Dynamic Ensemble Selection by K-Nearest Local Oracles with Discrimination Index

Marcelo T. Pereira[a,b], Alceu S. Britto Jr.[a]
[a]*Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, PR, Brazil*
[b]*Instituto Federal do Paraná (IFPR), Umuarama, PR, Brazil*
Emails: marcelo.pereira@ifpr.edu.br, alceu@ppgia.pucpr.br

Luiz S. Oliveira
Universidade Federal do Paraná (UFPR), Curitiba, PR, Brazil
Emails: lesoliveira@inf.ufpr.br

Robert Sabourin
École de Technologie Supérieure, Montreal, QC, Canada
Emails: robert.sabourin@etsmtl.ca

***Abstract*—This work describes a new oracle based Dynamic Ensemble Selection (DES) method in which an Ensemble of Classifiers (EoC) is selected to predict the class of a given test instance (x_t). The competence of each classifier is estimated on a local region (LR) of the feature space (Region of Competence - RoC) represented by the most promising k-nearest neighbors (or advisors) related to x_t according to a discrimination index (Đ) originally proposed in the Item and Test Analysis (ITA) theory. The Đ value is used to better define the advisors of the RoC since they will suggest the classifiers (local oracles) to compose the EoC. A robust experimental protocol based on 30 classification problems and 20 replications have shown that the proposed DES compares favorably with 15 state-of-the-art dynamic selection methods and the combination of all classifiers in the pool.**

***Index Terms*—Discrimination Index, Classical Test Theory, Dynamic Classifier Selection.**

I. Introduction

Multiple Classifier Systems (MCS) have been advocated as an interesting alternative to monolithic solutions in which a single classifier must deal with the large variability inherent to most of pattern recognition problems. In such a scenario, different strategies have been studied to select during the operational phase of the system the most promising classifier(s) given a test instance, producing very interesting MCS variants named Dynamic Classifier Selection (DCS) and Dynamic Ensemble Selection (DES) methods. The main difference between DCS and DES methods is that, in the first only one classifier is selected, and in the second, an ensemble of classifiers is chosen. Thus, in the case of DES, to predict a class for a given test instance, the output of each selected classifier must be combined by using fusion rules [1]. One may find in [2] a taxonomy about the main dynamic selection methods (DCS and DES) available in the literature, which was recently updated in [3]. For sake of simplicity, Dynamic Selection (DS) will be used in this paper to refer both DCS and DES methods.

978-1-5386-5541-2/18/$31.00 ©2018 IEEE

The success of a DS method depends on the adoption of an efficient criterion to evaluate the competence of the classifiers in recognizing the test pattern to be labeled. In the aforementioned taxonomy, the DS methods were categorized according to the criterion used to compute the competence of each classifier. We can find methods based on individual competence and those which consider the relationship between the classifiers that compose the pool. In fact, the competence measures for DS has been under investigation for years [4]. Many studies have focused on accuracy [5], diversity [6], probability information [7], oracle information [8], complexity measures [9], and meta-features such as complexity [9] and instance hardness [10]. However, the search for an efficient criterion to evaluate the competence of the classifiers is still an open problem.

With this in mind, we propose a new oracle-based DES method in which a discrimination index (Đ) [11] is used to better define the region of competence (RoC) in the feature space formed by local advisors (nearest neighbors of the test instance) which are responsible to indicate the classifiers (nearest oracles, as suggested in [8]) to compose the ensemble for a given test instance. This discrimination index is originally defined in the literature of Item and Test Analysis (ITA) belonging to the Classical Test Theory (CTT), as explained in section III of this paper. Originally, a professor (or examiner) can use this kind of index to rank questions in order to select the most promising ones to evaluate its students in an exam. Here, instead of professor, questions, and students; we have a competence measure, the set of advisors (nearest neighbors of x_t) and the classifiers, respectively.

Basically, our hypothesis is that by selecting the local advisors with the highest discrimination index Đ we can better estimate the classifier´s competence and, consequently, improve the DS accuracy. To evaluate our hypothesis, we have performed a set of experiments considering 30 classification problems, 20 replications and a comparison of the proposed method against 15 state-of-the-art DS methods.

This paper is organized as follows: sections II and III show a study of concepts involved with this work: Dynamic Ensemble Selection and Discrimination index. Section IV presents the proposed method. The experimental results and statistical comparisons are shown in section V. Finally, section VI presents our conclusions and future work.

II. Multiple Classifier Systems

In [2] the authors describe an MCS as composed of 3 possible modules, as follows: pool generation, selection and fusion.

A. Pool generation

The first module of an MCS is responsible to build a pool of classifiers $\mathcal{C}$. The pool consists of a set of monolithic classifiers c_i that can be trained using different strategies to generate diversity, such as the Bagging method [12] used in this work. The authors in [3] describe that we can generate a pool of diverse classifiers by considering different initializations, parameters, architectures, base classifiers (pools heterogeneous), training or feature sets. In fact, the classifiers need to be diverse, making different errors. Such a characteristic usually contributes to the pool accuracy. In addition, the pool can also be composed of weak classifiers as reported in [13], in which the author observed that a strong classification can be obtained from the combination of weak classifiers.

B. Selection

The selection module is facultative in an MCS, i.e., all classifiers can be combined. However, usually, a selection process can contribute to improve the classification results. The selection process can be static or dynamic. The former performs the selection during the training phase of the MCS, using the same classifier(s) for all testing samples, while the later executes the selection during the testing phase of the MCS, providing specific classifier(s) for each test sample x_t. Our focus in this work is on dynamic selection (DS) methods, which can either select a single classifier or an ensemble. The Dynamic Classifier Selection (DCS) methods choose the most competent classifier to be the one that will predict the label of the test instance.

The Dynamic Ensemble Selection (DES) selects, for each x_t, an Ensemble of Classifiers $EoC_t \subseteq \mathcal{C}$ that maximizes the competence measure θ (individually or in a group). The rationale behind the DES approaches is to avoid the risk related to the selection of just one classifier, distributing the risk of this over-generalization by choosing a group of classifiers instead of one individual classifier for each test pattern [8]. With respect to DS approaches, we can consider 3 important steps:

a) Region of Competence (RoC) Definition: the first step consists of defining the local region (LR_t) for instance x_t in the feature space in which the competence of the classifiers will be estimated, normally named as region of competence (RoC_t). The RoC_t is usually represented by the k-nearest neighbors of the test instance in a validation set, named in [3] as dynamic selection dataset (DSEL).

b) Classifiers' Competence (θ) Estimation: after the RoC_t definition, it is necessary to compute competence measure θ for each classifier $c_i \in \mathcal{C}$. As described in [2] and [3] the θ can be estimated considering each classifier individually, or in groups. For the first case, each classifier can be evaluated through the use of different sources of information like ranking, accuracy, probability, classifier behavior, oracle information, data complexity, and meta-learning based measures. The strategies based on groups of classifiers evaluate the relevance of each classifier together with others, i.e., the performance of the whole set is considered for each classifier added in the group. In the group strategy, we have measures of competence based on diversity, data handling, and ambiguity.

c) Selection Strategy: the last step in a DS method regards the strategy used to select the classifiers based on the information provided by the defined competence measure. It varies according to the information provided by a simple ranking [14] in which the best classifier(s) is(are) selected to the use of a meta-learning in which a classifier is used to make the selection [10].

C. Fusion

The third phase of an MCS consists in applying the selected classifiers to recognize a given testing pattern. In cases where all classifiers are used (without selection) or when an ensemble is selected, a fusion strategy is necessary. For the integration of the classifier outputs, there are different schemes available in the literature. Complete details regarding the combination methods can be found in Kittler et al. [1].

III. Item and Test Analysis

This section shows some background of the application of Item and Test Analysis (ITA) present on Classical Test Theory (CTT) [15]–[17]. The ITA is used to rank educational tests used to evaluate the student knowledge in a classroom. In this study, we have considered dichotomous tests within ITA. This kind of tests are characterized by having questions with only one true answer and one or several false answers, so that the student either hits or misses the question.

Suppose that we have a test answered by M students, containing N questions. Let $i = \{1, \ldots, M\}$ students, and $j = \{1, \ldots, N\}$ question, the Response (R) of each student i to each question j, is given by:

$$R_{ij} = \begin{cases} 1 & \text{if correct} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

From ITA, an important measure is the ability of a student for an item as denoted in Eq. (2). The student's ability measure (a_i) varies within the range [0, 1] and shows that closer to 1, more competent is the student.

$$a_i = \frac{\sum_j^N R_{ij}}{N} \tag{2}$$

A. Discrimination Index Ð

It is important to note that even the ITA having the same name within Classical Test Theory (CTT) and Item Response Theory (IRT), it is calculated and analyzed differently in each theory. This work focuses on ITA within the CTT, which provides a Discrimination index (Ð) for each question. In other words, a measure to calculate how effective a question could be to distinguish students with high performance (Hp) from those with low performance (Lp). According to Matlock [11], the number of students in each group is equal to $P = 0.27$ (proportion) of the total of students. This Ð measure is important because for an exam, the teacher needs to know the question that most separates the students and then infer some analysis. The discriminant calculation in CTT is done in 4 steps [11], as follows:

1) rank the students by their ability: Eq. (2),
2) separate the two extreme groups: Hp and Lp,
3) for each question j, compute the number of successes (sum of hits) that each group obtained, being: Sh_j and Sl_j, the successes of Hp and Lp group respectively,
4) compute the discrimination index $Ð_j$ as the difference of the success of two groups $\{Sh_j, Sl_j\}$ divided by the number of people in the largest group L:

$$Ð_j = \frac{Sh_j - Sl_j}{L} \tag{3}$$

In the second step, the proportion $P = 0.27$ is used according to Wiersma and Jurs [18], because they have shown that this value will maximize the difference in normal distributions while providing enough cases for analysis. This is confirmed in [11], where the authors pointed out that it is necessary to have enough students in each group to promote stability.

The value $Ð_j$ varies between [-1 and 1]. A value closer to 1 means a greater discrimination power of the item. Negative values would indicate that students with low performance matched that question more than the students in the other group. As this should not happen, usually in situations where the value of $Ð_j$ is negative, the question should be reviewed or canceled.

According to Ebel & Frisbie [19], Ð values greater than 0.4 are considered good. Values between 0.3 and 0.39 are reasonable. Values between 0.20 and 0.29 are items that require revision and less than 0.19 need to be revised or should be deleted.

B. Mapping Item Analysis to Dynamic Selection

The main inspiration for this work was the idea that CTT provides a simple way to calculate (among other measures): 1) the ability of students and 2) a way of measuring how much an item can discriminate (separate) the students. Bringing these concepts into DS, we can make the following associations:

- classifier $\rightarrow$ student: active element in the system that provides answers (correct or incorrect) for questions (or items),
- classifier competence $\rightarrow$ student ability to answer a given question (item),
- pool of classifiers $\rightarrow$ classroom, or a group of students,
- DSEL (validation dataset) $\rightarrow$ database of questions,
- test instance x_t $\rightarrow$ the specific question that professor wants to verify on his students,
- each instance in DSEL $\rightarrow$ an item (question) in an exam,
- region of competence $\rightarrow$ the questions in an exam related to a specific problem (x_t) applied to the students, or in other words, the professor will select only those questions that are related to a specific subject (x_t).

Considering these associations we can calculate Ð for each instance x_t inside region of competence RoC_t and verify which one has more power to distinguish the classifiers available in the pool. In this way, the main idea is to select advisors that have a high Ð, as described in section IV.

IV. Proposed DES Method

This section describes the proposed oracle-based DES method that uses a **DIS**crimination **i**ndex, named DISi. Our method comes from ITA (see section III) to evaluate the importance of each element in the RoC_t of a given test instance x_t. Inspired by the KNORA-Union method [8], we have adapted it to consider a mechanism to evaluate the importance of the local advisors used to define the classifiers to compose the ensemble. Since the proposed method can be considered as the selection module of an MCS, in this section we focus on the description of the main aspects related to how we dynamically select the classifiers to compose the ensemble EoC_t given a test instance x_t.

A. RoC Definition

The definition of the region of competence RoC_t for a given test instance x_t relies on the discrimination index (Ð) from ITA (section III). In the proposed method, the methodology for selecting the best instances to form the RoC_t considering Ð, starts with a local region (LR_t) of size $2 \times k$ (or $2k$), that is, the double of the k-nearest neighbors will be taken initially, however, after the Ð calculation, only the k most discriminant instances are selected to compose RoC_t.

Taking the instances with best Ð among the k-nearest neighbors of x_t in DSEL means that we are providing instances that better separate the classifiers into two groups: the good ones (local high performance) and the bad ones (local low performance) (section III) i.e., the competent ones and the incompetent ones. We expect that these instances having higher Ð are the best choice to suggest the classifiers in the oracle-based schema. In such a schema, for each instance in DSEL we have the most promising classifiers (those which correctly classify it). Acting as a local advisor, each nearest neighbor of x_t can suggest the classifiers to be used [2]. By adding a measure of discrimination in each nearest neighbor of x_t, it is possible to better improve the quality of the RoC_t based on which the classifiers will be selected.

Algorithm 1 shows the steps to calculate Ð and the choice of the most discriminating instances in the RoC_t. Initially, the algorithm receives the pool of classifiers $\mathcal{C}$, the dynamic selection dataset $DSEL$ (validation set in which for each

instance we know the best classifiers), the test instance x_t and the number of neighbors k. It starts by defining the local region LR_t which corresponds to the $2k$-nearest neighbors of x_t within $DSEL$. The number of instances to be captured is twice the k value passed to the algorithm because, at the end of the function, the k instances with the highest discrimination power will be selected.

Algorithm 1 Given x_t, select instances for the region of competence (RoC_t) ranked by Discrimination index

Input: pool of classifiers: $\mathcal{C}$, dynamic selection dataset: $DSEL$, query instance: x_t, number of neighbors: k
Output: region of competence: RoC_t

1: $LR_t \leftarrow$ KNN($DSEL$, x_t, size=$2k$)
2: **for** each c_i in pool $\mathcal{C}$ **do**
3: $\quad Acc_i \leftarrow$ get the accuracy of c_i on LR_t
4: **end for**
5: ${}_r\mathcal{C} \leftarrow$ rank: $\mathcal{C}$, ordered by: Acc, from: max to min
6: $Hp \leftarrow$ get the first 27% of classifiers from: ${}_r\mathcal{C}$
7: $Lp \leftarrow$ get the last 27% of classifiers from: ${}_r\mathcal{C}$
8: $L \leftarrow$ get the size of the largest group
9: **for** each ψ_j in neighbors LR_t **do**
10: $\quad Sh_j \leftarrow$ sum hits for Hp, for instance ψ_j
11: $\quad Sl_j \leftarrow$ sum hits for Lp, for instance ψ_j
12: $\quad Đ_j \leftarrow (Sh_j$ - $Sl_j)$ divided by L
13: **end for**
14: $RoC_t \leftarrow$ get the k best instances (highest Đ)
15: **return** RoC_t

Then, by considering the LR_t, the accuracy of each classifier is calculated (Acc_i). The classifiers are ranked (${}_r\mathcal{C}$) by accuracy and the best and the worst groups of classifiers (Hp and Lp) are chosen. The size of each group of classifiers will be 27% (according to ITA, as shown in section III) of the total classifiers in the pool.

Line 10 shows that the value Sh_j is calculated as the number of classifiers on group Hp that hits a given instance ψ_j (and the same routine is done for Sl_j on next line). In this way, the $Đ_j$ will be calculated, for each ψ_j, as the difference values Sh_j and Sl_j, divided by the size of the largest group L[1].

Finally, the vector $Đ_j$ has the discrimination value of each neighbor instance j. Then, the instances are ranked by the discrimination index, and the k neighbors that have the best values are chosen.

B. Classifiers Competence (θ) Definition

In our method, the competence measure θ_i is computed individually for each $c_i \in \mathcal{C}$. This competence is calculated based on the concept of nearest oracles, i.e., if a classifier c_i hits at least one instance of the RoC_t, then it will have competence θ_i equal to the number of hits, i.e., $\theta_i = Hits(c_i, RoC_t)$. This competence measure θ is the same used in the Knora-Union [2].

[1]Actually, the two groups are equal, then only the 27% of pool size is captured.

C. Selection Schema

In dynamic ensemble selection, only the competent classifiers will be selected for each instance x_t. Our method select, to compose $EoC_t \subseteq \mathcal{C}$, all classifiers that have competence measure θ_i different than zero. When there is no competent classifier to hit at least one instance in the RoC_t, all classifiers will be selected $EoC_t = \mathcal{C}$, and all competence is set to one, $\theta_i = 1$.

D. Fusion

For this final phase, we use the majority vote algorithm to predict the class ω, where $\omega_l \in \Omega$, given Ω as the set of classes in the dataset and $l = \{1..L\}$ the number of classes. To label x_t we follow the steps shown in the Eq. 4, 5, 6.

$$c_i(x_t) \rightarrow \hat{y}_i, \forall c_i \in EoC_t, \hat{y}_i \in \Omega \tag{4}$$

$$\Psi_l \leftarrow \Psi_l + \theta_i, \text{where} \quad l = \hat{y}_i \tag{5}$$

Eq. 5, the classifier competence θ_i will provide a number of votes for the class $\hat{y}_i$. Thus, a vector $\vec{\Psi}$ of size L receive the votes of each classifier:

Considering Eq. 4, 5, the final prediction ω_t for x_t will be the most voted class:

$$\omega_t \leftarrow arg\_max(\vec{\Psi}) \tag{6}$$

V. Experiments and Discussion

In this section we describe the experiments undertaken to evaluated the proposed DES method. The proposed method was compared against 16 different approaches, being: the fusion of all classifiers in the pool using the majority voting [1] rule (ALL); 8 DCS methods and 7 DES methods, respectively: DCS-Rank (Rank) [14], Overall Local Accuracy (OLA) [5], Local Class Accuracy (LCA) [5], APriori (API) [20], APosteriori (APO) [20], Multiple Classifier Behaviour (MCB) [21], Modified Local Accuracy (MLA) [22], Dynamic Selection on Complexity (DSOC) [9], DES-Clustering (CLU) [23], DES-KNN (KNN) [23], KNORA-Eliminate (KNE) [8], KNORA-Union (KNU) [8], Randomized Reference Classifier (RRC) [7], K-Nearest Output Profiles (KNOP) [24] and META-DES (MTD) [25]. The DesLib [26] was used as the library that provides the implementation of major of these DS methods.

Besides the aforementioned comparisons, we have computed the upper limit in terms of performance of each pool of classifiers with respect to the corresponding test dataset, which is known in the literature as the global oracle performance.

A. Experimental Protocol

To ensure a robust experimental protocol, 30 different datasets were extracted from 4 different sources: UC Irvine Machine Learning Repository (UCI) [27], Knowledge Extraction Evolutionary Learning (KEEL) dataset repository [28], Ludmila Kuncheva Collection (LKC) of real medical data [29] and finally a Matlab library for generation of artificial datasets (PRTOOLS) [30]. We chose these datasets because they were

used in the last dynamic selection review [3] and because we believe that 30 samples can generate statistically significant results when comparing different algorithms. Table I shows information about the selected datasets.

For each dataset, we perform 20 replications. For each replication, the dataset was partitioned into training, validation and testing sets with the proportions: $\{Tr, Va, Te\} = \{0.5, 0.25, 0.25\}$. This partition was made randomly but stratified, i.e., respecting the class proportions. The classifiers were trained by bootstrapping sample from Tr. Each bootstrapping sample has the proportion of 0.5 of the training set, which means 25% of the total dataset. Each bootstrapping was done randomly with replacement and stratified.

The pool was generated homogeneously with size $m = 100$, as also reported by [8], [9], [10], i.e., the pool size indicates the number of classifiers c_i that will be considered for the entire dynamic selection system. Perceptron was used as a base classifier (because it is a weak classifier [13]) with number-of-epochs $ne = 100$. All methods were run with the same pool for statistical comparison.

For every considered DS methods, the region of competence was selected using the k-nearest neighbor algorithm with the Euclidean distance and $k = 7$, previously defined in the following works [8]–[10]. All datasets had their features normalized (MinMax algorithm) with instances from train and validation sets. After normalization, all feature values ranges $[0, 1]$. Then, each query instance (x_t), in the generalization phase, is firstly normalized by the MinMax values collected in the training phase.

Finally, for ensemble-based methods, the combination of predictions were made by majority voting rule.

B. Results and Discussion

Table II shows the average and standard deviation of 20 replications for each dataset. This table only shows the top 10 algorithms since the Nemenyi critical difference (CD) in Fig. 2 shows they are equivalent. The last row (Tab. II) shows the average values for each column.

Table III summarizes the results of Tab. II (considering all algorithms):

a) Average accuracy (Acc) and standard deviation (in parenthesis): the proposed method obtained the best average accuracy considering 16 algorithms with $Acc = 81.67$, followed by DES-KNN and META-DES.

b) Friedman rank test (F.Rank) [31]: This test was conducted following the steps: for each dataset the algorithm with the highest Acc value receives the rank 1, the second best receives rank 2 and so on. If an algorithm has the same Acc of another (tie) then an average of the rankings is made and assigned this value for the methods in question. At the end, each algorithm will have a ranking for each of the 30 datasets, so the average of the rankings of each algorithm is calculated. This F.Rank varies $[1 - 30]$ (in our case we have 30 datasets) and the lower the F.Rank value, the better the algorithm. For this test, DISi got the first place with average $rank = 4.37$ followed by DES-KNN and DES-KNOP.

Table I
SELECTED DATASETS

No.	Dataset	#instances	#features	#classes	Source
01	Adult	690	14	2	UCI
02	Banana	2000	2	2	PRTools
03	Blood	748	4	2	UCI
04	CTG	2126	21	3	UCI
05	Diabetes	766	8	2	UCI
06	Ecoli	336	7	8	UCI
07	Faults	1941	27	7	UCI
08	German	1000	24	2	STATLOG
09	Glass	214	9	6	UCI
10	Haberman	306	3	2	UCI
11	Heart	270	13	2	STATLOG
12	ILPD	583	10	2	UCI
13	Ionosphere	351	34	2	UCI
14	Laryngeal1	213	16	2	UCI
15	Laryngeal3	353	16	3	LKC
16	Lithuanian	2000	2	2	LKC
17	Liver	345	6	2	PRTools
18	Magic	19020	10	2	UCI
19	Mammo	830	5	2	KEEL
20	Monk	432	6	2	KEEL
21	Phoneme	5404	5	2	KEEL
22	Segmentation	2310	19	7	KEEL
23	Sonar	208	60	2	UCI
24	Thyroid	692	16	2	LKC
25	Vehicle	846	18	4	STATLOG
26	Vertebral	300	6	2	UCI
27	WBC	569	30	2	UCI
28	WDVG	5000	21	3	UCI
29	Weaning	302	17	2	LKC
30	Wine	178	13	3	UCI

c) Sign Test [32]: pairwise count of Win, Tie and Loss (WTL) for DISi compared to each 16 selected DS methods. According to [32], a popular way to compare the overall performances of classifiers is to count the number of data sets on which an algorithm is the winner. We group the values of wins, ties, and losses separated by *':'*, and this value is shown in column (WTL). This column is ranked by the minimum value of wins. Here we can observe that the last 4 methods losses 30 times (on all datasets) compared with DISi. Figure 1 was built with WTL values. The vertical black line indicates the critical value n_c, calculated as:

$$n_c = \frac{n_{exp}}{2} + Z_\alpha \frac{\sqrt{n_{exp}}}{2} \tag{7}$$

where $n_{exp} = 30$ is the number of experiments (datasets). Knowing values $\alpha = \{0.1, 0.05, 0.01\}$ we have the values of $Z_\alpha = \{1.28, 1.64, 2.32\}$. Thus we get the values: $n_c = \{18.5054, 19.5050, 21.3535\}$, respectively. Figure 1 shows that at significance level 0.1, the DISi is better than all but DES-KNOP and DES-KNN. At significance 0.05 the critical value rises to 19.50 (represented by dashed line) and DISi still maintains superior in 12 of 16 methods, i.e., on 75% of cases. With $\alpha = 0.01$ (dotted line), the proposed method is superior in 8 from 16 methods, corresponding to 50% of cases.

d) Number of Wins (#Wins): following the same idea of Sign Test [32], we also count the number of data sets on wich an algorithm is the overall winner (took the first place, boldface in Tab. II), but in this case, considering all proposed

Table II
THE AVERAGE ACCURACY AND STANDARD DEVIATION OF 20 REPLICATIONS FOR EACH DATASET.

	DISi	KNN	KNOP	MTD	KNU	CLU	API	KNE	OLA	MCB	ORA
Adult	86.02 (2.76)	84.91 (2.25)	**86.66 (2.57)**	84.36 (2.29)	86.16 (2.90)	85.35 (2.31)	83.92 (1.95)	81.92 (3.21)	83.26 (3.31)	82.47 (2.66)	99.13 (0.74)
Banana	97.07 (0.73)	95.75 (1.31)	93.17 (1.53)	96.56 (0.76)	96.47 (0.64)	92.04 (1.94)	**97.38 (0.60)**	96.85 (0.63)	97.29 (0.53)	97.22 (0.64)	99.93 (0.12)
Blood	**77.89 (1.89)**	77.62 (1.51)	76.15 (3.09)	74.22 (3.24)	76.15 (2.63)	77.83 (2.21)	76.52 (2.08)	73.85 (3.63)	76.63 (2.51)	76.74 (2.54)	94.41 (3.76)
CTG	89.21 (0.76)	89.67 (0.94)	88.98 (1.02)	**89.97 (1.74)**	88.80 (1.07)	88.98 (0.80)	89.90 (0.78)	89.63 (1.31)	89.10 (0.97)	89.07 (1.01)	99.12 (0.44)
Diabetes	76.48 (2.34)	75.23 (2.70)	76.87 (2.76)	74.27 (2.28)	**77.03 (2.47)**	76.43 (2.46)	74.82 (3.19)	73.52 (3.15)	74.19 (2.50)	73.70 (2.39)	99.38 (0.52)
Ecoli	83.87 (2.95)	**84.70 (3.21)**	84.35 (3.26)	82.68 (2.55)	83.93 (2.83)	84.64 (2.99)	82.86 (3.33)	82.08 (3.87)	81.96 (3.71)	80.54 (4.69)	97.50 (1.09)
Faults	69.27 (1.70)	69.39 (1.86)	68.84 (2.00)	**70.12 (1.56)**	68.54 (2.78)	68.21 (1.98)	69.42 (1.36)	68.79 (1.96)	68.47 (2.04)	67.41 (1.69)	97.31 (1.03)
German	74.64 (2.48)	74.58 (2.51)	73.28 (3.24)	72.18 (3.52)	72.76 (3.00)	**75.38 (2.63)**	72.66 (2.86)	71.72 (2.99)	71.74 (3.12)	70.52 (3.12)	99.46 (0.57)
Glass	57.36 (5.59)	59.62 (5.39)	54.43 (4.63)	59.53 (5.92)	55.66 (4.95)	56.04 (3.78)	58.30 (5.19)	**62.17 (4.04)**	61.42 (5.25)	61.79 (5.54)	97.17 (2.63)
Haberman	61.25 (15.45)	50.39 (23.96)	53.49 (21.58)	56.45 (20.40)	52.76 (21.93)	50.39 (23.96)	55.86 (19.41)	62.76 (14.79)	63.22 (14.72)	**63.88 (14.32)**	73.82 (13.85)
Heart	82.91 (1.97)	82.09 (1.88)	83.13 (2.70)	80.30 (3.95)	82.91 (2.35)	**83.36 (2.29)**	80.37 (3.22)	77.31 (2.35)	76.27 (4.11)	76.12 (3.32)	98.88 (1.74)
ILPD	70.14 (2.17)	69.24 (2.90)	**71.66 (3.02)**	67.17 (3.80)	71.34 (2.66)	70.72 (1.94)	68.41 (2.67)	69.24 (3.95)	69.48 (3.02)	70.03 (2.71)	99.48 (0.63)
Ionosphere	86.25 (3.15)	87.27 (2.49)	87.10 (2.89)	**87.33 (2.81)**	86.31 (2.96)	86.65 (2.76)	85.45 (3.93)	87.05 (3.66)	85.45 (3.64)	85.23 (3.69)	99.72 (0.63)
Laryngeal1	83.02 (5.05)	81.89 (5.10)	**83.11 (4.26)**	80.38 (3.99)	82.64 (4.69)	81.98 (5.10)	80.09 (4.31)	78.68 (5.38)	79.72 (5.12)	79.06 (3.77)	100.00 (0.00)
Laryngeal3	72.67 (2.75)	72.61 (3.53)	72.33 (3.36)	70.85 (3.75)	72.73 (3.26)	72.39 (2.76)	**72.90 (3.28)**	69.43 (4.42)	69.49 (3.89)	68.64 (4.98)	97.61 (2.90)
Lithuanian	94.86 (1.11)	94.48 (1.04)	91.59 (1.48)	95.46 (0.95)	95.22 (1.01)	89.65 (1.69)	96.16 (0.76)	95.76 (1.05)	**96.27 (0.77)**	96.20 (0.77)	99.76 (0.31)
Liver	**67.73 (3.99)**	66.28 (3.52)	62.44 (3.50)	62.67 (5.70)	58.43 (3.29)	65.35 (4.13)	63.20 (5.11)	66.05 (5.24)	66.57 (4.46)	66.28 (3.56)	97.44 (3.45)
Magic	**82.46 (0.47)**	81.84 (0.74)	81.21 (0.95)	82.33 (0.57)	80.91 (0.69)	79.40 (0.72)	82.17 (0.55)	80.59 (0.78)	82.19 (0.62)	82.14 (0.64)	97.99 (1.23)
Mammo	**80.63 (2.71)**	80.48 (2.98)	80.48 (2.54)	77.49 (3.01)	80.24 (2.23)	80.31 (3.11)	78.84 (2.83)	76.04 (2.66)	79.06 (2.43)	78.91 (2.50)	98.86 (0.98)
Monk	84.12 (3.85)	85.23 (3.03)	84.35 (3.85)	**90.74 (3.76)**	82.36 (4.12)	82.31 (2.97)	84.07 (2.52)	88.06 (3.93)	84.21 (3.93)	83.66 (3.82)	99.58 (0.56)
Phoneme	81.18 (1.05)	78.98 (1.97)	77.39 (1.28)	**83.43 (0.84)**	77.69 (1.35)	75.24 (1.17)	83.08 (1.12)	83.25 (0.82)	82.03 (0.82)	81.99 (0.80)	97.65 (1.47)
Segmentation	92.52 (0.94)	93.47 (1.08)	93.67 (1.01)	**95.03 (0.97)**	92.52 (1.14)	91.69 (1.17)	93.80 (1.01)	94.97 (1.01)	93.71 (1.10)	93.93 (1.00)	99.20 (0.65)
Sonar	77.31 (5.57)	79.23 (4.78)	77.31 (5.28)	79.71 (4.85)	77.50 (5.86)	78.17 (5.59)	78.75 (6.22)	**80.19 (4.63)**	76.15 (6.87)	78.08 (6.22)	99.71 (0.70)
Thyroid	96.71 (1.01)	96.82 (1.18)	96.99 (0.93)	96.45 (1.07)	**96.99 (0.95)**	96.82 (1.25)	96.10 (1.56)	95.84 (1.40)	95.72 (1.69)	95.87 (1.52)	99.88 (0.24)
Vehicle	76.97 (2.54)	**77.51 (1.95)**	76.61 (2.62)	76.80 (2.11)	75.66 (2.75)	77.30 (2.23)	75.90 (1.85)	77.39 (2.58)	76.11 (2.29)	75.26 (1.74)	99.08 (0.82)
Vertebral	85.20 (4.10)	85.93 (2.95)	84.60 (3.76)	83.80 (3.60)	83.80 (4.04)	**86.00 (2.72)**	82.27 (5.25)	84.13 (3.79)	84.60 (3.68)	85.00 (2.87)	99.67 (0.59)
WBC	**97.54 (1.34)**	97.50 (1.22)	97.43 (1.41)	97.29 (1.52)	97.46 (1.30)	97.32 (1.28)	96.90 (1.32)	96.87 (1.30)	95.77 (1.53)	95.67 (1.54)	99.86 (0.37)
WDVG	85.17 (1.02)	84.95 (0.85)	84.77 (1.31)	84.33 (1.14)	83.78 (1.50)	**86.04 (0.90)**	83.69 (0.91)	83.48 (0.94)	83.73 (0.92)	83.05 (0.98)	99.36 (0.32)
Weaning	81.40 (4.65)	**82.20 (4.04)**	81.93 (4.27)	80.27 (3.34)	81.27 (4.23)	82.00 (4.16)	80.33 (4.49)	80.80 (5.00)	80.13 (4.47)	76.47 (5.09)	99.73 (0.55)
Wine	**98.30 (1.93)**	97.84 (1.88)	**98.30 (1.93)**	98.07 (2.12)	**98.30 (1.93)**	97.84 (2.02)	96.93 (2.25)	97.73 (2.33)	97.05 (2.46)	94.55 (3.64)	100.00 (0.00)
Mean	81.67 (10.38)	81.26 (11.18)	80.75 (11.30)	81.01 (11.32)	80.54 (11.70)	80.53 (11.08)	80.70 (11.01)	80.87 (10.47)	80.70 (10.22)	80.31 (10.16)	98.02 (4.74)

The first column shows dataset from Tab. I and other columns are the methods: DISi=Discrimation index, KNN=DES-KNN, KNOP=K-Nearest Output Profiles, MTD=META-DES, KNU=Knora-Union, CLU=DES-Clustering, API=APriori, KNE=Knora-Eliminate, OLA=Overall Local Accuracy, MCB=Multiple Classifier Behaviour, ORA=Global Oracle as upper limit. For each dataset, the maximum value is bold. The last row shows the average values for each column.

Table III
ACCURACY, FRIEDMAN RANK, WIN/TIE/LOSS AND NUMBER OF WINS

Alg.	Acc	Alg.	F.Rank	Alg.	WTL	Alg.	#Wins
DISi	81.67 (10.38)	DISi	4.37 (2.55)	DISi	–	DISi	6
KNN	81.26 (11.18)	KNN	5.07 (3.36)	KNN	18:0:12	MTD	6
MTD	81.01 (11.32)	KNOP	5.77 (3.57)	KNOP	18:1:11	CLU	4
KNE	80.87 (10.47)	MTD	6.43 (3.85)	KNU	19:2:9	KNN	3
KNOP	80.75 (11.30)	KNU	6.47 (2.99)	CLU	19:0:11	KNU	3
API	80.70 (11.01)	CLU	6.63 (4.37)	KNE	20:0:10	API	2
OLA	80.70 (10.22)	API	7.0 (3.19)	API	21:0:9	KNOP	3
KNU	80.54 (11.70)	KNE	7.57 (4.51)	MTD	21:0:9	KNE	2
CLU	80.53 (11.08)	OLA	7.87 (4.03)	OLA	23:0:7	MCB	1
MCB	80.31 (10.16)	MCB	8.67 (4.57)	MCB	23:0:7	OLA	1
RRC	78.63 (11.54)	RRC	9.6 (4.77)	RRC	25:0:5	RRC	1
ALL	78.12 (12.07)	ALL	10.8 (4.69)	ALL	26:0:4	ALL	0
Rank	78.02 (9.66)	DSOC	12.1 (3.66)	Rank	27:0:3	SB	0
DSOC	77.86 (12.18)	Rank	13.27 (4.23)	DSOC	29:0:1	APO	0
MLA	75.78 (12.50)	APO	14.9 (3.19)	MLA	30:0:0	DSOC	0
LCA	75.71 (12.53)	MLA	14.97 (2.3)	APO	30:0:0	LCA	0
APO	75.31 (12.74)	LCA	15.33 (2.01)	LCA	30:0:0	MLA	0
SB	74.23 (11.56)	SB	15.93 (1.91)	SB	30:0:0	Rank	0

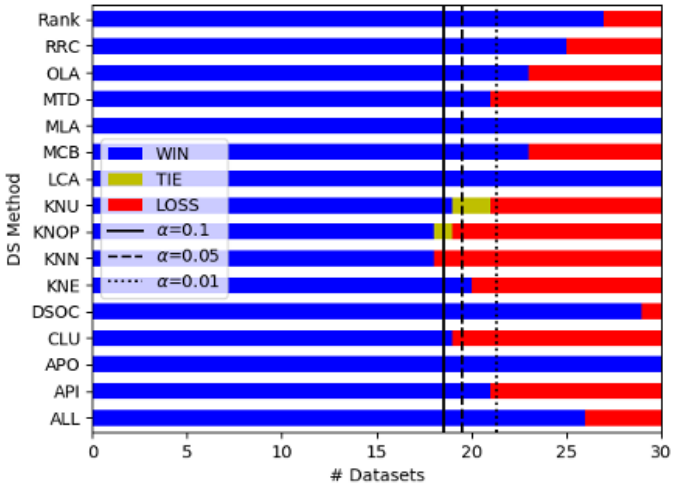

Figure 1. Win, Tie and Loss

methods and not only pairwise comparison, as was done in the above WTL item. As one can see the proposed method got first place tied with META-DES on 6 datasets. We can also see in this column that the last 7 methods did not score any win for the group of selected methods.

After Friedman rank test (Tab. III), Nemenyi critical difference (CD) was calculated [32], Fig. 2. This diagram shows that the results connected by the horizontal bar are statistically equivalent to the average of the Friedman rankings. In this Nemenyi diagram we see that there are 10 firstly equivalent methods: DISi, DES-KNN, KNOP, META-DES, Knora-Union, DES-Clustering, A Priori, Knora-Eliminate, Overal Local Accuracy and Multiple Classifier Behavior. This was one of the reasons for showing only these methods in Table II.

VI. CONCLUSIONS AND FUTURE WORK

We have proposed a new oracle-based DES method in which the local advisors (nearest neighbors) that composed the RoC of a given test instance x_t are evaluated using a proposed

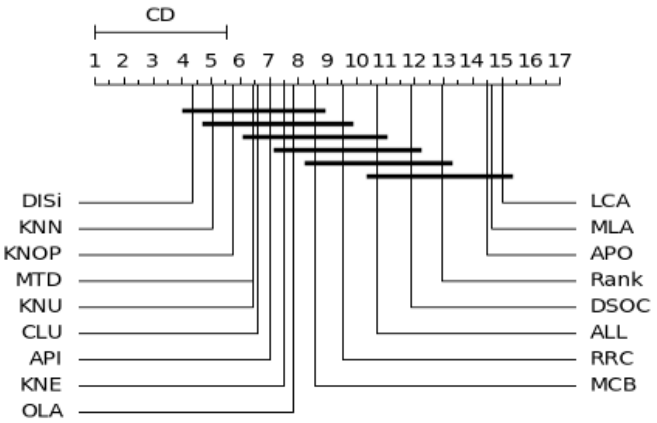

Figure 2. Nemenyi critical difference diagram. $CD = 4.8088$

schema based on a discriminant index used in the Item and Test Analysis (ITA), which is a concept originally presented in Classical Test Theory (CTT).

A robust experimental protocol based on 30 datasets, 20 replications and comparison against other 15 DS methods have shown that the proposed DES method is very promising. Based on the observed results, we confirmed our hypothesis that by selecting the local advisors with the highest discrimination index we can better estimate the classifier competence and, consequently, improve the DS accuracy.

As future work, we plan to better evaluate different sizes of neighborhood, or even consider a selection of neighbors with a variable k value. In addition, the discrimination index can be combined with measures of complexity to better rank the instances that form the RoC.

ACKNOWLEDGMENT

This research is partially supported by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, in portuguese) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico, in portuguese).

REFERENCES

[1] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, mar 1998.

[2] A. d. S. Britto, R. Sabourin, and L. E. S. Oliveira, "Dynamic selection of classifiers - A comprehensive review," *Pattern Recognition*, vol. 47, no. 11, pp. 3665–3680, 2014.

[3] R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Information Fusion*, vol. 41, pp. 195–216, 2017.

[4] B. Antosik and M. Kurzynski, "New Measures of Classifier Competence - Heuristics and Application to the Design of Multiple Classifier Systems," in *Computer Recognition Systems 4*. Springer Berlin Heidelberg, 2011.

[5] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405–410, apr 1997.

[6] R. Lysiak, M. Kurzynski, and T. Woloszynski, "Optimal selection of ensemble classifiers using measures of competence and diversity of base classifiers," *Neurocomputing*, vol. 126, pp. 29–35, feb 2014.

[7] T. Woloszynski and M. Kurzynski, "A probabilistic model of classifier competence for dynamic ensemble selection," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2656–2668, oct 2011.

[8] A. H. R. Ko, R. Sabourin, and A. d. S. Britto, "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recognition*, vol. 41, no. 5, pp. 1718–1731, may 2008.

[9] A. L. Brun, A. S. Britto, L. S. Oliveira, F. Enembreck, and R. Sabourin, "Contribution of data complexity features on dynamic classifier selection," in *2016 International Joint Conference on Neural Networks (IJCNN)*, vol. 2016-Octob. IEEE, jul 2016, pp. 4396–4403.

[10] R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "META-DES.oracle: Meta-learning and feature selection for dynamic ensemble selection," *Information Fusion*, vol. 38, pp. 84–103, nov 2017.

[11] S. Matlock-Hetzel, "Basic Concepts in Item and Test Analysis," in *Annual Meeting of the Southwest Educational Research Association*, no. January 1997, Austin, 1997, pp. 1–7.

[12] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[13] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, jun 1990.

[14] M. Sabourin, A. Mitiche, D. Thomas, and G. Nagy, "Classifier combination for hand-printed digit recognition," in *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*. IEEE Comput. Soc. Press, 1993, pp. 163–166.

[15] R. M. Kaplan and D. P. Saccuzzo, *Psychological testing: Principles, applications, and issues*. Nelson Education, 2017.

[16] L. Crocker and J. Algina, *Introduction to classical and modern test theory.* ERIC, 1986.

[17] F. M. Lord and M. R. Novick, *Statistical theories of mental test scores*. IAP, 2008.

[18] W. Wiersma and S. G. Jurs, *Educational measurement and testing*. Allyn and Bacon, 1985.

[19] R. L. Ebel, *Essentials of educational measurement.*, 3rd ed. Prentice-Hall, 1972.

[20] G. Giacinto and F. Roli, "Methods for dynamic classifier selection," *Proceedings - International Conference on Image Analysis and Processing, ICIAP 1999*, pp. 659–664, 1999.

[21] G. Giacinto and F. Roli, "Dynamic classifier selection based on multiple classifier behaviour," *Pattern Recognition*, vol. 34, no. 9, pp. 1879–1881, sep 2001.

[22] P. Smits, "Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 4, pp. 801–813, apr 2002.

[23] R. Soares, A. Santana, A. Canuto, and M. de Souto, "Using Accuracy and Diversity to Select Classifiers to Build Ensembles," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006, pp. 1310–1316.

[24] P. R. Cavalin, R. Sabourin, and C. Y. Suen, "Dynamic selection approaches for multiple classifier systems," *Neural Computing and Applications*, vol. 22, no. 3-4, pp. 673–688, 2013.

[25] R. M. O. Cruz, R. Sabourin, G. D. C. Cavalcanti, and T. Ing Ren, "META-DES: A dynamic ensemble selection framework using meta-learning," *Pattern Recognition*, vol. 48, no. 5, pp. 1925–1935, may 2015.

[26] R. M. O. Cruz, L. G. Hafemann, R. Sabourin, and G. D. C. Cavalcanti, "Deslib: A dynamic ensemble selection library in python," 2018.

[27] D. Dheeru and E. Karra Taniskidou, "UCI Machine Learning Repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[28] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework." *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, 2011.

[29] L. Kuncheva, "Ludmila kuncheva collection of real medical data sets," 2004, accessed: 12-April-2018. [Online]. Available: http://pages.bangor.ac.uk/~mas00a/activities/real_data.htm

[30] R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. De Ridder, D. Tax, and S. Verzakov, "A matlab toolbox for pattern recognition," *PRTools version*, vol. 3, pp. 109–111, 2000.

[31] M. Friedman, "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, dec 1937.

[32] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.