

# Forming Classifier Ensembles with Deterministic Feature Subspaces

Michał Koziarski\*, Bartosz Krawczyk\* and Michał Woźniak\*

\*Department of Systems and Computer Networks

Faculty of Electronics

Wrocław University of Science and Technology

Wrocław, Poland

Email: {michal.koziarski, bartosz.krawczyk, michal.wozniak}@pwr.edu.pl

**Abstract**—Ensemble learning is being considered as one of the most well-established and efficient techniques in the contemporary machine learning. The key to the satisfactory performance of such combined models lies in the supplied base learners and selected combination strategy. In this paper we will focus on the former issue. Having classifiers that are of high individual quality and complementary to each other is a desirable property. Among several ways to ensure diversity feature space division deserves attention. The most popular method employed here is Random Subspace approach. However, due to its random nature one cannot consider this approach as stable one or suitable for real-life applications. Therefore, we propose a new approach called Deterministic Subspace that constructs feature subspaces in a guided and repetitive manner. We present a general framework and three dedicated measures that can be used for selecting diverse and uncorrelated features for each base learner. This way we will always obtain identical sets of features, leading to creation of stable ensembles. Experimental study backed-up with statistical analysis prove the usefulness of our method in comparison to popular randomized solution.

**Index terms**— Machine learning, ensemble classification, feature subspaces, diversity, deterministic methods.

## I. INTRODUCTION

Contemporary machine learning deals with the ever-increasing complexity of problems, directly connected to the era of big data and data flood. Standard classifiers cannot properly capture the properties of analyzed data or by trying to do so finally become subject to the overfitting process. Therefore, methods that can take advantage of combining several learners to get at the same time advantages of complex decision boundary and simplified models are of high interest for both researchers and practitioners. Such approaches are known as ensemble learning, multiple classifier systems or classifier committees [1] and are being considered as one of the most efficient tools to handle pattern analysis process. It is assumed that we have at our disposal a number of models and combine their predictions in order to get a more efficient recognition system, as a set of weak models may overcome the limitations of using a single strong one.

For the ensemble to work efficiently one needs to supply a pool of diverse classifiers [2]. The diversity itself can be ensured on several different levels. One of the most popular is to train each learner on the basis of different features, in hope that such an embedding into lower dimensions will

at the same time simplify the training procedure and allow classifiers to explore different properties of supplied feature space [3]. Random Subspace (RS) [4] is the most popular implementation of this paradigm. This method assumes that each base learner is trained with a subset of randomly selected features, with the assumption that feature subsets may overlap. This provides simple, but efficient way of managing the diversity in the ensemble. However, there exist a significant drawback of this method, rooted in its randomized nature. Due to lack of any guidance when creating feature subspaces we obtain different sets for each run (e.g., when using cross-validation or repeating experiments). This significantly limits the usability of RS in real-life applications as we are not sure exactly what kind of model we should be using and how stable it is.

To overcome this limitation we introduce a novel approach for forming classifier ensembles based on feature subset named Deterministic Subspace (DS). It is based on the same idea as RS, namely creating a pool of diverse classifiers on the basis of reduced number of features. However, we remove the randomization from it and replace it with a fully guided search approach that guarantee a high stability and repetitiveness of the entire procedure. To obtain a number of equally-sized subspaces we propose a set of metrics dedicated to evaluating the discriminative power of each separate feature and the diversity among created subspaces. A round robin strategy is being employed to evenly distribute features with greedy approach. This leads to the creation of deterministic feature subsets of suitable discriminative power that lead to a creation of efficient ensemble in a guided and deterministic manner. Finally, the created classifiers are combined using a majority voting strategy. The proposed method is as flexible as original RS approach and can work with any kind of base learner.

The main contributions of this works are as follow:

- Novel Deterministic Subspace method for forming classifier ensembles.
- Set of metrics suitable for evaluating the quality and diversity of features being used.
- Greedy round robin approach for creating subspaces of evenly distributed features.
- Thorough experimental evaluation of the proposed approach backed-up by statistical tests.

The rest of the paper is organized as follows. Next section gives the necessary background in recent advances in ensemble classification. Section III gives the full details regarding the proposed DS method. Section IV, while the final section concludes the paper.

## II. RELATED WORKS ON ENSEMBLE CLASSIFICATION

Ensemble classifiers have several desirable properties for the process of pattern classification system design:

- Ensemble techniques allow to exploit local competencies of base learners, thus leading to a potential gain in accuracy of the combined system.
- Due to their structure they are highly flexible methods that could be easily adjusted by the user according to specific needs.
- They prevent us from selecting the worst model from the pool.
- They are easy to implement in parallel and distributed high-performance computing environments.

These properties made them highly regarded approaches for a variety of tasks including classification, regression and clustering. In this paper we will concentrate on their usage in supervised classification.

There are three major issues in designing multiple classifier systems:

- How to create a pool of classifiers characterized by a high individual accuracy and diversity [5].
- How to choose the topology of the ensemble.
- How to efficiently combine the outputs of individual classifiers in order to obtain better final predictions [6].

We will focus on the first issue, as this paper deals with the problem on how to form an efficient pool individuals. For such a group of learners to work well we must ensure that they display differing characteristics, as adding similar or identical models to the pool would not contribute to the quality of the ensemble, but would only increase its computational complexity.

There are three main approaches for introducing diversity into a pool of classifiers, depending whether we work with heterogeneous or homogeneous models:

- Different learning algorithms (different models or different parameters for the same model).
- Different inputs (training base classifiers on different data set partitions or choosing different attributes during training) .
- Different outputs (decompose the classification task e.g. into binary tasks).

Heterogeneous ensembles assume that using varying classifier models is enough to properly diversify the pool. By applying different learning paradigms we will get varying decision boundaries that combined together may promote their strong sides, while reducing their weaknesses. However, in some situations different classifiers may return similar or identical boundaries, thus making the selection process crucial. An entire family of dynamic classifier and ensemble selection

methods deserves mentioning, as they offer a flexible ensemble line-up for each incoming sample [7].

However, when the considered pool is homogeneous one needs to find an alternative way of introducing diversity. Input manipulation for each base classifier is the most straightforward approach. One can either work in the data or feature space. Former approach assumes that we introduce variance into training instances in order for classifiers to capture properties of different subsets of objects. Here Bagging [8] and Boosting [9] are the most popular solutions, but one can also train ensembles on the basis of clusters to preserve spatial relations among instances [10]. Latter solution introduces diversity by splitting the feature space. This can be done in either randomized manner [11] or in a guided way with the usage of feature selection [12] or global optimization methods [13].

Alternatively one may manipulate the outputs of classifiers in order to get a diverse set of learners. Here the most common solution is a multi-class decomposition, where a divide-and-conquer approach is being used to obtain simplified learners specialized on a reduced number of classes. Then a dedicated combination method like Error-Correcting Output Codes is being used to reconstruct the original multi-class task [14]. Two approaches deserving mentioning are binarization (in one-vs-one or one-vs-all manner) [15] and hierarchical decomposition [16].

Finally when using the same model one may initialize it with different or randomized parameters in hope that training process for each method will return diverse classifiers. This approach is based on the assumption of a complex search space during the classifier training procedure, as multiple starting points could end up in different local extrema, thus offering better coverage of the considered problem. Most popular examples are prematurely stopped neural networks or ensembles of Support Vector Machines with varying kernels [17].

## III. DETERMINISTIC FEATURE SUBSPACE APPROACH

Random Subspace method, while being hugely successful approach for ensemble classification [4], does not avoid some of the pitfalls associated with methods randomly selecting their inputs. Specifically, randomly selected subspaces may lack the discriminant power necessary for proper separation of different classes, which in turn can harm performance of whole ensemble. Additionally, even if individually strong subspaces are produced, we still do not have any guarantees on their diversity. In this section we present a novel approach of creating feature subspaces in a fully deterministic way, with the aim of, at least partially, mitigating mentioned issues.

### A. Deterministic subspace algorithm

The idea behind DS approach we propose is to assign features evenly between the subspaces, with some preference of individually strong predictors. To achieve this, we employ separate metrics of both individual feature quality and diversification between subspaces. We then greedily assign features

to subspaces using round robin strategy, based on weighted average of two above metrics. Pseudocode for the proposed method is presented in Algorithm 1.

---

**Algorithm 1** Deterministic Subspace algorithm

---

```

1: create  $k$  empty subspaces
2:
3: repeat
4:   for all subspaces do
5:     for all features not in current subspace do
6:       score =  $\alpha$  feature quality +  $(1 - \alpha)$  diversity
7:     end for
8:     add feature with highest score
9:   end for
10: until every subspace has  $n$  features
11:
12: return subspaces

```

---

### B. Subspace diversity measure

Many different approaches of measuring diversity in classifier ensembles exist in the literature. Most of them, however, are fairly computationally expensive, as they usually require to train classifiers and compute predictions [18]. We instead propose naive but fast approach of measuring evenness of feature spread between subspace.

Let us denote the set of created so far subspaces by  $S$  and current subspace, for which we are considering selecting additional feature  $f$ , by  $S_c$ . We define metric of diversity  $d$  as an average of two components: ratio of subspaces already containing considered feature  $d_f$  and nearest distance between subspaces after selecting said feature  $d_s$ .

$$d_f = 1 - \frac{|\{S_i : f \in S_i\}|}{|S|} \quad (1)$$

$$d_s = 1 - \max_{i \neq c} \frac{|S_c \cap S_i|}{|S_c|} \quad (2)$$

$$d = \frac{d_f + d_s}{2} \quad (3)$$

Using defined above metrics in the framework of deterministic subspace algorithm ensures even spread of features between subspaces. Furthermore, since its values are bound in range from 0 to 1, it naturally extends to the case in which we consider feature quality as well.

### C. Feature quality measures

Ideally, we would like to be able to measure the quality of whole subspace considered. It is especially important taking into account the fact that it can be easily shown that multiple weak features can have increased predictive power when combined. However, due to computational considerations such measures tend to be out of our reach: we have to either employ much less demanding search strategy for subspace creation or rely on rough estimations.

TABLE I  
DETAILS OF DATASETS USED THROUGHOUT THE EXPERIMENT.

No.	Name	Features	Objects	Classes
1	winequality	11	6497	11
2	vowel	13	990	11
3	vehicle	18	846	4
4	segment	19	2310	7
5	ring	20	7400	2
6	thyroid	21	7200	3
7	mushroom	22	5644	2
8	chronic kidney disease	24	157	2
9	automobile	25	159	6
10	wdbc	30	569	2
11	ionosphere	33	351	2
12	dermatology	34	358	6
13	texture	40	5500	11
14	biodegradation	41	1055	2
15	spectfheart	44	267	2
16	spambase	57	4597	2
17	sonar	60	208	2
18	splice	60	3190	3
19	optdigits	64	5620	10
20	mice protein expression	80	552	8
21	coil2000	85	9822	2
22	movement libras	90	360	15

Instead, we propose alternate strategy: giving higher preference to individually strong predictors. While it is not true that using individually stronger features must improve quality of subspace, and even more so whole ensemble, we hypothesize that reducing frequency of appearance of particularly weak features may result in increased performance, especially in cases when classifier used is not resistant to being trained on uninformative features. Furthermore, such strategy is relatively inexpensive, requiring only single computation of ranking between features.

During the experimental study, we evaluate performance of three different measures of features predictive power: accuracy on validation set, mutual information between the feature and target vector, and correlation between the two.

## IV. EXPERIMENTAL STUDY

In this section we present detailed description of conducted experiments, together with obtained results. Our goal was to compare the performance of proposed deterministic method with random subspace approach. We try to assess whether deterministic method can achieve at least as high accuracy without introducing randomness into the algorithm. We also investigate under what conditions, if any, deterministic approach may actually outperform random subspace method.

### A. Datasets

During the experiments 22 datasets with varying number of features and objects were used. All datasets were taken from UCI<sup>1</sup> and KEEL<sup>2</sup> repositories and are publicly available for download. Details of datasets used are presented in Table I.

<sup>1</sup><http://archive.ics.uci.edu/ml/datasets.html>

<sup>2</sup><http://sci2s.ugr.es/keel/datasets.php>

## B. Set-up

All experiments were implemented in Python programming language with usage of scikit-learn machine learning library<sup>3</sup>. In particular, all classification algorithms were taken from dedicated scikit-learn modules to ensure correctness of implementation. Repository with remaining code, sufficient to repeat conducted experiments is publicly available<sup>4</sup>.

Throughout the experiments, performance of three different classifiers was tested: linear Support Vector Machine, CART decision tree and k-nearest neighbors. For every classification algorithm default parameters, provided in corresponding scikit-learn modules, were used.

During the experiments number of features in single subspace  $n$  was fixed at half the total number of features, rounded down. Different numbers of subspaces  $k \in \{5, 10, \dots, 50\}$  were evaluated for both random and deterministic method. Additionally, parameter values  $\alpha \in \{0.0, 0.1, \dots, 0.9\}$  were tested for deterministic subspace approach. Three different feature quality metrics were evaluated, namely: 5-fold cross-validation accuracy on training set, mutual information between features and targets, and absolute correlation between the two. Simple majority voting was used in every case, for both methods. All tests were done employing  $5 \times 2$  cross-validation with combined F-test [19] performed to assess statistical significance of results.

## C. Results and discussion

Summary of the obtained results is presented in Figure 1. It shows averaged classification accuracy across different classifiers and measures of feature quality, with baseline accuracy achieved by random subspace method presented for reference. Figure 2 shows the number of datasets on which deterministic subspace achieved statistically significantly better results than random subspace method minus the number of datasets, on which results were significantly worse. Detailed tables showing the number of statistically significantly better and worse results are presented in Appendix A.

Results of experiments indicate a slightly better performance of proposed method against RS approach for small values of  $\alpha \in \{0.0, 0.1, \dots, 0.5\}$  and CART, k-NN and SVM classifiers, regardless of feature quality metric chosen. Using larger values of  $\alpha$  parameter, however, leads to significantly worse results. This indicates that whereas taking into account individual feature quality may have slight positive influence of quality of subspaces, there exist a threshold after which whole ensemble suffers due to lack of diversity.

## V. CONCLUSIONS AND FUTURE WORKS

Method of deterministic feature subspace creation was presented and tested throughout this paper. During the experimental study, its performance was evaluated and compared to random subspace approach. Proposed method, on average, achieved slightly better results compared to its random counterpart.

<sup>3</sup><http://scikit-learn.org/stable/>

<sup>4</sup><https://github.com/michalkoziarski/DeterministicSubspace>

It is worth noting that both individual feature quality and evenness of feature distribution measures are simplified means of estimating subspace quality and diversity of ensemble, respectively. Proposing alternative, computationally feasible metrics presents possible venue of further investigation. Additionally, different search strategies for feature selection could be considered, allowing using established, more computationally expensive metrics at the cost of depth of the search.

## ACKNOWLEDGMENT

This work was supported by the Polish National Science Center under the grant no. DEC-2013/09/B/ST6/02264.

## APPENDIX A. STATISTICAL SIGNIFICANCE TABLES

Tables containing number of datasets on which proposed deterministic approach achieved either statistically significantly better (indicated with plus sign) or worse (indicated with minus sign) results than random subspace method.

TABLE II  
CART CLASSIFIER, ACCURACY AS FEATURE QUALITY MEASURE.

k	$\alpha$									
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	+6/-2	+6/-0	+7/-0	+5/-0	+7/-0	+3/-1	+4/-4	+1/-6	+1/-8	+2/-9
15	+3/-0	+1/-0	+6/-0	+4/-0	+3/-0	+1/-3	+0/-7	+0/-9	+0/-9	+0/-11
10	+3/-0	+4/-0	+4/-1	+4/-2	+5/-0	+2/-3	+0/-11	+1/-12	+1/-12	+0/-15
30	+3/-0	+4/-0	+3/-0	+2/-1	+0/-0	+1/-3	+1/-11	+1/-12	+1/-11	+1/-12
25	+3/-1	+3/-0	+5/-1	+4/-1	+4/-0	+3/-2	+1/-8	+1/-12	+1/-11	+1/-12
20	+2/-1	+4/-1	+2/-0	+5/-0	+6/-0	+3/-2	+2/-10	+2/-12	+2/-12	+2/-14
35	+2/-0	+3/-0	+3/-1	+6/-0	+3/-0	+1/-1	+1/-12	+1/-15	+1/-13	+1/-13
40	+1/-1	+0/-1	+3/-0	+0/-2	+2/-0	+2/-6	+1/-10	+1/-10	+1/-13	+1/-14
45	+2/-0	+1/-0	+1/-0	+1/-0	+0/-2	+1/-3	+1/-13	+1/-12	+1/-13	+1/-13
50	+2/-0	+3/-0	+2/-1	+1/-0	+1/-0	+1/-1	+1/-13	+1/-13	+1/-14	+1/-15

TABLE III  
CART CLASSIFIER, CORRELATION AS FEATURE QUALITY MEASURE.

k	$\alpha$									
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	+4/-0	+7/-2	+6/-1	+4/-0	+6/-0	+5/-1	+2/-7	+2/-9	+2/-9	+1/-10
15	+4/-1	+3/-0	+3/-0	+4/-2	+3/-0	+0/-2	+0/-13	+0/-13	+0/-13	+0/-13
10	+4/-1	+4/-1	+4/-0	+6/-0	+4/-0	+2/-4	+1/-9	+1/-12	+1/-13	+1/-13
30	+2/-2	+2/-0	+1/-1	+3/-0	+1/-2	+2/-3	+1/-14	+1/-14	+1/-14	+1/-15
25	+0/-2	+0/-0	+1/-0	+0/-1	+0/-1	+1/-3	+1/-15	+1/-16	+1/-15	+1/-16
20	+2/-1	+4/-0	+2/-0	+3/-0	+1/-0	+1/-4	+1/-13	+1/-14	+1/-16	+1/-16
35	+3/-0	+1/-0	+4/-0	+3/-0	+1/-0	+3/-4	+1/-14	+1/-14	+1/-15	+1/-15
40	+0/-1	+3/-2	+1/-1	+2/-1	+2/-1	+1/-5	+1/-14	+1/-15	+1/-14	+1/-15
45	+0/-0	+0/-0	+0/-0	+1/-0	+0/-0	+1/-3	+1/-12	+1/-15	+1/-15	+1/-15
50	+3/-1	+3/-0	+2/-1	+1/-0	+2/-1	+2/-7	+1/-17	+1/-17	+1/-17	+1/-17

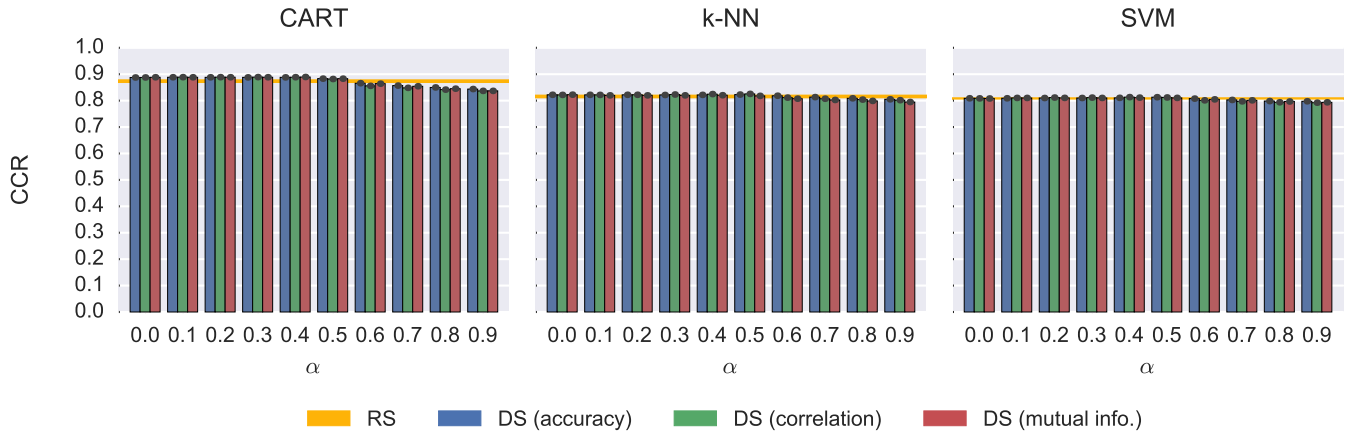


Fig. 1. Correct classification rates averaged over all datasets and examined number of subspaces.

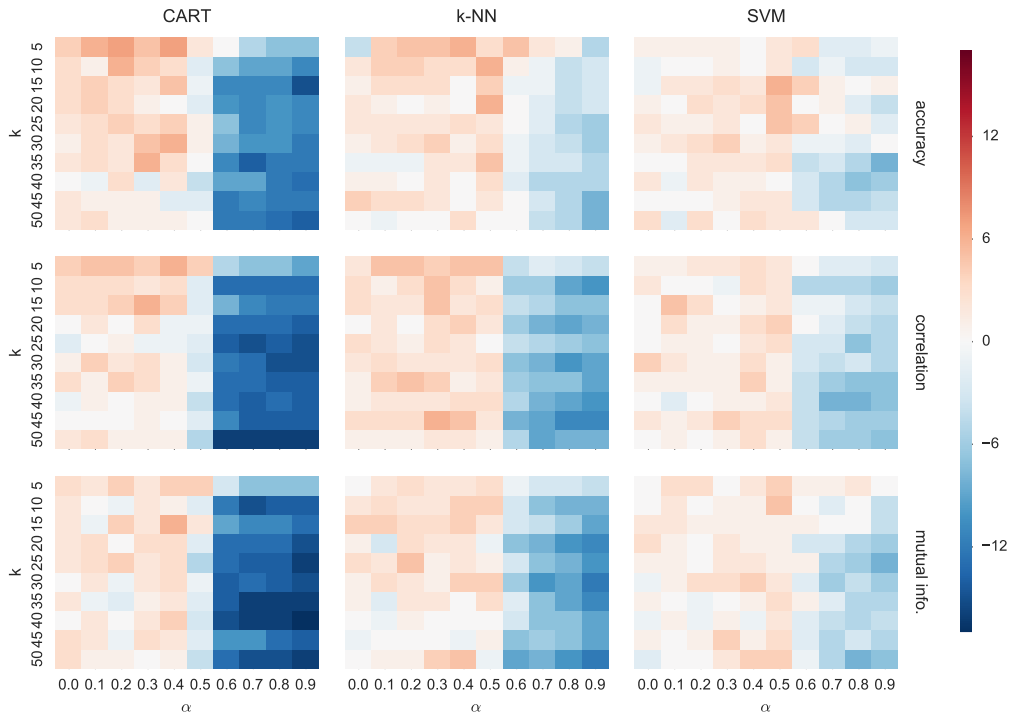


Fig. 2. Differences between the number of datasets on which proposed method achieved statistically significantly better and worst results and their relation to the number of subspaces  $k$  used.

TABLE IV  
CART CLASSIFIER, MUTUAL INFO. AS FEATURE QUALITY MEASURE.

k	$\alpha$										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	
5	+4/-1	+3/-1	+4/-0	+4/-2	+5/-1	+5/-1	+2/-5	+3/-10	+2/-9	+3/-10	
15	+2/-0	+1/-1	+1/-2	+2/-0	+2/-1	+1/-3	+0/-12	+0/-15	+0/-14	+0/-14	
10	+3/-1	+1/-2	+4/-0	+2/-0	+6/-0	+3/-1	+1/-10	+1/-12	+1/-12	+1/-14	
30	+2/-0	+3/-0	+0/-0	+4/-1	+3/-0	+1/-3	+1/-14	+1/-14	+1/-14	+1/-16	
25	+2/-0	+3/-0	+6/-2	+2/-0	+3/-0	+1/-6	+2/-15	+1/-15	+1/-15	+1/-17	
20	+1/-1	+3/-1	+1/-2	+2/-0	+4/-1	+2/-3	+1/-15	+1/-14	+1/-15	+1/-16	
35	+3/-1	+0/-1	+1/-3	+1/-0	+2/-0	+3/-5	+1/-15	+1/-17	+1/-17	+1/-17	
40	+1/-1	+2/-0	+1/-1	+2/-0	+4/-0	+3/-6	+1/-16	+1/-17	+1/-17	+1/-18	
45	+3/-0	+3/-1	+1/-2	+3/-0	+2/-0	+2/-4	+2/-12	+1/-11	+1/-14	+1/-15	
50	+3/-0	+1/-0	+1/-0	+0/-0	+2/-1	+1/-5	+1/-14	+1/-16	+1/-16	+1/-17	

TABLE V  
K-NN CLASSIFIER, ACCURACY AS FEATURE QUALITY MEASURE.

k	$\alpha$										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	
5	+0/-4	+4/-0	+5/-0	+5/-0	+6/-0	+3/-0	+6/-1	+4/-2	+4/-3	+3/-8	
15	+2/-0	+4/-0	+4/-0	+3/-0	+3/-0	+7/-1	+5/-4	+3/-4	+3/-7	+4/-7	
10	+2/-1	+3/-0	+3/-0	+3/-0	+1/-1	+5/-1	+4/-5	+5/-6	+4/-8	+5/-8	
30	+2/-0	+1/-0	+0/-0	+1/-0	+1/-1	+7/-1	+3/-3	+2/-4	+2/-6	+2/-5	
25	+3/-1	+2/-0	+2/-0	+2/-0	+3/-0	+4/-2	+3/-3	+2/-4	+1/-6	+1/-7	
20	+2/-1	+2/-0	+3/-0	+5/-1	+1/-0	+4/-1	+4/-5	+1/-4	+1/-5	+1/-7	
35	+0/-1	+1/-2	+1/-2	+4/-2	+4/-2	+7/-2	+5/-6	+4/-7	+4/-7	+2/-7	
40	+1/-0	+1/-0	+1/-1	+2/-0	+1/-1	+6/-3	+4/-6	+2/-7	+2/-7	+3/-8	
45	+4/-0	+3/-0	+3/-0	+2/-0	+3/-3	+5/-3	+4/-4	+3/-6	+3/-8	+2/-10	
50	+1/-1	+0/-1	+0/-0	+0/-0	+3/-0	+3/-3	+4/-4	+3/-7	+4/-9	+3/-11	

TABLE VI  
K-NN CLASSIFIER, CORRELATION AS FEATURE QUALITY MEASURE.

k	$\alpha$									
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	+3/-1	+6/-1	+6/-1	+5/-1	+6/-1	+6/-1	+2/-6	+3/-5	+2/-5	+2/-6
15	+3/-0	+1/-0	+3/-0	+5/-0	+3/-0	+2/-1	+2/-8	+2/-8	+1/-10	+1/-11
10	+3/-0	+3/-1	+2/-0	+5/-0	+3/-1	+5/-2	+2/-6	+3/-8	+1/-8	+1/-8
30	+1/-1	+4/-2	+1/-1	+3/-0	+4/-0	+5/-3	+3/-9	+3/-11	+3/-12	+4/-12
25	+4/-1	+3/-1	+2/-1	+4/-1	+4/-2	+3/-3	+2/-6	+2/-7	+2/-8	+2/-9
20	+3/-1	+5/-2	+3/-1	+2/-0	+2/-0	+7/-4	+3/-10	+2/-10	+2/-12	+2/-11
35	+3/-2	+4/-0	+5/-0	+4/-0	+3/-2	+5/-3	+3/-9	+2/-9	+2/-9	+2/-11
40	+1/-0	+2/-0	+2/-0	+2/-0	+3/-0	+5/-2	+2/-7	+0/-8	+0/-9	+0/-10
45	+4/-1	+3/-0	+3/-0	+6/-0	+5/-0	+5/-3	+2/-10	+2/-11	+1/-12	+1/-12
50	+1/-0	+1/-0	+1/-0	+2/-0	+2/-0	+5/-4	+1/-6	+1/-10	+2/-10	+2/-10

TABLE VII  
K-NN CLASSIFIER, MUTUAL INFO. AS FEATURE QUALITY MEASURE.

k	$\alpha$									
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	+0/-0	+2/-0	+3/-0	+2/-0	+2/-0	+3/-0	+3/-4	+3/-6	+3/-6	+3/-7
15	+2/-0	+3/-0	+2/-0	+2/-0	+4/-0	+5/-1	+3/-6	+3/-10	+2/-10	+2/-10
10	+4/-0	+4/-0	+4/-1	+4/-1	+4/-0	+5/-3	+4/-7	+3/-7	+3/-9	+2/-11
30	+3/-2	+0/-3	+5/-2	+3/-1	+3/-1	+4/-5	+2/-9	+2/-10	+2/-12	+1/-12
25	+4/-1	+2/-0	+5/-0	+2/-1	+2/-0	+4/-3	+3/-6	+1/-8	+1/-9	+1/-11
20	+2/-1	+3/-0	+2/-0	+1/-0	+4/-0	+6/-2	+2/-8	+1/-11	+1/-10	+0/-12
35	+1/-0	+0/-2	+2/-0	+2/-0	+1/-1	+5/-2	+2/-4	+1/-8	+1/-10	+0/-11
40	+1/-1	+1/-2	+2/-1	+1/-1	+1/-1	+3/-3	+3/-6	+1/-8	+1/-8	+1/-10
45	+0/-1	+0/-0	+0/-0	+0/-0	+0/-1	+4/-3	+1/-8	+2/-8	+2/-9	+0/-9
50	+3/-3	+2/-1	+2/-1	+5/-1	+6/-1	+5/-6	+1/-10	+2/-10	+1/-11	+1/-13

TABLE VIII  
SVM CLASSIFIER, ACCURACY AS FEATURE QUALITY MEASURE.

k	$\alpha$									
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	+1/-0	+1/-0	+1/-0	+1/-0	+0/-0	+2/-0	+3/-0	+2/-4	+1/-3	+1/-2
15	+0/-1	+0/-0	+1/-1	+1/-0	+1/-1	+2/-0	+2/-5	+2/-3	+2/-5	+2/-5
10	+1/-2	+2/-0	+3/-1	+3/-0	+2/-0	+6/-0	+5/-1	+5/-4	+4/-4	+5/-4
30	+2/-1	+2/-2	+3/-0	+3/-1	+5/-2	+6/-1	+4/-4	+5/-4	+4/-6	+4/-8
25	+3/-1	+1/-0	+1/-0	+3/-1	+0/-0	+5/-0	+5/-1	+4/-4	+4/-3	+4/-6
20	+2/-1	+2/-0	+3/-0	+4/-0	+3/-2	+4/-2	+2/-3	+3/-4	+3/-5	+4/-4
35	+1/-1	+1/-1	+2/-0	+2/-0	+2/-0	+4/-1	+2/-6	+2/-5	+2/-7	+1/-9
40	+2/-0	+0/-1	+2/-0	+1/-0	+2/-1	+2/-2	+1/-4	+1/-6	+1/-8	+1/-7
45	+1/-1	+1/-1	+2/-2	+1/-0	+2/-0	+3/-1	+3/-6	+2/-7	+2/-7	+2/-6
50	+3/-0	+1/-3	+3/-0	+2/-2	+4/-1	+5/-3	+5/-3	+5/-5	+5/-8	+5/-8

TABLE IX  
SVM CLASSIFIER, CORRELATION AS FEATURE QUALITY MEASURE.

k	$\alpha$									
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	+1/-0	+2/-1	+3/-1	+2/-0	+3/-0	+2/-0	+5/-5	+4/-6	+3/-5	+3/-6
15	+2/-0	+0/-0	+1/-1	+0/-1	+3/-0	+4/-2	+2/-7	+2/-7	+2/-7	+2/-8
10	+2/-2	+5/-0	+3/-0	+1/-1	+3/-2	+2/-0	+4/-5	+4/-5	+4/-7	+4/-8
30	+1/-1	+3/-0	+2/-1	+2/-1	+3/-0	+5/-1	+5/-5	+4/-6	+4/-8	+4/-9
25	+1/-1	+1/-0	+1/-0	+0/-0	+2/-0	+3/-0	+3/-6	+3/-6	+2/-9	+2/-7
20	+4/-0	+2/-0	+3/-2	+1/-0	+3/-0	+3/-2	+3/-6	+4/-8	+3/-6	+3/-8
35	+1/-0	+2/-1	+1/-0	+3/-2	+4/-0	+2/-1	+3/-7	+3/-9	+2/-9	+2/-9
40	+1/-1	+0/-2	+2/-2	+2/-1	+2/-1	+3/-1	+2/-6	+2/-10	+2/-10	+2/-9
45	+3/-1	+2/-1	+3/-1	+4/-0	+3/-0	+5/-2	+3/-7	+3/-8	+3/-9	+3/-9
50	+0/-0	+1/-0	+1/-1	+2/-1	+3/-0	+2/-1	+3/-7	+2/-8	+2/-8	+3/-10

TABLE X  
SVM CLASSIFIER, MUTUAL INFO. AS FEATURE QUALITY MEASURE.

k	$\alpha$									
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
5	+1/-1	+4/-1	+4/-1	+2/-2	+4/-2	+5/-1	+5/-4	+4/-3	+5/-3	+5/-5
15	+1/-1	+2/-0	+0/-0	+1/-0	+1/-0	+6/-1	+3/-3	+3/-5	+3/-3	+3/-7
10	+2/-0	+2/-0	+2/-1	+2/-1	+1/-0	+2/-1	+4/-3	+4/-4	+4/-4	+3/-7
30	+3/-0	+1/-0	+0/-0	+2/-0	+1/-0	+3/-2	+3/-6	+3/-6	+3/-8	+3/-9
25	+2/-1	+1/-0	+1/-1	+2/-0	+3/-1	+4/-3	+4/-3	+2/-7	+3/-9	+2/-10
20	+0/-1	+1/-0	+4/-1	+3/-0	+5/-1	+4/-2	+5/-7	+4/-10	+4/-8	+4/-10
35	+1/-0	+1/-1	+0/-1	+0/-0	+1/-0	+6/-3	+4/-3	+3/-5	+2/-7	+3/-8
40	+1/-2	+1/-0	+0/-1	+2/-1	+0/-1	+4/-2	+3/-5	+3/-7	+3/-8	+3/-7
45	+1/-0	+0/-0	+2/-1	+4/-0	+1/-0	+4/-1	+3/-5	+3/-8	+3/-6	+3/-8
50	+0/-2	+0/-0	+2/-2	+2/-0	+4/-0	+5/-1	+4/-5	+3/-8	+3/-11	+3/-10

## REFERENCES

- [1] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3–17, 2014.
- [2] S. Wang and X. Yao, "Relationships between diversity of classification ensembles and single-class performance measures," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 206–219, 2013.
- [3] W. M. Czarnecki, R. Józefowicz, and J. Tabor, "Maximum entropy linear manifold for learning discriminative low-dimensional representation," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I*, 2015, pp. 52–67.
- [4] 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.
- [5] T. Windeatt, "Accuracy/diversity and ensemble MLP classifier design," *IEEE Trans. Neural Networks*, vol. 17, no. 5, pp. 1194–1211, 2006.
- [6] B. Krawczyk and M. Woźniak, "Untrained weighted classifier combination with embedded ensemble pruning," *Neurocomputing*, vol. 196, pp. 14 – 22, 2016.
- [7] P. Trajdos and M. Kurzynski, "A dynamic model of classifier competence based on the local fuzzy confusion matrix and the random reference classifier," *Applied Mathematics and Computer Science*, vol. 26, no. 1, p. 175, 2016.
- [8] L. Rokach, "Decision forest: Twenty years of research," *Information Fusion*, vol. 27, pp. 111–125, 2016.
- [9] P. M. Álvarez, J. Luengo, and F. Herrera, "A first study on the use of boosting for class noise reparation," in *Hybrid Artificial Intelligent Systems - 11th International Conference, HAIS 2016, Seville, Spain, April 18-20, 2016, Proceedings*, 2016, pp. 549–559.
- [10] B. Cyganek, "One-class support vector ensembles for image segmentation and classification," *Journal of Mathematical Imaging and Vision*, vol. 42, no. 2-3, pp. 103–117, 2012.
- [11] J. Maudes, J. J. R. Diez, C. I. García-Osorio, and N. García-Pedrajas, "Random feature weights for decision tree ensemble construction," *Information Fusion*, vol. 13, no. 1, pp. 20–30, 2012.
- [12] A. M. P. Canuto, K. M. O. Vale, A. F. Neto, and A. Signoretti, "Reinsel: A class-based mechanism for feature selection in ensemble of classifiers," *Appl. Soft Comput.*, vol. 12, no. 8, pp. 2517–2529, 2012.
- [13] K. Nag and N. R. Pal, "A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification," *IEEE Trans. Cybernetics*, vol. 46, no. 2, pp. 499–510, 2016.
- [14] S. Özgür-Akyüz, T. Windeatt, and R. S. Smith, "Pruning of error correcting output codes by optimization of accuracy-diversity trade off," *Machine Learning*, vol. 101, no. 1-3, pp. 253–269, 2015.
- [15] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "DRWC-OVO: distance-based relative competence weighting combination for one-vs-one strategy in multi-class problems," *Pattern Recognition*, vol. 48, no. 1, pp. 28–42, 2015.
- [16] I. T. Podolak and A. Roman, "Theoretical foundations and experimental results for a hierarchical classifier with overlapping clusters," *Computational Intelligence*, vol. 29, no. 2, pp. 357–388, 2013.
- [17] T. Sun, L. Jiao, F. Liu, S. Wang, and J. Feng, "Selective multiple kernel learning for classification with ensemble strategy," *Pattern Recognition*, vol. 46, no. 11, pp. 3081–3090, 2013.

- [18] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, "Ensemble diversity measures and their application to thinning," *Information Fusion*, vol. 6, no. 1, pp. 49–62, 2005.
- [19] E. Alpaydin, "Combined 5 x 2 cv F test for comparing supervised classification learning algorithms," *Neural Computation*, vol. 11, no. 8, pp. 1885–1892, 1999.