1.91*	99.02 ± 0.28
Zoo	12.71 ± 0.00*	**94.43** ± 1.00	24.45 ± 0.00*	94.12 ± 1.13	90.13 ± 0.71*	92.91 ± 0.89*	93.74 ± 0.97	78.54 ± 2.77*	0.00 ± 0.00*	93.83 ± 0.00

Table 8
Best results (accuracy ± standard deviation) for the different ensemble approaches.

	Best-Bagging		*Best-AdaBoost*		*Best-Stacking*		*StackingC*
Dataset	*Type*	*Acc*	*Type*	*Acc*	*Type (Meta)*	*Acc*	*Acc*
Balance-Scale	*NB*	90.36 ± 0.27*	*BN*	90.92 ± 0.56*	*Ds.tab*	**93.80** ± 0.97	89.34 ± 0.36*
Breast-cancer	*K**	**73.95** ± 1.30	*DS*	71.61 ± 1.02*	*Ds.Tab*	72.73 ± 1.31	73.66 ± 1.78
Car	*KNN*	93.45 ± 0.51*	*Ds.Tab*	95.99 ± 0.38	*DT*	**98.68 ± 0.28**	94.91 ± 0.35*
Cmc	*RT*	52.94 ± 0.73	*Ds.Tab*	53.27 ± 0.71	*SVM*	54.76 ± 0.31	**54.94** ± 0.53
Colic	*DT*	**85.33** ± 0.36	*DT*	82.36 ± 1.46*	*BN*	84.96 ± 0.26	83.74 ± 0.51
Diabetes	*SVM*	**77.36** ± 0.50	*SVM*	76.86 ± 0.31	*SVM*	76.26 ± 0.25	75.72 ± 0.45
Ecoli	*BN*	85.62 ± 1.02	*NB*	**85.68 ± 0.41**	*RT*	83.09 ± 1.40	84.47 ± 0.98
Glass	*K**	75.58 ± 0.98	*DT*	74.07 ± 1.24*	*SVM*	75.58 ± 1.63	**76.64** ± 1.30
Hepatitis	*BN*	83.94 ± 0.73	*NB*	**84.00** ± 0.69	*SVM*	83.01 ± 0.30	82.69 ± 1.6
Iris	*SVM*	96.00 ± 0.82	*SVM*	**97.17** ± 0.80	*K**	95.33 ± 0.88	94.92 ± 0.81*
Liver-disorders	*RT*	**70.03** ± 1.63	*DT*	68.12 ± 1.15	*BN*	66.09 ± 1.32*	67.29 ± 1.43*
Lymph	*SVM*	**85.61** ± 1.68	*SVM*	83.85 ± 1.55*	*SVM*	84.01 ± 0.84	84.35 ± 0.99
Solar-flare-1	*Ds.Tab*	**72.33 ± 0.80**	*SVM*	71.98 ± 0.93	*SVM*	71.52 ± 0.90	71.21 ± 1.34
Solar-flare-2	*SVM*	75.15 ± 0.28	*SVM*	74.61 ± 0.54	*SVM*	75.07 ± 0.44	**75.46** ± 0.47
Soybean	*DT*	93.08 ± 0.50	*BN*	93.47 ± 0.39	*SVM*	**94.09 ± 0.28**	93.36 ± 0.50
Vehicle	*DT*	74.67 ± 1.36*	*DT*	**76.08** ± 0.93	*SVM*	75.61 ± 0.66	76.06 ± 0.57
Vote	*DT*	96.30 ± 0.36	*SVM*	95.95 ± 0.33	*DS*	96.07 ± 0.59	**96.36** ± 0.21
Vowel	*KNN*	98.65 ± 0.24	*KNN*	98.95 ± 0.20	*SVM*	99.04 ± 0.17	**99.05** ± 0.24
Wine	*BN*	98.74 ± 0.37	*SVM*	97.68 ± 0.82*	*BN*	**99.09** ± 0.62	98.24 ± 0.66
Zoo	*KNN*	96.29 ± 0.65	*NB*	**96.91** ± 0.33	*K**	96.04 ± 0.70	96.16 ± 1.35

and Geometric mean to evaluate the performance of the classifiers. For multi-class datasets, the one-vs-rest approach (Ramírez et al., 2018) is applied to calculate the geometric mean as it is more common for non-binary datasets described in (Espíndola & Ebecken, 2005).

4.4. Method

With the aim of evaluating the proposed approach, three different classification groups are selected to compare against HBoost:

1- Ten base classifiers in Table 5
2- Four traditional ensemble classifiers include Bagging, Boosting, Stack Generalization, and StackingC (Seewald, 2002) which is a more efficient version of Stack Generalization.
3- Random Forest (Tin Kam, 1995) and the Classifier Subset Selection (CSS) presented in (Mendialdua et al., 2015).

Our comparison approach is as follows: Firstly, we report the results of each above method separately. Then in Section 4.5.1,

Table 10
Results obtained by the CSS (Mendialdua et al., 2015).

Dataset	Type (Meta)	Accuracy	Geometric mean*
Balance-Scale	*RF*	94.70 ± 1.04	94.83 ± 0.60
Breast-cancer	*NB, RF*	74.83 ± 0.70	61.57 ± 2.31
Car	*RF*	98.62 ± 1.72	98.19 ± 1.38
Cmc	*NB*	53.51 ± 2.20	60.21 ± 4.03
Colic	*DT*	86.74 ± 0.84	81.47 ± 1.93
Diabetes	*K**	77.86 ± 0.56	71.44 ± 0.96
Ecoli	*SVM*	85.24 ± 3.50	65.01 ± 4.46
Glass	*RF*	75.84 ± 4.70	77.50 ± 3.52
Hepatitis	*DT, RF*	86.52 ± 1.12	72.53 ± 5.93
Iris	*KNN, K**	96.73 ± 0.38	95.85 ± 0.70
Liver-disorders	*NB*	71.51 ± 1.27	68.48 ± 1.82
Lymph	*BN*	86.49 ± 1.35	84.64 ± 5.51
Solar-flare-1	*KNN*	72.54 ± 3.31	65.66 ± 3.26
Solar-flare-2	*SVM*	73.77 ± 2.91	70.09 ± 3.04
Soybean	*RF, SVM*	89.18 ± 13.71	97.88 ± 0.25
Vehicle	*RF*	77.01 ± 5.03	86.62 ± 2.13
Vote	*BN*	96.90 ± 0.31	95.80 ± 1.08
Vowel	*KNN, RF*	93.03 ± 19.35	99.48 ± 0.12
Wine	*KNN, NB*	99.10 ± 2.28	99.29 ± 0.49
Zoo	KNN,NB	94.84 ± 7.00	90.45 ± 2.23

*implemented by us

Table 9
Best results (geometric mean ± standard deviation) for the different ensemble approaches.

	Best-Bagging		*Best-AdaBoost*		*Best-Stacking*		*StackingC*
Dataset	*Type*	*G-Mean*	*Type*	*G-Mean*	*Type (Meta)*	*G-Mean*	*G-Mean*
Balance-Scale	*NB*	62.99 ± 0.18*	*NB*	82.72 ± 2.51*	*NB*	**94.61** ± 0.39	62.31 ± 0.24*
Breast-cancer	*BN*	61.00 ± 1.92	*DS*	58.16 ± 1.95*	*BN*	**64.74 ± 0.20**	57.34 ± 1.73*
Car	*SVM*	92.91 ± 0.32*	*Ds.Tab*	94.67 ± 0.92	*DT*	**97.97** ± 0.64	93.64 ± 0.63*
Cmc	*BN*	62.20 ± 0.22	*NB*	61.77 ± 0.17*	*BN*	**64.92** ± 0.26	62.56 ± 0.38
Colic	*RT*	82.25 ± 0.59	*DS*	80.35 ± 1.27*	*BN*	**83.06 ± 0.09**	80.76 ± 0.49
Diabetes	*BN*	**72.48** ± 1.05	*BN*	71.78 ± 0.95	*NB*	72.46 ± 0.44	69.09 ± 0.71*
Ecoli	*NB*	66.44 ± 1.28	*NB*	**67.61 ± 0.42**	*RT*	65.36 ± 1.07	66.70 ± 0.96
Glass	*K**	78.85 ± 1.47	*DT*	79.80 ± 2.11	*SVM*	**79.82 ± 2.05**	79.33 ± 1.81
Hepatitis	*NB*	**78.01 ± 0.84**	*NB*	71.92 ± 1.93*	*BN*	76.81 ± 2.46	68.56 ± 4.69*
Iris	*SVM*	96.98 ± 0.61	*SVM*	**97.87 ± 0.60**	*K**	96.48 ± 0.67	95.98 ± 1.01*
Liver-disorders	*DT*	67.23 ± 2.43	*DT*	66.61 ± 1.53	*NB*	62.83 ± 0.73*	**69.00** ± 1.92
Lymph	*BN*	**92.11** ± 1.02	*BN*	91.36 ± 0.60	*K**	82.26 ± 3.35*	76.26 ± 8.52*
Solar-flare-1	*KNN*	63.20 ± 1.07*	*KNN*	66.55 ± 3.52*	*NB*	**73.04** ± 1.55	60.28 ± 2.86*
Solar-flare-2	*BN*	73.35 ± 0.84*	*BN*	73.62 ± 0.43	*NB*	**75.47 ± 0.27**	70.49 ± 0.95*
Soybean	*BN*	97.68 ± 0.18	*BN*	97.61 ± 0.28	*SVM*	**97.86** ± 0.16	97.75 ± 0.15
Vehicle	*DT*	82.29 ± 0.99	*DT*	**83.39** ± 0.68	*SVM*	83.03 ± 0.51	83.21 ± 0.43
Vote	*DT*	**96.27** ± 0.31	*NB*	95.88 ± 0.52	*DS*	96.11 ± 0.56	96.26 ± 0.25
Vowel	*K**	99.03 ± 0.22	*KNN*	99.42 ± 0.11	*SVM*	**99.50 ± 0.10**	99.48 ± 0.13
Wine	*BN*	99.16 ± 0.27	*SVM*	**98.39** ± 0.63	*BN*	99.37 ± 0.45	98.96 ± 0.47
Zoo	*KNN*	94.61 ± 1.05	*NB*	94.48 ± 0.08	*NB*	**95.52** ± 1.36	93.65 ± 2.55*

these results are compared with the results of the proposed method. As mentioned in Section 4.3, all of the experiments are implemented in Weka with 10-fold cross-validation.

4.5. Obtained results

4.5.1. Single classifiers

As mentioned in Section 4.4, different classification methods are selected to compare with. Firstly, all base classifiers in Table 5 are evaluated independently for each dataset and the results for the accuracy and geometric mean are reported in Tables 6 and 7, respectively. Best results are shown in bold font and the standard deviations of the accuracies and geometric means of 10 simulations are also shown using the "±" sign corresponding to each percentage. According to (Bouckaert & Frank, 2004), we have conducted the corrected pair *t*-test for the obtained results at a significant level $\alpha = 0.05$ using the Weka experiment environment. The numbers marked with an asterisk (*) are those that have a statistically significant difference with the best result (Boldface). The no asterisk numbers draw with the best result.

As can be seen, accuracy and G-Mean are different for most datasets. It means that there are differences between TPR and TNR. For example, as it is specified in the tables above, the SVM algorithm classifies the instances of the "Liver-disorders" dataset with an accuracy of 57.88%. On the other hand, the G-Mean for this dataset is 1.66% which means that SVM could not classify both positive and negative classes uniformly. (It classifies all of the 200 positive instances correctly, whereas just one instance from 145 negative instances is correctly classified). In other words, the significant difference between accuracy and geometric mean shows that the TPR has been sacrificed for TNR or vice versa.

4.5.2. Traditional ensemble classifiers

In the next step, four famous ensemble classifiers are selected to compare with the proposed approach. The selected classifiers are Bagging, Boosting, Stack Generalization and Stack-

Table 11
Comparison of Accuracy for HBoost and other approaches.

Dataset	*Single Classifiers (best)*	*Traditional Ensembles (best)*				*State-of-the-art Ensembles*		*Our approach*
		Bagging	*AdaBoost*	*Stacking*	*StackingC*	*Random Forest*	*CSS*	*HBoost*
Balance-Scale	90.22 ± 0.43*	90.36 ± 0.27*	90.92 ± 0.56*	93.80 ± 0.97	89.34 ± 0.36*	81.64 ± 0.50*	94.70 ± 1.04	**95.71** ± 0.61
Breast-cancer	74.23 ± 0.96*	73.95 ± 1.30*	71.61 ± 1.02*	72.73 ± 1.31*	73.66 ± 1.78*	70.45 ± 1.43*	74.83 ± 0.70*	**76.47 ± 0.52**
Car	93.68 ± 0.48*	93.45 ± 0.51*	95.99 ± 0.38*	98.68 ± 0.28	94.91 ± 0.35*	94.36 ± 0.30*	98.62 ± 1.72	**99.12** ± 0.44
Cmc	53.17 ± 0.68*	52.94 ± 0.73*	53.27 ± 0.71	54.76 ± 0.31	54.94 ± 0.53	51.13 ± 0.87*	53.51 ± 2.20*	**57.38** ± 0.77
Colic	85.38 ± 0.29	85.33 ± 0.36	82.36 ± 1.46*	84.96 ± 0.26*	83.74 ± 0.51*	86.03 ± 0.49	86.74 ± 0.84	**87.23** ± 0.60
Diabetes	76.86 ± 0.31*	77.36 ± 0.50	76.86 ± 0.31*	76.26 ± 0.25*	75.72 ± 0.45*	76.11 ± 0.52	77.86 ± 0.56	**78.29** ± 0.31
Ecoli	85.68 ± 0.41	85.62 ± 1.02	85.68 ± 0.41	83.09 ± 1.40*	84.47 ± 0.98	**86.16** ± 0.78	85.24 ± 3.50	85.44 ± 0.93
Glass	74.59 ± 1.07*	75.58 ± 0.98*	74.07 ± 1.24*	75.58 ± 1.63*	76.64 ± 1.30*	**79.03 ± 1.06**	75.84 ± 4.70*	78.86 ± 1.18
Hepatitis	85.10 ± 1.57	83.94 ± 0.73*	84.00 ± 0.69	83.01 ± 0.30*	82.69 ± 1.6*	84.13 ± 1.20	86.52 ± 1.12	**87.47** ± 0.97
Iris	96.08 ± 0.40	96.00 ± 0.82	97.17 ± 0.80	95.33 ± 0.88	94.92 ± 0.81*	94.25 ± 0.81*	96.73 ± 0.38	**97.73** ± 0.98
Liver-disorders	66.67 ± 1.39*	70.03 ± 1.63*	68.12 ± 1.15*	66.09 ± 1.32*	67.29 ± 1.43*	72.58 ± 0.94	71.51 ± 1.27*	**74.67** ± 1.02
Lymph	87.36 ± 0.80	85.61 ± 1.68*	83.85 ± 1.55*	84.01 ± 0.84*	84.35 ± 0.99*	83.31 ± 1.17*	86.49 ± 1.35*	**88.41** ± 0.88
Solar-flare-1	71.13 ± 0.72	72.33 ± 0.80	71.98 ± 0.93	71.52 ± 0.90	71.21 ± 1.34	69.97 ± 1.11*	**72.54 ± 3.31**	72.41 ± 0.86
Solar-flare-2	75.49 ± 0.19	75.15 ± 0.28	74.61 ± 0.54	75.07 ± 0.44	75.46 ± 0.47	73.01 ± 0.56*	73.77 ± 2.91*	**76.49** ± 0.37
Soybean	93.14 ± 0.48	93.08 ± 0.50	93.47 ± 0.39	94.09 ± 0.28	93.36 ± 0.50	93.21 ± 0.57	89.18 ± 13.71*	**95.10** ± 0.57
Vehicle	73.83 ± 0.54*	74.67 ± 1.36*	76.08 ± 0.93*	75.61 ± 0.66*	76.06 ± 0.57	75.26 ± 0.80*	77.01 ± 5.03*	**79.18** ± 1.16
Vote	96.64 ± 0.21	96.30 ± 0.36	95.95 ± 0.33	96.07 ± 0.59	96.36 ± 0.21	96.60 ± 0.20	**96.90 ± 0.31**	96.09 ± 0.21
Vowel	98.69 ± 0.26	98.65 ± 0.24	98.95 ± 0.20	99.09 ± 0.17	99.05 ± 0.24	97.83 ± 0.23*	93.03 ± 19.35*	**99.13** ± 0.10
Wine	98.81 ± 0.19	98.74 ± 0.37	97.68 ± 0.82	**99.09** ± 0.62	98.24 ± 0.66	97.68 ± 0.34*	99.10 ± 2.28	98.81 ± 0.32
Zoo	96.04 ± 0.00	96.29 ± 0.65	96.91 ± 0.33	96.04 ± 0.70	96.16 ± 1.35	93.94 ± 1.15*	94.84 ± 7.00*	**97.18** ± 0.66

Table 12
Comparison of Geometric mean for HBoost and other approaches.

Dataset	*Single Classifiers (best)*	*Traditional Ensembles (best)*				*State-of-the-art Ensembles*		*Our approach*
		Bagging	*AdaBoost*	*Stacking*	*StackingC*	*Random Forest*	*CSS*	*HBoost*
Balance-Scale	62.90 ± 0.29*	62.99 ± 0.18*	82.72 ± 2.51*	94.61 ± 0.39	62.31 ± 0.24*	61.65 ± 2.37*	94.83 ± 0.60	**95.13** ± 0.48
Breast-cancer	62.11 ± 2.24*	61.00 ± 1.92*	58.16 ± 1.95*	64.74 ± 0.20	57.34 ± 1.73*	53.31 ± 1.79*	61.57 ± 2.31*	**65.77** ± 1.06
Car	93.67 ± 0.84*	92.91 ± 0.32*	94.67 ± 0.92*	97.97 ± 0.64	93.64 ± 0.63*	90.87 ± 1.06*	98.19 ± 1.38	**99.11** ± 0.77
Cmc	61.76 ± 0.20*	62.20 ± 0.22*	61.77 ± 0.17*	64.92 ± 0.26	62.56 ± 0.38*	59.53 ± 0.82*	60.21 ± 4.03*	**66.96** ± 0.25
Colic	81.85 ± 0.34*	82.25 ± 0.59*	80.35 ± 1.27*	83.06 ± 0.09*	80.76 ± 0.49*	83.52 ± 0.58*	81.47 ± 1.93*	**85.85** ± 0.64
Diabetes	71.81 ± 0.94*	72.48 ± 1.05*	71.78 ± 0.95*	72.46 ± 0.44*	69.09 ± 0.71*	71.30 ± 0.76*	71.44 ± 0.96*	**75.11** ± 0.51
Ecoli	67.61 ± 0.42*	66.44 ± 1.28*	67.61 ± 0.42*	65.36 ± 1.07*	66.70 ± 0.96*	65.95 ± 0.92*	65.01 ± 4.46*	**71.28** ± 1.03
Glass	78.70 ± 1.22*	78.85 ± 1.47*	79.80 ± 2.11*	79.82 ± 2.05*	79.33 ± 1.81*	83.16 ± 0.57	77.50 ± 3.52*	**83.53** ± 2.17
Hepatitis	78.28 ± 0.83	78.01 ± 0.84	71.92 ± 1.93*	76.81 ± 2.46	68.56 ± 4.69*	64.49 ± 2.83*	72.53 ± 5.93*	**78.35** ± 2.46
Iris	97.04 ± 0.30	96.98 ± 0.61	97.87 ± 0.60	96.48 ± 0.67	95.98 ± 1.01*	95.66 ± 0.62*	95.85 ± 0.70	**98.05** ± 0.62
Liver-disorders	65.76 ± 1.30*	67.23 ± 2.43*	66.61 ± 1.53*	62.83 ± 0.73*	69.00 ± 1.92*	70.15 ± 0.99	68.48 ± 1.82*	**73.46** ± 1.02
Lymph	92.25 ± 0.45	92.11 ± 1.02	91.36 ± 0.60	82.26 ± 3.35*	76.26 ± 8.52*	51.38 ± 6.16*	84.64 ± 5.51*	**93.17** ± 0.71
Solar-flare-1	62.29 ± 0.65*	63.20 ± 1.07*	66.55 ± 3.52*	73.04 ± 1.55	60.28 ± 2.86*	64.73 ± 2.52*	65.66 ± 3.26*	**74.20** ± 1.13
Solar-flare-2	73.37 ± 0.46*	73.35 ± 0.84*	73.62 ± 0.43	75.47 ± 0.27	70.49 ± 0.95*	71.21 ± 0.79*	70.09 ± 3.04*	**76.07** ± 0.99
Soybean	97.81 ± 0.07	97.68 ± 0.18	97.61 ± 0.28	97.86 ± 0.16	97.75 ± 0.15	96.79 ± 0.44	97.88 ± 0.25	**98.18** ± 0.20
Vehicle	81.53 ± 0.41*	82.29 ± 0.99*	83.39 ± 0.68*	83.03 ± 0.51*	83.21 ± 0.43*	82.61 ± 0.60*	86.62 ± 2.13*	**88.23 ± 0.37**
Vote	**96.39** ± 0.21	96.27 ± 0.31	95.88 ± 0.52*	96.11 ± 0.56	96.26 ± 0.25	96.35 ± 0.20	95.80 ± 1.08*	96.35 ± 0.29
Vowel	99.49 ± 0.13	99.03 ± 0.22	99.42 ± 0.11	**99.50 ± 0.10**	99.48 ± 0.13	98.79 ± 0.13	99.48 ± 0.12	**99.50 ± 0.10**
Wine	99.21 ± 0.12	99.16 ± 0.27	98.29 ± 0.63	**99.37** ± 0.45	98.96 ± 0.47	98.38 ± 0.26	99.29 ± 0.49	99.21 ± 0.36
Zoo	94.43 ± 1.00	94.61 ± 1.05	94.48 ± 0.08	95.52 ± 1.36	93.65 ± 2.55	89.54 ± 3.55*	90.45 ± 2.23*	**96.41** ± 1.11

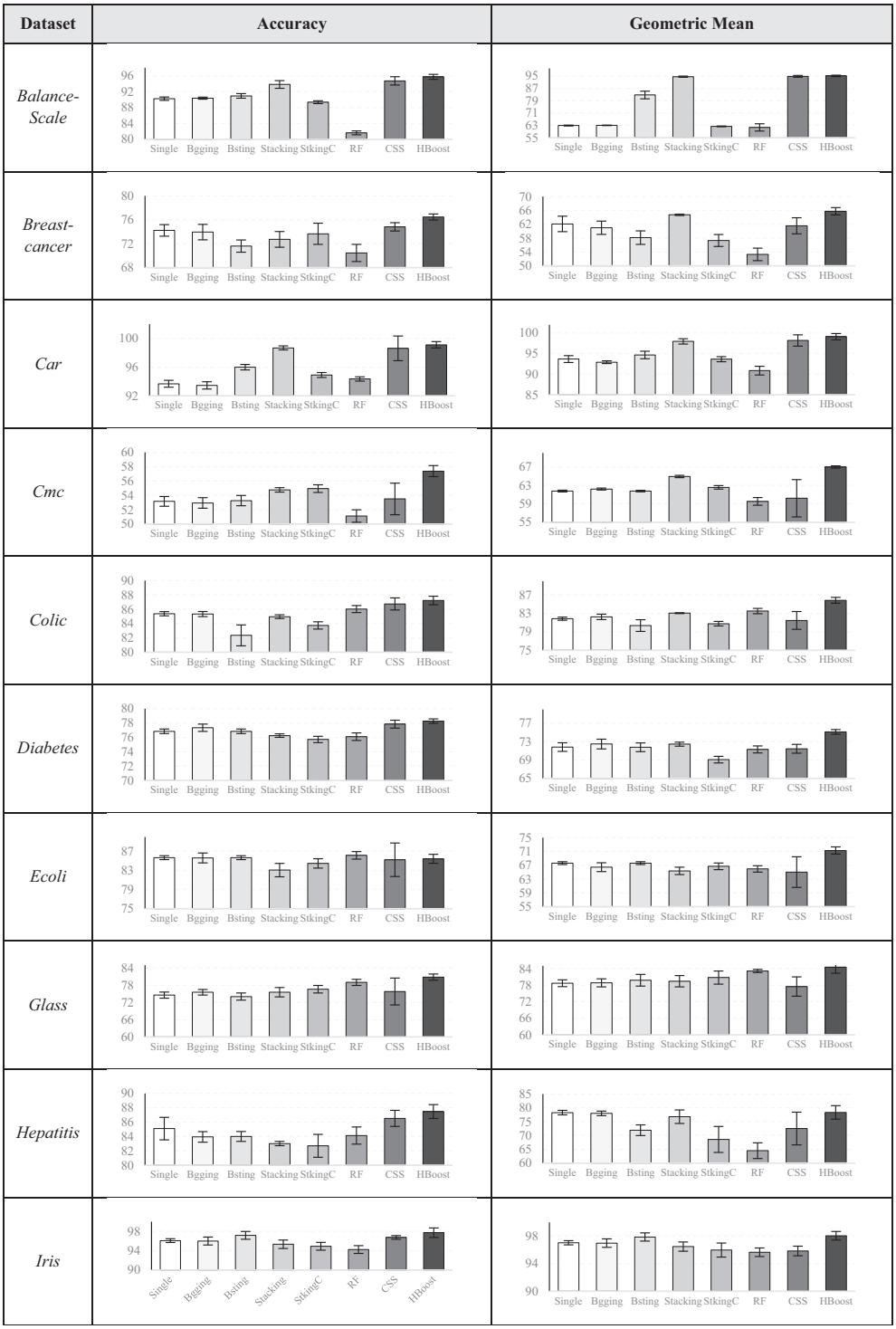


Fig. 3. (PART I)- Obtained results for the HBoost and other approaches.

ingC. Since Bagging and Boosting are homogeneous approaches, we set each classifier in Table 5 as the base classifier and record the obtained output. For stacking, we consider all of 10 classifiers as the base classifiers and then set each of them as Meta-Classifier.

The number of iterations in Bagging and Boosting is 10 as it is the most common and default value in many machine learning frameworks and packages such as Weka and R. Table 8 and Table 9 show the best accuracy and geometric mean of the obtained results. As can be observed, there is not a unique classification algorithm that outperform the other algorithms for all dataset. However, Stack Generalization shows a good performance in geometric mean. This could be as a result of using multiple learning algorithms that causes increasing the diversity.

4.5.3. Random Forest and classifier Subset Selection

Once the comparison with the aforementioned ensemble approached was conducted, a comparison with modern and the state-of-the-art ensemble techniques is applied. We have selected

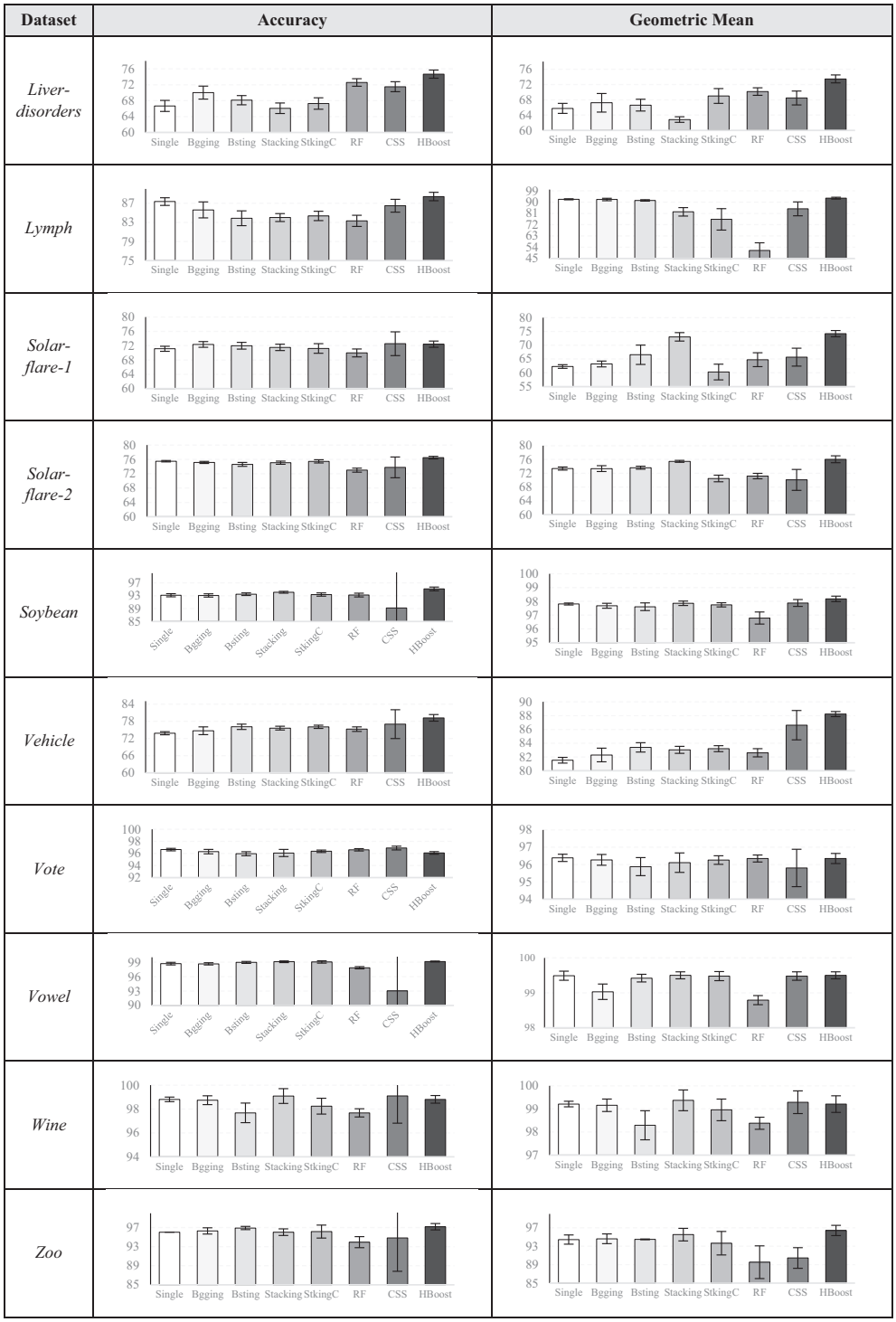


Fig. 3 (*continued*)

two ensemble generation techniques with different approaches: Random Forest (RF) and Classifier Subset Selection (CSS). The random forest belongs to homogeneous classifiers and extends the Bagging approach with the decision tree as its base classifier (Muzammal, Talat, Sodhro, & Pirbhulal, 2020). It has been largely used in ensemble communities.

The second approach is the so-called CSS (Mendialdua et al., 2015) which uses heterogeneous classifiers. As was stated previously in Section 2, the authors have argued that although heterogeneous classifiers can increase the potential diversity, selecting all of them might decrease the overall quality of the model. They convert the problem of selecting the best base classifiers to the Feature Subset Selection (FSS) problem. Afterward, by applying an evolutionary algorithm called Estimation of Bayesian Network Algorithm –EBNA- have found the best combination of the base classifiers. They called their approach Classifier Subset Selection (CSS). Table 10 shows their results (although they have reported both "Best Results" and "average ± std.dev", we have selected the latter as it is more confident). The first three columns are directly reported from the paper. The results of the geometric

Table 13
Number of participant classifiers for the best obtained result. The maximum value for each cell is 10.

Dataset	*1R*	*NB*	*DS*	*KNN*	*DT*	*BN*	*K**	*Ds.Tab*	*RT*	*SVM*	**Total (out of 100)**
Balance-Scale	4	2	1	2	1	3	1	6	5	1	**26**
Breast-cancer	3	3	2	6	2	4	0	3	3	2	**28**
Car	7	1	2	0	0	1	1	7	3	2	**24**
Cmc	5	6	6	3	0	3	0	7	0	1	**31**
Colic	5	4	5	2	3	0	2	8	1	2	**32**
Diabetes	7	4	7	2	4	3	4	1	5	1	**38**
Ecoli	0	5	0	0	5	0	0	4	4	3	**21**
Glass	2	7	8	1	4	2	0	3	1	2	**30**
Hepatitis	7	1	3	2	0	3	1	2	3	0	**22**
Iris	4	3	6	1	1	2	0	1	3	2	**23**
Liver-disorders	3	3	6	2	4	7	1	1	2	1	**30**
Lymph	4	2	3	0	3	0	0	1	2	3	**18**
Solar-flare-1	1	7	8	2	0	0	0	0	1	1	**20**
Solar-flare-2	4	1	9	3	1	1	2	1	2	5	**29**
Soybean	1	3	10	3	3	3	0	3	2	2	**30**
Vehicle	0	3	5	0	2	7	0	1	0	1	**19**
Vote	1	1	0	2	0	1	0	2	1	0	**8**
Vowel	0	1	10	9	1	0	0	5	0	0	**26**
Wine	5	1	9	1	2	0	0	4	2	0	**24**
Zoo	9	7	1	1	3	0	0	1	0	2	**24**

mean were obtained by an implementation done by us, since we only had access to the mean accuracy and standard deviation.

4.5.4. Hboost and other approaches

Finally, we applied HBoost to each dataset. To do this, each dataset is split into 10 folds. Eight folds are used for training, one fold for validation set and one fold for the test set. The average experimental results of 10 simulation runs in terms of percentage accuracy and geometric mean obtained by the HBoost and comparing methods performed on 20 datasets are reported in Table 11 and Table 12. It is observed from the summarized experimental results (shown in Table 11), that for 15 out of 20 datasets, HBoost dominates all the other methods in terms of the accuracy measures. For geometric mean, it reaches to 18 out of 20 datasets. It means that the geometric mean has influenced more than the accuracy. We believe that this is the effectiveness of using heterogeneous classifiers. Using different learning algorithms results in more coverage of dataset with appropriate classifiers and consequently, the minority class is more likely to be classified correctly than using one algorithm for Boosting. From a statistical point of view, it can be seen that HBoost has never been a loser for both accuracy and geometric mean.

For a better comparison, the chart model for each dataset is depicted in Fig. 3. As it can be observed, although our approach has not reached the best standard deviation for some datasets, the obtained results are acceptable in comparison with other approaches. HBoost uses different types of classifiers and for each type, construct 10 classifiers. If a classifier with high standard deviation participates in the final decision-making process, other stable classifiers suppress it and compensate its negative impact with their low standard deviations.

In Table 13, the type and the number of participant classifiers in the best obtained result for each dataset are presented. The last column equals to the total number of classifiers that construct the model. Since there are 10 different learning algorithms and each algorithm generates 10 homogeneous classifiers, the maximum value for this column is 100.

There are some interesting aspects in Table 13 that are better to be highlighted. Naïve Bayes and Decision Table classifiers exist in almost all datasets (Decision Table exists in all datasets but Solar-flare-1). It means that the combination of these classifiers with other classifiers can produce a high diversity.

On the other hand, some other classifiers participate just in some datasets. For example, KStar appears only in seven datasets which means that KStar algorithm makes low diversity in a combination with other algorithms. It is also clear that for each dataset, more than half of classifiers are pruned from the classifier pool. For example, the total number of participant classifiers for dataset "Ecoli" is 21 which means that 79 classifiers have been eliminated in the pruning step of the HBoost algorithm. This makes it possible to create a simpler ensemble model than the one in which all of the classifiers are included.

5. Conclusions and future work

In this paper, a heterogeneous ensemble classifier is proposed that is applied to the Boosting method to reduce the bias error. The intuition for applying heterogeneous algorithms on Boosting is to benefit the advantages of both different learning algorithms and low-biased classifiers. In other words, we use multiple learning algorithms to increase the diversity between the classifiers and for each algorithm and we construct a model by Boosting method to reduce the bias error. Since some classifiers may correlate with each other, we categorized them into some groups and apply the ensemble pruning based on the entropy measure to reach maximum diversity. Then by simplifying the survived classifiers in each group, the final decision is generated by combining the classifier's outputs.

Experimental results with 20 datasets from the UCI ML repository show the better performance of HBoost in comparison to different base classifiers and ensemble methods. The improvement of geometric mean was more than the accuracy and it means that our approach is also a competitive one for imbalanced datasets.

It is worth mentioning that we do not claim that the HBoost always has a better performance than the other ensemble methods. We believe that when both false positives (type I error) and false negatives (type II error) are essential, our approach can be considered as an efficient one. Nevertheless, it is certainly possible that for some datasets, other approaches outperform the HBoost.

For future works, it is better to investigate other combination methods such as considering a meta classifier to combine the base classifiers' output. It is also suggested to consider the dynamic pruning of the classifiers. That is, instead of eliminating some classifiers for all instances, it is better to postpone the impact of the classifiers in the decision-making process until a new instance comes. This is the way that dynamic ensemble pruning methods use (Cruz et al., 2019).

CRediT authorship contribution statement

Hamid Reza Kadkhodaei: Conceptualization, Software, Validation, Writing - original draft, Writing - review & editing, Visualization. **Amir Masoud Eftekhari Moghadam:** Validation, Formal analysis, Writing - original draft, Supervision, Project administration. **Mehdi Dehghan:** Methodology, Formal analysis, Investigation, Data curation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Bouckaert, R. R., & Frank, E. (2004). Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms. Paper presented at the Advances in Knowledge Discovery and Data Mining, Berlin, Heidelberg.

Bramer, M. (2013). *Principles of Data Mining* (Second ed.). Springer Publishing Company Incorporated.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140. https://doi.org/10.1007/bf00058655.

Cavalcanti, G. D. C., Oliveira, L. S., Moura, T. J. M., & Carvalho, G. V. (2016). Combining diversity measures for ensemble pruning. *Pattern Recognition Letters, 74*, 38–45. https://doi.org/10.1016/j.patrec.2016.01.029.

Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). SMOTEBoost: Improving Prediction of the Minority Class in Boosting, Knowledge Discovery in Databases: PKDD 2003, Berlin, Heidelberg.

Coelho, A. L. V., & Nascimento, D. S. C. (2010). On the evolutionary design of heterogeneous Bagging models. *Neurocomputing, 73*(16), 3319–3322. https://doi.org/10.1016/j.neucom.2010.07.008.

Cruz, R. M. O., Oliveira, D. V. R., Cavalcanti, G. D. C., & Sabourin, R. (2019). FIRE-DES++: Enhanced online pruning of base classifiers for dynamic ensemble selection. *Pattern Recognition, 85*(1), 149–160. https://doi.org/10.1016/j.patcog.2018.07.037.

Cruz, R. M. O., Sabourin, R., & Cavalcanti, G. D. C. (2017). META-DES.Oracle: Meta-learning and feature selection for dynamic ensemble selection. *Information Fusion, 38*, 84–103. https://doi.org/10.1016/j.inffus.2017.02.010.

Cruz, R. M. O., Sabourin, R., & Cavalcanti, G. D. C. (2018). Prototype selection for dynamic classifier and ensemble selection. *Neural Computing and Applications, 29*(2), 447–457. https://doi.org/10.1007/s00521-016-2458-6.

Cruz, R. M. O., Sabourin, R., Cavalcanti, G. D. C., & Ing Ren, T. (2015). META-DES: A dynamic ensemble selection framework using meta-learning. *Pattern Recognition, 48*(5), 1925–1935. https://doi.org/10.1016/j.patcog.2014.12.003.

Dai, Q., Ye, R., & Liu, Z. (2017). Considering diversity and accuracy simultaneously for ensemble pruning. *Applied Soft Computing, 58*, 75–91. https://doi.org/10.1016/j.asoc.2017.04.058.

de Souza, E. N., & Matwin, S. (2012, 2012//). Improvements to AdaBoost Dynamic. Paper presented at the Advances in Artificial Intelligence, Berlin, Heidelberg.

Dos Santos, E. M., Sabourin, R., & Maupin, P. (2008). A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. *Pattern Recognition, 41*(10), 2993–3009. https://doi.org/10.1016/j.patcog.2008.03.027.

Dua, D., & Graff, C. (2019). *UCI Machine Learning Repository. Retrieved from.* Irvine, CA: University of California, School of Information and Computer Science.

Espíndola, R., & Ebecken, N. (2005). On extending f-measure and g-mean metrics to multi-class problems. Sixth international conference on data mining, text mining and their business applications, 35, 25-34. doi:10.2495/DATA050031.

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*(1), 119–139. https://doi.org/10.1006/jcss.1997.1504.

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn. *Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media.

Golestani, A., Ahmadian, K., Amiri, A., & JahedMotlagh, M. (2007, 11-13 July 2007). A Novel Adaptive-Boost-Based Strategy for Combining Classifiers Using Diversity Concept. Paper presented at the 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007), Melbourne, Qld., Australia

Jankowski, N. (2013, 16-19 April 2013). Fast heterogeneous boosting. Paper presented at the 2013 IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL), Singapore.

Kadkhodaei, H., & Moghadam, A. M. E. (2016, 27-28 Jan. 2016). An entropy based approach to find the best combination of the base classifiers in ensemble classifiers based on stack generalization. Paper presented at the 2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA).

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning, 51*(2), 181–207. https://doi.org/10.1023/A:1022859003006.

LeCun, Y., Cortes, C., & Burges, C. J. C. (2014). MNIST handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist, 2.

Lertampaiporn, S., Thammarongtham, C., Nukoolkit, C., Kaewkamnerdpong, B., & Ruengjitchatchawalya, M. (2013). Heterogeneous ensemble approach with discriminative features and modified-SMOTEbagging for pre-miRNA classification. *Nucleic Acids Research, 41*(1). https://doi.org/10.1093/nar/gks878 e21.

Mendialdua, I., Arruti, A., Jauregi, E., Lazkano, E., & Sierra, B. (2015). Classifier Subset Selection to construct multi-classifiers by means of estimation of distribution algorithms. *Neurocomputing, 157*, 46–60. https://doi.org/10.1016/j.neucom.2015.01.036.

Muzammal, M., Talat, R., Sodhro, A. H., & Pirbhulal, S. (2020). A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks. *Information Fusion, 53*, 155–164. https://doi.org/10.1016/j.inffus.2019.06.021.

Obregon, J., Kim, A., & Jung, J.-Y. (2019). RuleCOSI: Combination and simplification of production rules from boosted decision trees for imbalanced classification. *Expert Systems with Applications, 126*, 64–82. https://doi.org/10.1016/j.eswa.2019.02.012.

Oliveira, D. V. R., Cavalcanti, G. D. C., & Sabourin, R. (2017). Online pruning of base classifiers for Dynamic Ensemble Selection. *Pattern Recognition, 72*, 44–58. https://doi.org/10.1016/j.patcog.2017.06.030.

Palit, I., & Reddy, C. K. (2012). Scalable and Parallel Boosting with MapReduce. *IEEE Transactions on Knowledge and Data Engineering, 24*(10), 1904–1916. https://doi.org/10.1109/TKDE.2011.208.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine, 6*(3), 21–45. https://doi.org/10.1109/MCAS.2006.1688199.

Porwik, P., Doroz, R., & Wrobel, K. (2019). An ensemble learning approach to lip-based biometric verification, with a dynamic selection of classifiers. *Expert Systems with Applications, 115*, 673–683. https://doi.org/10.1016/j.eswa.2018.08.037.

Ramírez, J., Górriz, J. M., Ortiz, A., Martínez-Murcia, F. J., Segovia, F., Salas-Gonzalez, D., ... Puntonet, C. G. (2018). Ensemble of random forests One vs. Rest classifiers for MCI and AD prediction using ANOVA cortical and subcortical feature selection and partial least squares. *Journal of Neuroscience Methods, 302*, 47–57. https://doi.org/10.1016/j.jneumeth.2017.12.005.

Reddy, C. K., & Park, J.-H. (2011). Multi-resolution boosting for classification and regression problems. *Knowledge and Information Systems, 29*(2), 435–456. https://doi.org/10.1007/s10115-010-0358-0.

Rokach, L. (2009). Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis, 53*(12), 4046–4072. https://doi.org/10.1016/j.csda.2009.07.017.

Russell, S. J. (2010). Artificial intelligence : a modern approach: Third edition. Upper Saddle River, N.J. : Prentice Hall, [2010] ©2010.

Sabzevari, M., Martínez-Muñoz, G., & Suárez, A. (2018). Vote-boosting ensembles. *Pattern Recognition, 83*, 119–133. https://doi.org/10.1016/j.patcog.2018.05.022.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning, 5*(2), 197–227. https://doi.org/10.1007/BF00116037.

Seewald, A. K. (2002). *How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness.* Machine Learning.

Seiffert, C., Khoshgoftaar, T. M., Hulse, J. V., & Napolitano, A. (2010). RUSBoost: A Hybrid Approach to Alleviating Class Imbalance. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 40*(1), 185–197. https://doi.org/10.1109/TSMCA.2009.2029559.

Sultana, N., & Islam, M. M. (2020). *Meta Classifier-Based Ensemble Learning For Sentiment Classification.* Singapore: Algorithms for Intelligent Systems.

Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics.* https://doi.org/10.1016/j.aci.2018.08.003. in press.

Tin Kam, H. (1995, 14-16 Aug. 1995). Random decision forests. Paper presented at the Proceedings of 3rd International Conference on Document Analysis and Recognition.

van Rijn, J. N., Holmes, G., Pfahringer, B., & Vanschoren, J. (2018). The online performance estimation framework: Heterogeneous ensemble learning for data streams. *Machine Learning, 107*(1), 149–176. https://doi.org/10.1007/s10994-017-5686-9.

Volna, E., & Kotyrba, M. (2017). Enhanced ensemble-based classifier with boosting for pattern recognition. *Applied Mathematics and Computation, 310*, 1–14. https://doi.org/10.1016/j.amc.2017.04.019.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann Publishers Inc..

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82. https://doi.org/10.1109/4235.585893.

Woźniak, M., Graña, M., & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. *Information Fusion, 16*, 3–17. https://doi.org/10.1016/j.inffus.2013.04.006.

Xing, H.-J., & Liu, W.-T. (2020). Robust AdaBoost based ensemble of one-class support vector machines. *Information Fusion, 55*, 45–58. https://doi.org/10.1016/j.inffus.2019.08.002.

Zhao, Q. L., Jiang, Y. H., & Xu, M. (2010). *Incremental Learning by Heterogeneous Bagging Ensemble.* Berlin, Heidelberg: Advanced Data Mining and Applications.