Sesmero, M.P., Alonso-Weber, J.M., Sanchis, A. (2020). CCE: An ensemble architecture based on coupled ANN for solving multiclass problems. *Information Fusion*, 58, pp. 132-152.

DOI: 10.1039/c3lc51419f

CCE: An Ensemble Architecture based on coupled ANN for solving multiclass problems

M. Paz Sesmero[1], Juan M. Alonso-Weber[2], Araceli Sanchis[3]
Computer Science Department
Universidad Carlos III de Madrid
Avenida de la Universidad 30, Leganés 28911, Madrid (Spain)
{[1]msesmero, [3]masm}@inf.uc3m.es, [2]jmaw@ia.uc3m.es

Abstract

The resolution of multiclass classification problems has been usually addressed by using a *"divide and conquer"* strategy that splits the original problem into several binary subproblems. This approach is mandatory when the learning algorithm has been designed to solve binary problems and a multiclass version cannot be devised.

Artificial Neural Networks, ANN, are binary learning models whose extension to multiclass problems is rather straightforward by using the standard *1-out-of N* codification of the classes. However, the use of a single ANN can be inefficient in terms of accuracy and computational complexity when the data set is large, or the number of classes is high.

In this work, we exhaustively describe CCE, a new classifier ensemble based on ANN. Each member of this new ensemble is a couple of multiclass ANN's. Each ANN is trained using different subsets of the dataset ensuring these subsets to be disjoint. This new approach allows to combine the benefits of the *divide and conquer* methodology, with the use of multiclass ANNs and with the combination of individual classification modules that give a complete answer to the addressed problem. The combination of these elements results in a classifier ensemble in which the diversity of the base classifiers provides high accuracy values. Moreover, the use of couples of ANN proves to be tolerant to labeling noise and computationally efficient.

The performance of CCE has been tested on various datasets and the results show the higher performance of this approach with respect to other used classification systems.

Keywords: Ensemble of classifiers; Multiclass-classification tasks; Artificial Neural Networks; Diversity

1. Introduction

A classifier is a system that, given an input example, assigns that example to one class or category [1]. The most common way of establishing the mapping function is to deduce it from a set of previously categorized instances by applying a specific learning algorithm. Despite many classification algorithms such as the SVM family or logistic regression are defined only to binary problems [2] [3], many real classification problems require algorithms that may classify instances that belong to more of two categories. When the set of potential categories is of finite cardinality and contains more than two elements, the mapping task is called multi-class classification.

Most research in *Machine Learning* has been focused on studying and comparing the performance of different learning algorithms such as decision trees [4], artificial neural networks [5], inductive logic programming [6], [7] or Bayesian algorithms [7]. Given that each learning algorithm has both, advantages and drawbacks the main conclusion of these studies is that there is not a single approach that can claim to be superior to any other [8]. So, the strategy of combining different classification models has attracted the interest of the Machine Learning Community. These kind of approaches are known as hybrid methods, multiple experts, mixture of experts, ensemble methods or ensembles of classifiers [9]. Given that today there are many unanswered questions about ensembles of classifiers, the improvement of classic algorithms and the design of new methodologies are one of the main focus of interest in the Machine Learning Community [10], [11], [12][13].

The main idea behind the ensembles of classifiers is to use the predictions of a pool of individual classifiers (base learners) in order to obtain a system that is more accurate than the base learners that make it up [14]. Therefore, to obtain the final ensemble decision a procedure to combine the individual decisions must be established. There are two main strategies for combining the base learner decisions: fusion and selection [15]. Classifier selection assumes that each base learner has a region of the space in which it is the most reliable. So, when an instance must be classified, the ensemble decision coincides with the decision given by the classifier (or subset of base classifiers) that is expert in the region of the space to which the instance belongs. In classifier fusion, the ensemble decision is obtained by combining the decisions from all the base learners. Classifier fusion algorithms include combining rules such as the average, majority voting or the weighting methods and, more complex integration models, such as meta-learning methods [16] [17].

The main premise to obtain a good ensemble when the final decision is obtained using fusion methods is that the error rate of the base learners must be low, and the errors made by one member of the ensemble must be compensated by the correct predictions from other base learners. That is, the members of the ensemble must be both accurate and diverse [18]. However, the more accurate the base learners are, the more similar they are likely to be. And, the more diverse the classifiers are, the more uncertain the individual predictions are [19]. So, one of the main challenges in the field of the ensembles of classifiers is to achieve a balance between these two conditions.

An important drawback of the classifier ensembles is that they usually require longer training times than single classifiers. This potential time increase depends on the number of base learners and how are they built. Therefore, a second challenge related to the design of a good classifier ensemble is to seek for a good compromise between the required ensemble training time and its accuracy.

To analyze the influence of both, diversity and accuracy of the base learners on the accuracy of the ensemble, we have designed two different ensemble architectures that are complementary. Moreover, these architectures have been conceived with the goal to obtain classifier systems that are both accurate, and efficient in terms of training time.

The first proposal, named BCE –Binary-Complementary Ensemble–, builds base learners that are highly accurate but not very diverse. To achieve this goal, the members of the ensemble have been implemented with two coupled classifiers: a binary classifier and a multiclass classifier. The binary classifier is trained to distinguish whether an example belongs to a certain class. On the other hand, the second classifier, named complementary classifier, is a multi-class classifier with *k-1* outputs (where *k* is the number of classes). The goal of this classifier is to label the instances that have been labelled as negative by the corresponding binary classifier. Moreover, with the double goal of promoting diversity among the classifiers and improving the learning time in terms of both accuracy and computational cost, each one of the classifiers that compose a base learner, is trained using a specific feature subset. The experimental evaluation of BCE over different domains indicates that on domains with a large number of features, BCE is equal or more accurate than other traditional classifiers, but clearly much more efficient. On the contrary, in domains where the number of features is relatively small, the accuracy of BCE is not good. However, when the feature selection process is switched off, the accuracy of BCE is equivalent or even better than those obtained with other classification models. The details and the experimental evaluation of this architecture can be found in [20].

The second architecture, named CCE –Complementary-Complementary Ensemble–, generates base learners that are relatively accurate but highly diverse. To accomplish this requirement, we propose transforming the initial learning task into a pool of new pairwise disjoint subproblems. So, an initial classification problem with instances belonging to one of *k* classes is transformed in a pool of pairwise disjoint subproblems. One of these subproblems works with the instances that belong to *j* classes ($2 \leq j \leq k\text{-}2$) and the other one with the examples that belong to the (*k-j*) remaining classes. A preliminary description of the conceptual framework of CCE was presented in [21]. In this preliminary work the accuracy of the proposed ensemble was tested only on the popular MNIST database [22]. The good results obtained in this domain have motivated us to evaluate whether this methodology can be successfully applied to resolve other multiclass classification problems. The present work accomplishes this task, including also an exhaustive comparison with other well-established methods.

Machine Learning literature collects different metrics such as accuracy, precision, recall or F_measure, that can be used to analyze various aspects of the multi-class classification systems [23]. To evaluate all the different characteristics of the proposed classification systems, we have chosen to use the accuracy, the diversity (or disagreement degree between the members of the ensemble) using the *Q-statistic*, and the computational cost measuring the training time.

In this research, we describe the theoretical basis of CCE and indicate, in an intensive way, how the base classifiers are constructed and how the final decision of the ensemble is generated. To test the viability of this proposal, we present an exhaustive experimental analysis in which CCE is evaluated in 20 benchmark classification tasks according to the parameters indicated above (accuracy, diversity and training time). In addition, an analysis of its tolerance to labeling noise is presented. The experimental analysis also includes a study in which the CCE performance is compared with those obtained with:

- BCE. Because it is a system that encourages the precision of the base learners versus their diversity.
- A single ANN. Since all the base learners are ANN, this model can be considered the standard classification model.
- An OAA –One Against All– architecture, an OAO architecture, and ECOC –Error-Correcting Output Codes–. These architectures are the tree basic approaches to solve multiclass problems using binary decomposition.

- *Bagging* and *Boosting*: Because, as is stated in the literature, these systems are the most used ensembles of classifiers. In addition, although Bagging and Boosting seem to be quite old, they are still one of the best alternatives.

To avoid biases that may be attributed to different implementations of the same algorithm, [24] [3] all the systems used in the comparison have been implemented in C ++ by the authors of this work. Moreover, all the models are based in Artificial Neural Networks, they use the same set of parameters and, in the case of the ensembles, they use the same combination method. With these premises, it is guaranteed that no system (in particular CCE) benefits from a favorable choice of parameters.

It is worth mentioning that this paper extends the experimental work shown in previous works [20], [21]. The aim is to propose an exhaustively tested system; therefore, we have now included:

- A completely new evaluation for the CCE architecture, as we have included more domains rather than using only a specific version of MNIST.
- *Boosting* and the OAO architecture are now included for comparison.
- A thorough test of the tolerance to noise of our system and of the other baseline methods, based on training with examples with incorrect class labels.

The remainder of the paper is organized as follows: First, some related work on ensemble classifiers is given. Then Section 3 describes the theoretical framework proposed for our ensemble model and presents the architecture of CCE. Section 4 introduces the data sets, the method and the measures used to evaluate CCE. Section 5 shows the experimental evaluation. Last, Section 6 presents some conclusions that have been derived from this work.

2. Ensembles of Classifiers.

As previously mentioned, an ensemble of classifiers is a pool of classifiers whose outputs are combined in an attempt to reach a more accurate decision than the best of its members [25].

The main phases to develop an ensemble are the following [26]: 1) *decomposition phase*: if ensembles are applied to multi-class problems, then the first phase can be to split the classification task into several sub-problems; 2) *generation phase*: build the base classifiers; 3) *pruning phase*: some base classifiers obtained from the generation phase can be dismissed if they are redundant; 4) and finally the *integration phase* provides a strategy to get the final answer of the ensemble, combining the answer from each individual classifier.

To achieve the main goal of the ensemble (to outperform each of its members), the members of the ensemble must be both accurate and diverse [18], [27]. Empirical and theoretical studies have proved that an effective way to achieve accurate and diverse base learners is varying the set of hypotheses that are accessible by each base learner. According to [28], there are two ways to achieve this goal: i) Altering the training data that the base learner receives and/or ii) varying the learning algorithm.

Approaches that alter the training data are usually subdivided into three groups [29]. Some methods, such as *Bagging* [30], *Boosting* [31] or the ensemble model proposed in [32], build each member of the ensemble using different subsets of patterns. Other methods, such as the

Decision Forest proposed in [33] or the ensembles presented in [34], build each base learner using all the patterns contained in the training set, but with an alteration of the set of attributes that describe them. The third alternative consists in manipulating the output targets, that is, in decomposing the original classification problem into new subproblems (decomposition phase). This category includes methods like OAO –One Against One– [35], [36], OAA –One Against All – [37], OAHO –One Against High Order– [38] or ECOC [39], that solve multiclass problems by converting them into several binary subproblems. Finally, some methods build each base leaner combining or modifying some of these processes. So, in *Random Forest* [40] diversity is induced training each member of the ensemble (tree) on a different sample of instances and choosing the best split attribute among a subset of features. Other methods as RFW –*Random Feature Weights*– [41] alter the training data assigning a weight to each attribute.

On the other hand, methods that change the learning algorithm can be subdivided in two groups: Approaches that use different versions of the same learning algorithm and approaches that use different learning algorithms. Among the strategies for generating diversity using a single learning algorithm are the proposed in [42] where the pool of ANN that compose the ensemble are trained starting from different initial weights, and Randomization where diversity is achieved by varying the criterion used to expand the decision tree nodes. Approaches where diversity is obtained using different learning algorithms seek to increase the performance of the base learners by exploiting the strength of each algorithm. In these systems, known as heterogeneous ensembles, the ensemble is generated by combining several base learners that are trained using ANN, decision trees, Bayesian models, and so on. Stacking [43] and most of its variants [44] achieve diversity by applying this approach.

More detailed surveys of these and other recent ensemble approaches can be found in [45] and [16].

3. Complementary-Complementary Ensemble Architecture

CCE is an ensemble architecture for classifying multi-class patterns that seeks to increase the diversity among the base learners by decomposing the learning task into two coupled disjoint multi-class subproblems. According to this requirement, when the application domain has instances belonging to *k* classes, one of the classifiers that compose the base learner is trained with the instances that belong to *j* classes ($2 \leq j \leq k-2$) and the other one is trained with the instances that belong to the (k-j) remaining classes. For example, in a four-class problem ($k=4$; $j=2$) in which the classes are labelled with $\{c_1, c_2, c_3, c_4\}$, the distribution of classes used to build the different base learners (BL_i) is shown in **Table 1**.

Table 1. Class distribution scheme of CCE for a four-class problem.

Base Learner	Classifier #1	Classifier #2
BL$_1$	$\{c_1, c_2\}$	$\{c_3, c_4\}$
BL$_2$	$\{c_1, c_3\}$	$\{c_2, c_4\}$
BL$_3$	$\{c_1, c_4\}$	$\{c_2, c_3\}$

Taking into account the possible values of *j* and omitting the dual combinations[1], the potential number of base learners, $L^{(1)}$, is bounded by Eq. (1).

$$L^{(1)} = 2^{k-1} - (k+1) \qquad (1)$$

Given that the number of possible base learners increases exponentially with the number of classes, the use of an *over-produce and choose* [26] methodology, in which all the candidates for base learners are built and then the subgroup that offers a higher degree of diversity or accuracy is selected, can be an unfeasible alternative due to its high computational cost. To overcome this difficulty, we propose the use of an *ad hoc* technique that first heuristically picks the number and the structure (class distribution) of the base learners, and then builds these learners. In the next epigraph, we show how to choose the structure of the base learners to achieve good classification performance with short training time.

3.1. Design of the Base Learners

The base learners that integrate CCE are composed by two multiclass coupled classifiers. So, in a *k* class domain, the first classifier is trained with the instances that belong to *j* classes ($2 \leq j \leq k-2$) and the other one is trained with the instances that belong to the (k-j) remaining classes.

The proposed class decomposition scheme allows that some classifiers must classify examples that do not belong to any of the learnt classes. That is, given an example that belongs to class c_i, those classifiers that have not been trained using examples belonging to this class give a decision that is always erroneous. If we consider the class distribution scheme shown in **Table 1**, we can observe that this unfavorable scenario appears in one of the two classifiers that composes each base learner. For example, no pattern belonging to c_1 is correctly classified by the classifiers identified as *Classifier#2* because no pattern belonging to c_1 is used during the training phase of these classifiers (following the same example, *Classifier#2* of BL_1 is trained using only the patterns belonging to classes c_3 and c_4).

To delimit the number of arbitrary decisions that are given by each one of the classifiers that compose a base learner and, at the same time, to limit their training time, we propose to fix the value of *j* to *k/2*. This restriction does not only affect to the expected accuracy and the training time of the classifiers that compose each base learner but also reduces the number of possible base learners that can be built. Consequently, the number of potential base learners is reduced from $L^{(1)}$ to $L^{(2)}$, with $L^{(2)}$ defined in Eq. (2).

$$\mathrm{L}^{(2)} = \begin{cases} \frac{1}{2}\,\frac{\mathrm{k}!}{\left(\frac{\mathrm{k}}{2}\right)!\left(\frac{\mathrm{k}}{2}\right)!}, & \text{if } k \text{ is an even number} \\ \frac{1}{2}\,\frac{(\mathrm{k}+1)!}{\left(\frac{\mathrm{k}+1}{2}\right)!\left(\frac{\mathrm{k}+1}{2}\right)!} & \text{if } k \text{ is an odd number} \end{cases} \qquad (2)$$

The constraint imposed on the value of *j* considerably reduces the number of possible base learners (for example, when k=6, $L^{(1)}$=25, $L^{(2)}$=10; k=8, $L^{(1)}$=119, $L^{(2)}$=35; k=10, $L^{(1)}$=501, $L^{(2)}$=126, and so on) but when the number of classes is high, the number of learners that can be built is

[1] Note that the combination [{c_1, c_2}∈ classifier #1; {c_3, c_4}∈ classifier #2] is equivalent to [{c_3, c_4}∈ classifier#1; {c_1, c_2}∈ classifier #2]. Therefore, only one of these dual combinations is considered.

still very high. As an alternative approach applicable in the cases in which L$^{(2)}$ is greater than 20 (that is, when k>6), we propose to reduce the number of base learners to the number of classes ($L^{(3)}$= k).

This new restriction on the number of base learners implies choosing from the group of candidates which specific base learners will constitute the ensemble. A possibility is to build all the *candidates* for the base learners and then to select those that, when combined, offer more diversity or accuracy. Depending on the value of L$^{(2)}$ –number of possible candidates– and L$^{(3)}$ –bounded ensemble size– either an exhaustive search, or another selection algorithm can be applied [46]. Another alternative is to use an *ad hoc* technique that first selects the class distribution of the prefixed number of base learners, and then builds them. To avoid generating all the candidates for the base learners and therefore, to reduce the computational cost, CCE follows the second approach. Next, we show how to choose the structure of the base learners to achieve a classification performance as best as possible.

3.2. Assignment of Classes to Base Learners

As noted above, one of the main difficulties of the CCE architecture is the high number of arbitrary decisions that are given by each one of the classifiers that compose a base learner. In a balanced domain (all classes have the same number of instances) where the two classifiers that integrate a base learner are trained with half of the classes (j =k/2), the 50% of the decisions given by each classifier will be fictitious. Ideally, it is expected that when these classifiers are combined, the output of the classifier that was trained with examples belonging to the class that must be predicted prevails over the fictitious output of the other classifier. Experimentally we have observed that, in a great number of cases, this premise is satisfied. Nevertheless, analyzing the global output given by different base learners in different domains we have observed that a high number of wrong classifications have their origin in that this assumption is not satisfied. Therefore, one of the main challenges of the CCE architecture is to reduce this mislabeling. Experimentally we have realized that an effective way of reducing this type of error is to guarantee that, given any two classes, there is at least a base classifier that has been trained using samples from both classes. This restriction allows that when the domain contains two classes that can easily be confused, there is, at least, a classifier that has been trained to distinguish them.

To satisfy the previous requirements and establish the class distribution of the base learners, a simple *trial and error* algorithm is used. This algorithm can be summarized as follows:

1. Build, in a random way, a tentative class distribution for all the $L^{(3)}$ base learners[2].
2. Check that there are no two learners with the same class distribution.
3. Verify that given any two classes there is at least one classifier that has been trained using examples that belong to both classes.
4. Repeat the process until requirements 2 and 3 are satisfied.

Considering a domain of ten classes and setting $L^{(3)}=k$, a possible scheme of the CCE topology is shown in **Table 2.a**. Additionally, and based on this topology, **Table 2.b** shows the number of classifiers built using examples belonging to the classes c_i and c_j.

[2] Note that $L^{(3)} = L^{(2)}$ when $k \leq 6$; otherwise $L^{(3)}=k$.

Table 2 a) A possible class distribution scheme of CCE for a ten-class problem. b) Number of classifiers that, based on Table 2a, have been built using examples belonging to the classes c_ic_j.

Base Learner	Classifier #1	Classifier #2
BL$_1$	$\{c_0, c_5, c_7, c_8, c_9\}$	$\{c_1, c_2, c_3, c_4, c_6\}$
BL$_2$	$\{c_0, c_4, c_7, c_8, c_9\}$	$\{c_1, c_2, c_3, c_5, c_6\}$
BL$_3$	$\{c_1, c_3, c_7, c_8, c_9\}$	$\{c_0, c_2, c_4, c_5, c_6\}$
BL$_4$	$\{c_1, c_3, c_4, c_5, c_8\}$	$\{c_0, c_2, c_6, c_7, c_9\}$
BL$_5$	$\{c_0, c_1, c_2, c_4, c_6\}$	$\{c_3, c_5, c_7, c_8, c_9\}$
BL$_6$	$\{c_2, c_5, c_6, c_7, c_9\}$	$\{c_0, c_1, c_3, c_4, c_8\}$
BL$_7$	$\{c_4, c_5, c_6, c_7, c_8\}$	$\{c_0, c_1, c_2, c_3, c_9\}$
BL$_8$	$\{c_2, c_3, c_6, c_7, c_8\}$	$\{c_0, c_1, c_4, c_5, c_9\}$
BL$_9$	$\{c_1, c_3, c_4, c_5, c_7\}$	$\{c_0, c_2, c_6, c_8, c_9\}$
BL$_{10}$	$\{c_0, c_1, c_3, c_4, c_6\}$	$\{c_2, c_5, c_7, c_8, c_9\}$

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
c_0	5	5	3	6	3	5	3	4	6
c_1		4	8	7	4	4	2	3	3
c_2			4	3	4	8	4	3	5
c_3				5	4	4	4	5	3
c_4					5	5	3	4	2
c_5						4	6	5	5
c_6							4	3	3
c_7								7	7
c_8									6

In conclusion, CCE is an ensemble of classifiers that is designed to resolve multi-class problems and whose architecture adopts a configuration in which:

i. When *k>6*, the number of base learners is fixed to *k*. Otherwise, the number of base learners is computed according to Eq (2).

ii. Each base learner is composed by two complementary multiclass classifiers. Each one of these classifiers is trained with instances that belong to *k/2* classes. When *k* is an even number the first one is trained with instances that belong to *k/2* classes while the second one later is trained with instances that belong to *k/2 +1* classes.

iii. Given any two classes, there is, at least one classifier that has been trained using examples that belong to both classes.

Fig. 1 shows the construction scheme of CCE for a problem of four classes.

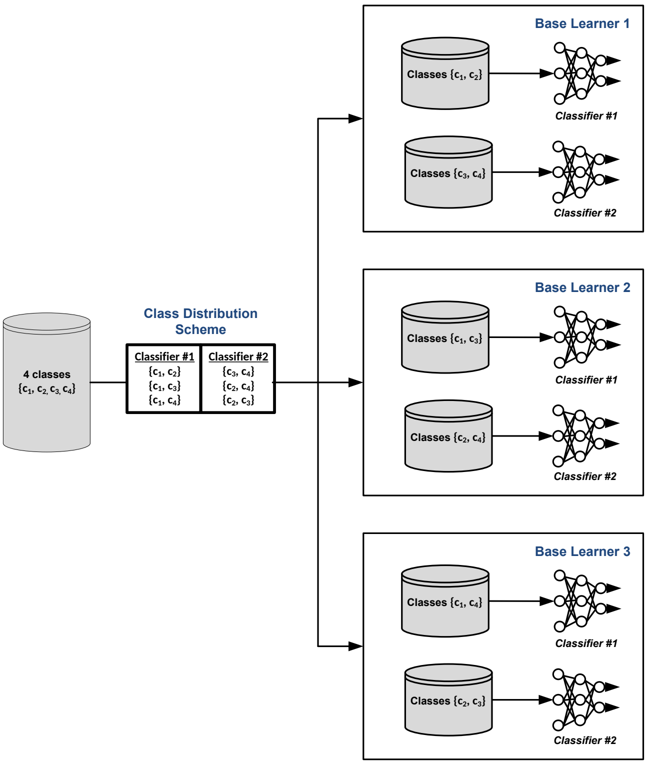

Fig. 1. CCE design for a four-class problem.

3.3. Base Learner Combination Method

Once the methodology for creating the class distribution of the base learners has been fixed, the next step is to determine the strategies for obtaining both the output of each base learner and the final decision of the ensemble.

To obtain the output given by a base learner, a parallel combination scheme is applied. As a result, each base learner produces an output ($Y_i(x) = \{y_0, y_2,.., y_{k-1}\}$), in which the y_j component is generated by the first classifier (*Classifier #1*) when the examples belonging to class c_j have been used to train this classifier. Otherwise, the y_j component is generated by the second classifier (*Classifier #2*). Using the example of BL_1 in Table2.a, the components $\{y_0, y_5, y_7, y_8, y_9\}$ are generated by *Classifier #1* and components $\{y_1, y_2, y_3, y_4, y_6\}$ are generated by *Classifier #2*.

Due to the members of CCE supply different solutions to the classification problem and all of them are equally reliable, the ensemble decision should be taken in a collaborative way using a fusion strategy. For both its simplicity and its effectiveness in large and complex data sets [47], the CCE output is calculated by averaging the outputs that are associated with each class and choosing the class that attains the maximum value. Mathematically, the process is described through Eq. (3).

$$C(\bar{x}) = \underset{i=0}{\overset{k-1}{\arg\max}} \left(\frac{\sum_{j=1}^{L} y_{ij}}{L} \right) \qquad (3)$$

where: y_{ij} is the i^{th} output of the j^{th} base learner, k is the number of categories, L is the number of base learners, $\bar{x}$ is the input instance and $C(\bar{x}) \in [0,k-1]$ the class assigned to $\bar{x}$.

Algorithm 1 compiles the details of the CCE design.

Algorithm 1: CCE

```
Input:
    Training dataset: $D_t = \{(x_i, y_i)\}_{i=1}^{N}, y_i \in \{c_1,\ldots, c_k\}$
    Number of base learners: $L^{(3)}$

Output:
    Classifier Ensemble E with classification function $C_E$
Process:
  repeat
    1. for i=1 to i=$L^{(3)}$
           $CDS_i = \{C_{\#1}^{l}, C_{\#2}^{l}\}$        # $C_{\#1}^{l} \cup C_{\#2}^{l} = \{c_1,.., c_k\}; C_{\#1}^{l} \cap C_{\#2}^{l} = \phi$
       end for
    2. Verify that $\forall$i, j | i$\neq$j $CDS_i \neq CDS_j$
    3. Verify that $\forall\{c_i, c_j\}$ i$\neq$j $\exists$ $CDS_k$ | $\{c_i,c_j\} \subset C_{\#1}^{k}$ or $\{c_i,c_j\} \subset C_{\#2}^{k}$
   until requirements 2 and 3 are satisfied;
   E=Ø;
   for i=1,    ,$L^{(3)}$
     1. Generate $D_t^1$ and $D_t^2$        # $D_t^1$=U{($\mathbf{x_i}$, $\mathbf{y_i}$) | $y_i$=$c_i$ AND $c_i \in C_{\#1}^{l}$} ; $D_t^2$={($\mathbf{x_j}$, $\mathbf{y_j}$) | $y_j$=$c_j$ AND $c_j \in C_{\#2}^{l}$}.
     2. Train a classifier, $Cl^{\#1}$, from $D_t^1$
     3. Train a classifier, $Cl^{\#2}$, from $D_t^2$
     4. Build a base learner, $BL_l$ from $Cl^{\#1}$ AND $Cl^{\#1}$              # $Y_l(\boldsymbol{x}) = \{y_0,\ldots..,y_{k-1}\}$
     5. E=E$\cup$ $BL_l$
   end for

    $C_E(\overline{x}) = \underset{j=0}{\overset{k-1}{\arg max}}\left(\sum_{l=1}^{L^3} y_{jl}\right)$ where $y_{jl}$, is the $j^{th}$ output of the $l^{th}$ base learner and $k$ is the number of
    classes.
end
```

4. Experimental Evaluation

This section describes the data sets (Sec. 4.1) and the method and the procedures (Sec. 4.2) used to evaluate CCE.

4.1. Selected Data Sets

To test the performance of CCE when solving multiclass problems, we have selected 20 different datasets. A summary of these datasets is shown in Table 3.

Table 3. Summary of the evaluated datasets.

Data set	Number of Instances	Number of Features	Number of Classes	Num. Instances max/min class	Imbalance Ratio	Source
YEAST	1472	8	10	462/5	94.2	[48]
GLASS	211	9	6	76/9	8.44	[48], [49]
SHUTTLE	58000	9	7	45586/10	4558.60	[48], [49]
WINERED	1599	11	6	681/10	68.10	[48]
VOWEL	990	12	11	90/90	1.00	[48], [50]
PENDIGITS	10992	16	10	1144/1055	1.08	[48], [49]
SEGMENTATION	2310	18	7	330/330	1.00	[48], [50]
SATIMAGE	6435	36	6	1533/626	2.45	[48], [50]
TEXTURE	5500	40	11	500/500	1.00	[48], [50]
SENSORLESS	58483	48	11	5319/5314	1.00	[48], [49]
SYNTHETIC	600	60	6	100/100	1.00	[48]
OPTDIGITS	5620	64	10	572/554	1.03	[48], [50]
AUTOMOBILE	159	75	6	48/3	3.05	[48], [50]
LIBRAS	360	90	15	24/24	1.00	[48], [50]
MFEAT-FAC	2000	216	10	200/200	1.00	[48]
SEMEION	1592	256	10	162/155	1.04	[48]
IMBALANCED SEMEION	1236	256	10	162/40	4.05	[48], [51]
USPS	7291	256	10	1194/542	2.20	[48], [49]
MNIST	60000	784	10	6742/5421	1.24	[22]
ASISTENTUR	1006	1024	9	478/22	21.73	[51]

4.2. Experimental Setup

To obtain a global vision of the behavior of CCE, we analyze the following aspects: ensemble performance (classification accuracy and training time), diversity of the base learners, and the tolerance in the presence of examples with incorrect class labels. Moreover, to test how well CCE works, its performance is compared to that obtained by a single ANN, by BCE [20] and by the usual ANN classifier ensembles: the OAA *architecture* [52], *Bagging* [30], ECOC [39], *Boosting* with *re-sampling* [53] and the OAO *architecture* [35].

4.2.1. Designing the Comparison

For all the classification models, the ANN's used are one-hidden-layer *Multilayer Perceptron* (MLP) trained with the *Back-Propagation* algorithm. For each data set, the ANN architecture and topology have been fixed with the objective of achieving a single MLP with a good generalization capacity. This thesis is supported by [54] who points that finding the adequate parameters for an optimal generalization capacity is more critical in the case of a single ANN than in the case of an ANN ensemble. So, the parameter search (number of hidden units, number of iterations and learning rate) has been performed on a single ANN using a cross validation scheme with only the training set. To establish a fair basis for comparison, the same final parameters have been used to train the ANN of the other systems. The details of the used ANN, the training parameters, and the number of base learners of each ensemble are summarized in Table 4.

It is worth mentioning that the number of the base learners of:

- OAA/BCE is equal to the number of classes.
- ECOC has been fixed according to the error-correcting codes proposed in [39]
- *Bagging* and *Boosting* have been fixed to 15. To set this value we have attempted to reach a compromise between the accuracy improvement and the computational cost of training an ANN. According to the experimental setup conducted in [55], when *Bagging* and *Boosting* are implemented with ANN, the largest error generalization reduction occurs when using approximately 10 base learners.
- *CCE* has been calculated according to Eq (2) when the number of classes is lower than *6*. Otherwise the number of base classifiers is equal to the number of classes.
- OAO is equal to $\binom{k}{2}$, where *k* is the number of classes.

Table 4. Parameters of the evaluated models**.**

	Number of base classifiers					Number of Hidden units	Number of Iterations	Learning Rate
	CCE	OAA/BCE	Bagging/Boosting	ECOC	OAO			
Yeast	10	10	15	15	45	8	400	0.25
Glass	10	6	15	31	15	7	300	0.3
Shuttle	7	7	15	63	21	10	500	0.25
WineRed	10	6	15	31	15	20	1000	0.025
Vowel	11	11	15	14	55	20	500	0.050
Pendigits	10	10	15	15	45	8	200	0.025
Segmentation	7	7	15	63	21	10	500	0.025
Satimage	10	6	15	31	15	15	600	0.050
Texture	11	11	15	14	55	20	300	0.250
Sensorless	11	11	15	14	55	20	300	0.02
Synthetic	10	6	15	31	15	15	300	0.025
Optdigits	10	10	15	15	45	30	400	0.050
Automobile	10	6	15	15	10	20	500	0.025
Libras	15	15	15	15	105	20	300	0.250
Mfeat-fac	10	10	15	15	45	20	150	0.3
Semeion	10	10	15	15	45	20	300	0.025
Imbalanced Semeion	10	10	15	15	45	20	300	0.025
Usps	10	10	15	15	45	30	100	0.025
Mnist	10	10	15	15	45	100	500	0.025
Asistentur	9	9	15	15	36	30	2000	0.025

4.2.2. Ensemble Performance Evaluation

To measure the accuracy of the different classification models, we have performed 5 replications of a *2-fold stratified cross validation*. In each replication, the dataset is randomly partitioned into two stratified and equal-sized subsets, $S_i^{(1)}$ and $S_i^{(2)}$. Furthermore, to reduce the variations that are due to the randomness of the ANN, each classification model has been trained on each dataset ten times. So, the accuracy of each model is computed following Eq. (4).

$$Ac = \frac{1}{10}\sum_{j=i}^{10}\frac{1}{5}\sum_{i=1}^{5}\frac{1}{2}\left(\frac{TP_{ij}^{(1)}}{N}+\frac{TP_{ij}^{(2)}}{N}\right) \qquad (4)$$

where $\frac{TP_{ij}^{(1)}}{N}$ is the ratio of correctly identified instances by the model when it is trained on $S_i^{(1)}$ and tested on $S_i^{(2)}$ and $\frac{TP_{ij}^{(2)}}{N}$ is the proportion of correctly identified instances by the model when it is trained on $S_i^{(2)}$ and tested on $S_i^{(1)}$ in the *j-th* execution.

Given that, for multi-class classification problems the F_1-score is accepted as an evaluation criterion that complements the accuracy measure, Appendix 1 gathers the F_1-score values computed on each data set.

To statistically compare the accuracy of CCE with that obtained by the baseline classification models, we have applied the *F-test* over the results obtained in the 5 runs of the *2-fold stratified cross validation*. According to [56], if $p_i^{(j)}$ is the difference between the error rates of the two classifiers on fold *j* of run *i*, and $s_i^2 = \left(p_i^{(1)} - \overline{p}_i\right)^2 + \left(p_i^{(2)} - \overline{p}_i\right)^2$ is the estimated variance on run *i* (where $\overline{p}_i = \left(p_i^{(1)} + p_i^{(2)}\right)/2$), then the statistic given in Eq. (3) approximately follows an *F distribution* with 10 and 5 degrees of freedom.

$$F - test = \frac{\sum_{i=1}^{5}\sum_{j=1}^{2}\left(p_i^{(j)}\right)^2}{2\sum_{i=1}^{5}s_i^2} \qquad (3)$$

Consequently, the null hypothesis of equal error rate can be rejected when the computed *F-Test* value is equal to or greater than the tabled critical one-tailed value of the *F distribution*. At the 0.05 level of significance, this value is equal to 4.735 [57].

In addition, to analyze the performance of CCE over the 20 datasets and compare it with the global performance of the baseline classification models, we have applied the Wilcoxon Signed-Ranks test [58]. This test is a non-parametric procedure which computes, for each data set, the differences in performance of two classifiers by subtracting the classifier 2 performance score from the classifier 1 performance score. Then, it ranks the differences ignoring the signs and finally compares the sum of the ranks for the positive differences *(∑R+)* and the sum of the ranks for the negative differences *(∑R-)*.

According to this test, the null hypothesis that both classifiers perform equally well is rejected at the α confidence level when min *(∑R+, ∑R-)* is less or equal than the one-tailed critical *T*-value at the pre-specified level of significance and *n* is equal to the number of signed ranks (number of data sets). For *n*=20 and α=0.05 this value is equal to 60.

Finally, to estimate the cost-effectiveness of CCE, we have measured and analyzed its training time.

4.2.3. Diversity Evaluation

To study the relationship between the diversity of the base learners and the ensemble accuracy, some well-known measures of diversity have been computed: The *Q* statistic, that computes the "coefficient of association" for two classifiers [59], the correlation coefficient (ρ) that indicates the strength and direction of a linear relationship between two classifiers [59], the kappa statistic (κ), that measures the degree of similarity between two classifiers while subtracting the probability that the similarity occurs by chance [60], and the fail/non-fail disagreement measure, which measures the percentage of instances for which the classifiers make different predictions being one of them the correct one [61]. Table 5 shows a summary of these measures and the relationship between the obtained value and the diversity between the ensemble members (the greater/lower the value is, the more diverse the base classifiers are).

Table 5. Summary of the diversity measures used. Monotonically increasing/decreasing measures are identified with an ascending/descending arrow.

Name	Symbol	Definition	↑/↓
Q statistic	*Q*	$\frac{N^{11}N^{00}-N^{01}N^{10}}{N^{11}N^{00}+N^{01}N^{10}}$	↓
correlation coefficient	ρ	$\frac{N^{11}N^{00}-N^{01}N^{10}}{\sqrt{\left(N^{11}+N^{10}\right)\left(N^{01}+N^{00}\right)\left(N^{11}+N^{01}\right)\left(N^{10}+N^{00}\right)}}$	↓
kappa degree-of-agreement statistic	κ	$\frac{\frac{\sum_{i=1}^{k}N_{ii}}{N}-\sum_{i=1}^{k}\left(\frac{N_{i*}}{N}\frac{N_{*i}}{N}\right)}{1-\sum_{i=1}^{k}\left(\frac{N_{i*}}{N}\frac{N_{*i}}{N}\right)}$	↓
fail/non-fail disagreement measure	*Dis*	$\frac{N^{01}+N^{01}}{N^{11}+N^{10}+N^{01}+N^{00}}$	↑

where:

N is the cardinality of the test set.

k is the number of classes.

N^{ab} is the number of instances in the data set, correctly (a=1) or incorrectly (a=0) classified by the classifier *i*, and correctly (b=1) or incorrectly (b=0) by the classifier *j*.

N_{ij} is the number of instances in the data set, labelled as class *i* by the first classifier and as class *j* by the second classifier.

4.2.4. Tolerance to Noise

One of the most important requirements in any classification system is its tolerance to the noise. To determine the performance of a classifier in the presence of noise, some authors [62], [63], propose to randomly change the class that is assigned to a fraction of the instances in both the training and the testing set. Nevertheless, in this work, the labeling errors are exclusively induced on the training instances.

In this experimental phase, we use the previously mentioned 20 benchmark classification tasks (Table 3), the *5x2cv* scheme and four rates of noise: 10%, 15%., 20% and 25%.

The Fig. 2 shows the noise injection process when the classifier system is built using a cross-validation process with two folds.

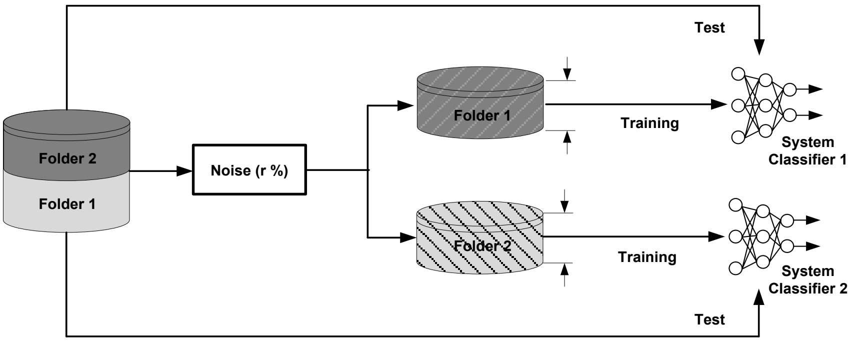

Fig. 2. Noise injection process in a two-folds cross validation scheme.

5. Experimental Results

Once the implemented classification models and the experimental methodology have been described, this section shows the obtained experimental results. First, in Section 5.1, we test the accuracy and the training time of CCE. In addition, the obtained values are compared with other baseline classification systems. Section 5.2, shows the diversity of the base learners and an analysis of the relationship between base learner diversity and ensemble accuracy. Finally, section 5.3, shows the CCE performance in the presence of noise.

5.1. CCE performance

As was noted above, to evaluate the accuracy of the different classification models on each dataset, a *10x5x2 cv* scheme is employed. Moreover, to determine if CCE is statistically better, equal or worse than every other of the baseline methods, the *5x2 cv F-test* with a significance level of 0.05 is computed.

Table 6 shows a summary of the accuracy for the different classification models that have been evaluated and the conclusions of the statistical comparison between CCE and the other implemented classifier models using F-test. Additionally, the last row compiles the number of times that the standard model (of the corresponding column) performs significantly better (win), equal (tie) or worse (loss) than CCE and the conclusions of the statistical comparison using Wilcoxon signed-ranks test.

Table 6. Summary of the different values of accuracy. The ✓/~/× symbols indicate that the standard classifier is significantly better/equal/worse than CCE. The best values are shown in bold.

Data Set	CCE	Standard Classifiers						
		BCE	ANN	OAA	Bagging	ECOC	Boosting	OAO
Yeast	56.85	56.89 ~	56.27 ~	57.13 ~	59.21 ~	48.32 ×	**59.63** ~	57.91 ~
Glass	**68.53**	65.96 ~	66.12 ~	67.87 ~	67.83 ~	67.57 ~	67.99 ~	66.12 ~
Shuttle	99.64	99.53 ~	99.53 ~	99.51 ~	**99.66** ~	99.54 ~	99.18 ~	99.53 ~
WineRed	59.42	59.26 ~	59.27 ~	59.12 ~	59.58 ~	55.83 ×	**59.74 ~**	59.15 ~
Vowel	88.58	73.76 ×	79.13 ×	86.29 ~	83.82 ×	87.05 ~	**93.21** ✓	90.14 ~
Pendigits	97.73	93.65 ×	92.27×	96.44×	93.83×	98.03 ~	**99.11** ✓	98.09 ~
Segmentation	**94.14**	93.08 ×	93.73 ~	92.89 ×	93.67 ~	92.88 ×	93.43 ~	94.10 ~
Satimage	**87.77**	86.14 ×	87.16 ~	86.21 ×	87.44 ~	85.35 ×	84.98 ×	87.52 ~
Texture	99.54	98.63 ×	99.52 ~	99.49 ~	**99.65** ~	99.56 ~	99.62 ~	99.27 ~
Sensorless	94.33	84.82 ×	92.66 ×	92.69 ×	93.71 ×	91.08 ×	**98.77** ✓	92.66 ~
Synthetic	95.90	**97.02** ~	96.26 ~	94.58 ×	96.85 ~	94.77 ~	96.33 ~	95.60 ~
Optdigits	97.72	95.37 ~	96.13 ×	97.08 ×	97.68 ~	97.31 ×	**97.99** ~	97.41 ×
Automobile	67.11	**69.29** ~	64.28 ×	64.42 ×	65.13 ~	63.33 ×	67.01 ~	67.55 ~
Libras	77.34	71.39 ~	72.99 ×	73.15 ~	77.08 ~	76.33 ~	**81.27 ~**	78.16 ~
Mfeat-fac	96.87	96.96 ~	96.42 ×	96.66 ~	97.00 ~	96.77 ~	**97.12 ~**	96.50 ~
Semeion	**90.67**	90.12 ~	86.10 ×	87.09 ×	90.56 ~	88.06 ×	90.45 ~	89.22 ×
Imbalanced Semeion	89.51	**90.70** ~	84.71 ×	85.70 ×	89.12 ~	87.07 ×	90.09 ~	87.89 ×
Usps	96.22	95.26 ×	95.19 ×	95.98 ~	95.91 ×	95.86 ~	**96.97** ✓	96.20 ~
Mnist	**97.09**	96.91 ×	95.26 ×	96.56 ×	96.38 ×	96.95 ×	97.03 ×	96.38 ×
Asistentur	94.43	**94.70** ~	93.27 ~	92.87 ×	94.36 ~	94.31 ~	94.69 ~	91.32 ×
win/tie/loss		0/12/8 ×	0/9/11 ×	0/9/11 ×	0/15/5 ×	0/10/10 ×	4/14/2~	0/15/5 ×

As it can be appreciated from the results in Table 6:

- The best accuracy values are achieved by *Boosting* (in 9 of the 20 data sets: Yeast, WineRed, Vowel, Pendigits, Sensorless, Optdigits, Libras, Mfeat-fac, and Ups), CCE (in 5 of the 20 data sets: Glass, Segmentation, Satimage, Semeion and Mnist), BCE (in 4 of the 20 data sets: Synthetic, Automobile, Imbalanced-Semeion and Asistentur), and *Bagging* (in two data sets: Shuttle and Texture).
- CCE is significantly more accurate than OAA and ANN in 11 data sets, ECOC in 10 data sets, BCE in 8 data sets, *Bagging* and OAO in 5 data sets, and *Boosting* in two. Only *Boosting* outperforms CCE with statistical significance in four domains (Vowel, *Pendigits*, *sensorless and usps*).
- For the Mnist data set, the mean accuracy rate achieved by CCE is statistically better than those obtained by the rest of implemented models.
- The comparison over all data sets reveals that CCE is statistically better than ANN, OAA, ECOC, BCE, *Bagging*, and OAO. Additionally, CCE is statistically equivalent to *Boosting*.

To evaluate the quality of CCE and to verify whether it outperforms its base learners, Fig. 3 shows the relation between the accuracy of the ensemble and the mean accuracy of the base learners. Each display shows the values obtained using a *10x5x2 cv* scheme that delivers 100 points in each plot. Additionally, similar graphs for BCE, *Bagging* and *Boosting* are shown. (Note that this representation only is possible with systems in which the base learners provide a complete answer to the classification problem. Therefore, for the single ANN and the ensembles based on binary decomposition –ECOC, OAA and OAO– this graphical representation is unfeasible).

In each graph, the points that lie above the dashed diagonal represent a better accuracy of each ensemble with respect to the mean accuracy of its base learners. It can be seen, that sometimes, both *Bagging* and *Boosting* are less accurate than some of their base classifiers. By the contrary, CCE and BCE always outperform their base learners. On the other hand, when the values for CCE, BCE, *Bagging* and *Boosting* are compared, it is possible to appreciate that the four ensembles have a similar global accuracy (y axis values) but, as expected, the base learners of CCE are less accurate than the base learners of the other ensembles (x axis values). Let us remember that CCE attends to build base learners that are relatively accurate but highly diverse.

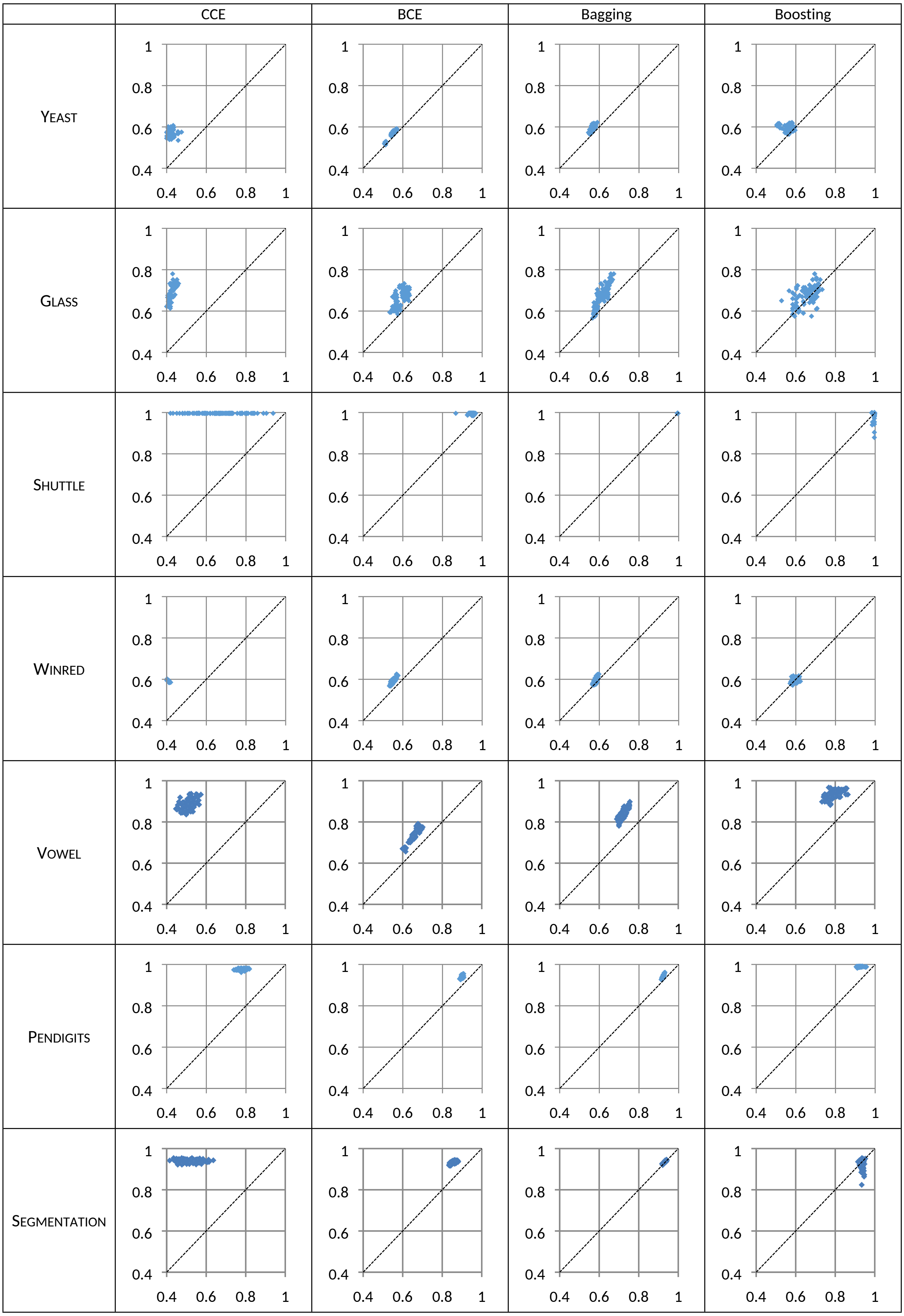

Fig. 3.a. Relationship between the accuracy of CCE/BCE/Bagging/Boosting (y axis) and the mean accuracy of the base learners (x axis). Datasets: YEAST, GLASS, SHUTTLE, WINERED, VOWEL, PENDIGITS and SEGMENTATION.

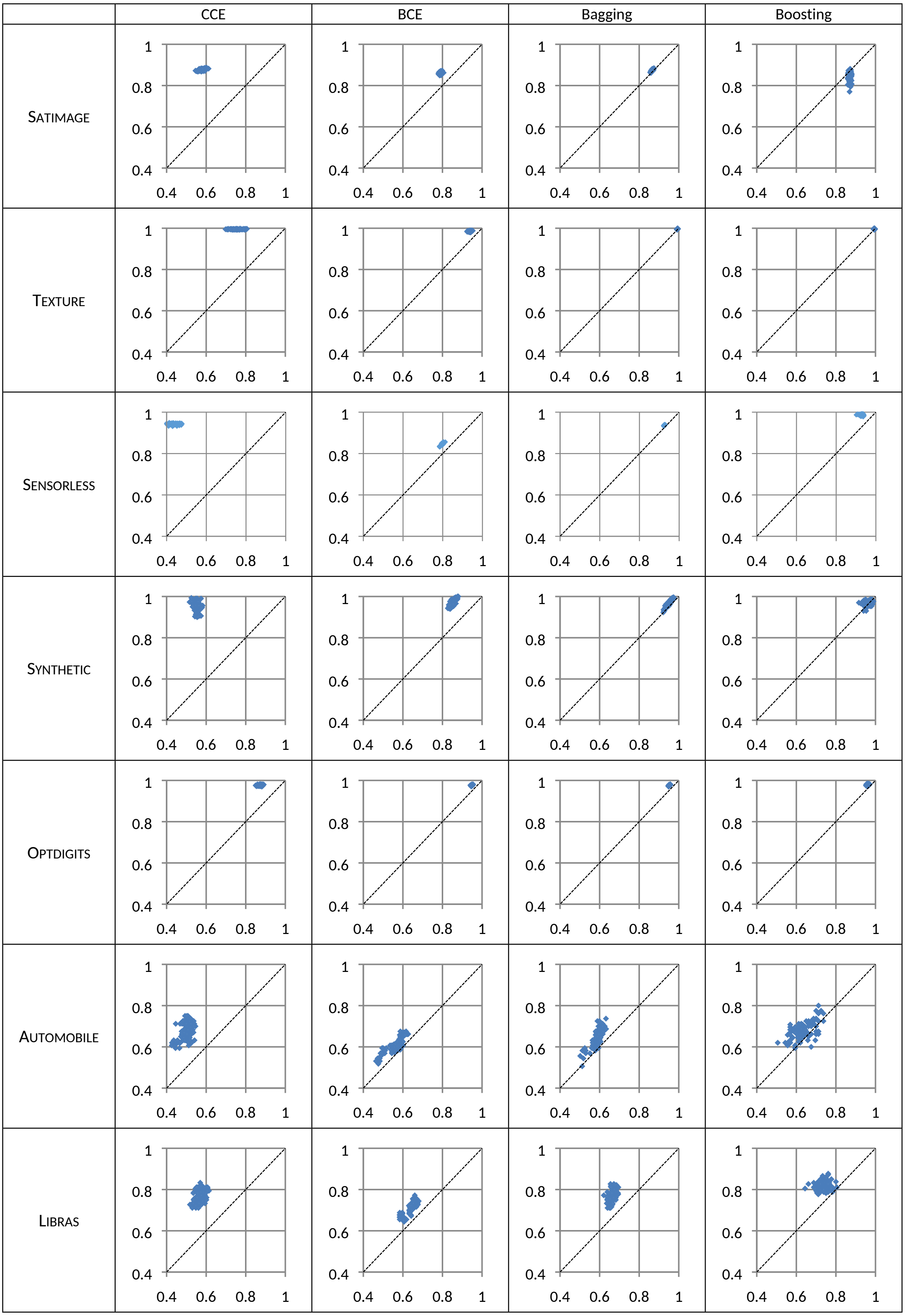

Fig 3.b. Relationship between the accuracy of CCE/BCE/Bagging/Boosting (y axis) and the mean accuracy of the base learners (x axis). Datasets: SATIMAGE, TEXTURE, SENSORLESS, SYNTHETIC, OPTDIGITS, AUTOMOBILE and LIBRAS.

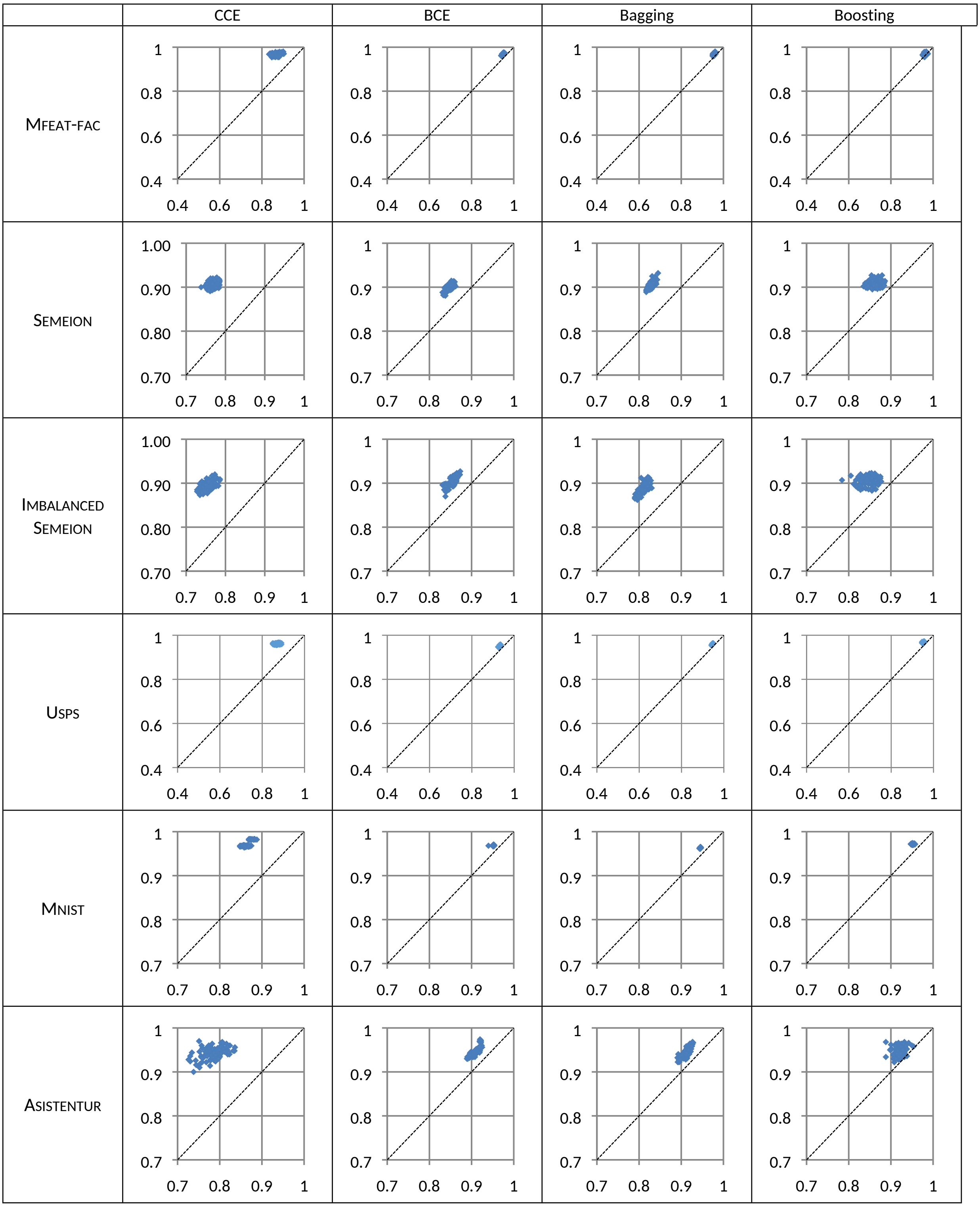

Fig 3.c. Relationship between the accuracy of CCE/BCE/Bagging/Boosting (y axis) and the mean accuracy of the base learners (x axis). Datasets: MFEAT-FAC, SEMEION, IMBALANCED SEMEION, USPS, MNIST, and ASISTENTUR.

Although the accuracy is the main criterion used to measure the quality of the classification systems, another criterion that must receive some attention, specially in the case of large data sets, is the computational cost of training each system. According to [64], when two or more systems deliver comparable accuracy rates, excessive training times may be undesirable.

Fig. 4. shows the training time of the implemented classification models when they are measured on a computer cluster.

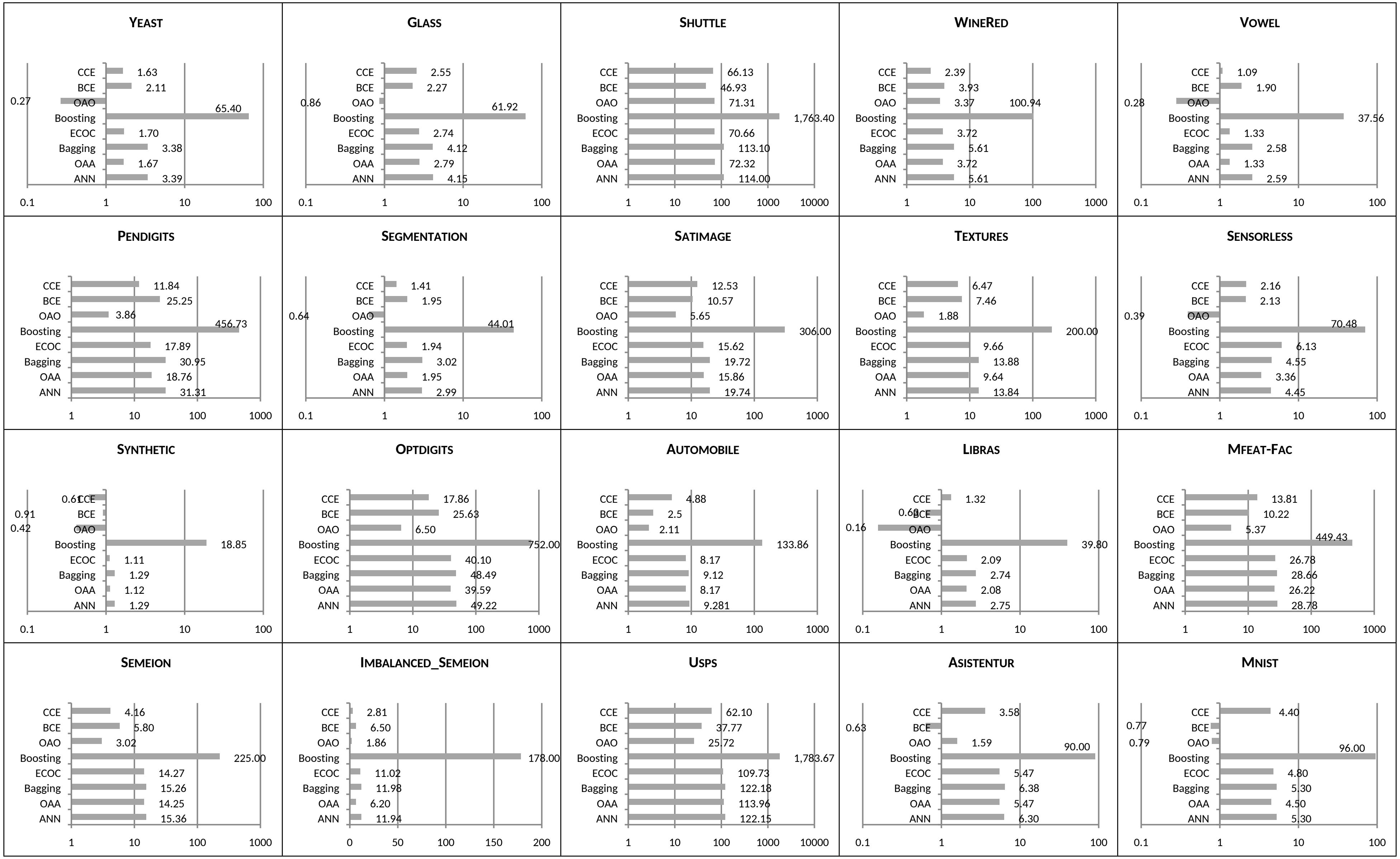

Fig. 4. Ensemble training time measured on a computer cluster based on Intel(R) Xeon(R) CPUs E5-2640 @ 2.50GHz. Note than for GLASS and AUTOMOBILE, time is measured in deciseconds; for ASISTENTUR and SENSORLESS time is measured in minutes; for MNIST time is measured in hours. In all the other domains, times are given in seconds

It is worth mentioning that the independent nature of the base learners that compose CCE, BCE, *Bagging*, ECOC, OAA and OAO allows for a very natural parallelization of the construction process of the ensembles. Using clusters or CPUs with multiple cores it is possible to reduce the training time of the whole ensemble to the training time of a base learner. This is reflected in the graphs in Fig. 4. A special case is *Boosting*, which has the specific inconvenience of requiring a strictly sequential training of its base learners, as each one uses a resampled data set based on the training of the previous base learner.

The evaluation of the training time shows that the system with the lowest training time is OAO followed by BCE and CCE. The reduced training time of these systems is due to the structure of their base learners. While OAO and CCE decrease the training time due to the reduced number of examples that are used for training each classifier, BCE decreases its training time due to the reduced number of attributes that are used to describe the instances.

If we consider simultaneously the accuracy and the training time, we can conclude that CCE outperforms the other analyzed systems: CCE is more accurate than those systems that require less training time (OAO and BCE), and it requires a much shorter training time than the system with a comparable accuracy rate (*Boosting*).

5.2. Study of the Diversity

Once CCE has been analyzed according to its performance, in this section we study the relation between the diversity of base learners and the accuracy of the ensemble. At first, the global ensemble diversity is calculated by using the four measures mentioned in the section 4.2.3: the *Q statistic*, the *correlation coefficient* (ρ), the *kappa statistic* (κ) and the *fail/non-fail disagreement* measure. Then, we measure the correlation between the diversity and the gain (ensemble accuracy minus mean accuracy of its base learners) of the ensemble. The relationship between the *Q statistic* and the gain of the ensemble is illustrated in Fig 5.

Given that *Q* is a monotonically decreasing diversity measure [65], values plotted in Fig 5 indicate that *Boosting* and CCE are the most diverse systems (the clouds of points are distributed close to the best possible theoretical value). Moreover, CCE is the system that presents the biggest gain (the values in the y axis are the highest). Finally, *Bagging* and BCE are the system that present the lowest diversity and gain values. These findings suggest that the accuracy of both *Bagging* and BCE appear to depend on the accuracy of the base learners but not on their diversity. By the contrary the accuracy of both *Boosting* and CCE appears to depend more on the diversity than on the accuracy of the base learners.

To check whether the gain of the different ensembles is a consequence of the diversity of their base learners, we compute the correlation between both measures using the *Spearman's* correlation coefficient (*RCC*) [61]. This coefficient is defined by Eq.5.

$$RCC = 1 - 6\sum_{i=1}^{N} \frac{(Rank(x_i) - Rank(y_i))^2}{N(N^2 - 1)} \qquad (5)$$

Where *N* is the number of evaluated ensembles (10x5x2=100), *X* and *Y* are the diversity and the gain of the ensembles respectively, and *Rank(x$_i$)* and *Rank(y$_i$)* are the i^{th} value of *X* and *Y* when they are ranked in descending order. The different values of *RCC* are compiled in Table 7.

To determine the significance of *RCC* and to analyze whether the measured variables (diversity and gain) are correlated, the *RCC* value must be evaluated with the *Table of Critical Values for Spearman's Rho* [57]. According to this table, at the 0.05 level of significance, the hypothesis that the diversity and the gain of the ensemble are unrelated is rejected when $|RCC| \geq 0.197$.

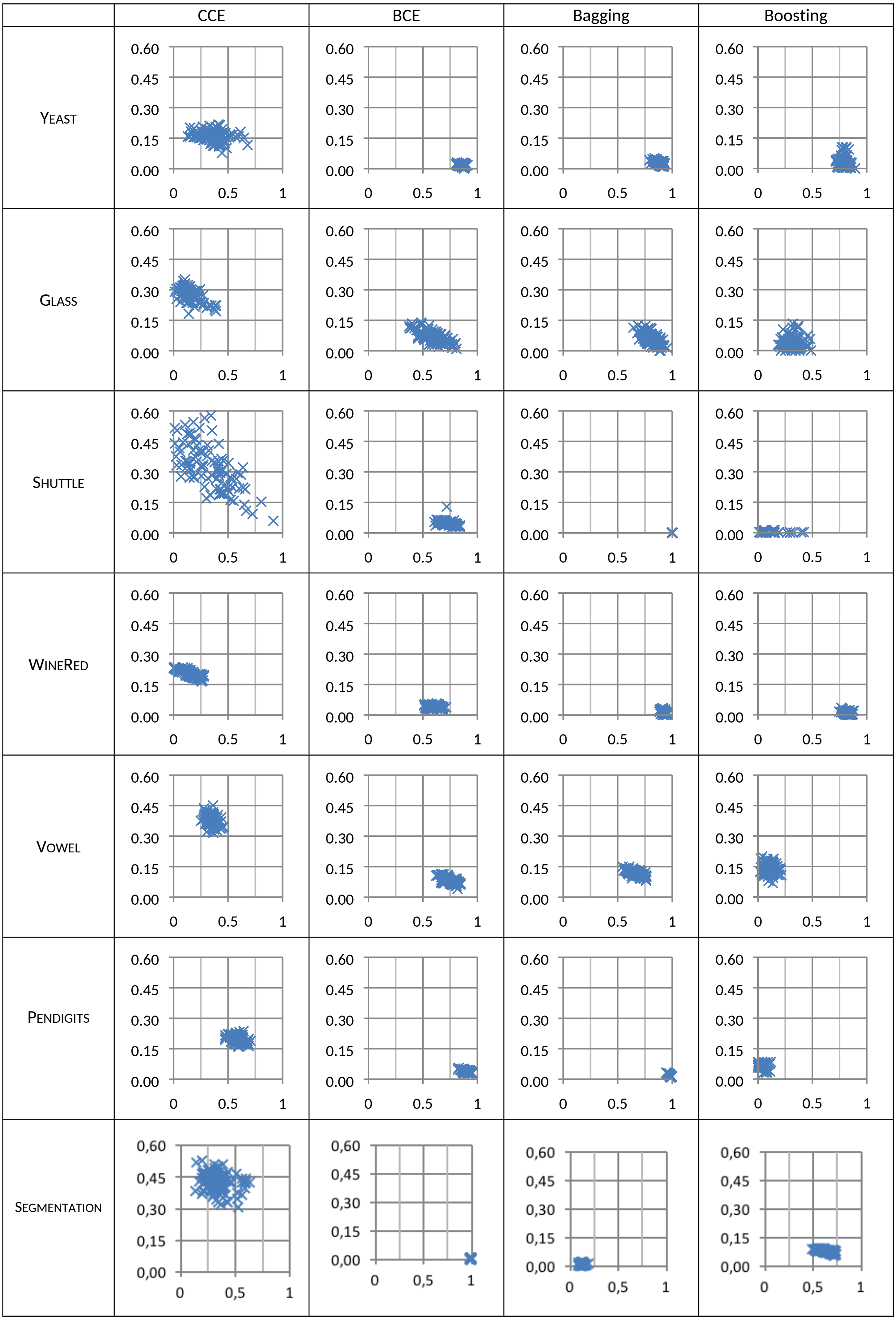

Fig. 5.a. Relationship between the diversity (x axis) and the gain (y axis) of the ensemble. Diversity is quantified using the Q statistic. Datasets: YEAST, GLASS, SHUTTLE, WINERED, VOWEL, PENDIGITS and SEGMENTATION

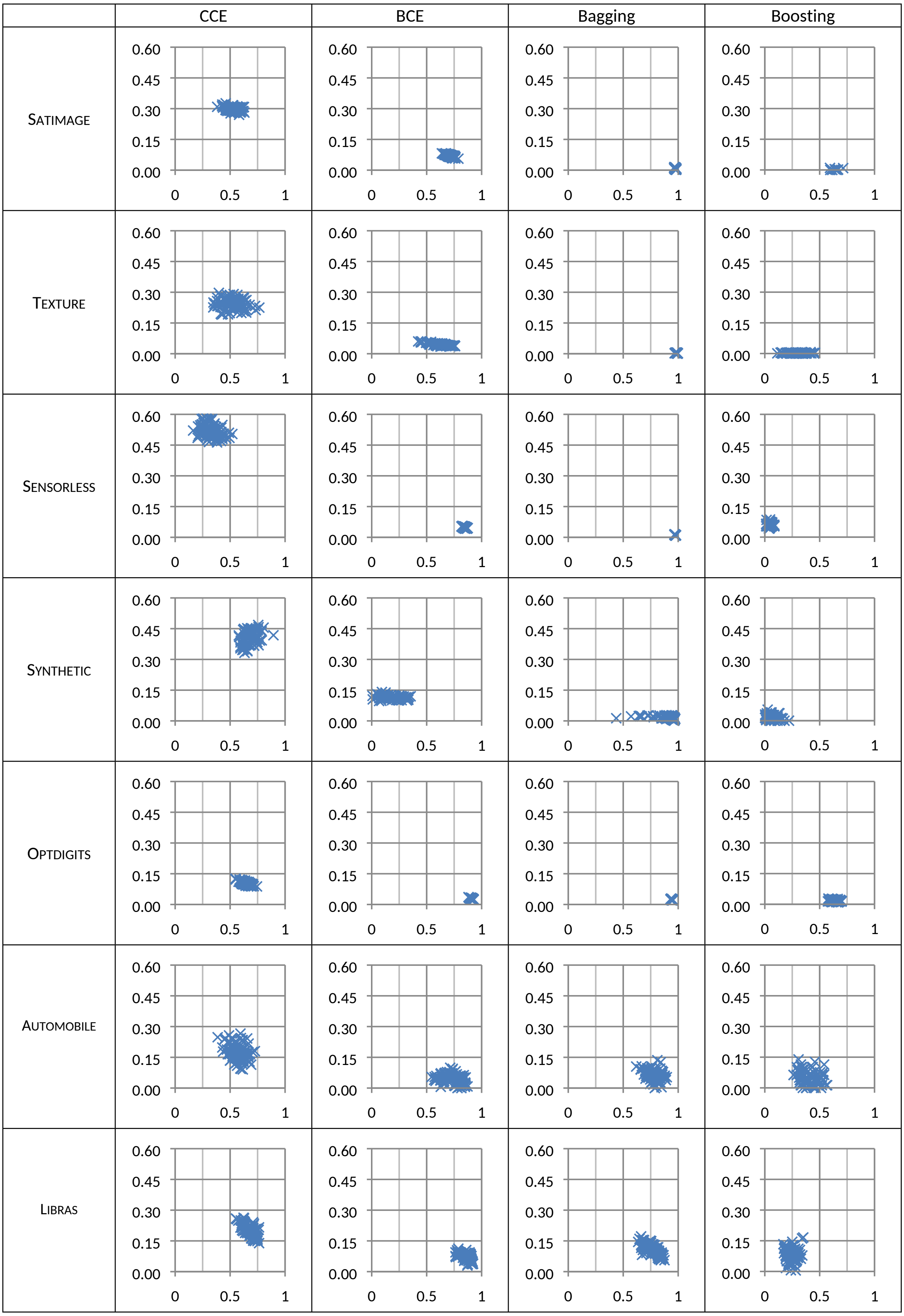

Fig. 5.b. Relationship between the diversity (x axis) and the gain (y axis) of the ensemble. Diversity is quantified using the Q statistic. Datasets: Satimage, Texture, Sensorless, Synthetic, Optdigits, Automobile and Libras.

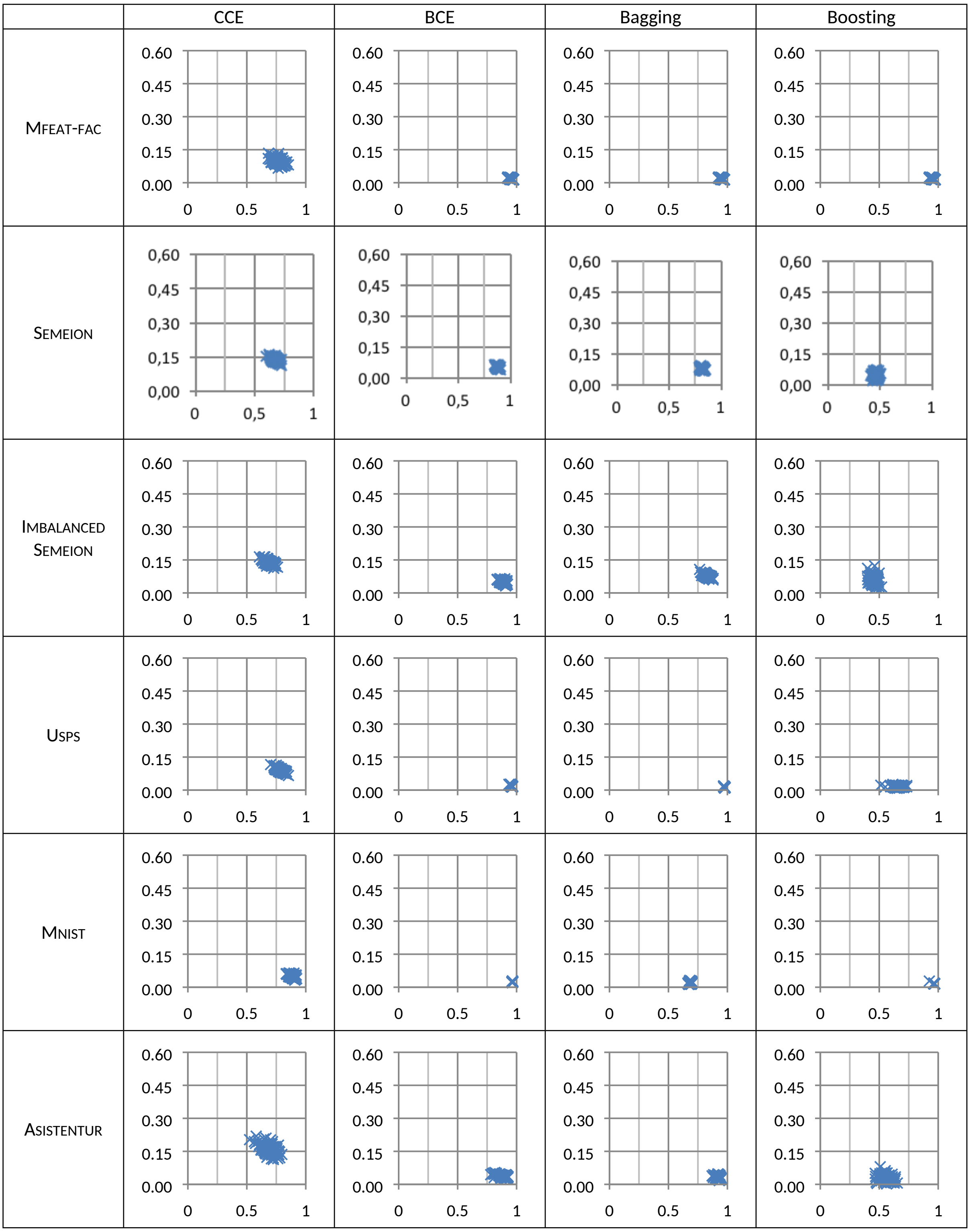

Fig. 5.c. Relationship between the diversity (x axis) and the gain (y axis) of the ensemble. Diversity is quantified using the Q statistic. Datasets: MFEAT-FAC, SEMEION, IMBALANCED-SEMEION, USPS, MNIST and ASISTENTUR.

Table 7. Spearman´s correlation coefficient for the diversity and the gain of the ensemble. The shaded values represent the values that support the hypothesis that the diversity and the gain of the ensemble are unrelated. Monotonically increasing/decreasing measures are identified with an ascending/descending arrow.

	Q (↓)				ρ (↓)				κ(↓)				dis (↑)			
	CCE	*BCE*	*Bagging*	*Boosting*	CCE	*BCE*	*Bagging*	*Boosting*	CCE	*BCE*	*Bagging*	*Boosting*	CCE	*BCE*	*Bagging*	*Boosting*
Yeast	-0.246	0.465	**0.66**	0.185	-0.128	-0.458	-0.476	**0.480**	-0.651	**-0.668**	-0.616	0.618	-0.195	**-0.201**	-0.117	0.122
Glass	-0.591	**0.718**	0.605	-0.554	-0.679	**-0.74**	-0.676	0.683	-0.616	-0.631	-0.637	**0.644**	-0.362	-0.369	-0.348	**-0.581**
Shuttle	-0.677	**0.978**	0.871	-0.362	-0.641	-0.58	-0.976	**0.979**	-0.618	-0.698	-0.729	**0.87**	-0.214	-0.102	**0.397**	-0.36
WineRed	**-0.769**	-0.159	-0.551	-0.198	-0.159	**-0.288**	-0.202	0.213	**-0.551**	-0.548	-0.499	0.5	**-0.198**	-0.192	-0.182	-0.187
Vowel	-0.34	0.679	**-0.695**	0.04	-0.359	**0.71**	-0.679	0.066	**-0.635**	0.454	-0.556	-0.142	**0.633**	-0.453	0.555	0.139
Pendigits	-0.414	0.77	**0.839**	0.005	-0.506	-0.553	-0.797	**0.804**	**-0.819**	-0.808	-0.744	0.751	**-0.13**	-0.078	0.017	-0.014
Segmentation	-0.233	**-0.386**	-0.209	-0.269	-0.164	**-0.347**	-0.23	-0.289	**-0.651**	-0.312	-0.22	-0.04	0.646	0.312	0.239	-0.428
Satimage	-0.244	-0.219	**-0.537**	0.461	-0.182	-0.243	**-0.554**	0.389	-0.512	-0.41	-0.463	**0.681**	**0.464**	0.413	0.467	**-0.662**
Textures	-0.107	-0.271	**-0.634**	**-0.02**	0.085	-0.268	**-0.559**	-0.026	**-0.766**	-0.209	-0.588	0.039	**0.772**	0.21	0.644	0.072
Sensorless	-0.331	0.71	**0.728**	-0.182	-0.629	-0.589	-0.653	**0.668**	-0.689	-0.684	-0.697	**0.728**	-0.107	-0.045	0.19	-0.184
Synthetic	0.287	-0.403	**-0.564**	-0.008	0.347	-0.506	**-0.564**	0.063	**-0.298**	-0.182	-0.091	0.136	**0.218**	0.195	0.105	0.139
Optidigits	**-0.736**	-0.698	-0.449	-0.118	-0.486	**-0.508**	-0.079	-0.12	**-0.965**	-0.781	-0.625	0.053	**0.968**	0.809	0.663	-0.041
Autos	-0.246	-0.310	**-0.378**	-0.252	-0.251	-0.308	**-0.407**	-0.24	-0.394	**-0.447**	-0.26	-0.265	0.387	**0.456**	0.258	-0.024
Libras	-0.588	-0.573	**-0.745**	-0.145	-0.608	-0.557	**-0.744**	-0.123	-0.612	-0.419	**-0.75**	0.157	0.617	0.422	**0.752**	-0.143
Mfeat-fac	-0.559	-0.117	-0.877	**0.880**	**-0.703**	-0.618	-0.676	0.689	**-0.701**	-0.643	-0.509	0.521	-0.175	**-0.204**	0.057	-0.05
Semeion	**-0.647**	-0.479	-0.505	-0.023	**-0.524**	-0.384	-0.471	-0.009	**-0.808**	-0.473	-0.411	0.058	**0.811**	0.525	0.475	-0.054
Imbalanced Semeion	**-0.689**	-0.536	-0.668	-0.255	-0.615	-0.514	**-0.626**	-0.189	**-0.709**	-0.433	-0.495	-0.108	**0.711**	0.478	0.584	0.11
Usps	**-0.854**	0.798	-0.583	-0.189	**-0.872**	-0.807	-0.73	0.731	-0.583	**-0.658**	-0.264	0.300	-0.189	**-0.24**	-0.147	0.159
Mnist	-0.659	**-0.697**	-0.374	0.229	-0.547	**-0.741**	-0.329	0.378	**-0.825**	**-0.825**	-0.304	0.165	0.829	**0.834**	0.522	0.023
Asistentur	**-0.523**	-0.324	-0.455	-0.178	**-0.315**	-0.274	-0.288	-0.235	**-0.915**	-0.36	-0.361	0.235	**0.916**	0.506	0.507	-0.228

The values on Table 7 show that, with some exceptions, for CCE, BCE and *Bagging* the relationship between the diversity and the gain of the ensemble is statistically significant. Therefore, it is possible to conclude than the increase on the diversity between the base learners is associated with an increase on the gain of the ensemble. On the contrary, although the experimental results show that the base learners of *Boosting* are the most diverse, there is no statistical relationship between the diversity and the gain of the ensemble. This observation suggests that i) the *Boosting* accuracy depends mainly of the accuracy of the base learners and ii) on the analyzed domains, the diversity of the base learners of *Boosting* can be qualified as a "bad" diversity [66].

On the other hand, the values on Table 7 show that when diversity is computed using the disagreement measure (*dis*) or the kappa statistic (κ), CCE presents the highest values of RCC. That is, according to these measures, CCE is the system in which the degree of relationship between the diversity of the base learners and the gain of ensemble is the strongest. On the contrary, when diversity is computed using the *Q statistic* (*Q*) the system that presents a high degree of correlation between diversity and gain is *Bagging*. Finally, when diversity is computed using the correlation coefficient (ρ), the systems with the highest values of RCC are BCE and *Bagging*. These results reveal that is difficult to determine the relationship between the diversity and the gain of the ensembles because of their strong dependency on the measure used to compute the diversity.

5.3. Noise Resilience

To analyze the performance of CCE in the presence of noise, we add noise by randomly changing the class that is assigned to a fraction of the training examples. According to the scheme shown in Fig 2., once the classification models have been built, they are tested using a noiseless data set.

Tables 8-11 show the accuracy on the different domains (see Table 3) when the evaluated classification models are built with a noise rate of 10%, 15%, 20% and 25% respectively. As in the previous section, the results of the statistical comparison (applying the *F-test* and Wilcoxon Signed-Ranks test with level of significance of 0.05) between CCE and the baseline classification systems are represented.

Table 8. Summary of the different values of accuracy when the training set has a noise rate of 10%. The ~/× symbols indicate that the standard classifier is significantly equal/worse than CCE. The best values are shown in bold.

Data Set	CCE	Standard Classifiers						
		BCE	ANN	OAA	Bagging	ECOC	Boosting	OAO
Yeast	57.07	55.94 ~	56.03 ~	56.94 ~	59.05 ~	46.84 ×	**59.30** ~	57.54 ~
Glass	66.35	44.19 ×	65.16 ~	66.31 ~	**67.11** ~	65.36 ~	65.69 ~	66.04 ~
Shuttle	99.56	84.35 ×	99.41 ~	99.29 ~	**99.64** ~	99.31 ~	99.61 ~	99.40 ~
WineRed	59.37	58.71 ~	58.99 ~	58.7 ×	59.36 ~	56.15 ×	**59.65** ✓	59.28 ~
Vowel	**84.79**	66.28 ×	75.59 ×	82.45 ×	81.53 ×	81.38 ×	79.67 ×	83.96 ~
Pendigits	97.39	92.56 ×	92.16 ×	97.49 ~	93.84 ×	**97.87** ~	97.86 ✓	97.14 ~
Segmentation	**93.17**	92.56 ~	93.14 ~	92.42 ~	93.19 ~	91.90 ×	91.47 ~	93.32 ~
Satimage	86.84	85.27 ×	86.52 ~	85.55 ~	87.15 ~	84.43 ×	**87.43** ~	86.46 ~
Texture	99.33	97.81 ×	98.90 ×	99.08 ~	**99.46** ~	99.23 ~	97.87 ×	98.56 ~
Sensorless	93.29	83.66 ×	92.19 ×	92.08 ×	93.25 ~	89.57 ×	**94.11** ✓	92.80 ~
Synthetic	94.64	**96.67** ~	95.12 ~	92.88 ~	95.90 ~	95.78 ~	78.22 ×	94.05 ~
Optdigits	97.48	97.36 ~	94.49 ×	94.69 ×	**97.52 ~**	96.23 ×	94.30 ×	96.54 ×
Automobile	**63.76**	56.34 ~	62.67 ~	62.64 ~	62.80 ~	60.63 ×	59.86 ×	62.56 ~
Libras	**74.91**	60.79 ×	70.29 ~	69.28 ~	74.89 ~	70.77 ~	74.18 ~	73.53 ~
Mfeat-fac	95.96	96.40 ~	93.01 ×	94.91 ×	96.55 ~	94.75 ×	92.51 ×	94.72 ×
Semeion	88.70	87.94 ~	81.67 ×	83.46 ×	**88.78** ~	84.74 ×	86.11 ×	86.81 ×
Imbalanced Semeion	88.01	**88.56** ~	80.59 ×	82.42 ×	87.65 ~	84.64 ×	86.17 ×	86.40 ×
Usps	**95.97**	95.15 ×	94.56 ×	95.37 ×	95.78 ×	95.11 ×	93.90 ×	95.37 ×
Mnist	**96.41**	96.36 ~	93.97 ×	93.07 ×	94.41 ×	95.34 ×	93.84 ×	95.34 ×
Asistentur	91.17	89.84 ~	87.77 ×	88.98 ×	**92.49 ~**	88.23 ×	88.24 ×	87.91 ×
win/tie/loss		0/11/9 ×	0/9/11 ×	0/10/10 ×	0/16/4 ~	0/6/14 ×	3/6/11 ×	0/13/7 ×

Table 9. Summary of the different values of accuracy when the training set has a noise rate of 15%. The ~/× symbols indicate that the standard classifier is significantly equal/worse than CCE. The best values are shown in bold.

Data Set	CCE	Standard Classifiers						
		BCE	ANN	OAA	Bagging	ECOC	Boosting	OAO
Yeast	57.15	56.40 ~	57.37 ~	57.98 ~	59.17 ~	49.08 ×	**59.69** ✓	58.51 ~
Glass	66.18	31.28 ×	64.27 ~	65.47 ~	**67.01** ~	65.39 ~	63.22 ×	65.23 ~
Shuttle	99.58	82.82 ×	99.25 ~	99.31 ~	**99.64** ~	99.25 ~	99.60 ~	99.29 ~
WineRed	58.77	57.97 ~	58.73 ~	58.28 ~	59.18 ~	55.28 ×	**59.59** ~	58.77 ~
Vowel	**81.94**	61.49 ×	72.80 ×	79.44 ~	80.31 ~	78.21 ×	76.36 ×	81.74 ~
Pendigits	97.29	92.40 ×	92.44 ×	97.65 ✓	94.36 ×	**97.80** ✓	97.30 ~	96.83 ×
Segmentation	92.65	92.07 ~	93.11 ~	92.57 ~	**93.16** ~	91.68 ×	92.55 ~	91.84 ~
Satimage	86.38	85.19 ×	86.26 ~	85.26 ×	86.88 ~	84.00 ×	**87.26** ~	85.95 ~
Texture	99.25	97.91 ×	98.69 ~	99.05 ~	**99.38** ~	99.00 ~	98.52 ✓	98.31 ×
Sensorless	92.51	83.02 ×	91.92 ~	91.55 ~	92.83 ~	88.65 ×	**93.55** ✓	91.73 ×
Synthetic	95.10	95.49 ~	95.30 ~	93.55 ×	**96.04** ~	93.76 ~	86.02 ×	94.43 ~
Optdigits	**97.29**	97.20 ~	93.41 ×	93.31 ×	97.24 ~	95.06×	95.30 ×	96.07 ×
Automobile	**61.18**	59.37 ~	58.51 ×	59.30 ~	60.08 ~	57.10 ×	56.65 ×	61.17 ~
Libras	72.68	56.88 ×	67.20 ×	66.15 ×	**73.28 ~**	66.24 ~	72.27 ~	71.19 ~
Mfeat-fac	95.04	95.47 ~	91.04 ×	92.43 ×	**95.99** ✓	92.84 ×	89.25 ×	93.07 ×
Semeion	89.46	89.17 ~	83.39 ×	84.50 ×	89.99 ✓	81.00 ×	87.51 ~	**87.45** ×
Imbalanced Semeion	86.56	86.92 ~	77.97 ×	79.26 ×	**87.24** ✓	75.95 ×	83.53 ×	84.19 ×
Usps	**95.68**	94.96 ×	94.05 ×	94.74 ×	95.48 ×	94.29 ×	94.69 ×	94.92 ×
Mnist	**96.17**	96.10 ~	91.83 ×	93.92 ×	95.78 ×	94.08 ×	93.65 ×	94.73 ×
Asistentur	88.74	89.28 ~	84.38 ×	86.12 ×	**90.62** ✓	85.01 ×	84.99 ×	87.12 ×
win/tie/loss		0/11/9 ×	0/9/11 ×	1/9/10 ×	4/13/3 ~	1/5/14 ×	3/7/10 ×	0/10/10 ×

Table 10. Summary of the different values of accuracy when the training set has a noise rate of 20%. The ✓/~/× symbols indicate that the standard classifier is significantly better/equal/worse than CCE. The best values are shown in bold.

Data Set	CCE	Standard Classifiers						
		BCE	ANN	OAA	Bagging	ECOC	Boosting	OAO
Yeast	57.30	54.19 ~	55.10 ~	56.18 ~	58.77 ~	44.99 ×	59.26 ~	56.81 ~
Glass	66.19	26.36 ×	64.86 ~	**66.78** ~	66.40 ~	64.57 ~	63.02 x	65.90 ~
Shuttle	99.43	84.87 ×	99.26 ~	99.28 ~	**99.62** ~	99.25 ~	99.60 ~	99.46 ~
WineRed	58.82	58.05 ~	58.57 ~	58.64 ~	58.84 ~	55.64 ×	58.73 ~	**58.98** ~
Vowel	**78.33**	59.72 ×	71.84 ×	76.82 ~	76.85 ×	75.01 ×	76.50 ~	77.03 ~
Pendigits	97.23	92.97 ×	92.28 ×	97.58 ~	94.34 ×	**97.59 ~**	97.09 ~	96.55 ×
Segmentation	92.46	92.71 ~	92.85 ~	92.39 ~	**92.90** ~	91.73 ~	92.73 ~	91.89 ~
Satimage	86.19	84.80 ×	85.93 ~	84.99 ×	86.74 ~	83.66 ×	**87.13 ~**	85.53 ~
Texture	99.13	97.52 ×	98.37 ~	98.84 ~	**99.30** ~	98.78 ~	98.83 ×	98.02 ~
Sensorless	92.19	82.62 ×	91.43 ~	90.48 ~	92.48 ✓	87.15 ×	**93.14** ✓	90.61 ~
Synthetic	93.49	94.04 ~	93.57 ~	89.91 ~	**95.75 ~**	94.53 ~	72.10 ×	90.95 ~
Optdigits	**96.93**	96.83 ~	92.17 ×	90.92 ×	96.87 ~	93.02 ×	95.58 ×	95.69 ×
Automobile	**58.90**	57.62 ~	55.74 ~	55.61 ~	56.84 ~	53.93 ×	55.13 ×	58.09 ~
Libras	**71.94**	50.85 ×	65.22 ×	64.12 ×	71.79 ~	63.84 ×	69.59 ~	69.79 ~
Mfeat-fac	93.73	94.82 ~	89.07 ×	90.11 ×	**95.39** ✓	90.82 ×	89.58 ×	92.16 ×
Semeion	85.13	84.00 ~	75.19 ×	76.69 ×	**85.75 ~**	78.80 ×	80.12 ×	82.37 ×
Imbalanced Semeion	85.78	84.32 ~	74.93 ×	76.38 ×	**85.81** ~	79.51 ×	81.72 ×	83.50 ×
Usps	**95.57**	94.91 ×	93.61 ×	94.21 ×	95.42 ~	93.55 ×	94.95 ×	94.56 ×
Mnist	**95.78**	95.76 ~	90.00 ×	92.42 ×	95.49 ×	92.30 ×	92.84 ×	94.06 ×
Asistentur	86.96	86.41 ~	80.51 ×	82.68 ×	**89.20** ✓	81.19 ×	81.36 ×	83.81 ×
win/tie/loss		0/11/9 ×	0/10/10 ×	0/11/9 ×	2/14/3 ~	0/6/14 ×	1/8/11 ×	0/12/8 ×

Table 11. Summary of the different values of accuracy when the training set has a noise rate of 25%. The ✓/~/× symbols indicate that the standard classifier is significantly better/equal/worse than CCE. The best values are shown in bold.

Data Set	CCE	Standard Classifiers						
		BCE	ANN	OAA	Bagging	ECOC	Boosting	OAO
Yeast	57.95	56.34 ~	56.69 ~	57.27 ~	58.94 ~	49.02 ×	**59.10 ~**	57.77 ~
Glass	62.27	27.37 ×	61.71 ~	62.17 ~	**64.46 ~**	60.18 ~	59.15 ~	60.46 ~
Shuttle	99.40	85.55 ×	99.29 ~	99.31 ~	99.58 ~	99.33 ~	**99.60** ~	99.13 ~
WineRed	57.89	57.08 ~	57.76 ~	57.60 ~	57.86 ~	54.83 ×	58.02 ~	**58.22** ~
Vowel	**76.07**	53.93 ×	68.49 ×	74.44 ~	73.18 ×	71.13 ×	74.87 ~	74.76 ~
Pendigits	97.08	92.80 ×	92.57 ×	**97.21** ~	94.39 ×	97.16 ~	96.62 ×	96.43 ×
Segmentation	91.56	91.76 ~	92.16 ~	91.80 ~	**92.44** ✓	90.55 ~	92.39 ✓	91.17 ~
Satimage	85.55	84.58 ×	85.49 ~	84.54 ×	86.52 ~	82.89 ×	**87.09 ~**	85.05 ~
Texture	**99.00**	97.30 ×	97.88 ~	98.45 ~	99.00 ~	98.25 ~	98.76 ~	97.86 ×
Sensorless	91.77	82.34 ×	91.02 ×	87.89 ×	92.05 ✓	85.54 ×	**93.18** ✓	89.31 ×
Synthetic	94.11	93.85 ~	94.94 ~	92.63 ~	**95.65** ~	92.33 ~	88.71 ~	92.45 ~
Optdigits	96.62	**96.74** ~	91.02 ×	88.94 ×	96.45 ~	91.49 ×	95.84 ×	95.05 ×
Automobile	**56.38**	45.59 ×	53.62 ~	54.28 ~	55.81 ~	53.52 ~	50.18 ~	56.08 ~
Libras	68.00	39.19 ×	62.58 ×	61.74 ×	68.15 ~	58.34 ×	**69.13** ~	64.92 ~
Mfeat-fac	93.28	94.31 ~	87.15 ×	87.53 ×	**94.57** ✓	87.67 ×	91.04 ~	89.96 ×
Semeion	88.28	88.29 ~	80.61 ×	81.53 ×	**88.32** ~	77.65 ×	85.14 ×	85.87 ×
Imbalanced Semeion	82.62	**84.40 ~**	72.01 ×	73.31 ×	83.52 ~	68.47 ×	78.05 ×	80.20 ×
Usps	**95.39**	94.70 ×	93.02 ×	93.52 ×	94.70 ×	92.25 ×	94.99 ~	94.04 ×
Mnist	**95.41**	95.36 ~	88.07 ×	90.24 ×	95.00 ~	89.77 ×	92.45 ~	93.39 ×
Asistentur	84.41	85.10 ~	77.43 ×	79.72 ×	**86.54** ✓	78.12 ×	79.22 ×	83.08 ×
win/tie/loss		0/10/10 ×	0/9/11 ×	0/10/10 ×	4/13/3 ~	0/7/13 ×	2/13/5 ×	0/10/10 ×

As it can be appreciated from the results in Tables 8-11:

- The ensembles which show more resilience to noise are *Bagging* and CCE. By the contrary, *Boosting* and BCE are the most prone to fail with the labeling noise.
- For a noise level of 10%, 15%. 20% and 25% the *Bagging* accuracy degrades around 0.9%, 1.21%, 2.2% and 3% respectively.
- In CCE, for a noise level of 10%, the accuracy is reduced in a 1.4% and for a noise level of 15%, 20% and 25% it is reduced in a 2%, a 3% and a 3.8% respectively.
- On the other hand, for a noise level of 10%, the accuracy of ANN, OAA, ECOC and OAO degrades around 1.7%. For higher noise levels, the accuracy is degraded between 2.3 and 6.5%.
- Finally, the accuracy of BCE is reduced in a 4.1%, a 5.4%, a 6.8% and an 8% respectively, and the accuracy of *Boosting* is reduced in a 4.2%, 4.6%, 6.4% and a 6.2% respectively.
- In four of the 20 domains, VOWEL, AUTOMOBILE, USPS and MNIST, CCE always offers the best values of accuracy.
- The comparison over all data sets reveals that, in the presence of noise, CCE, is statistically better than BCE, ANN, OAA, ECOC, OAO and *Boosting* and statistically equivalent to *Bagging*.

This study allows to consider CCE as the best alternative in multiclass problems with different noise levels affecting the class label.

6. Conclusions and Future Work

This work demonstrates that the pairwise combination of ANNs that are trained with disjoint subsets of data is a worthy approach to resolve multiclass problems. A limitation of this schema is that the pool of possible base learners increases exponentially with the number of classes. To mitigate this disadvantage certain rules have been defined from which the CCE architecture is derived.

We have tested the performance of CCE in twenty different domains, and to evaluate their performance we have compared CCE against a single ANN, OAA, ECOC, *Bagging*, *Boosting*, OAO and BCE. The results of the experiments carried out indicate that, according to the *F-test*, CCE significantly outperforms OAA and ANN in 11 data sets, ECOC in 10 data sets, BCE in 8 data sets, and *Bagging* and OAO in 2. *Boosting* outperforms CCE with statistical significance in 4 domains. For the MNIST data set, one of the most difficult classification tasks, the mean accuracy rate achieved by CCE is statistically better than those obtained by the rest of implemented models.

Regarding to the training time, we have experimentally proved that CCE is more efficient than most standard classification models. Only OAO, and sometimes BCE, require a lower training time than CCE.

If we consider simultaneously the accuracy and the training time, we can conclude that CCE outperforms the other analyzed systems: CCE is more accurate than those systems that require less training time (OAO and BCE) and it requires a much shorter training time than the system with a comparable accuracy rate (*Boosting*).

With the objective of estimating the quality of CCE and checking whether it outperforms every one of its members we have studied the relationship between its accuracy and the average accuracy of its base learners. Furthermore, to examine the influence of the diversity on the accuracy of the ensemble, we have computed the correlation between the diversity and the gain of CCE. This study shows that the members of CCE are relatively accurate but quite diverse and that the relationship between diversity and the gain is statistically significant.

In the last set of experiments, we have analyzed the CCE performance in the presence of four levels of labeling noise. In this context, the experimental results show that CCE is as accurate as *Bagging* and less prone to noise than the other implemented classification systems. This resilience, together with the shorter training time highlights CCE as a very good option for classification in multiclass problems.

Finally, it is worth mentioning that when the Wilcoxon signed-ranks test is used to compare the global performance of CCE with the baseline classifiers, the results reveal that, in the absence of labeling noise, CCE is statistically better than ANN, OAA, ECOC, Bagging OAO and BCE and statistically equivalent to Boosting. On the other hand, in presence of labeling noise, the results indicate that CCE is statistically better than BCE, ANN, OAA, ECOC, OAO and Boosting and statistically equivalent to Bagging. Since it is not easy to know in advance the proportion of mislabeled training examples, we can conclude that CCE is the best alternative in all cases.

Some final conclusions about the results can be extracted. From all the experiments, we determined that, in multiclass problems, CCE offers a high correctly classified instance rate, is tolerant to labeling noise and is computational efficient. Moreover, considering these three parameters simultaneously, we can conclude that CCE outperforms many other classifier systems.

Our future work is directed towards different goals. First, we intend to develop a more theoretical study about the relationship between the accuracy of CCE and the number of base learners. This study should include an exhaustive analysis of the effect of the class distribution scheme on the CCE performance. Furthermore, we intend to analyze the results obtained when the assignment of classes to base learners is done using other search techniques as, for example, Genetic Algorithms or Simulated Annealing.

On the other hand, we intend to analyze the dependence between CCE and the algorithm used in the construction of the base learners. Moreover, to increase the diversity of the base learners we think it would be interesting to incorporate a feature selection process into the CCE design. Finally, our work could be completed with an exhaustive study of the influence that both integration and selection methods have on the classification process.

Acknowledgments

This research was supported by the Spanish MINECO under projects TRA2016-78886-C3-1-R and RTI2018-096036-B-C22.

Appendix 1

This appendix shows the micro-average of the F1-score (eq 6) over the different evaluated models when the training set has a noise rate of a) 0%, b) 10%, c) 15%, d) 20% and e) 25%:

$$F_1^{\mu} = \frac{2\sum_{i=1}^{k} TP_i}{2\sum_{i=1}^{k} TP_i + \sum_{i=1}^{k} FP_i + \sum_{i=1}^{k} FN_i} \qquad (6)$$

where:

- TP_i: Is the number of instances of class c_i that are properly identified.
- FN_i: Is the number of instances of class c_i that are incorrectly classified.
- FP_i: Is the number of instances that are incorrectly identified as examples of class c_i.

	F_1^{μ} (Noise Rate: 0%)								F_1^{μ} (Noise Rate: 10%)								F_1^{μ} (Noise Rate: 15%)							
	CCE	Standard Classifiers							CCE	Standard Classifiers							CCE	Standard Classifiers						
		BCE	ANN	OAA	Bagging	ECOC	Boosting	OAO		BCE	ANN	OAA	Bagging	ECOC	Boosting	OAO		BCE	ANN	OAA	Bagging	ECOC	Boosting	OAO
Yeast	0.57	0.57	0.56	0.57	0.59	0.51	**0.60**	0.58	0.57	0.56	0.56	0.57	0.59	0.49	**0.59**	0.58	0.57	0.56	0.57	0.58	0.59	0.51	**0.60**	0.59
Glass	0.68	0.66	0.66	0.68	0.68	**0.69**	0.68	0.67	0.65	0.63	0.65	0.66	0.67	**0.67**	0.65	0.66	0.65	0.62	0.64	0.65	0.67	**0.68**	0.63	0.65
Shuttle	1.00	1.00	0.99	0.99	**1.00**	1.00	0.99	1.00	1.00	0.99	0.99	0.99	**1.00**	0.99	1.00	0.99	1.00	0.99	0.99	0.99	**1.00**	0.99	1.00	0.99
Winered	0.60	0.59	0.59	0.59	0.60	0.59	0.60	0.59	0.59	0.59	0.59	0.59	0.59	0.59	**0.60**	0.59	0.59	0.58	0.59	0.58	0.59	0.58	**0.60**	0.59
Vowel	0.89	0.74	0.79	0.86	0.84	0.89	**0.92**	0.90	**0.85**	0.66	0.76	0.82	0.82	0.84	0.80	0.84	**0.83**	0.64	0.74	0.80	0.78	0.82	0.80	0.83
Pendigits	0.98	0.94	0.92	0.96	0.94	0.98	**0.99**	0.98	0.97	0.93	0.92	0.97	0.94	**0.98**	0.98	0.97	0.97	0.92	0.93	0.98	0.94	**0.98**	0.97	0.97
Segmentation	0.94	0.93	0.94	0.93	0.94	0.93	0.93	**0.95**	0.93	0.94	0.93	0.92	**0.93**	0.92	0.91	0.93	0.93	0.92	0.93	0.93	**0.93**	0.92	0.93	0.92
Satimage	**0.88**	0.86	0.87	0.86	0.87	0.86	0.85	0.88	0.87	0.85	0.87	0.86	0.87	0.86	**0.87**	0.86	0.87	0.85	0.86	0.85	0.87	0.85	**0.87**	0.86
Texture	1.00	0.99	1.00	0.99	**1.00**	1.00	1.00	0.99	0.99	0.98	0.99	0.99	**0.99**	0.99	0.97	0.99	0.99	0.98	0.99	0.99	**0.99**	0.99	0.98	0.98
Sensorless	0.94	0.85	0.93	0.93	0.94	0.93	**0.99**	0.94	0.93	0.84	0.92	0.92	0.93	0.92	**0.94**	0.93	0.93	0.83	0.92	0.92	0.93	0.91	**0.94**	0.92
Synthetic	0.96	0.97	0.96	0.95	**0.97**	0.97	0.96	0.96	0.95	0.97	0.95	0.93	**0.96**	0.96	0.78	0.94	0.95	0.95	0.95	0.93	0.96	**0.96**	0.86	0.94
Optdigits	0.98	0.98	0.96	0.97	0.98	0.98	**0.98**	0.97	0.97	0.97	0.94	0.95	**0.98**	0.97	0.94	0.97	**0.97**	0.97	0.93	0.93	0.97	0.97	0.95	0.96
Automobile	**0.67**	0.61	0.64	0.64	0.65	0.28	0.67	0.67	**0.64**	0.56	0.63	0.63	0.63	0.29	0.60	0.63	**0.61**	0.59	0.58	0.59	0.60	0.59	0.56	0.61
Libras	0.77	0.51	0.73	0.73	0.77	0.80	**0.81**	0.78	0.75	0.61	0.70	0.69	0.75	**0.77**	0.74	0.74	0.73	0.57	0.68	0.65	0.74	**0.75**	0.69	0.70
Mfeat-fac	**0.97**	0.97	0.96	0.97	0.97	0.97	**0.97**	0.97	0.96	0.96	0.93	0.95	**0.97**	0.96	0.92	0.95	0.95	0.95	0.91	0.92	**0.96**	0.94	0.89	0.93
Semeion	**0.91**	0.90	0.86	0.87	0.91	0.88	0.90	0.89	0.89	0.88	0.82	0.83	**0.89**	0.85	0.86	0.87	0.89	0.89	0.83	0.85	**0.90**	0.86	0.87	0.87
Imbalanced Semeion	0.90	0.89	0.85	0.86	0.89	0.87	**0.90**	0.88	**0.88**	0.89	0.81	0.82	0.88	0.85	0.86	0.86	0.87	0.87	0.78	0.79	**0.87**	0.82	0.84	0.84
Usps	0.96	0.96	0.95	0.96	0.96	0.97	**0.97**	0.96	0.96	0.95	0.95	0.95	0.96	**0.96**	0.93	0.95	**0.96**	0.95	0.94	0.95	0.95	0.96	0.94	0.95
Mnist	**0.97**	0.97	0.95	0.97	0.96	0.97	0.97	0.96	**0.96**	0.96	0.93	0.95	0.96	0.96	0.94	0.95	**0.96**	0.96	0.92	0.94	0.96	0.95	0.94	0.95
Asistentur	0.94	0.95	0.94	0.93	0.94	0.94	**0.95**	0.93	0.91	0.90	0.88	0.89	**0.92**	0.91	0.88	0.89	0.89	0.89	0.84	0.86	**0.91**	0.88	0.85	0.87

	CCE	F_1^{μ} (Noise Rate: 20%)							CCE	F_1^{μ} (Noise Rate: 25%)						
		Standard Classifiers								Standard Classifiers						
		BCE	ANN	OAA	Bagging	ECOC	Boosting	OAO		BCE	ANN	OAA	Bagging	ECOC	Boosting	OAO
Yeast	0.57	0.54	0.55	0.56	0.59	0.48	**0.59**	0.57	0.58	0.56	0.56	0.57	0.59	0.51	**0.59**	0.58
Glass	0.66	0.62	0.65	0.67	0.66	**0.67**	0.63	0.66	0.61	0.58	0.62	0.62	**0.64**	0.64	0.62	0.60
Shuttle	0.99	0.99	0.99	0.99	**1.00**	0.99	1.00	0.99	0.99	0.99	0.99	0.99	1.00	0.99	**1.00**	0.99
Winered	0.59	0.58	0.59	0.59	0.59	0.58	0.59	0.59	0.58	0.57	0.58	0.58	0.58	0.58	0.58	**0.58**
Vowel	0.78	0.60	0.72	0.77	0.77	**0.79**	0.76	0.77	**0.77**	0.57	0.69	0.75	0.74	0.75	0.75	0.75
Pendigits	0.97	0.93	0.92	**0.98**	0.94	0.97	0.97	0.97	0.97	0.93	0.93	0.97	0.94	**0.98**	0.97	0.96
Segmentation	0.92	0.94	0.93	0.92	**0.93**	0.92	0.93	0.92	0.92	0.92	0.92	0.92	**0.92**	0.91	0.92	0.91
Satimage	0.86	0.85	0.86	0.85	0.87	0.85	**0.87**	0.86	0.85	0.85	0.85	0.85	0.87	0.84	**0.87**	0.85
Texture	0.99	0.98	0.98	0.99	**0.99**	0.99	0.99	0.98	**0.99**	0.97	0.98	0.98	0.99	0.99	0.99	0.98
Sensorless	0.92	0.83	0.91	0.90	0.92	0.90	**0.93**	0.91	0.92	0.82	0.91	0.90	0.92	0.89	**0.93**	0.89
Synthetic	0.93	0.94	0.93	0.90	**0.96**	0.94	0.72	0.91	0.93	0.94	0.95	0.93	**0.96**	0.95	0.88	0.92
Optdigits	**0.97**	0.97	0.92	0.91	0.97	0.95	0.96	0.96	0.97	**0.97**	0.91	0.89	0.96	0.95	0.96	0.95
Automobile	**0.59**	0.58	0.56	0.56	0.57	0.31	0.55	0.58	**0.56**	0.46	0.54	0.54	0.55	0.56	0.50	0.56
Libras	**0.72**	0.51	0.65	0.64	0.72	0.71	0.70	0.70	**0.71**	0.39	0.65	0.64	0.71	0.69	0.69	0.67
Mfeat-fac	0.94	0.95	0.89	0.90	**0.95**	0.93	0.88	0.92	0.93	0.94	0.87	0.87	**0.95**	0.91	0.91	0.90
Semeion	0.85	0.84	0.75	0.77	**0.86**	0.79	0.80	0.82	0.88	0.88	0.81	0.81	**0.88**	0.84	0.85	0.86
Imbalanced Semeion	0.86	0.84	0.75	0.76	**0.86**	0.80	0.82	0.84	0.83	**0.84**	0.72	0.73	0.83	0.76	0.78	0.80
Usps	**0.96**	0.95	0.94	0.94	0.95	0.95	0.95	0.95	**0.95**	0.95	0.93	0.94	0.95	0.95	0.95	0.94
Mnist	**0.96**	0.96	0.90	0.92	0.95	0.94	0.94	0.94	**0.95**	0.95	0.88	0.90	0.95	0.92	0.94	0.93
Asistentur	0.87	0.86	0.80	0.83	**0.89**	0.85	0.81	0.85	0.84	0.85	0.77	0.80	**0.87**	0.83	0.79	0.83

Statistical comparison using Wilcoxon Signed-Ranks test shows that:

- In absence of labeling noise, CCE is statistically equivalent to ECOC and *Boosting* but statistically better than BCE, ANN, OAA, OAO and *Bagging*.
- In presence of labeling noise, CCE is statistically better than BCE, ANN, OAA, OAO, ECOC and *Boosting* and statistically equivalent to *Bagging*.

References:

[1] T. G. Dietterich, “Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms.,” *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, Sep. 1998.

[2] N. García-Pedrajas and D. Ortiz-Boyer, “An empirical study of binary classifier fusion methods for multiclass classification,” *Inf. Fusion*, vol. 12, no. 2, pp. 111–130, Apr. 2011.

[3] J. Wainer, “Comparison of 14 different families of classification algorithms on 115 binary datasets,” 2016.

[4] J. Quinlan, “Induction of decision trees,” *Mach. Learn.*, vol. 1, pp. 81–106, 1986.

[5] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*. The MIT Press, 1988.

[6] R. S. Michalski, “A Theory and Methodology of Inductive Learning,” in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, T. J. Carbonell, and T. M. Mitchell, Eds. TIOGA Publishing Co., Palo Alto, 1983, pp. 83–134.

[7] T. M. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[8] R. Ranawana, “Multi-Classifier Systems: Review and a roadmap for developers,” *Int. J. Hybrid Intell. Syst.*, vol. 3, no. 1, pp. 35–61, 2006.

[9] T. K. Ho, “Multiple Classifier Combination: Lessons and Next Steps,” *Hybrid Methods Pattern Recognit.*, vol. 47, pp. 171–198, 2002.

[10] P. Pławiak, “Novel genetic ensembles of classifiers applied to myocardium dysfunction recognition based on ECG signals,” *Swarm Evol. Comput.*, 2018.

[11] J. Martins, L. S. Oliveira, R. Sabourin, and A. S. Britto, “Forest species recognition based on ensembles of classifiers,” *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, vol. 2018–November, pp. 371–378, 2018.

[12] C. J. Tan, C. P. Lim, and Y. N. Cheah, “A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models,” *Neurocomputing*, vol. 125, pp. 217–228, 2014.

[13] M. Uriz, D. Paternain, H. Bustince, and M. Galar, “A first approach towards the usage of classifiers’ performance to create fuzzy measures for ensembles of classifiers: A case study on highly imbalanced datasets,” *IEEE Int. Conf. Fuzzy Syst.*, vol. 2018–July, pp. 1–8, 2018.

[14] T. G. Dietterich, “Ensemble Methods in Machine Learning,” in *Multiple Classifier Systems. Lecture Notes in Computer Science*, vol. 1857, Springer Berlin Heidelberg, 2000, pp. 1–15.

[15] L. I. Kuncheva, “Switching Between Selection and Fusion in Combining Classifiers: An Experiment,” *IEEE Trans. Syst. Man. Cybern.*, vol. 32, no. 2, pp. 146–156, 2002.

[16] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, pp. 1–18, 2018.

[17] R. P. W. Duin and D. M. J. Tax, “Experiments with Classifier Combining Rules,” *Mult. Classif. Syst. Lect. Notes Comput. Sci.*, vol. 1857, pp. 16–29, 2000.

[18] L. K. Hansen and P. Salamon, “Neural Network Ensembles,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 19, pp. 993–1001, 1990.

[19] A. Chandra, H. Chen, and X. Yao, “Trade-Off Between Diversity and Accuracy in Ensemble Generation,” *Multi-objective Mach. Learn. Stud. Comput. Intell.*, vol. 16, no. 2006, pp. 429–464, 2006.

[20] M. P. Sesmero, J. M. Alonso-Weber, G. Gutierrez, A. Ledezma, and A. Sanchis, “An ensemble approach of dual base learners for multi-class classification problems,” *Information Fusion*, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S156625351400102X.

[21] M. P. Sesmero, J. M. Alonso-Weber, G. Gutierrez, and A. Sanchis, “CCE: An Approach to Improve the Accuracy in Ensembles by Using Diverse Base Learners,” in *International Conference on Hybrid Artificial Intelligence Systems, HAIS 2014*, 2014, pp. 630–641.

[22] Y. LeCun, “THE MNIST DATABASE of handwritten digits.” [Online]. Available: http://yann.lecun.com/exdb/mnist. [Accessed: 27-Sep-2018].

[23] A. Kota, T. Ozcelebi, J. J. Lukkien, and A. Liotta, “Runtime evaluation of cognitive systems for non-deterministic multiple output classification problems,” vol. 100, pp. 1005–1016, 2019.

[24] M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim, “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems ?,” *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, 2014.

[25] L. I. Kuncheva and C. J. Whitaker, “Ten Measures of Diversity in Classifier Ensembles: Limits for Two Classifiers,” in *Proceedings of IEEE Workshop on Intelligent Sensor*, 2001, p. 10/1-10/10.

[26] F. Roli and G. Giacinto, “Methods for Designing Multiple Classifier Systems,” *Mult. Classif. Syst. Lect. Notes Comput. Sci.*, vol. 2096, pp. 78–87, 2001.

[27] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. 2012.

[28] G. Brown, J. Wyatt, R. Harris, and X. Yao, “Diversity creation methods: a survey and categorisation,” *Inf. Fusion*, vol. 6, no. 1, pp. 5–20, Mar. 2005.

[29] T. G. Dietterich, “Machine-Learning Research,” *AI Mag.*, vol. 18, no. 4, pp. 97–137, 1997.

[30] L. Breiman, “Bagging Predictors,” *Mach. Learn.*, vol. 24, pp. 123–140, 1996.

[31] R. E. Schapire, “The Strength of Weak Learnability,” *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, Jun. 1990.

[32] E. Mayhua-López, V. Gómez-Verdejo, and A. R. Figueiras-Vidal, “A new boosting design of Support Vector Machine classifiers,” *Inf. Fusion*, vol. 25, pp. 63–71, 2015.

[33] T. K. Ho, “The Random Subspace Method for Constructing Decision Forests,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.

[34] G. Zenobi and P. Cunningham, “Using Diversity in Preparing Ensembles of Classifiers Based on Different Feature Subsets to Minimize Generalization Error,” *Mach. Learn. ECML 2001*, vol. 2167, no. 1995, pp. 576–587, 2001.

[35] J. Friedman, “Another approach to polychotomous classifcation,” 1996.

[36] A. Fernández, M. Elkano, M. Galar, J. A. Sanz, S. Alshomrani, H. Bustince, and F. Herrera, “Enhancing evolutionary fuzzy systems for multi-class problems: Distance-based relative competence weighting with truncated confidences (DRCW-TC),” *Int. J. Approx. Reason.*, vol. 73, pp. 108–122, 2016.

[37] T. Hastie and R. Tibshirani, “Classification by Pairwise Coupling,” *Ann. Stat.*, vol. 26, no. 2, pp. 451–471, 1998.

[38] Y. L. Murphey, H. Wang, and G. Ou, “OAHO: An Effective Algorithm for Multi-Class Learning from Imbalanced Data,” in *Proceedings of International Joint Conference on Neural Networks*, 2007, pp. 406–411.

[39] T. G. Dietterich and G. Bakiri, “Solving Multiclass Learning Problems via Error-Correcting Output Codes,” *J. Artif. Intell. Res.*, vol. 2, pp. 263–286, 1995.

[40] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[41] J. Maudes, J. J. Rodríguez, C. García-Osorio, and N. García-Pedrajas, “Random feature weights for decision tree ensemble construction,” *Inf. Fusion*, vol. 13, no. 1, pp. 20–30, 2012.

[42] J. F. Kolen and J. B. Pollack, “Backpropagation is Sensitive to Initial Conditions,” *Complex Syst.*, vol. 4, no. 3, pp. 269–280, 1990.

[43] D. Wolpert, “Stacked Generalization,” *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[44] M. P. Sesmero, A. Ledezma, and A. Sanchis, “Generating ensembles of heterogeneous classifiers using Stacked Generalization,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 5, no. February, pp. 21–34, 2015.

[45] M. Re and G. Valentini, “Ensemble methods: a review,” in *Data Mining and Machine Learning for Astronomy*, Chapman & Hall, 2012, pp. 563–594.

[46] A. J. C. Sharkey, N. E. Sharkey, and U. Gerecke, “The ‘Test and Select’ Approach to Ensemble Combination,” *Mult. Classif. Syst. Lect. Notes Comput. Sci.*, vol. 1857, pp. 30–44, 2000.

[47] N. C. Oza and K. Tumer, “Classifier ensembles: Select real-world applications,” *Inf. Fusion*, vol. 9, no. 1, pp. 4–20, Jan. 2008.

[48] A. Frank and A. Asuncion, “UCI Machine Learning Repository,” *University of California, Irvine, School of Information and Computer Sciences*, 2010. [Online]. Available: http://archive.ics.uci.edu/ml/. [Accessed: 27-Sep-2018].

[49] C. Chang and C. Lin, “LIBSVM : A Library for Support Vector Machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–39, 2011.

[50] J. Alcalá-Fdez, L. Sánchez, S. García, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, “KEEL: a software tool to assess evolutionary algorithms for data mining problems,” *Soft Comput.*, vol. 13, no. 3, pp. 307–318, 2009.

[51] M. P. Sesmero, A. Ledezma, J. M. Alonso-Weber, G. Gutierrez, and A. Sanchis, “Control Learning and Systems Optimization Group.” [Online]. Available: http://www.caos.inf.uc3m.es/datasets/. [Accessed: 09-Apr-2018].

[52] E. L. Allwein, R. E. Schapire, and Y. Singer, “Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers,” *J. Mach. Learn. Res.*, vol. 1, pp. 113–141, 2000.

[53] G. I. Webb, “MultiBoosting: a technique for combining boosting and wagging,” *Mach. Learn.*, vol. 40, no. 2, pp. 159–196, 2000.

[54] G. Zhang, “Neural networks for classification: a survey,” *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.*, vol. 30, no. 4, pp. 451–462, Nov. 2000.

[55] H. Schwenk and Y. Bengio, “Boosting neural networks,” *Neural Comput.*, vol. 12, pp. 1869–1887, 2000.

[56] E. Alpaydin, “Combined 5 × 2 cv F Test for Comparing Supervised Classification Learning Algorithms,” *Neural Comput.*, vol. 11, pp. 1885–1892, 1999.

[57] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Second Edi. Chapman & Hall/CRC, 2000.

[58] J. Demšar, “Statistical Comparisons of Classifiers overMultiple Data Sets,” *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[59] H. Chen, “Diversity and Regularization in Neural Network Ensembles,” Birmingham, 2008.

[60] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, “Ensemble diversity measures and their application to thinning,” *Inf. Fusion*, vol. 6, no. 1, pp. 49–62, 2005.

[61] A. Tsymbal, M. Pechenizkiy, and P. Cunningham, “Diversity in Search Strategies for Ensemble Feature Selection,” *Inf. Fusion*, vol. 6, no. 1, pp. 83–98, Mar. 2005.

[62] T. G. Dietterich, “An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization,” *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, 2000.

[63] N. García-Pedrajas, C. García-Osorio, and C. Fyfe, “Nonlinear boosting projections for ensemble construction,” *J. Mach. Learn. Res.*, vol. 8, pp. 1–33, 2007.

[64] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih, “A Comparison of Prediction Accuracy, Complexity , and Training Time of Thirty-three Old and New Classification Algorithms,” *Mach. Learn.*, vol. 40, no. 3, pp. 203–228, 2000.

[65] L. I. Kuncheva and C. J. Whitaker, “Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy,” *Mach. Learn.*, vol. 51, pp. 181–207, 2003.

[66] G. Brown and L. I. Kuncheva, “‘Good’ and ‘bad’ diversity in majority vote ensembles,” *Lect. Notes Comput. Sci.*, vol. 5997 LNCS, pp. 124–133, 2010.