Information Fusion 41 (2018) 195–216

Contents lists available at ScienceDirect

Information Fusion

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

Dynamic classifier selection: Recent advances and perspectives

Rafael M.O. Cruz[a,*], Robert Sabourin[a], George D.C. Cavalcanti[b]

[a] LIVIA, École de Technologie Supérieure, University of Quebec, Montreal, Québec, Canada
[b] Centro de Informática, Universidade Federal de Pernambuco, Recife, PE, Brazil

ARTICLE INFO

Article history:
Received 5 July 2017
Revised 1 September 2017
Accepted 10 September 2017
Available online 11 September 2017

Keywords:
Multiple classifier systems
Ensemble of classifiers
Dynamic classifier selection
Dynamic ensemble selection
Classifier competence
Survey

ABSTRACT

Multiple Classifier Systems (MCS) have been widely studied as an alternative for increasing accuracy in pattern recognition. One of the most promising MCS approaches is Dynamic Selection (DS), in which the base classifiers are selected on the fly, according to each new sample to be classified. This paper provides a review of the DS techniques proposed in the literature from a theoretical and empirical point of view. We propose an updated taxonomy based on the main characteristics found in a dynamic selection system: (1) The methodology used to define a local region for the estimation of the local competence of the base classifiers; (2) The source of information used to estimate the level of competence of the base classifiers, such as local accuracy, oracle, ranking and probabilistic models, and (3) The selection approach, which determines whether a single or an ensemble of classifiers is selected. We categorize the main dynamic selection techniques in the DS literature based on the proposed taxonomy. We also conduct an extensive experimental analysis, considering a total of 18 state-of-the-art dynamic selection techniques, as well as static ensemble combination and single classification models. To date, this is the first analysis comparing all the key DS techniques under the same experimental protocol. Furthermore, we also present several perspectives and open research questions that can be used as a guide for future works in this domain.

1. Introduction

Multiple Classifier System (MCS) is a very active area of research in machine learning and pattern recognition. In recent years, several studies have been published demonstrating its advantages over individual classifier models based on theoretical [1–3] and empirical [4–6] evaluations. They are widely used to solve many real-world problems, such as face recognition [7], music genre classification [8], credit scoring [9,10], class imbalance [11], recommender system [12,13], software bug prediction [14,15], intrusion detection [16,17], and for dealing with changing environments [18–20].

Several approaches are currently used to construct an MCS, and they have been presented in many excellent reviews covering different aspects of MCS [3,6,21–23]. One of the most promising MCS approaches is Dynamic Selection (DS), in which the base classifiers[1] are selected on the fly, according to each new sample to be classified. DS has become an active research topic in the multiple classifier systems literature in past years. This has been due to the fact that more and more works are reporting the superior performance of such techniques over traditional combination approaches, such as majority voting and Boosting [24–27]. DS techniques work by estimating the competence level of each classifier from a pool of classifiers. Only the most competent, or an ensemble containing the most competent classifiers is selected to predict the label of a specific test sample. The rationale for such techniques is that not every classifier in the pool is an expert in classifying all unknown samples; rather, each base classifier is an expert in a different local region of the feature space [28].

In dynamic selection, the key is how to select the most competent classifiers for any given query sample. Usually, the competence of the classifiers is estimated based on a local region of the feature space where the query sample is located. This region can be defined by different methods, such as applying the K-Nearest Neighbors technique, to find the neighborhood of this query sample, or by using clustering techniques [29,30]. Then, the competence level of the base classifiers is estimated, considering only the samples belonging to this local region according to any selection criteria; these include the accuracy of the base classifiers in this local region [30–32] or ranking [33] and probabilistic models [25,34]. At this point, the classifier(s) that attained a certain competence level is (are) selected.

* Corresponding author.
E-mail addresses: rmoc@cin.ufpe.br, rafaelmenelau@gmail.com (R.M.O. Cruz), robert.sabourin@etsmtl.ca (R. Sabourin), gdcc@cin.ufpe.br (G.D.C. Cavalcanti).
URL: http://www.livia.etsmtl.ca (R.M.O. Cruz), http://www.cin.ufpe.br/~viisar (G.D.C. Cavalcanti)

[1] The term base classifier refers to a single classifier belonging to an ensemble or a pool of classifiers.

http://dx.doi.org/10.1016/j.inffus.2017.09.010
1566-2535/

In this paper, we present an updated taxonomy of dynamic classifier and ensemble selection techniques, taking into account the following three aspects: (1) The selection approach, which considers, whether a single classifier is selected (this is known as Dynamic Classifier Selection (DCS)) or an ensemble is selected (this for its part is known as Dynamic Ensemble Selection (DES)); (2) The method used to define the local region in which the local competences of the base classifiers are estimated, and (3) The selection criteria used to estimate the competence level of the classifier. We review and categorize the state-of-the-art dynamic classifier and ensemble selection techniques based on the proposed taxonomy.

We also discuss the increasing use of dynamic selection techniques, considering different classification contexts, such as One-Class Classification (OCC) [35], concept drift [36–38], One-Versus-One (OVO) decomposition problems [39–41], as well as the application of DS techniques to solve complex real-world problems such as signature verification [42], face recognition [7,43–45], music classification [8] and credit scoring [10]. In particular, we describe how the properties of dynamic selection techniques can be used to handle the intrinsic properties of each problem.

An experimental analysis is conducted comparing the performance of 18 state-of-the-art dynamic classifier and ensemble selection techniques over multiple classification datasets. The DS techniques are also compared against the baseline methods, namely, (1) Static Selection (SS), i.e., the selection of an ensemble of classifiers during the training stage of the system [46]; (2) Single Best (SB), which corresponds to the performance of the best classifier in the pool according to the validation data, and majority voting (MV), which corresponds to the majority voting combination of all classifiers in the pool without any pre-selection of classifiers. To allow a fair comparison of the techniques, all the DS and static techniques were evaluated using the same experimental protocol, i.e., the same division of datasets, as well as the same pool of classifiers. The performance of the DS techniques was also compared with those of the best classification models according to [4], including Support Vector Machine (SVM) and Random Forests.

The contributions of this paper in relation to other reviews in classifiers ensembles are:

1. It proposes an updated taxonomy of dynamic selection techniques.
2. It discusses the use of dynamic selection techniques on different contexts, including One-Versus-One decomposition (OVO), and One-Class Classification (OCC).
3. It reviews the use of DS techniques to solve complex real-world problems such as image classification and biomedical applications.
4. It presents an empirical comparison between several state-of-the-art dynamic selection techniques over several classification datasets under the same experimental protocol.
5. It discusses the most recent findings in this field, and examines the open questions that can be addressed in future works.

This work is organized as follows: Section 2 presents an overview of multiple classifier system approaches. In Section 3, we propose an updated dynamic selection taxonomy, and discuss each component of a DS technique. In Section 4, we describe the most relevant dynamic selection methods and categorize them according to the proposed taxonomy. Section 5 presents a review of several real-world applications that use dynamic selection techniques to achieve a higher classification accuracy. An empirical comparison between the state-of-the-art DS techniques is conducted in Section 6. The conclusion and perspectives for future research in dynamic selection are given in the last section.

2. Basic concepts

This section presents the main concepts comprised in DS approaches. They provide the background needed to understand how DS techniques work as well as the main challenges involved in this class of techniques. The following mathematical notation is used in this paper:

- $C = \{c_1, \ldots, c_M\}$ is the pool consisting of M base classifiers.
- $\mathbf{x}_j$ is a test sample with an unknown class label.
- $\theta_j = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ is the region of competence of $\mathbf{x}_j$, and $\mathbf{x}_k$ is one instance belonging to θ_j.
- $P(\omega_l \mid \mathbf{x}_j, c_i)$ Posterior probability obtained by the classifier c_i for the instance $\mathbf{x}_j$.
- $W_k = \frac{1}{d_k}$, and d_k is the Euclidean distance between the query $\mathbf{x}_j$ and its neighbour sample $\mathbf{x}_k$.
- $\delta_{i,j}$ is the estimated competence of the base classifier c_i for the classification of $\mathbf{x}_j$.
- $\Omega = \{\omega_1, \ldots, \omega_L\}$ is the set of L class labels in the classification problem.
- ω_l is the class predicted by c_i.
- $S(\mathbf{x}_k) = \{S_1(\mathbf{x}_k), \ldots, S_L(\mathbf{x}_k)\}$ is the vector of class supports obtained by the classifier c_i for a given sample $\mathbf{x}_k \in \theta_j$.
- $\tilde{\mathbf{x}}_j$ is the output profile of an instance $\mathbf{x}_j$.
- ϕ_j is the set containing the most similar output profiles of the query sample, $\tilde{\mathbf{x}}_j$, computed in the decision space.

A multiple classifier system is essentially composed of three stages [24]: (1) Generation, (2) Selection, and (3) Aggregation or Fusion (Fig. 1). In the generation stage, a pool of classifiers is trained to create a set of classifiers that are both accurate and diverse. In the selection stage, an ensemble, containing the most competent classifiers, is selected. It must be mentioned that the selection stage is optional, as it is not used by some MCS algorithms. In the last phase, the outputs of the base experts are aggregated to give the final decision of the system.

Several works have been proposed for each of the three phases. In Fig. 2, we present the taxonomy of an MCS, considering the main approaches for classifier generation, selection and integration. The selection phase is highlighted as it is the main focus of this review. The taxonomy of the three MCS stages is described in the following sections.

2.1. Classifier generation

The goal of the classifier generation step is to create a pool, $C = \{c_1, \ldots, c_M\}$, containing M classifiers that are both accurate and diverse. The base classifiers should be different, since there is no reason to combine experts that always present the same output. There are six main strategies to generate a diverse pool of classifiers [3,47]. Similar to [47], the methods are ordered here such that the lower an item appears on the list, the more successful the combination stages will be, as the generated base experts will be more diverse and informative:

1. **Different initializations:** If the training process is initialization dependent, different initializations may result in different classifiers. This holds, for instance, for neural networks, where the initial configuration of weights changes the final model [48,49].
2. **Different parameters:** In this case, the base experts are generated with different configurations of the hyper-parameters, which leads to different decision boundaries; for example, the combination of Support Vector Machine (SVM) classifiers trained with distinct values of cost and kernel scale.
3. **Different architectures:** Like training multiple Multi-Layer Perceptron (MLP) neural networks with different numbers of hidden layers.

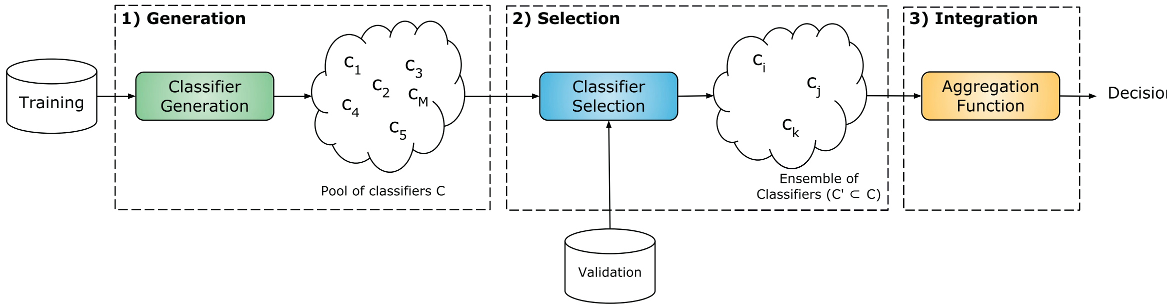


Fig. 1. The three possible phases of an MCS system. In the first stage, a pool of classifiers $C = \{c_1, \dots, c_M\}$ (M is the number of classifiers) is generated. In the second phase, an Ensemble of Classifiers (EoC), $C' \subseteq C$ is selected. In the last phase, the decisions of the selected base classifiers are aggregated to give the final decision.

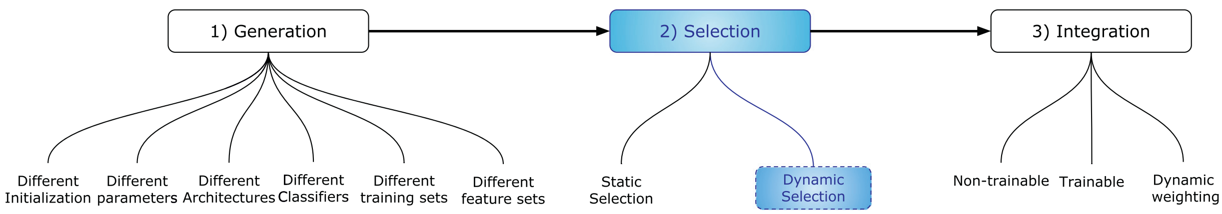


Fig. 2. Taxonomy of an MCS system considering the three main phases. The selection stage is highlighted since it is the focus of this review.

4. **Different classifier models:** This method involves the combination of different classification models (decision tree, K-Nearest Neighbor (K-NN) and SVM, for example). In this case, diversity is due to the intrinsic properties of each model, which change the way the decision boundaries are generated. Systems that use different classifier models are often called *heterogeneous* models.
5. **Different training sets:** In this case, each base classifier is trained using a different distribution of the training set. The Bagging [50,51], Boosting [52,53] and clustering-based classifier generation approaches [10,54] are examples of generation methods that are based on this paradigm.
6. **Different feature sets:** This methodology is used in applications where the data can be represented in distinct feature spaces. In face recognition, for example, multiple feature extraction methods can be applied to extract distinct sets of features based on a face image [43]. The same principle applies to handwriting recognition [55–57] and music genre classification [58]. Each base expert can be trained based on a different feature extraction method. Furthermore, different feature spaces can be generated based on one feature space through the selection of a subset of features, just as in the random subspace method [59,60].

Empirical studies of different generation strategies can be found in [61,62]. It should be mentioned that more than one strategy can be used together. For example, in the Random Forest [63,64] and Rotation Forest [65] techniques, each decision tree can be trained using different feature sets, different divisions of the dataset, as well as different configurations of hyper-parameters.

2.2. Selection

For the selection stage, it can be conducted either in a static or dynamic fashion. Fig. 3 presents the differences in the selection approaches. In static selection methods, the EoC, C', is selected during the training phase, according to a selection criterion estimated in the validation dataset. The same ensemble C' is used to predict the label of all test samples in the generalization phase. The most common selection criteria used for selecting static ensembles are diversity [66–70] and classification accuracy [46,71]. Many search algorithms have been considered for static selection, such as greedy search [72,73], evolutionary algorithms [71,74,75] and other heuristic approaches [69,76].

In contrast, in dynamic selection, a single classifier or an ensemble is selected specifically to classify each unknown example. Based on a pool of classifiers C, dynamic selection techniques consist in finding a single classifier c_i, or an ensemble of classifiers $C' \subseteq C$, having the most competent classifiers for the classification of a specific query, $\mathbf{x}_j$. The rationale for dynamic selection techniques is that each base classifier is an expert in distinct regions of the feature space. The method aims to select the most competent classifiers in the local region where $\mathbf{x}_j$ is located.

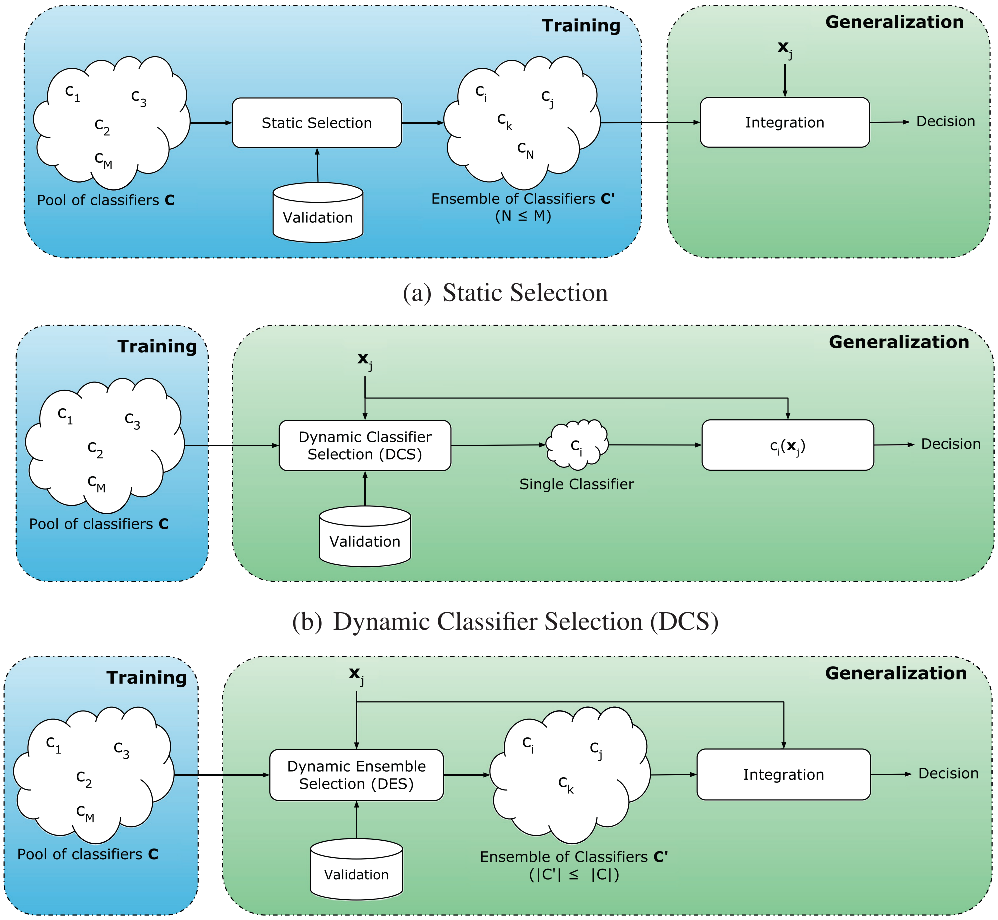
2.3. Aggregation

The aggregation phase consists in fusing the outputs obtained by the selected classifiers according to a combination rule. The combination of the base classifiers can be performed based on the class labels, such as in the Majority Voting scheme, or by using the scores obtained by the base classifier for each of the classes in the classification problem. In this approach, the scores obtained by the base classifiers are interpreted as fuzzy class memberships [77] or the posterior probability that a sample $\mathbf{x}_j$ belongs to a given class [78].

There are three main strategies for the aggregation phase: non-trainable, trainable and dynamic weighting.

2.3.1. Non-trainable

Several non-trainable rules for combining classifiers have been proposed [47,79]. Examples of such aggregation methods are the Sum, Product, Maximum, Minimum, Median and Majority voting schemes [79], Borda count [80], Behavior Knowledge space [81], Decision Templates [82] and Dempster-Shafer combination [83,84]. The effectiveness of different aggregation rules have been analyzed in several works [6,47,80,85–87]. As reported in [47], the problem with non-trainable combination rules is that they require certain assumptions about the base classifiers in order to obtain a good performance. For instance, the Majority Voting and Product rule are effective if the base classifiers are independent, while the Sum rule produces good results when the base classifiers have independent noise behavior.



(a) Static Selection

(b) Dynamic Classifier Selection (DCS)

(c) Dynamic Ensemble Selection (DES)

Fig. 3. Differences between static selection, dynamic classifier selection (DCS) and dynamic ensemble selection (DES). In static selection, the EoC is selected based on the training or validation data. In the dynamic selection approaches, the selection is based on each test data $\mathbf{x}_j$.

2.3.2. Trainable

The rationale for trained combiners is that instead of using just a fixed combination rule, such as the Majority Voting or the Sum rule, the combination may be adapted to the specificity of the classification problem. In this strategy, the outputs of the base classifier are used as input features for another learning algorithm, which learns the aggregation function based on the training data [88].

Several works have demonstrated the fact that trainable combiners outperform non-trainable ones. For instance, in [55–57] an MLP neural network used to combine the outputs of the base experts trained using distinct feature sets outperformed all non-trainable combination rules for the handwritten digit and character recognition. The effectiveness of trained combination rules for small and large sample size data was studied by Raudys in [89] and [90], respectively. An interesting discussion about the benefits of trainable combination schemes is presented by Duin in the paper “The combining classifier: to train or not to train?” [47].

Another interesting trainable approach is the Mixture of Experts (ME) [91,92]. In this methodology, the base classifiers and the aggregation function are trained together. The problem with this approaches lies in the fact they are only defined for neural network ensembles [24]. The main foundations and algorithms in the mixture of experts community are presented in two recent surveys [92,93].

2.3.3. Dynamic weighting

Dynamic weighting is essentially, in essence, similar to dynamic selection methods. They are all based on the local competence of the base classifiers in the region where the query sample $\mathbf{x}_j$ is located. However, instead of selecting a subset of classifiers, the outputs of all classifiers are aggregated to give the final decision such that the most competent classifier receives a higher weight value, and so on.

Examples of dynamic weighting schemes are the local classifier weighting by quadratic programming [94], the dynamic integration of classifiers [37,95], and the fuzzy dynamic classifier aggregation [96]. Moreover, a hybrid dynamic selection and weighting scheme is also possible [25,26,97,98]. In this approach, the base classifiers that presented a certain competence level are first selected. Then, their decisions are weighted based on their estimated competence levels. Experimental results conducted in [25,97,99] demonstrate that the hybrid dynamic selection and weighting approaches usually present the best classification performances when compared to performing only dynamic weighting.

2.4. The Oracle

An important concept in the MCS literature is the concept of the Oracle. The Oracle is an abstract model defined in [1], which always selects the classifier that predicted the correct label, for the given query sample, if such a classifier exists. Although it is possi-

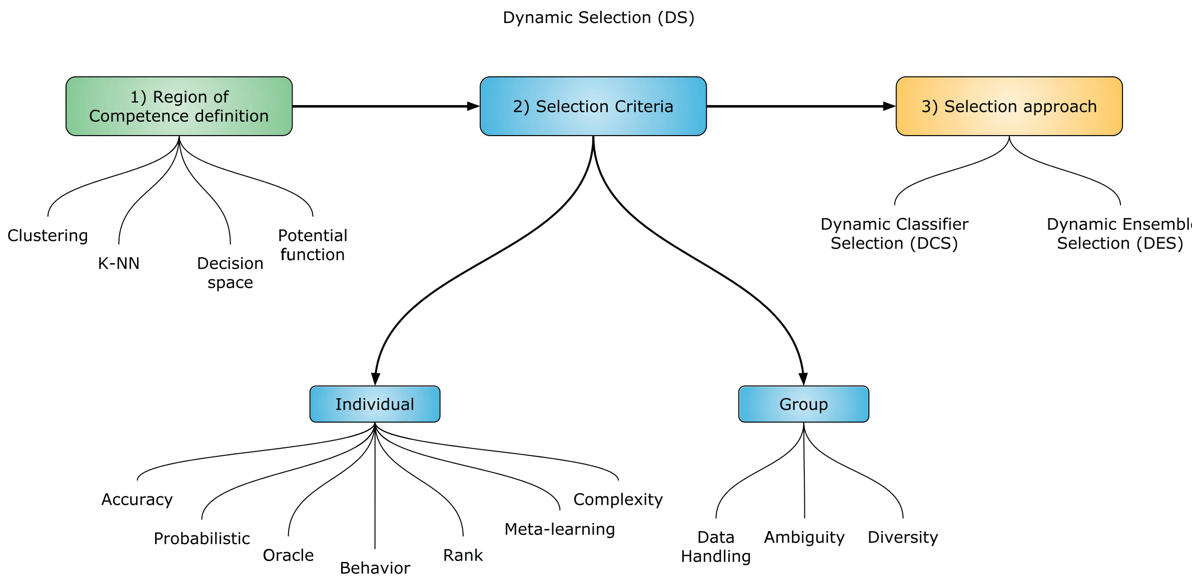


Fig. 4. Taxonomy of dynamic selection systems. The taxonomy of the selection criteria is based on the previous taxonomy proposed by Britto et al. [24].

ble to achieve results higher than the Oracle by working on the supports given by the base classifier [21,100], from a dynamic selection point of view, the Oracle is regarded in the literature as a possible upper limit for the performance of MCS, and as such, it is widely used to compare the performances of different dynamic selection schemes [101]. The Oracle can thus measure how close a DCS technique is to the upper limit performance, for a given pool of classifiers, and indicates whether there is still room for improvement in terms of classification accuracy.

It has however been shown that there is a significant performance gap between DS schemes and the Oracle [26,27,101]. Didaci et al. [101] stated that the Oracle is too optimistic to be considered as an upper bound for dynamic selection techniques. In fact, the Oracle can correctly classify instances that should not be correctly classified based on the Bayesian decision theory [21].

3. Dynamic selection

In dynamic selection, the classification of a new query sample usually involves three steps:

1. Definition of the region of competence; that is, how to define the local region surrounding the query, $\mathbf{x}_j$, in which the competence level of the base classifiers is estimated.
2. Determination of the selection criteria used to estimate the competence level of the base classifiers, e.g., Accuracy, Probabilistic, and Ranking.
3. Determination of the selection mechanism that chooses a single classifier (DCS) or an ensemble of classifiers (DES) based on their estimated competence level.

Fig. 4 presents the taxonomy of DS, considering these three aspects. DS methods can be improved by working on each of these points. For instance, the approaches proposed in [102–104] aim to improve DS techniques by obtaining better estimates of the regions of competence, while several works are based on new criteria for estimating the competence level of the base classifiers [26,27,30,31,34,105,106]. These three aspects are detailed in the following sections.

3.1. Region of competence definition

The definition of a local region is of fundamental importance to DS methods, since the performance of all DS techniques is very sensitive to the distribution of this region [104,107,108]. Indeed, many recent papers have pointed out that it is possible to improve the performance of DS methods just by working on better defining these regions [102–104,109–111].

Usually, the local regions are defined using the K-NN technique [26,31], via clustering methods (e.g., K-Means) [29,30], using the decisions of the base classifiers [106,112,113] or a competence map that is defined through the use of a potential function [25,114]. In all cases, a set of labeled samples, which can be either the training or validation set, is required. This set is called the dynamic selection dataset (DSEL) [27].

3.1.1. Clustering

In techniques that use clustering to define the region of competence [29,30,115,116], the first step is to define the clusters in DSEL. Next, the competence of each base classifier is estimated for all clusters. During the generalization stage, given a new test sample, $\mathbf{x}_j$, the distance between the test sample and the centroid of each cluster is calculated. The competence of the base classifiers are then measured based on the samples belonging to the nearest cluster.

The advantage of using the clustering technique is that all the rankings and classifier selections are estimated during the training phase. For each cluster, the EoC is defined a priori. Hence, DS techniques based on clustering are much faster during the generalization phase. In addition, only the distance between the query sample and the centroids of each cluster needs to be estimated, rather than all instances in DSEL.

3.1.2. K-Nearest Neighbors

In the case of the K-NN technique, the K-Nearest neighbors of the query sample, $\mathbf{x}_j$, are estimated, using the dynamic selection dataset (DSEL). The set with the K-Nearest Neighbors is called the region of competence and is denoted by $\theta_j = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$. Then, the competence of the base classifiers is estimated taking into account only the instances belonging to this region of competence.

The advantage of using K-NN over clustering is that K-NN allows a more precise estimation of the local region, which leads to many different configurations of EoC according to the classification of the new instances [30,116]. However, there is a higher computational cost involved when using K-NN rather than clustering, since the distance between the query and the whole DSEL needs to be estimated prior to estimating the classifiers' competence. This is a problem especially when dealing with large sized datasets [107].

Since the definition of the region of competence plays a very important role in the accuracy of DS techniques, some works have evaluated different versions of the K-NN algorithm for a better estimation of such regions. In [107], the authors considered an adaptive K-NN proposed in [117,118], which shifts the region of competence from the class border to the class centers. Samples that are more likely to be noise were less likely to be selected to compose the region of competence. In [39], the authors used the K-Nearest Neighbors Equality (K-NNE) [119] to estimate the regions of competence. Didaci and Giacinto [103] evaluated the impact of an adaptive neighborhood for dynamic classifier selection techniques. They evaluated the choice of the better suited distance metric to compute the neighborhood as well as a suitable choice of neighborhood size.

3.1.3. Potential function model

These methods are inspired by the work of Rastrigin and Erenstein [114], which is one of the first works to provide a methodology for dynamic selection. They differ from the majority of the other techniques in the DS literature in that they use the whole dynamic selection dataset for the computation of competence, rather than just only the neighborhood of the test sample. However, the influence of each data point in $\mathbf{x}_k \in$ DSEL is weighted by its Euclidean distance to the query $\mathbf{x}_j$ using a potential function model. Usually, a Gaussian potential function is considered (Eq. (1)). Hence, the points that are closer to the query have a higher influence on the estimation of the classifiers' competence.

$$K(\mathbf{x}_k, \mathbf{x}_j) = exp(-d(\mathbf{x}_k, \mathbf{x}_j)^2) \quad (1)$$

Several DS techniques have been proposed using the potential function model: Dynamic Ensemble Selection based on Kullback–Leibler divergence (DES-KL) [34], the technique based on the randomized reference classifier (RRC) [25], the DCS methods based on logarithmic and exponential functions [120].

Using this class of methods to define the regions of competence has the advantage of removing the need to set the neighborhood size a priori as the potential function $K(\mathbf{x}_k, \mathbf{x}_j)$ is used to reduce the influence of each data point based on its Euclidean distance to the query. However, its drawback is the increased computational cost involved in computing the competence of the base classifier since the whole DSEL, and not just the neighborhood of the query sample, is used for the competence estimation.

3.1.4. Decision space

The DS techniques in this category are based on the behavior of the pool of classifiers using the classifiers' predictions as information. They are inspired by the Behavior Knowledge Space (BKS) [81], often called the "decision space" [106,112], since it is based on the decisions made by the base classifiers.

An important aspect of this class of techniques is the transformation of the test and training sample into output profiles. This transformation can be conducted by using the hard decisions of the base classifiers (e.g., the class labels predicted), such as in the BKS method, or by using the estimated posterior probabilities of the base classifiers, as suggested in [106,113,121]. The output profile of an instance $\mathbf{x}_j$ is denoted by $\tilde{\mathbf{x}}_j = \left\{\tilde{\mathbf{x}}_{j,1}, \tilde{\mathbf{x}}_{j,2}, \ldots, \tilde{\mathbf{x}}_{j,M}\right\}$, where each $\tilde{\mathbf{x}}_{j,i}$ is the decision yielded by the base classifier c_i for the sample $\mathbf{x}_j$.

Then, the competence region is calculated by the similarity between the output profile of the query, $\tilde{\mathbf{x}}_j$, and the output profiles of the samples in DSEL. The set with the most similar output profiles, denoted by ϕ_j, is used to estimate the competence level of the base classifiers. Examples of techniques that use a competence region defined in the decision space are the Multiple Classifier Behavior (MCB) [112], K-Nearest Output Profiles (KNOP) [106,121] and META-DES [27,99].

3.2. Selection criteria

The criterion used to measure the competence level of the base classifiers for the classification of $\mathbf{x}_j$ is a key component of any dynamic selection technique. Based on [24], the criteria can be organized into two groups (Fig. 4): individual-based and group-based measures. The former presents the measures where the individual performance of the base classifier is used to estimate its level of competence. The competence of each base classifier c_i is measured independently of the performance of the other base classifiers in the pool. This category can be divided into several other subgroups according to the type of information that is used to measure the competence of the base classifiers, namely, Ranking [31,33], Accuracy [31,102], Probabilistic [25,66,122], Behavior [106,112], Oracle [26], Data complexity [123] and Meta-learning [27,124,125]. It must be mentioned that the system based on meta-learning, however, presents a different perspective of how the competence of a base classifier can be "learned" based on different sources of information.

The group-based measures are composed of metrics that take into account the interaction between the classifiers in the pool. This category can be further divided into three subgroups [24]: Diversity [30,71], Data Handling [126] and Ambiguity [127]. These measures are not directly related to the notion of competence of a base classifier, but rather, to the notion of relevance, i.e., whether the base classifier works well in conjunction with other classifiers in the ensemble. In these techniques, they are based on the performance of the base classifier in relation to the performance of a pre-selected ensemble of classifiers. For instance, in [30], first, an ensemble with the most accurate classifiers is selected (thus, the local accuracy is the criterion used to select the most competent classifiers individually). Next, the system checks the base classifiers that are more diverse, in relation to the pre-selected base classifiers, in order to add more diversity to the EoC.

3.3. Selection approach

Regarding the selection approach, dynamic selection techniques can select either a single classifier, dynamic classifier selection (DCS) or select an ensemble of classifiers (dynamic ensemble selection (DES)). Early works in dynamic selection started with the selection of a single classifier rather than an ensemble of classifiers (EoC). In such techniques, only the classifier that attained the highest competence level is used for the classification of the given test sample. Examples of DCS methods are the A Priori and A Posteriori methods [101], as well as the Multiple-Classifier Behaviour (MCB) [112].

However, given the fact that selecting only one classifier can be highly error-prone, some researchers decided to select a subset of the pool of classifiers rather than just a single base classifier. All base classifiers that obtained a certain competence level are used to compose the EoC, and their outputs are aggregated to predict the label of $\mathbf{x}_j$. Examples of DES techniques are the K-Nearest Oracles (KNORA) [26], the K-Nearest Output Profiles (KNOP) [106], the method based on the Randomized Reference Classifier (RRC) DES-RRC [25] and the META-DES framework [27,99]. Another reason for selecting an EoC rather than a single classifier model is that, frequently, several base classifiers present the same competence level locally. In such cases, the question then is why all of them are not selected, rather than one being randomly chosen.

Table 1
Categorization of DS methods. They are based on their year of publication. All methods presented in this table are later considered in our comparative study (Section 6).

Technique	Region of competence definition	Selection criteria	**Selection approach**	Reference	Year
Classifier Rank (DCS-Rank)	K-NN	Ranking	DCS	Sabourin et al. [33]	1993
Overall Local Accuracy (OLA)	K-NN	Accuracy	DCS	Woods et al.[31]	1997
Local class accuracy (LCA)	K-NN	Accuracy	DCS	Woods et al.[31]	1997
A Priori	K-NN	Probabilistic	DCS	Giacinto[129]	1999
A Posteriori	K-NN	Probabilistic	DCS	Giacinto[129]	1999
Multiple Classifier Behavior (MCB)	K-NN	Behavior	DCS	Giacinto et al.[112]	2001
Modified Local Accuracy (MLA)	K-NN	Accuracy	DCS	P.C. Smits[32]	2002
DES-Clustering	Clustering	Accuracy & Diversity	DES	Soares et al.[30,116]	2006
DES-KNN	K-NN	Accuracy & Diversity	DES	Soares et al.[30,116]	2006
K-Nearest Oracles Eliminate (KNORA-E)	K-NN	Oracle	DES	Ko et al.[26]	2008
K-Nearest Oracles Union (KNORA-U)	K-NN	Oracle	DES	Ko et al.[26]	2008
Randomized Reference Classifier (RRC)	Potential function	Probabilistic	DES	Woloszynski et al.[25]	2011
Kullback-Leibler (DES-KL)	Potential function	Probabilistic	DES	Woloszynski et al.[34]	2012
DES Performance (DES-P)	Potential function	Probabilistic	DES	Woloszynski et al.[34]	2012
K-Nearest Output Profiles (KNOP)	K-NN	Behavior	DES	Cavalin et al.[106]	2013
META-DES	K-NN	Meta-Learning	DES	Cruz et al.[27]	2015
META-DES.Oracle	K-NN	Meta-Learning	DES	Cruz et al.[130]	2016
Dynamic Selection On Complexity (DSOC)	K-NN	Accuracy & Complexity	DCS	Brun et al.[123]	2016

4. Dynamic selection techniques

In this section, we present a review of the most relevant dynamic selection algorithms. The DS techniques were chosen taking into account their importance in the literature by the introduction of new concepts in the area (i.e., methods that introduced different ways of defining the competence region or selection criteria), their number of citations, as well as the availability of source code. Minor variations of an existing technique, such as different versions of the KNORA-E technique proposed in [102] and [98], were not considered. Furthermore, we gave more emphasis to the techniques proposed in the last four years, since they were published after the last reviews in MCS [21,22,24,128].

Table 1 categorizes the key dynamic selection techniques described in this review according to our proposed taxonomy. These techniques are used in our experimental evaluation conducted in Section 6. They are detailed in the next sections [2]. Moreover, in Section 4.19, we present the use of DS techniques in different pattern recognition contexts (e.g., One-Class Classification and One-Versus-One decomposition).

4.1. Modified Classifier Ranking (DCS-Rank)

In the Modified Classifier Ranking method [31,33], the ranking of a single base classifier c_i is simply estimated by the number of consecutive correctly classified samples in the region of competence θ_j. The classifier that correctly classifies the highest number of consecutive samples is considered to have the highest "rank", and is selected as the most competent classifier for the classification of $\mathbf{x}_j$.

4.2. Overall Local Accuracy (OLA)

In this method [31], the level of competence, $\delta_{i,j}$, of a base classifier c_i is simply computed as its classification accuracy in the region of competence θ_j (Eq. (2)). The classifier presenting the highest competence level is selected to predict the label of $\mathbf{x}_j$.

$$\delta_{i,j} = \frac{1}{K}\sum_{k=1}^{K} P(\omega_l \mid \mathbf{x}_k \in \omega_l, c_i) \tag{2}$$

[2] Code for all 18 DS the techniques is available upon request.

4.3. Local classifier accuracy (LCA)

The LCA technique [31] is similar to the OLA, with the only difference being that in the former, the local accuracy is estimated in respect of output classes ω_l (ω_l is the class assigned for $\mathbf{x}_j$ by c_i) for the whole region of competence (Eq. (3)). The classifier presenting the highest competence level, $\delta_{i,j}$, is selected to predict the label of $\mathbf{x}_j$.

$$\delta_{i,j} = \frac{\sum_{\mathbf{x}_k \in \omega_l} P(\omega_l \mid \mathbf{x}_k, c_i)}{\sum_{k=1}^{K} P(\omega_l \mid \mathbf{x}_k, c_i)} \tag{3}$$

4.4. A Priori

The A Priori method [129] considers the probability of correct classification of the base classifier c_i, in θ_j taking into account the supports obtained by the base classifier c_i. Hence, the vector containing the posterior probabilities for each class is considered instead of only the label assigned to each $\mathbf{x}_k \in \theta_j$. Moreover, this method also weights the influence of each sample, $\mathbf{x}_k$, in the region of competence according to its Euclidean distance to the query $\mathbf{x}_j$. The closest samples have a higher influence on the computation of the competence level $\delta_{i,j}$. Eq. (4) demonstrates the calculation of the competence level $\delta_{i,j}$ using the A Priori method:

$$\delta_{i,j} = \frac{\sum_{k=1}^{K} P(\omega_l \mid \mathbf{x}_k \in \omega_l, c_i) W_k}{\sum_{k=1}^{K} W_k} \tag{4}$$

The classifier with the highest value of $\delta_{i,j}$ is selected. However, this selected classifier is only used to predict the label of $\mathbf{x}_j$ if its competence level is significantly better than that of the other base classifiers in the pool (i.e., when the difference in competence level is higher than a predefined threshold). Otherwise, all classifiers in the pool are combined using the majority voting rule.

4.5. A Posteriori

The A Posteriori method [129] works similarly to the A Priori. The only difference is that it takes into account the class predicted by the base classifier c_i, for the test sample $\mathbf{x}_j$ during the competence estimation (Eq. (5)).

$$\delta_{i,j} = \frac{\sum_{\mathbf{x}_k \in \omega_l} P(\omega_l \mid \mathbf{x}_k, c_i) W_k}{\sum_{k=1}^{K} P(\omega_l \mid \mathbf{x}_k, c_i) W_k} \tag{5}$$

The classifier with the highest value of $\delta_{i,j}$ is selected. As in the A Priori method, the selected classifier is only used to predict the label of $\mathbf{x}_j$ if its competence level is significantly better than that of the other base classifiers in the pool (i.e., when the difference in competence level is higher than a predefined threshold). Otherwise, all classifiers in the pool are combined using the majority voting rule.

4.6. Multiple Classifier Behavior (MCB)

The MCB technique is based on the behavior knowledge space (BKS) [81] and the classifier local accuracy. Given a new test sample $\mathbf{x}_j$, its region of competence, θ_j, is estimated. Next, the output profiles of the test sample as well as those of the region of competence are computed using the BKS algorithm.

The similarity between the output profile of the test sample $\tilde{\mathbf{x}}_j$ and those from its region of competence, $\tilde{\mathbf{x}}_k \in \theta_j$, are calculated (Eq. (6)). Samples with similarities lower than a predefined threshold are removed from the region of competence θ_j. Hence, the size of the region of competence is variable, since it also depends on the degree of similarity between the query sample and those in its region of competence. After all the similar samples are selected, the competence of the base classifier, $\delta_{i,j}$, is simply estimated by its classification accuracy in the resulting region of competence.

$$S(\tilde{\mathbf{x}}_j, \tilde{\mathbf{x}}_k) = \frac{1}{M}\sum_{i=1}^{M} T(\mathbf{x}_j, \mathbf{x}_k) \tag{6}$$

$$T(\mathbf{x}_j, \mathbf{x}_k) = \begin{cases} 1 & \text{if} \quad c_i(\mathbf{x}_j) = c_i(\mathbf{x}_k), \\ 0 & \text{if} \quad c_i(\mathbf{x}_j) \neq c_i(\mathbf{x}_k). \end{cases} \tag{7}$$

Similar to the A Priori and A Posteriori techniques (Sections 4.4 and 4.5), the decision is made as follows: If the selected classifier is significantly better than the others in the pool (difference in competence level is higher than a predefined threshold), it is used for the classification of $\mathbf{x}_j$. Otherwise, all classifiers in the pool are combined using the majority voting rule.

4.7. Modified Local Accuracy (MLA)

Proposed by Smits [32], this technique aims to solve the problem of defining the size of the region of competence (i.e., the number of instances selected to compose the region of competence). When the value of K is too high, instances that are not similar to $\mathbf{x}_j$ may be included in the region of competence, while a low value of K may lead to insufficient information. To tackle this issue, the MLA algorithm weights each instance in θ_j by its distance to $\mathbf{x}_j$ (Eq. (8)):

$$\delta_{i,j} = \sum_{k=1}^{K} P(\omega_l \mid \mathbf{x}_k \in \omega_l, c_i) W_k \tag{8}$$

The classifier presenting the highest competence level, $\delta_{i,j}$, is selected to predict the label of $\mathbf{x}_j$.

4.8. DES-clustering (DES-KMEANS)

In this method [30], the K-Means algorithm is applied to DSEL in order to sub-divide this set into several clusters. For each cluster produced, the classifiers are ranked in decreasing order of accuracy and in increasing order of diversity. The Double Fault measure [67] is used to measure the diversity of the base classifiers. The N most accurate and the J most diverse classifiers are associated to each cluster.

Given a new sample $\mathbf{x}_j$ of unknown class, its Euclidean distance to the centroid of each cluster is calculated. Then, the set of N most accurate and J most diverse classifiers associated to the nearest cluster is used to compose the ensemble of classifiers C'.

4.9. DES-KNN

The first step in this technique is to compute the region of competence θ_j. Then, the base classifiers are ranked in decreasing order of accuracy and in increasing order of diversity based on the samples belonging to θ_j. The Double Fault measure [67] was used, since it presented the highest correlation with ensemble accuracy in the study conducted by Shipp and Kuncheva [131]. Then, the N most accurate classifiers and the J most diverse classifiers are selected to compose the EoC, C'. The values of J and N, ($J \leq N$) must be defined prior to applying this method.

4.10. KNORA-Eliminate

The KNORA-Eliminate technique [26] explores the concept of Oracle, which is the upper limit of a DCS technique. Given the region of competence θ_j, only the classifiers that correctly recognize all samples belonging to the region of competence are selected. In other words, all classifiers that achieved a 100% accuracy in this region (i.e., that are local Oracles) are selected to compose the ensemble C'. Then, the decisions of the selected base classifiers are aggregated using the majority voting rule. If no base classifier is selected, the size of the region of competence is reduced, and the search for the competent classifiers is restarted.

4.11. KNORA-Union

The KNORA-Union technique [26] selects all classifiers that are able to correctly recognize at least one sample in the region of competence. This method also considers that a base classifier can participate more than once in the voting scheme when it correctly classifies more than one instance in the region of competence. The number of votes of a given base classifier c_i is equal to the number of samples in the region of competence, θ_j, for which it predicted the correct label. For instance, if a given base classifier c_i predicts the correct label for three samples belonging to θ_j, it gains three votes for the majority voting scheme. The votes collected by all base classifiers are aggregated to obtain the ensemble decision.

4.12. Randomized reference classifier (DES-RRC)

This method is based on the randomized reference classifier, in order to decide whether or not the base classifier c_i performs significantly better than the random classifier. The level of competence of c_i is computed based on two parts: a source of competence C_{src} and a Gaussian potential function $K(\mathbf{x}_k, \mathbf{x}_j)$ (Eq. (1)), which is used to reduce the influence of each data point in DSEL based on its Euclidean distance to $\mathbf{x}_j$. Thus, the competence level of a base classifier, c_i, for the classification of the query, $\mathbf{x}_j$, is estimated using Eq. (9).

$$\delta_{i,j} = \sum_{\mathbf{x}_k \in DSEL} C_{src}\, K(\mathbf{x}_k, \mathbf{x}_j) \tag{9}$$

The source of competence C_{src} is estimated based on the concept of randomized reference classifier (RRC) proposed in [105][3]. The base classifiers that presented a competence level higher than the random classifier are selected to compose the ensemble C'. The base classifiers with a level of competence $\delta_{i,j}$ higher than the competence of a random classifier $\frac{1}{L}$ are selected to compose the ensemble C'

[3] The Matlab code for this technique is available at: http://www.mathworks.com/matlabcentral/fileexchange/28391-a-probabilistic-model-of-classifier-competence.

4.13. Dynamic ensemble selection performance (DES-P)

Proposed by Woloszynski et al. [34], this method works as follows: First, the local performance of a base classifier c_i is calculated using the region of competence θ_j. The competence of the base classifier is then calculated by the difference between the accuracy of the base classifier c_i, in the region of competence θ_j (denoted by $\hat{P}(c_i \mid \theta_j)$), and the performance of the random classifier, that is, the classification model that randomly chooses a class with equal probabilities. For a classification problem with L classes, the performance of the random classifier is ($RC = \frac{1}{L}$). Hence, the competence level $\delta_{i,j}$ in this technique is calculated according to Eq. (10).

$$\delta_{i,j} = \hat{P}(c_i \mid \theta_j) - \frac{1}{L} \tag{10}$$

The base classifiers with a positive value of $\delta_{i,j}$, i.e., that obtain a local accuracy higher than the random classifier, are selected to compose the ensemble C'.

4.14. Kullback–Leibler divergence (DES-KL)

The DES-KL method [34] measures the competence of the base classifiers from an information theory perspective. For each instance, $\mathbf{x}_k$, from the whole DSEL, the source of competence C_{src} is calculated as the Kullback–Leibler (KL) divergence between the uniform distribution and the vector of class supports, $S(\mathbf{x}_k) = \{S_1(\mathbf{x}_k), \ldots, S_L(\mathbf{x}_k)\}$, estimated by the base classifier, c_i. Then, a Gaussian potential function is applied to weight the source of competence based on the Euclidean distance between $\mathbf{x}_k$ and the query sample $\mathbf{x}_j$ (Eq. (11)).

$$\delta_{i,j} = \sum_{\mathbf{x}_k \in DSEL} C_{src}\, exp(-d(\mathbf{x}_k, \mathbf{x}_j)^2) \tag{11}$$

Since the KL divergence is always positive, the signal of C_{src} is set as positive if the base classifier c_i predicted the correct label for the instance $\mathbf{x}_k$, and negative otherwise. After computing the KL divergence for all samples in DSEL, the base classifiers, c_i, with a positive value of $\delta_{i,j}$ are selected to compose the EoC, C'.

4.15. K-Nearest Output Profiles (KNOP)

The K-Nearest Output Profiles (KNOP) technique [106] works similarly to the KNORA-U technique, with the difference being that KNORA-U works in the feature space, while KNOP works in the decision space. First, the output profiles' transformation is applied over the input $\mathbf{x}_j$, giving its output profile $\tilde{\mathbf{x}}_j$. Then, the similarity between $\tilde{\mathbf{x}}_j$ and the output profiles from the dynamic selection dataset, is computed and stored in the set, ϕ_j. Similarly to the KNORA-U rule, each time a base classifier performs a correct prediction, for a sample belonging to ϕ_j, it gains one vote. The votes obtained by all base classifiers are aggregated to obtain the ensemble decision.

4.16. META-DES

The META-DES framework is based on the assumption that the dynamic ensemble selection problem can be considered as a meta-problem [124]. This meta-problem uses different criteria regarding the behavior of a base classifier c_i, in order to decide whether it is competent enough to classify a given test sample $\mathbf{x}_j$. The meta-problem is defined as follows [27]:

- The **meta-classes** are either "competent" (1) or "incompetent" (0) to classify $\mathbf{x}_j$.
- Each set of **meta-features** f_i corresponds to a different criterion for measuring the level of competence of a base classifier.
- The meta-features are encoded into a **meta-features vector** $v_{i,j}$.
- A **meta-classifier** λ is trained based on the meta-features $v_{i,j}$ to predict whether or not c_i will achieve the correct prediction for $\mathbf{x}_j$, i.e., if it is competent enough to classify $\mathbf{x}_j$.

In other words, a meta-classifier, λ, is trained, to predict whether a base classifier c_i is competent enough to classify, a given test sample $\mathbf{x}_j$. After the pool of classifiers is generated, the framework performs a meta-training stage, in which, the meta-features are extracted from each instance belonging to the training and the dynamic selection dataset (DSEL). Then, the extracted meta-features are used to train the meta-classifier λ. Thus, the advantage of using meta-learning is that multiple criteria can be encoded as different sets of meta-features in order to estimate the competence level of the base classifiers. In addition, the selection rule is learned by the meta-classifier using the meta-features extracted from the training data.

When an unknown sample, $\mathbf{x}_j$, is presented to the system, the meta-features are calculated according to $\mathbf{x}_j$, and presented to the meta-classifier. The competence level $\delta_{i,j}$ of the base classifier c_i for the classification of $\mathbf{x}_j$ is estimated by the meta-classifier.

The impact of the meta-classifier as well as a variation of the META-DES framework was evaluated in [97]. Four classifier models for the meta-classifier were evaluated: MLP, SVM, a Naive Bayes (NB) classifier and Random Forest (RF). Experimental results demonstrated that the performance of the MLP, SVM and NB were statistically equivalent. However, since an NB obtained the highest number of wins in the experimental analysis, it was selected as the overall best classification model for the meta-classifier. Three versions of the META-DES framework were evaluated: Dynamic selection, Dynamic weighting and Hybrid. In the dynamic selection approach, only the classifiers that attain a certain level of competence are used to classify a given query sample. In the dynamic weighting approach, all classifiers in the pool are used for classification; however, their decisions are weighted based on their estimated competence levels. Classifiers that attain a higher level of competence for the classification of the given query sample have a greater impact on the final decision. The hybrid approach conducts both steps: first, the base classifiers that obtained a certain competence level are selected; then, their outputs are aggregated using a weighted majority voting scheme based on their respective competence level.

4.17. META-DES.Oracle (META-DES.O)

An improvement to the META-DES was proposed in [99]. In this new version of the framework, a total of 15 sets of meta-features were considered. Following that, a meta-features selection scheme using a Binary Particle Swarm Optimization (BPSO) was conducted in order to optimize the performance of the meta-classifier, λ. The difference between the level of competence estimated by the meta-classifier and that estimated by the Oracle was used as the fitness function for the BPSO meta-features selection scheme, so that the difference between the behavior of the meta-classifier and that of the Oracle in estimating the competence level of the base classifiers was minimized. The new framework was called META-DES.Oracle since it was based on the Oracle definition.

Experimental results conducted in [99] demonstrated that the META-DES.Oracle dominates the classification results compared against previous DES techniques. Its performance is statistically better when compared to any of the 10 state-of-the-art techniques, including the META-DES. This can be explained by two factors: state-of-the-art DES techniques are based only on one criterion to estimate the competence of the base classifier; this criterion could be local accuracy, ranking, probabilistic models, etc. In addition, through the BPSO meta-features selection scheme, only the meta-features that are relevant for the given classification problem are selected and used for the training of the meta-classifier λ.

4.18. Dynamic Selection on Complexity (DSOC)

Brun et al. [123] proposed an interesting dynamic classifier selection approach which takes into account data complexity measures from Ho and Basu [132], together with the local accuracy estimates of the base classifiers, to perform dynamic selection. The proposed system is called Dynamic selection on complexity (DSOC).

DSOC aims to select the base classifier c_i that presents not only a high local performance, but also that was trained using a data distribution that presents similar complexity measures, regarding the shape of the decision boundary and the overlap between the classes, extracted from the neighborhood of the query sample $\mathbf{x}_j$. Three complexity measures were considered: The Fisher's discriminant ratio (F1), the intra/inter class ratio (N2) and the non-linearity of the 1-NN classifier (N4). More details about these complexity metrics are given in [132].

The base classifier is selected taking into consideration three features:

- f1: The similarity in terms of complexity, which is measured by the differences in F1, N2 and N4 between the training data DS_i and the region of competence of θ_j.
- f2: The distance of $\mathbf{x}_i$ and the centroid of its class, predicted by the base classifier c_i in its training data DS_i.
- f3: The local classifier accuracy based on the region of competence θ_j.

where f1 and f2 are related to the concept of complexity, and can be seen as measures of pertinence of the base classifier, while f3 represents the local competence of the base classifier. The three features are combined using Eq. (12). The base classifier with the highest value of $\delta_{i,j}$ is selected for the classification of $\mathbf{x}_j$.

$$\delta_{i,j} = (1 - f_{1i}) + (1 - f_{2i}) + f_{3i} \tag{12}$$

One important aspect covered in [123] is the size of the region of competence θ_j, since a large number of samples from at least two different classes is required in order to extract the data complexity features. This is different from the other dynamic selection methods, which usually consider a smaller neighborhood [27]. For the size of the neighborhood, the authors varied its size between 20 and 50, considering 30 UCI classification datasets. Ultimately, the size was set to 30, which presented the best performance.

Although a significant boost in classification performance from DSOC was observed for some datasets, its overall performance was not significantly superior when compared to the baseline DS techniques. A study was conducted in order to understand why the method works for some datasets, and not for others. When there was an overlap between the complexity measured for the neighborhood of the training sample and that for the training bootstraps, the contributions of this approach were more evident. The authors pointed out that this problem is due to the classifier generation approach. As they used the Bagging technique, each bootstrap used to train a base classifier was chosen randomly. Therefore, there was no guarantee that each bootstrap would have a significant difference in relation to the complexity measures. Hence, the authors mentioned that future work should include the use of a different pool generation model in order to obtain a better coverage of the problem space. Thus, it will be more likely to find a classifier that was trained with a bootstrap with similar complexity measures to the neighborhood of the test instance.

4.19. Dynamic selection in different contexts

4.19.1. One-Versus-One decomposition (OVO)

Another context where dynamic selection has recently shown a lot of promise is in One-Versus-One (OVO) decomposition strategies [133]. OVO works by dividing a multi-class classification problem into as many binary problems as there are combinations between pairs of classes [133]. Each base classifier is trained solely to distinguish between one pair of classes. When a new query sample is presented for classification, the outputs of all base classifiers are combined to predict its label. However, since each base classifier is only trained for a pair of classes, the majority of base classifiers might not even be trained for the corresponding class, and their decision may hinder the performance of the system. This is called "the non-competent classifier" problem, which is a crucial problem in OVO strategies [133].

Dynamic selection represents an interesting way of solving this non-competent classifier problem, since it provides a methodology for estimating the competence of the base classifiers on-the-fly, while thus, avoiding non-competent classifiers which may hinder the system decision during the generalization phase. Five dynamic selection techniques were proposed in this context:

- The Dynamic-OVO strategy [40] was the first use of DS in OVO. This method works by applying the K-Nearest Neighbors algorithm to find the neighborhood of the query sample. Then, only the classifiers that were trained considering the classes presented in this neighborhood are used in the combination scheme. Hence, a base classifier is considered competent if it was trained with any of the classes presented in the neighborhood of the query sample.
- The Distance-based Relative Competence Weighting combination (DRCW-OVO) [134] is an updated version of the Dynamic-OVO [40] to further reduce the impact of non-competent classifiers by using a weighting mechanism. The outputs of the selected classifiers are weighted depending on the closeness of the query instance to the nearest neighbors of each class in the problem. The greater the distance, the lower the weight of the classifier, and vice versa.
- The DYNOVO technique [39] performs dynamic classifier selection in each sub-problem of the OVO decomposition, and selects the best base classifiers to classify the query sample. In this case, the Overall Local Accuracy (OLA) [31] was used in each sub-problem of the OVO-decomposition. Moreover, the authors also considered the use of the K-Nearest Neighbors Equality (K-NNE) in order to estimate the region of competence.
- Zhang et al. [41] recently proposed the OVO-DES-P, which combines the OVO decomposition with the DES-Performance technique (Section 4.13). The system works by training an ensemble of classifiers for each sub-problem in the OVO decomposition. During the generalization phase, the DES-P technique is applied to select an EoC with the most competent classifiers in each binary classification problem of the OVO decomposition.
- The DRCW-OVO-DES-P [41] is a combination of the OVO-DES-P with the DRCW-OVO. The OVO-DES-P is applied to estimate the competence level of each base classifier, and select an EoC with the most competent classifier in each binary classification problem of the OVO decomposition. Then, the distance relative weighting method (DRCW-OVO) is applied to weight the decision of the base classifiers depending on the closeness of the query instance to the nearest neighbors of each class in the problem. The greater the distance, the lower the weight of the classifier, and vice versa. The selected base classifiers are combined using the weighted majority voting scheme.

Table 2
Application using DS.

Application	**Pool generation**	DS methods	Ref.
Credit Scoring	Bagging and Boosting	OLA, LCA, KNORA-E, KNORA-U	[9,10]
Customer Classification	Different training sets	LCA, OLA, KNORA-E	[126]
Music Classification	Different feature sets	KNORA-E, KNORA-U, OLA, LCA	[8,98]
Watch list screening	Different feature sets	Distance based DS	[7,43,44]
Face recognition	Different feature sets	OLA, LCA, Distance based DS	[44,45]
Handwriting recognition	Bagging	KNORA-E, KNORA-U, KNOP, OLA, LCA	[26,98,113]
Signature verification	Random subspaces	KNOP	[42,121]
Forest species	Different feature sets	MCB	[137]
Remote sensor images	Bagging	MLA	[32]
Time series forecasting	Heterogeneous classifiers	MCB	[138]
Antibiotic Resistance	Bagging	LCA	[37]
Bioprosthetic-hand	Heterogeneous classifiers	DES-PRC	[139–141]

Classification results demonstrated that the DRCW-OVO-DES-P strategy outperformed the other dynamic selection approaches in the context of OVO decomposition classification.

4.19.2. One-Class Classification (OCC)

One-class classification (OCC) is one of the most difficult problems in machine learning. It is based on the assumption that during the training stage, only objects originating from one class are present, with no access to counterexamples, making it difficult to train an efficient classifier, since there is no data available to properly estimate its parameters [35,135]. It can therefore be considered as an ill-posed problem. As has been mentioned in several works [27,99,106], dynamic selection techniques outperform strong classification models, such as SVMs when dealing with ill-posed problems. Therefore, the application of DS in the context of OCC is expected to improve generalization performance.

As reported in [35], dynamic selection techniques work well in this context because during the training stage it is possible to generate a diverse pool of classifiers. The flexibility of DS techniques then allows the most competent classifier to be selected to classify each new test sample. Three One-Class DCS (OCDCS) methods were proposed by adapting three DCS methods proposed in [136]: One-Class Entropy Measure Selection (OCDCS-EM), which is based on the entropy measure, One-Class Minimum Difference Selection (OCDCS-MD), and the Fuzzy Competence Selection (OCDCS-FC). These three OCDCS methods are based on the potential function model to weight the decisions obtained by each sample in the reference set according to its distance to the given query instance.

An experimental evaluation demonstrated that the OCDCS-EM works well for small datasets, while the Fuzzy Competence (OCDCS-FC) is the best choice for large datasets. However, only DCS methods were proposed for OCC. Since DES usually presents better results than DCS, it is reasonable to think that a next step in this direction would be to adapt DES techniques for one-class classification problems.

5. Applications

In this section, we present a review of real-world applications using dynamic selection techniques. Moreover, we also discuss how the authors adapt traditional DS techniques to the intrinsic characteristics of their applications; these includes aspects such as imbalanced distributions in customer classification and credit scoring [126] and the lack of validation samples in face recognition applications [7,45].

Table 2 lists several real-world applications of DS techniques. Based on usage statistics respecting the use of DS techniques in different applications, the KNORA-E and KNORA-U methods are the most commonly used in different applications. This may be explained by the fact these techniques represent a good trade-off between simplicity and classification accuracy. Moreover, we can see that DES methods are more popular than DCS ones, which may account for the many works which point out that DES techniques usually achieve higher classification accuracy. Another interesting fact is that the majority of the related works apply DS techniques for a pool of classifiers trained using different feature spaces [7,8,43,45,137].

5.1. Credit scoring

Credit scoring is one of the most studied problems in pattern recognition. Its difficulty comes from the observation that the problem is often heavily imbalanced, since there are way fewer samples from customers with poor credit scores [126]. Moreover, high accuracy in this problem is very important, since even just 1% in improvement in classification accuracy should greatly increase the profits of financial institutions [142].

Many works consider the use of DS techniques for the credit scoring problem [9,10,126]. In [126], the authors proposed a dynamic ensemble selection system that takes into account the cost of misclassifications of each class. The proposed approach, called Dynamic Classifier Ensemble for Imbalanced Distributions (DCEID), uses cost-sensitive versions of the LCA and OLA techniques in dealing with the imbalanced nature of this problem. Another recent use of DS techniques for credit scoring was proposed by Xiao et al. [10]. The authors proposed a new ensemble generation method to increase the diversity between the members of the pool. Then, two DCS and two DES schemes were considered: LCA and OLA as DCS, KNORA-E and KNORA-U as DES methods. Similar to previous works, the authors concluded that DES techniques presented the best classification performance.

An extensive comparison between DS techniques and other classification schemes considering eight credit scoring datasets was conducted by Lessmann et al. [9]. However, the dynamic selection techniques considered in their analysis did not improve the classification accuracy when compared to traditional credit scoring approaches.

5.2. Music genre classification

In [8], the authors investigated the use of dynamic ensemble selection techniques for music genre classification. In their solution, first, a pool of weak classifiers was generated, trained with distinct segments of the audio signal and different feature extraction methods. A total of 13 distinct feature extraction methods were considered, each corresponding to a different musical aspect, such as harmony, timbre and rhythm. In the generalization phase, the KNORA-E and KNORA-U techniques were considered for the classification of each unknown music sample. In the experiments conducted using the Latin Music Dataset [143], the use of DES achieved a recognition accuracy of about 70%, which significantly improved the classification performance, as compared to the 54% seen with of the best single classifier model.

5.3. Image recognition

5.3.1. Face recognition and watch list screening

The face recognition problem presents an interesting use of dynamic selection techniques, since in such applications, we do not have enough data available to have a dynamic selection dataset (DSEL). Thus, the authors in [7,43] adapted the dynamic selection techniques considering just the query sample and a pool of SVM classifiers trained using different feature extraction methods. The algorithm is based on the distance between the query sample and the support vectors of the SVM classifier for the negative samples (i.e., face images belonging to different users). The higher the distance between the support vectors, the more competent the SVM-Feature Extraction pair is. Therefore, the system is not only selecting the SVM, but also choosing which feature extraction method is more suitable for the classification of an unknown face image.

5.3.2. Handwriting recognition

Dynamic ensemble and classifier selection techniques have been used to solve several image recognition problems. Ko et al. [26] used the KNORA-E and KNORA-U techniques for the NIST SD19 handwritten recognition dataset. Although the performance of those techniques did not present the highest classification performance, their classification results were among the best achieved for this problem to that point. In addition, the proposed KNORA techniques outperformed static ensemble selection schemes such as GA and MVE [46].

Furthermore, in [113], the authors applied the KNOP technique for four handwriting recognition datasets: NIST Letters, NIST Digits, Japanese Vowels, and Arabic Digits. Experiments demonstrated that the proposed approach outperformed the state-of-the art methods for the Arabic and Japanese datasets. The accuracy achieved in the NIST Letters and Digits datasets were comparable to the state-of-the-art results obtained by SVM and MLP classifiers [144,145]. However, in another publication, Cavalin et al. [106] demonstrated that the use of dynamic selection outperforms both SVM and MLP classifiers when the training dataset is small (less than 5000 examples).

5.3.3. Signature verification

Batista et al. [42] evaluated four dynamic selection techniques for the signature verification problem: KNORA-UNION, KNORA-ELIMINATE and their corresponding versions using output profile, namely, OP-ELIMINATE and OP-UNION. The system was based on an ensemble of Hidden Markov Models (HMMs) used as feature extractors from the signature image. Each HMM was trained using different numbers of states and codebook sizes in order to learn signatures from different levels of perception. The features extracted using HMMs were merged in a feature vector. For each writer, the random subspace method was used to train a pool of 100 Gaussian SVM classifiers.

The proposed approach was applied to two well-known signature verification datasets: GPDS [146] and Brazilian signature composed of random, simple and skilled forgeries. Experimental results demonstrated that dynamic selection can significantly reduce the overall error rates as compared to other combination methods. In the majority of the experiments conducted, the OP-ELIMINATE presented the best performance. It must be noted however, that the OP-UNION method worked better when the SVM classifiers were trained with a limited number of signature samples, since classifiers trained with fewer signatures are less accurate. Thus, more classifiers are needed to form a robust EoC. Nevertheless, in all cases, dynamic selection provided a better classification performance when compared to static ensemble techniques.

5.3.4. Forest species recognition

Martins et al. [137] used the DS for the forest species recognition problem. The system was based on multiple features extraction methods, such as texture (Gabor filters and Local Binary Patterns), as well as key point-based features (SIFT and SURF) to generate a diverse pool of classifiers. Then, several static and dynamic selection methods were evaluated. The MCB technique presented the best result, achieving a 93.03% accuracy.

5.4. Time series forecasting

Sergio et al. [138] proposed a dynamic selection of regressors for the time series forecasting. An adaptation of the MCB technique for regression was used to perform the dynamic selection steps. The proposed system was called Dynamic Selection of Forecast Combiners (DS-FC). The pool of classifiers was composed of four regression models: a feed-forward neural network with one hidden layer, a feed-forward neural network with two hidden layers, a deep belief network (DBN) with two hidden layers and a support vector machine for regression (SVR).

The proposed DS-FC was used to forecast eight time series with chaotic behavior considering short and long term series. The proposed dynamic selection scheme outperformed static combination schemes in six out of eight time series. Moreover, it also presented better results than most of the state-of-the-art time series forecasting techniques.

5.5. Biomedical

5.5.1. Antibiotic resistance

Tsymbal et al. [37] proposed a dynamic ensemble method to tackle the problem of antibiotic resistance. This is a typical example of a changing environment (concept drift), where pathogen sensitivity may change over time as new pathogen strains develop resistance to previously used antibiotics. A dynamic classifier method was proposed to deal with this problem, with the authors considering a variation of the LCA technique, in which the distance between the neighbors are also taken into account.

Three dynamic approaches were considered: Dynamic Voting (DV), Dynamic Selection (DS) and Dynamic Voting with Selection (DVS). The methodology was evaluated considering gradual and abrupt drift scenarios. Experimental results demonstrated that the approaches using dynamic selection presented the best performance. Furthermore, the dynamic approaches always obtained a better result than the best base classifier model and ensemble technique.

5.5.2. Bioprosthetic hand

Kurzynski et al. [139,147] proposed a dynamic ensemble selection system for the recognition of electromyography (EMG) signals for the control of a bioprosthetic hand. The proposed solution was based on the estimation of the classifiers competence using the probabilistic randomized reference classifier proposed in [25]. The pool of classifiers consisted of seven different classification models, including linear and quadratic discriminant classifiers (LDC and QDC) and MLP neural networks.

Moreover, in [140,141], the authors proposed a method for the control of a bioprosthetic hand using a two-stage MCS and DES for the recognition of EMG and mechanomiographic (MMG) signals indicating the patient's movement intention. Additionally, feedback information coming from bioprosthesis sensors was used to calibrate the competence of the base classifiers estimated using the

Table 3
Summary of the 30 datasets used in the experiments.

Database	No. of instances	Dimensionality	No. of classes	Source
Adult	48,842	14	2	UCI
Banana	1000	2	2	PRTOOLS
Blood transfusion	748	4	2	UCI
Breast (WDBC)	568	30	2	UCI
Cardiotocography (CTG)	2126	21	3	UCI
Ecoli	336	7	8	UCI
Steel Plate Faults	1941	27	7	UCI
Glass	214	9	6	UCI
German credit	1000	20	2	STATLOG
Haberman's Survival	306	3	2	UCI
Heart	270	13	2	STATLOG
ILPD	583	10	2	UCI
Ionosphere	315	34	2	UCI
Laryngeal1	213	16	2	LKC
Laryngeal3	353	16	3	LKC
Lithuanian	1000	2	2	PRTOOLS
Liver Disorders	345	6	2	UCI
MAGIC Gamma Telescope	19,020	10	2	KEEL
Mammographic	961	5	2	KEEL
Monk2	4322	6	2	KEEL
Phoneme	5404	6	2	ELENA
Pima	768	8	2	UCI
Satimage	6435	19	7	STATLOG
Sonar	208	60	2	UCI
Thyroid	215	5	3	LKC
Vehicle	846	18	4	STATLOG
Vertebral Column	310	6	2	UCI
WDG V1	5000	21	3	UCI
Weaning	302	17	2	LKC
Wine	178	13	3	UCI

RRC technique during the operation phase. The two-stage technique developed provided the state-of-the-art results for control of the bioprosthetic hand, considering several types of movements, such as hook, power and pinch.

6. Comparative study

In this section, we present an empirical comparison between 18 state-of-the-art techniques under the same experimental protocol. First, we compare the results of each DS techniques (Section 6.3). The DS techniques are then also compared against the baseline methods, namely, (1) Static Selection (SS), that is, the selection of an EoC during the training stage, and its combination using a majority voting scheme [46]; (2) single best (SB), which corresponds to the performance of the best classifier in the pool according to the validation data, and (3) majority voting (MV), which corresponds to the majority voting combination of all classifiers in the pool without any pre-selection of classifiers. We also compare the performance of DS techniques with the best classification models according to [4] (Section 6.4).

We evaluate a wide range of DS techniques covering all points of the proposed DS taxonomy: classifier and ensemble selection techniques, techniques based on K-NN, clustering, and competence maps through a potential function (explained in Section 3.1), as well as different information sources (criteria), to estimate the competence level of the base classifiers, such as, local accuracy, ranking, probabilistic models and oracle (explained in Section 3.2).

For dynamic classifier selection (DCS), the following techniques were evaluated: Local Class Accuracy (LCA) [31], Overall Local Accuracy (OLA) [31], Modified Local Accuracy (MLA) [32], Modified Classifier Ranking (RANK) [31,33], Multiple Classifier Behavior (MCB) [112], A Priori [101,129] , A Posteriori [101,129] and the Dynamic Selection on Complexity (DSOC). For dynamic ensemble selection, the following techniques were considered: K-Nearest Oracles Eliminate (KNORA-E) [26], K-Nearest Oracles Union (KNORA-U) [106], Randomized Reference Classifier (DES-RRC) [105], K-Nearest Output Profiles (KNOP) [106,113], Dynamic Ensemble Selection Performance (DES-P) [34], Dynamic Ensemble Selection Kullback–Leibler (DES-KL) [34], DES Clustering [30], DES KNN [30], Meta Learning for Dynamic Selection (META-DES) [27] and META-DES.Oracle [99]. All the DS methods considered in this evaluation are detailed in Section 4 and summarized in Table 1.

6.1. Datasets

The comparative study was performed using a test bed composed of 30 classification problems proposed in [27]. Sixteen datasets were taken from the UCI machine learning repository [148], four from the STATLOG project [149], four from the Knowledge Extraction based on Evolutionary Learning (KEEL) repository [150], four from the Ludmila Kuncheva Collection of real medical data [151], and two artificial datasets generated with the Matlab PRTOOLS toolbox [152].

6.2. Experimental setup

To ensure a fair comparison of the results obtained in this analysis with previous results from the DES literature, the same experimental setup from previous works [27,97] was used. For each dataset, the experiments were carried out using 20 replications. For each replication, the datasets were randomly divided as follows: 50% for training, 25% for the dynamic selection dataset (DSEL) and 25% for the test set. The divisions were performed while maintaining the prior probabilities of each class. Similar to our previous works [27,97], the pool of classifiers C was composed of 100 Perceptrons generated using the Bagging technique [50]. The size of the region of competence (neighborhood size) K was equally set to 7 for all the techniques based on K-NN. Table 3.

For the DES-KMEANS and DES-KNN, the number of base classifiers selected using accuracy (N) and diversity (J) was set to 30% of the whole pool for both cases, as suggested in [30,116]. In addition, the number of clusters in the DES-KMEANS was set to 6 following

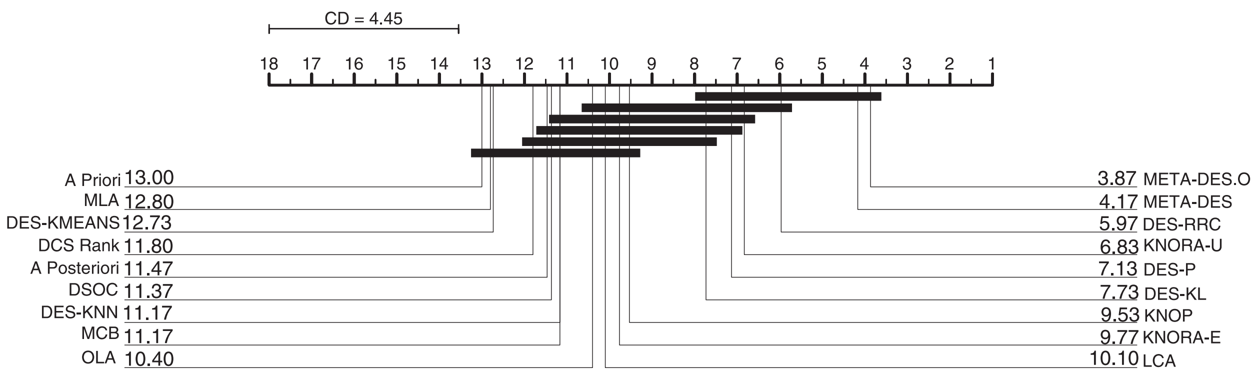


Fig. 5. Average rank of the 18 dynamic selection methods over the 30 datasets. The best algorithm is the one presenting the lowest average rank. Techniques in which the difference in average ranks is lower than the critical difference are connected by a black bar.

Table 4
Overall results considering the 30 classification datasets. The average ranks and accuracy for each DS technique are presented. Standard deviation is presented in parenthesis.

DS method	Avg. rank	DS method	Mean accuracy
META-DES.O	3.87(3.54)	META-DES.O	83.92(9.13)
META-DES	4.17(2.98)	META-DES	83.24(8.94)
DES-RRC	5.97(4.66)	DES-P	82.26(9.26)
KNORA-U	6.83(4.11)	DES-RRC	82.11(8.76)
DES-P	7.13(3.69)	KNORA-U	81.69(9.82)
DES-KL	7.73(4.92)	DES-KL	81.52(8.77)
KNOP	9.53(3.98)	KNOP	80.81(8.92)
KNORA-E	9.77(3.88)	KNORA-E	80.36(10.75)
LCA	10.10(4.66)	OLA	79.87(10.67)
OLA	10.40(4.95)	DCS Rank	79.69(10.38)
MCB	11.17(4.74)	LCA	79.57(9.84)
DES-KNN	11.17(4.40)	MCB	79.56(9.70)
DSOC	11.37(5.74)	DSOC	79.33(9.44)
A Posteriori	11.47(5.56)	DES-KNN	79.29(10.23)
DCS Rank	11.80(4.20)	A Priori	78.57(11.18)
DES-KMEANS	12.73(3.84)	DES-KMEANS	78.49(10.40)
MLA	12.80(4.60)	A Posteriori	78.14(11.53)
A Priori	13.00(4.53)	MLA	77.34(9.78)

the experiments conducted by Soares et al. [30]. Lastly, the values of the hyper-parameters K_p and h_c for the META-DES framework were set to 5 and 80% according to the results presented in [27]. Moreover, the BPSO optimization scheme of the META-DES.Oracle was conducted using the V-Shaped transfer function [99,153].

6.3. Comparison of dynamic selection techniques

The Friedman rank [154] test was used for the statistical comparison of the dynamic selection techniques over the 30 classification datasets. The average ranks were calculated as follows: For each dataset, the method that achieved the best performance received rank 1, the second best rank 2, and so forth. In case of a tie, i.e., two methods presented the same classification accuracy for the dataset, their average ranks were summed and divided by two. The average rank was then obtained, considering all datasets. The best performing algorithm was the one presenting the lowest average rank. Next, the critical difference (CD) value was calculated using the Bonferonni–Dunn post-hoc test recommended in [155]. The performance of two techniques was deemed statistically different when their difference in average rank was higher than the critical difference CD. Table 4 presents the average accuracy of all DS methods as well as their average ranking.

We use the critical difference diagram proposed in [155] in order to have a visual illustration of the statistical test. The CD diagram with the results of the Bonferonni–Dunn post-hoc test is shown in Fig. 5. Techniques in which the difference in average ranks is lower than the critical difference are connected by a black bar (i.e., the results are statistically equivalent according to the ranking analysis).

Based on the ranking analysis, we can clearly see that DES techniques outperform DCS ones. Among the top 10 techniques, 8 are DES. The only DES methods that did not achieve a lowest average rank in comparison with the DCS ones were the DES-KNN and DES-Clustering. Interestingly, these two techniques take into account diversity measures to increase the diversity in the EoC, after the most competent classifiers are selected. This result may indicate that adding diversity to EoC does not provide classification benefits at the instance level, i.e., for the classification of a single instance. As reported in [156], we should promote the consensus in the EoC, for the classification of a single query $\mathbf{x}_j$, rather than diversity. Our experimental analysis supports this hypothesis.

The six techniques with the lowest average rankings (DES-P, DES-KL, KNORA-U, DES-RRC, META-DES and META-DES.Oracle) were considered equivalent in this analysis. However, one problem with the ranking analysis is that the result of the comparison between two techniques changes according to the other techniques that were considered in the test. This problem can cause several type I errors according to [157]. For this reason, we performed a new test considering only these top 6 techniques. The CD diagram considering only the top 6 techniques is shown in Fig. 6. Moreover, the classification results obtained for the 30 datasets by the top 6 DS techniques are presented in Table 5.

Furthermore, for a better comparison of the top DS techniques, we conducted two additional analysis: an $n \times n$ comparison considering the hypothesis of equality between all existing pairs of algorithms using Bergman–Hommel procedure [158–160], and a pairwise test using the Wilcoxon Sign Test as recommended by Benavoli et al. [157]. The results of both tests for each pairwise comparison are presented in Table 6. Hypotheses that are rejected at a $\alpha = \{0.1, 0.05, 0.01\}$ are marked with a •, ••, and •••, respectively. The Bergmann–Hommel test was conducted using the JAVA code published by Garcia and Herrera [159][4].

Based on the results, we can see that the two versions of the META-DES framework and the Randomized Reference Classifier (DES-RCC) presented the best results. According to the Bergmann–Hommel test, these three techniques are statistically equivalent. However, the Wilcoxon Sign test shows that both the META-DES and the META-DES.Oracle outperform the DES-RRC method, with a level of significance $\alpha = 0.01$.

[4] Code available at http://sci2s.ugr.es/keel/multipleTest.zip.

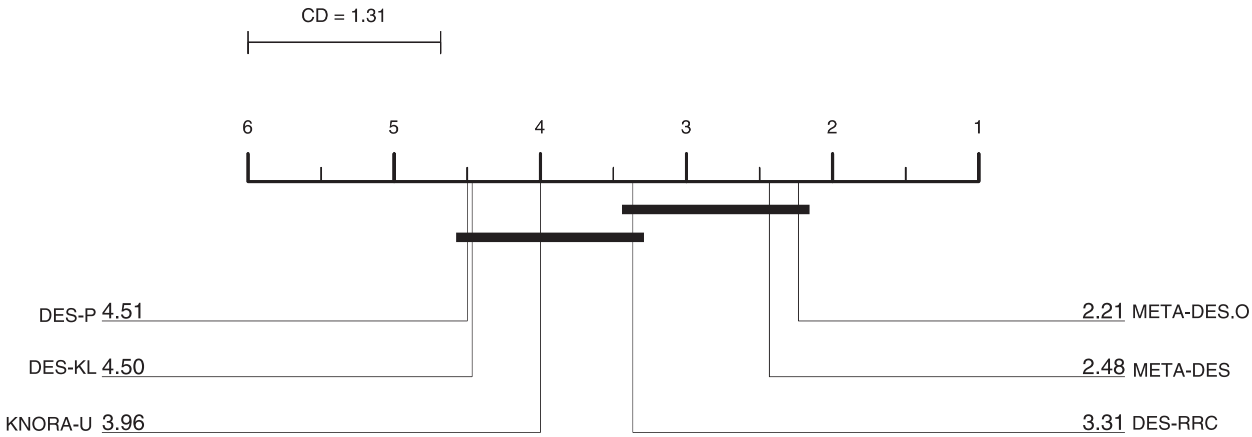


Fig. 6. Average rank of the top six dynamic selection methods over the 30 datasets.

Table 5
Mean and standard deviation results for the top 6 DS techniques. The best results for each dataset are highlighted in bold.

Dataset	DES-RRC	META-DES	META-DES.O	DES-KL	DES-P	KNORA-U
Pima	77.64(2.73)	**79.03(2.24)**	77.53(2.24)	77.97(2.64)	76.87(1.87)	78.84(2.18)
Liver	68.01(4.14)	70.08(3.49)	**72.02(4.72)**	67.11(5.62)	67.46(3.84)	61.29(3.76)
Breast	96.94(0.61)	97.40(1.07)	96.71(0.86)	97.13(0.59)	96.78(0.78)	**97.41(1.02)**
Blood	78.02(1.41)	79.14(1.03)	**79.38(1.76)**	78.83(1.09)	77.72(1.53)	79.25(3.36)
Banana	86.56(1.76)	91.78(2.68)	**94.54(1.16)**	85.58(2.40)	93.61(1.80)	92.40(2.87)
Vehicle	**83.34(1.81)**	82.75(1.70)	82.87(1.64)	82.99(1.74)	82.85(1.97)	83.12(1.70)
Lithuanian	85.34(1.57)	93.18(1.32)	94.97(2.00)	83.50(2.82)	91.90(3.49)	**95.63(2.64)**
Sonar	80.77(5.09)	80.55(5.39)	**81.63(3.90)**	78.15(6.28)	79.49(6.66)	77.34(1.94)
Ionosphere	88.80(2.48)	**89.94(1.96)**	89.94(1.97)	88.42(2.78)	88.42(2.62)	88.42(1.67)
Wine	98.77(1.57)	99.25(1.11)	**99.52(1.11)**	98.26(2.45)	98.48(2.30)	98.03(1.62)
Haberman	75.98(2.50)	**76.71(1.86)**	72.03(2.67)	75.04(2.17)	75.57(2.54)	74.51(2.27)
CTG	85.41(1.02)	84.62(1.08)	**86.37(1.10)**	84.98(1.39)	85.11(1.39)	86.22(2.20)
Vertebral	86.76(3.68)	86.89(2.46)	84.90(5.33)	84.19(6.40)	86.76(3.47)	**88.03(2.24)**
Faults	68.20(0.95)	67.21(1.20)	**69.32(1.18)**	67.91(1.29)	68.03(1.36)	68.58(1.98)
WDVG1	84.63(0.48)	84.56(0.36)	**84.72(0.49)**	84.61(0.53)	84.59(0.54)	84.18(1.10)
Ecoli	80.66(3.58)	77.25(3.52)	**81.57(3.47)**	79.95(4.30)	79.83(4.26)	77.12(3.41)
GLASS	66.04(4.23)	**66.87(2.99)**	66.46(4.22)	63.32(4.27)	63.13(5.30)	62.05(2.88)
ILPD	68.58(1.89)	69.40(1.64)	69.79(3.15)	**69.94(2.49)**	68.02(2.21)	69.86(1.58)
Adult	87.16(1.53)	87.15(2.43)	**87.74(2.04)**	86.06(2.90)	87.10(2.76)	80.21(2.26)
Weaning	78.96(3.29)	79.00(3.78)	81.73(3.14)	78.74(4.44)	78.08(4.07)	**82.02(3.65)**
Laryngeal1	87.21(3.79)	79.67(3.78)	**87.42(2.98)**	86.54(4.25)	86.35(3.82)	82.38(4.45)
Thyroid	**97.61(0.96)**	96.78(0.87)	96.99(2.14)	**97.04(1.09)**	96.98(1.12)	96.71(1.89)
Laryngeal3	**72.74(1.87)**	72.65(2.17)	73.67(0.75)	72.67(2.37)	73.34(2.81)	72.36(1.25)
German	75.83(2.36)	75.55(1.31)	**76.58(1.99)**	73.88(2.15)	74.72(3.50)	73.16(1.80)
Heart	83.99(3.64)	84.80(3.36)	**86.44(3.38)**	82.83(4.13)	83.27(3.68)	84.15(4.05)
Segmentation	96.38(0.75)	96.21(0.87)	**96.65(0.83)**	96.20(0.89)	96.22(0.87)	96.64(1.07)
Phoneme	74.65(1.55)	80.35(2.58)	**85.05(1.08)**	77.13(1.32)	81.64(0.53)	79.94(3.33)
Monk2	80.98(2.58)	83.24(2.19)	**94.45(1.88)**	80.85(2.68)	79.93(2.57)	77.88(4.25)
Mammographic	**85.00(1.32)**	84.82(1.55)	80.72(2.56)	84.12(2.26)	84.98(1.86)	82.91(2.27)
Magic	**86.20(1.84)**	84.35(3.27)	86.02(2.20)	83.56(1.22)	83.54(1.34)	82.99(1.25)
Average	82.11(8.76)	83.20(8.94)	**83.92(9.13)**	81.52(8.77)	81.29(9.08)	81.69(9.82)
Avg. rank	3.31(1.54)	2.48(0.80)	**2.21(1.13)**	4.50(0.98)	4.51(1.06)	3.96(1.72)

6.4. Comparison with different classification approaches

In this section, we compare the results obtained by DS techniques against monolithic classifier models. The objective of this study is to determine whether the performance of DS methods in relation to the best off-the-shelf classifiers. Three single classifier models were considered: Multi-Layer Perceptron (MLP) Neural Network, Support Vector Machine with Gaussian Kernel (SVM) and K-Nearest Neighbor classifier. As ensemble methods, we considered the Random Forest [63] and Adaboost [52] techniques. These classifiers were selected based on a recent study [4] that ranked the best classifiers in a comparison considering a total of 179 classifiers over 121 classification datasets.

Furthermore, as reported by Britto et al. [24], usually, the performances of dynamic selection techniques are compared with those of the best classifier in the pool, Single Best (SB); the selection of the best base classifiers in the pool, Static Selection (SS), and the majority voting combination of all classifiers in the pool, Majority Vote (MV). We also included these three methods since they are extensively used as baseline methods in the dynamic selection literature.

All classifiers were evaluated using the Matlab PRTOOLS toolbox [152]. The dynamic selection dataset (DSEL) was used as the validation set in the training process of the classifiers, and as a result, all methods were trained using the same amount of data available. The distribution of the test set remained the same. The hyper-parameters of the classifiers were set as follows:

1. Single Best (SB): The base classifier with the highest classification accuracy in the validation set is selected for classification.

Table 6
Pairwise comparison of the top six DS techniques. (a) Comparison with the adjusted p-values calculated using the Bergmann-Hommel procedure. (b) Pairwise comparison using the Wilcoxon Sign-test. The hypothesis are ordered in an ascending order according to the p-value. Hypothesis that are rejected at a $\alpha = \{0.1, 0.05, 0.01\}$ are marked with a •, ••, ••• respectively.

(a) Hypothesis	Berg p	(b) Hypothesis	Wilcoxon p
META-DES.O vs DES-P	2.88E−5 •••	META-DES vs DES-KL	5.21E−6 •••
META-DES.O vs DES-KL	2.88E−5 •••	META-DES vs DES-P	5.30E−5 •••
META-DES vs DES-P	2.56E−4 •••	META-DES.O vs DES-KL	4.19E−4 •••
META-DES vs DES-KL	2.56E−4 •••	META-DES.O vs DES-P	0.0010 •••
META-DES.O vs KNORA-U	0.0020 •••	DES-RRC vs DES-KL	0.0073 •••
META-DES vs KNORA-U	0.0085 •••	META-DES.O vs DES-RRC	0.0093 •••
DES-RRC vs DES-P	0.0908 •	META-DES.O vs KNORA-U	0.0107 ••
DES-RRC vs DES-KL	0.0908 •	META-DES vs DES-RRC	0.0136 ••
META-DES.O vs DES-RRC	0.1366	META-DES vs KNORA-U	0.0148 ••
META-DES vs DES-RRC	0.3379	DES-RRC vs DES-P	0.0159 ••
DES-RRC vs KNORA-U	0.5352	META-DES.O vs META-DES	0.0333 ••
DES-P vs KNORA-U	1.0000	DES-KL vs DES-P	0.2844
DES-KL vs KNORA-U	1.0000	DES-P vs KNORA-U	0.3252
META-DES.O vs META-DES	1.0000	DES-RRC vs KNORA-U	0.3286
DES-KL vs DES-P	1.0000	DES-KL vs KNORA-U	0.8882

2. Majority Voting (MV): The outputs of all base classifiers in the pool are combined using the majority voting rule [79].
3. Static Selection (SS): A GA ensemble selection approach based on the majority voting accuracy presented in [46]. The parameters of the GA method were set according to [46]. The validation set was used for the computation of the majority voting accuracy.
4. AdaBoost: We set the number of iterations of the algorithms to 100. The Perceptron classifier was used as the weak model. The Multi-Class Adaboost [161] was used for the multi-class problems.
5. Random Forest (RF): The number of trees was set to 200. The number of leaves was set to the square root of the number of features as recommended in [64,162,163].
6. Multi-Layer Perceptron (MLP): We varied the number of neurons in the hidden layer from 10 to 100 at 10 point intervals. The configuration that achieved the best results in the validation data was used. The MLP training process was conducted using the Levenberg–Marquadt algorithm [164]. The training was stopped if the performance on the validation set decreased or failed to improve for five consecutive epochs (early stopping).
7. Support Vector Machine with a Gaussian Kernel (SVM): A grid search was performed in order to set the values of the regularization parameter, c, and the Kernel spread parameter γ.
8. K-Nearest Neighbors (K-NN): For the K-Nearest Neighbors classifier, we considered a neighborhood size $K = 7$ so that the same neighborhood size was used for both the K-NN and the DS techniques. In addition, we also considered the performance of the 1-NN as the baseline for this method.

Table 7
Overall results considering the 30 classification datasets. The average ranks and accuracy for each algorithm is presented. Standard deviation is presented in parenthesis.

Algorithm	Avg. Rank	Algorithm	Accuracy
META-DES.O	5.43(4.92)	META-DES.O	83.92(9.13)
META-DES	5.70(4.28)	META-DES	83.24(8.94)
DES-RRC	7.67(6.23)	DES-P	82.26(9.26)
DES-P	9.17(5.27)	SVM	82.22(10.24)
KNORA-U	9.33(6.40)	DES-RRC	82.11(8.76)
DES-KL	9.90(6.42)	KNORA-U	81.69(9.82)
SVM	11.07(8.14)	DES-KL	81.52(8.77)
KNOP	13.07(5.86)	KNOP	80.81(8.92)
KNORA-E	13.23(5.62)	RF	80.78(10.98)
RF	13.77(9.50)	KNORA-E	80.36(10.75)
LCA	14.30(6.42)	OLA	79.87(10.67)
OLA	14.60(7.02)	DCS Rank	79.69(10.38)
MV	14.93(6.62)	LCA	79.57(9.84)
SS	14.97(6.38)	MCB	79.56(9.70)
MCB	15.03(7.49)	MV	79.51(9.39)
AdaBoost	15.43(7.63)	SS	79.40(10.12)
DES-KNN	15.53(6.49)	DES-KNN	79.29(10.23)
DCS Rank	16.33(5.77)	AdaBoost	79.23(10.32)
A Posteriori	16.40(7.91)	MLP	79.20(11.74)
SB	16.47(6.04)	SB	79.06(9.98)
DSOC	16.87(7.93)	DSOC	79.00(9.44)
MLP	16.90(8.45)	A Priori	78.57(11.18)
7-NN	17.40(8.59)	DES-KMEANS	78.49(10.40)
DES-KMEANS	17.50(6.13)	A Posteriori	78.14(11.53)
MLA	18.20(7.41)	7-NN	77.42(13.06)
A Priori	18.30(6.24)	MLA	77.34(9.78)
1-NN	20.50(8.10)	1-NN	76.64(11.98)

Table 7 presents the average accuracy of all classification methods, as well as their average ranking. In comparison to the baseline methods (SB, SS and MV), we can see that the majority of the DS techniques improve upon the SB (only the MLA presented both a lower ranking and average accuracy). With respect to SS and MV, 66% of the DS considered presented better results. Furthermore, the top DS techniques (Table 5) also presented a higher average accuracy and a better ranking when compared to the Random Forests and AdaBoost techniques, which are classical static ensemble techniques.

The SVM and RF classifier presented very high recognition accuracy. However, it must be pointed out that the DS techniques in this paper were all evaluated using a pool of weak, linear classifiers. All methods considered in this work could also benefit from the use of a pool of SVM classifiers, such as in the following works [7,25,30,34,156]. It is also possible to train a Random Forest, and apply dynamic selection for classification, instead of Majority Voting.

Moreover, the hyper-parameters of the SVM classifier were optimized for each dataset, while for the DS techniques, the values of the hyper-parameters were set based on previous publications [26,102]. An optimization of the hyper-parameter (e.g., neighborhood size K for the DS methods based on the KNN), as well as the evaluation of DS techniques using a different base classifier model such as SVM, could further illustrate the benefits of DS techniques. The use of techniques to estimate the best number of classifiers in the pool, such as in [165], could also be employed for improving the classification performance of DS algorithms.

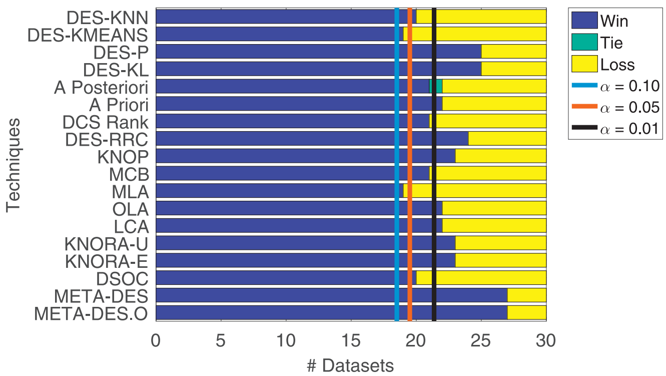


Fig. 7. Pairwise comparison between the results achieved using the different DS techniques and the 7-NN. The analysis is based on wins, ties and losses. The vertical lines illustrate the critical values considering a confidence level $\alpha = \{0.10, 0.05, 0.01\}$.

Another interesting observation is the dominance of DS techniques over the K-NN method. All DS techniques presented a better ranking and average accuracy when compared to the 1-NN, and only the MLA technique presented a lower classification accuracy and lower rank than the K-NN using the same neighborhood size. This is an interesting finding, since the majority of the DS techniques in this study (14 methods) use the K-NN method in the process of estimating the local competence of the base classifiers.

A pairwise analysis was conducted based on the Sign test [155], computed on the number of wins, ties and losses obtained by each DS, compared to the 7-NN (i.e., same neighborhood size). The null hypothesis, H_0, meant that both techniques obtained statistically equivalent results. A rejection in H_0 meant that the classification performance obtained by a corresponding DS technique was significantly better at a predefined significance level α. In this case, the null hypothesis, H_0, is rejected when the number of wins is greater than or equal to a critical value, denoted by n_c. The critical value is computed using Eq. (13):

$$n_c = \frac{n_{exp}}{2} + z_\alpha \frac{\sqrt{n_{exp}}}{2} \tag{13}$$

where $n_{\exp}$ is the total number of experiments. We ran the test considering three levels of significance: $\alpha = \{0.10, 0.05, 0.01\}$. Fig. 7 shows the results of the Sign test, and the different bars represent the critical values for each significance level. We can see that at a 0.1 significance level, all DS techniques obtained a significant number of wins. Using an $\alpha = 0.05$, only two DS methods (DS-KMEANS and MLA) did not present a significant number of wins. Moreover, even restricting the test to a significance level of 0.01, we could see that 12 DS methods out of 18 obtained a significant number of wins.

We therefore believe that this point should be further investigated in order to understand why and when DS methods that are based on K-NN present a significant boost in classification accuracy even when the same neighborhood size is considered in both approaches.

7. Conclusion and perspectives

In this paper, we presented an updated taxonomy of dynamic selection systems. The key points of a dynamic selection systems are analyzed: 1) the methodology used for the definition of the region of competence used to estimate the local competences of the base classifiers; 2) the source of information used to estimate the competence of the base classifiers, and 3) The selection approach used to determine whether a single base classifier or an ensemble of classifiers is selected. Then, we present the state-of-the-art dynamic selection techniques and categorize them based on the proposed taxonomy. We also present a review of DS methods used in different contexts such as One-Class Classification and One-Versus-One decomposition, as well as the application of DS techniques to solve complex real-world problems, such as face recognition and music genre classification.

A comparative study is carried out to compare several DCS and DES algorithms under the same testing conditions. The experimental analysis clearly demonstrates that DES techniques outperform DCS ones. Eight of the top 10 techniques are DES methods. The only exceptions are the two techniques that are based on diversity. This finding may indicate that increasing diversity at the instance level may not improve the generalization performance of DS. Another interesting finding relates to the comparison between DS and the K-NN technique, having both techniques using the same neighborhood. Results show that the majority of DS methods present statistically superior performance based on the Sign test. Moreover, the DS methods usually improve upon the baseline methods (SB and MV), as well as on other classic classifier ensemble techniques such as AdaBoost.

In general, the methods based on meta-learning obtained the best performance, which can be explained by the fact they use several sources of information to perform the dynamic selection scheme. In addition, since the selector mechanism in the meta-learning approaches is based on another classification model, these methods also adapt to the characteristics of each classification problem (e.g., imbalance, noise, etc.). However, they also present a significant increase in complexity, since it is not only multiple sets of meta-features that need to be extracted, but there is also the cost involved in using the meta-classifier for the local competence estimation.

Notwithstanding all recent contributions that have been made in this field, the definition of an ideal DS technique is still far from finalized. There are several open research questions that need to be properly answered, as well as challenges to further improve the performance of DS techniques. Therefore, we present some perspectives and our points of view on these topics.

7.1. The Oracle in dynamic selection

Many authors have used the notion of the Oracle to generate a pool or ensemble of classifiers. For example, dos Santos et al. [71,73] used the Oracle performance in a greedy search algorithm to generate the pool of classifiers. Although the proposed pool achieved a higher Oracle performance, the classification accuracy obtained by the pool generated using the Oracle performance was inferior to the pool of classifiers generated using the majority voting accuracy or diversity as search criteria.

In a recent paper, Souza et al. [166] studied the relationship between the Oracle and DS schemes. To that end, a new pool generation method was proposed, which guarantees that the Oracle performance is always 100% for the training dataset. The analysis showed that even though the Oracle for the training dataset is always 100%, the performance of the DCS techniques for the training set (in memorization) had about an 85% accuracy. That means that although the presence of the correct classifier is guaranteed in the pool, the DCS techniques struggled to select correct classifier deemed by the Oracle.

This difficulty is characterized by the notable difference in accuracy rates between the Oracle and the DCS techniques reported in several works [26,27,101], which suggests that the Oracle, as intuitive as it may be, is not the best guide for measuring the performance of a given classifier pool for dynamic selection. The reason for this is that the Oracle perceives the classification problem globally, whereas DS techniques rely on local information to select the best classifier for each test instance. Thus, the Oracle information may not be very much relevant for dynamic selection schemes. Based on this analysis, the authors propose a new metric, called the Hit-rate, which takes into account the local information from the DS methods. They argue that the Hit-rate should be used instead of the Oracle, as it covers both local and global information regarding the given pool of classifiers.

7.2. Pool generation

In the majority of DS publications, the pool of classifiers is generated using either well known ensemble generation methods such as Bagging, or by using heterogeneous classifiers [25,34]. The problem with such generation approaches is that they were proposed for static combination methods. In other words, they use a global approach in generating the base classifiers. Since these techniques look at the problem globally rather than locally, they do not guarantee the presence of local experts. For this reason, the DS methods may not be able to select the competent classifiers locally [166]. To the best of our knowledge, there is no classifier generation procedure that is adapted to dynamic selection techniques. We believe that the definition of a classifier generation procedure that takes into account the local information is a very promising research direction in order to improve the performance of all DS techniques.

Another research direction in terms of the generation of a pool of classifiers concerns the pool size. Normally, a large pool of classifiers is considered when DS techniques are evaluated. For instance, in [27,123,167] a pool composed of 100 base classifiers was used, while in [156], a pool composed of 1000 SVMs was employed. An interesting work conducted by Roy et al. [165,168] proposed a meta-regression model to predict the best size of the pool of classifiers based on complexity measures extracted from the classification problem. Results demonstrate that DS usually presents better classification results using an average 20 base classifiers. Hence, using smaller pools of classifiers can not only improve the accuracy, but also, reduce the computational costs involved in DS techniques.

7.3. Region of competence definition

As reported in [108] and [107], the definition of the region of competence plays a very important role in the classification performance of a DS system since the local competence of the base classifiers is estimated based on the samples belonging to this region. Hence, we believe an interesting research direction is to study the relationship between the samples belonging to the region of competence and the selection of the base classifiers; in other words, how the distribution of the region of competence changes the way the competence of the base classifiers is estimated. This relationship can be used in defining new ways of demarcating the region of competence and selecting classifiers, and additionally should be taken into account during the classifier generation procedure.

Furthermore, the majority of the dynamic selection techniques use a fixed neighborhood size. This value of K is often used for multiple classification problems, regardless of their complexity. Another interesting future work would involve the prediction of the best K value according to the specificity of the complexity of the classification problem, or, having a variable neighborhood size which could also change dynamically, based on the location of the query sample in the feature space. In this case, one could for instance, have a higher K value for samples that are closer to the decision border in order to reduce the influence of noisy samples.

Another interesting aspect regarding the region of competence is the work conducted by Oliveira et al. [109], in which the authors studied the estimated regions of competence in order to know whether or not the query sample is located in region with borderline samples of different classes (called indecision region). In the context of DS, a sample is located in an indecision region when its region of competence contains instances from different classes. The authors demonstrated that in such cases, many DS techniques can select classifiers with decision boundaries that do not cross the region of competence, assigning all samples in the region of competence to the same class. This may cause problems specially when dealing with imbalanced datasets, in which the majority of the samples in the region of competence belong to a single class (majority class).

In order to deal with this problem, an online pruning framework was proposed in [109], which pre-selects classifiers with decision boundaries that crosses the region of competence of the test instance, when the test instance is located in an indecision region. The first step of the framework is to determine whether or not query instance is located in an indecision region. If yes, the pruning mechanism is employed to pre-select the classifiers that cross the region of competence. Then, a DS technique is applied to select the most competent classifiers among the pre-selected pool. If the query is located in a safe region, the DS technique is used for classification. The proposed online pruning framework significantly improved the classification accuracy of the 9 DS techniques considered in the experimental analysis. This result demonstrates that in order to be considered competent, a base classifier should not only obtain a high competence level, estimated by the DS method, but must also cross the region of competence in the case of indecision regions.

However, one of the problems pointed out in this work is the fact that some samples were located in an indecision region, and had no base classifiers with decision boundaries crossing its region of competence, which lead us to think that an ensemble generation technique that maximizes the number of base classifiers with decision boundaries crossing the indecision regions as a very promising research direction. That also supports our hypothesis that, when working with DS, we need an ensemble generation method that generates local experts rather than global ones.

7.4. Prototype selection and generation for DS

The rationale for using PS techniques is that the performance of DS techniques is very dependent on the distribution of DSEL. When the samples in this set are not representative enough of the query sample, the DS technique may not select the most competent classifiers to predict its label. This phenomenon may occur as a result of a high degree of overlap between different classes or may be due to the presence of noise [108]. Another important aspect of editing the distribution of DSEL is that it can also significantly reduce the computational complexity involved in applying dynamic selection techniques, since the definition of the region of competence is conducted using the K-NN technique, which can be very costly when dealing with large datasets.

Recent works have pointed out that the use of Prototype Selection (PS) [169] techniques can significantly improve the classification accuracy of several dynamic selection techniques [104,107]. In this case, the PS techniques are applied to edit the distribution of DSEL in the training stage. The edited DSEL, denoted by $DSEL'$, is then used for extracting the regions of competence during the generalization phase. In [104], we evaluated the impact of six Prototype Selection techniques in relation to the classification accuracy and reduction of computational time of several dynamic selection techniques. The experimental analysis demonstrated that PS techniques, such as the Relative Neighborhood Graph, significantly improve the classification performance of all DS methods considered in the analysis. Moreover, it can also significantly reduce the size of DSEL and improve the recognition performance of several DS techniques.

However, as reported in [104], the techniques that present the best results for DS are not the ones that obtained the best classification accuracy when the 1-NN is considered as the classification scheme. For instance, the Generational Genetic Algorithm (GGA) and the CHC Adaptive Search Algorithm obtained the worse performance in the experimental study with DS techniques; however, they were among the top 5 best performing algorithms when the 1-NN was considered as the classification scheme [169]. The performance obtained by the CHC method was significantly worse when compared to the other PS techniques, as well as the baseline result (i.e., the system without using PS).

This may be attributable to the fact that these techniques use the classification accuracy of the 1-NN technique in the fitness function for editing the dataset. An interesting direction for future work would involve the use of the performance of DS techniques, (e.g., the accuracy of the OLA technique) as a criterion within those techniques in order to adapt the distribution of DSEL for the use of DS techniques, rather than the 1-NN classifier. Moreover, the use of Prototype Generation techniques [170] should also be evaluated for improving the performance of DS techniques, especially when dealing with imbalanced distributions [107].

7.5. Diversity for DS

One of the most studied aspects in MCS is the concept of diversity. It is known that we need diversity in the classifier ensemble, since combining classifiers that always produce the same decision will not improve the recognition rate of the system. Diversity is often measured by the difference in the classifiers' decision, such as in the Q-statistics and in the disagreement measures [131].

To date, few dynamic selection techniques have utilized diversity together with different competence measures to perform ensemble selection [30,171]. However, the two DS techniques that take into account diversity information did not present a good overall performance in our experimental analysis (as shown in Section 6).

In our opinion, when dynamic selection is considered, a diverse pool of classifiers is required, since, intuitively, a pool of diverse classifiers means that there are several classifiers specialized in different regions of the feature space. Consequently, the classifier pool has a better coverage of the whole feature space [172]. However, after selecting the EoC, we believe that we need to promote consensus, rather than diversity, among the selected classifiers. Intuitively, the non-competent classifiers, for the corresponding local region, will present high diversity when compared to the competent ones, since they performed differently at the local level. The addition of diverse classifiers at the ensemble level can hinder the EoC decision. Moreover, if there is no consensus among the selected base classifiers, the system may end up randomly selecting the class of the query instance. This point is discussed by Sağlam and Street in a recent publication [156], who evaluate the concept of distant diversity.

Another important point to be investigated is the impact of diversity for dynamic ensembles, rather than static ones. The analysis conducted in [70,131] considered only static combination rules (e.g., Average, Product and Majority voting). In the case of DS techniques, the impact of diversity could be analyzed at the pool level, i.e., before selecting the base classifiers, as well as at the instance level, i.e., after the base classifiers are selected for the classification of the query.

7.6. Cost-sensitive dynamic selection

In many real-world classification tasks, such as medical diagnosis and credit analysis, it is crucial to take into account misclassification costs associated with each class. Furthermore, different classification costs can stem from the imbalanced nature of some classification problems. In other words, due to the imbalanced nature of the problem, the system may require different costs for the majority and minority classes.

Recently, several cost-sensitive ensemble approaches have been proposed, such as cost-sensitive trees ensemble [173], boosting [174] and the cost-sensitive ensemble methods based on the ROC space [175,176]. However, these approaches are all based on static ensembles. To date the dynamic selection literature considers all classes with the same cost, and no rejection mechanism has been proposed for DS [24]. Classification systems that deal with such applications often require a built-in rejection mechanism to avoid committing errors in very risky predictions. Thus, we believe that a definition of cost-sensitive DCS and DES techniques is another promising research direction for dynamic selection.

7.7. Imbalanced datasets

Imbalanced learning has recently attracted much attention from the pattern recognition community since this kind of data is very common in real-world applications, e.g., biomedical data and spam detection. Classification in the presence of class imbalance is challenging since the usual method of training and selecting standard classification models is based on classification accuracy. However, if we take into account the classification accuracy in such cases, the minority class could be totally ignored.

In our opinion, dynamic selection techniques can bring many benefits to this type of problem since they perform a local classification. The selection of the ensemble of classifiers is performed taking into account only the neighborhood of the query sample, rather than the whole dataset. Thus, we believe that the classifier selection scheme will not be biased towards the majority class.

To the best of our knowledge, there is just one publication that discusses the use of dynamic selection for imbalanced distributions [126]. However, the proposed system was only applied to two credit scoring datasets, which is not enough to evaluate whether or not DS can cope with class imbalance. In addition, this paper did not take into consideration the use of data preprocessing such as SMOTE and RAMO [177]. Hence, another interesting future work would involve the evaluation of DS techniques for imbalanced distributions, and possibly, the definition or adaptation of DS techniques for this kind of application.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the École de technologie supérieure (ÉTS Montréal) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

References

[1] L.I. Kuncheva, A theoretical study on six classifier fusion strategies, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2) (2002) 281–286.
[2] T.G. Dietterich, Ensemble methods in machine learning, in: International Workshop on Multiple Classifier Systems, Springer, 2000, pp. 1–15.
[3] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[4] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. 15 (2014) 3133–3181.
[5] D. Opitz, R. Maclin, Popular ensemble methods: an empirical study, J. Artif. Intell. Res. 11 (1999) 169–198.

[6] R. Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (3) (2006) 21–45.
[7] S. Bashbaghi, E. Granger, R. Sabourin, G. Bilodeau, Dynamic selection of exemplar-svms for watch-list screening through domain adaptation, in: Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM), 2017, pp. 738–745.
[8] P.R.L. de Almeida, E.J. da Silva Júnior, T.M. Celinski, A. de Souza Britto, L.E.S. de Oliveira, A.L. Koerich, Music genre classification using dynamic selection of ensemble of classifiers, in: Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on, IEEE, 2012, pp. 2700–2705.
[9] S. Lessmann, B. Baesens, H.-V. Seow, L.C. Thomas, Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research, Eur. J. Oper. Res. 247 (1) (2015) 124–136.
[10] H. Xiao, Z. Xiao, Y. Wang, Ensemble classification based on supervised clustering for credit scoring, Appl. Soft Comput. 43 (2016) 73–86.
[11] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Trans. Syst. Man. Cybern. Part C 42 (4) (2012) 463–484.
[12] C. Porcel, A. Tejeda-Lorente, M. Martínez, E. Herrera-Viedma, A hybrid recommender system for the selective dissemination of research resources in a technology transfer office, Inf. Sci. 184 (1) (2012) 1–19.
[13] M. Jahrer, A. Töscher, R. Legenstein, Combining predictions for accurate recommender systems, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 693–702.
[14] D. Di Nucci, F. Palomba, R. Oliveto, A. De Lucia, Dynamic selection of classifiers in bug prediction: an adaptive method, IEEE Trans. Emerg. Topics Comput. Intell. 1 (3) (2017) 202–212.
[15] A. Panichella, R. Oliveto, A. De Lucia, Cross-project defect prediction models: L'union fait la force, in: Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), 2014 Software Evolution Week-IEEE Conference on, IEEE, 2014, pp. 164–173.
[16] G. Giacinto, F. Roli, L. Didaci, Fusion of multiple classifiers for intrusion detection in computer networks, Pattern Recognit. Lett. 24 (12) (2003) 1795–1803.
[17] G. Giacinto, R. Perdisci, M. Del Rio, F. Roli, Intrusion detection in computer networks by a modular ensemble of one-class classifiers, Inf. Fus. 9 (1) (2008) 69–82.
[18] B. Krawczyk, L.L. Minku, J. Gama, J. Stefanowski, M. Woźniak, Ensemble learning for data stream analysis: a survey, Inf. Fus. 37 (2017) 132–156.
[19] L.I. Kuncheva, Classifier ensembles for changing environments (2004) 1–15.
[20] R. Polikar, L. Udpa, S. Udpa, S. Member, S. Member, V. Honavar, Learn++: an incremental learning algorithm for supervised neural networks, IEEE Trans. Syst. Man Cybern. (C), Special Issue on Knowledge Management 31 (2001) 497–508.
[21] M. Wozniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fus. 16 (2014) 3–17.
[22] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (1) (2010) 1–39.
[23] Y. Ren, L. Zhang, P.N. Suganthan, Ensemble classification and regression-recent developments, applications and future directions, IEEE Comput. Intell. Mag. 11 (1) (2016) 41–53.
[24] A.S. Britto, R. Sabourin, L.E.S. de Oliveira, Dynamic selection of classifiers - a comprehensive review, Pattern Recognit. 47 (11) (2014) 3665–3680.
[25] T. Woloszynski, M. Kurzynski, A probabilistic model of classifier competence for dynamic ensemble selection, Pattern Recognit. 44 (2011) 2656–2668.
[26] A.H.R. Ko, R. Sabourin, u.S. Britto Jr., From dynamic classifier selection to dynamic ensemble selection, Pattern Recognit. 41 (2008) 1735–1748.
[27] R.M.O. Cruz, R. Sabourin, G.D.C. Cavalcanti, T.I. Ren, META-DES: a dynamic ensemble selection framework using meta-learning, Pattern Recognit. 48 (5) (2015) 1925–1935.
[28] X. Zhu, X. Wu, Y. Yang, Dynamic classifier selection for effective mining from noisy data streams, in: Proceedings of the 4th IEEE International Conference on Data Mining, 2004, pp. 305–312.
[29] L.I. Kuncheva, Clustering-and-selection model for classifier combination, in: Fourth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies, 2000, pp. 185–188.
[30] R.G.F. Soares, A. Santana, A.M.P. Canuto, M.C.P. de Souto, Using accuracy and diversity to select classifiers to build ensembles, in: Proceedings of the International Joint Conference on Neural Networks, 2006, pp. 1310–1316.
[31] K. Woods, W.P. Kegelmeyer Jr., K. Bowyer, Combination of multiple classifiers using local accuracy estimates, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 405–410.
[32] P.C. Smits, Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection, IEEE Trans. Geosci. Remote Sens. 40 (4) (2002) 801–813.
[33] M. Sabourin, A. Mitiche, D. Thomas, G. Nagy, Classifier combination for hand-printed digit recognition, International Conference on Document Analysis and Recognition (1993) 163–166.
[34] T. Woloszynski, M. Kurzynski, P. Podsiadlo, G.W. Stachowiak, A measure of competence based on random classification for dynamic ensemble selection, Inf. Fus. 13 (3) (2012) 207–213.
[35] B. Krawczyk, M. Wozniak, Dynamic classifier selection for one-class classification, Knowl. Based Syst. 107 (2016) 43–53.
[36] P.R.L. de Almeida, L.S. Oliveira, A. de Souza Britto Jr, R. Sabourin, Handling concept drifts using dynamic selection of classifiers, in: Tools with Artificial Intelligence (ICTAI), 2016, pp. 989–995.
[37] A. Tsymbal, M. Pechenizkiy, P. Cunningham, S. Puuronen, Dynamic integration of classifiers for handling concept drift, Inf. Fus. 9 (1) (2008) 56–68.
[38] W. Qu, Y. Zhang, J. Zhu, Q. Qiu, Mining multi-label concept-drifting data streams using dynamic classifier ensemble, in: Asian Conference on Machine Learning, Springer, 2009, pp. 308–321.
[39] I. Mendialdua, J. Martínez-Otzeta, I. Rodriguez-Rodriguez, T. Ruiz-Vazquez, B. Sierra, Dynamic selection of the best base classifier in one versus one, Knowl. Based Syst. 85 (2015) 298–306.
[40] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Dynamic classifier selection for one-vs-one strategy: avoiding non-competent classifiers, Pattern Recognit. 46 (12) (2013) 3412–3424.
[41] Z.-L. Zhang, X.-G. Luo, S. García, J.-F. Tang, F. Herrera, Exploring the effectiveness of dynamic ensemble selection in the one-versus-one scheme, Knowl. Based Syst. 125 (2017) 53–63.
[42] L. Batista, E. Granger, R. Sabourin, Dynamic selection of generative–discriminative ensembles for off-line signature verification, Pattern Recognit. 45 (4) (2012) 1326–1340.
[43] S. Bashbaghi, E. Granger, R. Sabourin, G.-A. Bilodeau, Robust watch-list screening using dynamic ensembles of svms based on multiple face representations, Mach. Vis. Appl. (2017) 1–23.
[44] S. Bashbaghi, E. Granger, R. Sabourin, G.-A. Bilodeau, Dynamic ensembles of exemplar-svms for still-to-video face recognition, Pattern Recognit 69 (2017) 61–81.
[45] C. Pagano, Adaptive classifier ensembles for face recognition in video-surveillance, Ph.D. thesis, École de technologie supérieure, 2015.
[46] D. Ruta, B. Gabrys, Classifier selection for majority voting, Inf. Fus. 6 (1) (2005) 63–81.
[47] R.P.W. Duin, The combining classifier: to train or not to train?, Proceedings of the 16th International Conference on Pattern Recognition 2 (2002) 765–770.
[48] L.K. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (10) (1990) 993–1001.
[49] S.-B. Cho, J.H. Kim, Combining multiple neural networks by fuzzy integral for robust classification, IEEE Trans. Syst. Man Cybern. 25 (2) (1995) 380–384.
[50] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
[51] M. Skurichina, R.P.W. Duin, Bagging for linear classifiers, Pattern Recognit. 31 (1998) 909–930.
[52] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: Proceedings of the Second European Conference on Computational Learning Theory, 1995, pp. 23–37.
[53] J. Feng, L. Wang, M. Sugiyama, C. Yang, Z.-H. Zhou, C. Zhang, Boosting and margin theory, Front. Electr. Electron. Eng. 7 (1) (2012) 127–133.
[54] A. Rahman, B. Verma, Novel layered clustering-based approach for generating ensemble of classifiers, IEEE Trans. Neural Netw. 22 (5) (2011) 781–792.
[55] R.M. O. Cruz, G.D. C. Cavalcanti, T.I. Ren, An ensemble classifier for offline cursive character recognition using multiple feature extraction techniques, Proceedings of the International Joint Conference on Neural Networks (2010a) 744–751.
[56] R.M. O. Cruz, G.D. C. Cavalcanti, T.I. Ren, Handwritten digit recognition using multiple feature extraction techniques and classifier ensemble, in: 17th International Conference on Systems, Signals and Image Processing, 2010, pp. 215–218.
[57] A. Rahman, B. Verma, Effect of ensemble classifier composition on offline cursive character recognition, Inf. Process. Manage. 49 (4) (2013) 852–864.
[58] A. Schindler, R. Mayer, A. Rauber, Facilitating comprehensive benchmarking experiments on the million song dataset., in: ISMIR, 2012, pp. 469–474.
[59] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 832–844.
[60] M. Skurichina, R.P.W. Duin, Bagging, boosting and the random subspace method for linear classifiers, Pattern Anal. Appl. 5 (2) (2002) 121–135.
[61] R.P. Duin, D.M. Tax, Experiments with classifier combining rules, in: International Workshop on Multiple Classifier Systems, 2000, pp. 16–29.
[62] W. Wang, P. Jones, D. Partridge, Diversity between neural networks and decision trees for building multiple classifier systems, in: International Workshop on Multiple Classifier Systems, 2000, pp. 240–249.
[63] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[64] L. Rokach, Decision forest: twenty years of research, Inf. Fus. 27 (2016) 111–125.
[65] J.J. Rodríguez, L.I. Kuncheva, C.J. Alonso, Rotation forest: a new classifier ensemble method, IEEE Trans. Pattern Anal. Mach. Intell. 28 (10) (2006) 1619–1630.
[66] G. Giacinto, F. Roli, Design of effective neural network ensembles for image classification purposes, Image Vis. Comput. 19 (9–10) (2001) 699–707.
[67] G. Giacinto, F. Roli, Design of effective neural network ensembles for image classification purposes, Image Vis. Comput. 19 (9–10) (2001) 699–707.
[68] M. Aksela, Comparison of classifier selection methods for improving committee performance, in: International Workshop on Multiple Classifier Systems, 2003, pp. 84–93.
[69] R.M.O. Cruz, G.D.C. Cavalcanti, I.R. Tsang, R. Sabourin, Feature representation selection based on classifier projection space and oracle analysis, Expert Syst. Appl. 40 (9) (2013) 3813–3827.
[70] G. Brown, L.I. Kuncheva, "Good" and "bad" diversity in majority vote ensembles, in: International Workshop on Multiple Classifier Systems, Springer, 2010, pp. 124–133.
[71] E.M. dos Santos, R. Sabourin, P. Maupin, Overfitting cautious selection of classifier ensembles with genetic algorithms, Inf. Fus. 10 (2) (2009) 150–162.
[72] I. Partalas, G. Tsoumakas, I. Vlahavas, Focused ensemble selection: a diversi-

ty-based method for greedy ensemble selection, in: Proceeding of the 18th European Conference on Artificial Intelligence, 2008, pp. 117–121.
[73] E.M. dos Santos, R. Sabourin, Classifier ensembles optimization guided by population oracle, in: IEEE Congress on Evolutionary Computation, 2011, pp. 693–698.
[74] B. Gabrys, D. Ruta, Genetic algorithms in classifier fusion, Appl. Soft Comput. 6 (4) (2006) 337–347.
[75] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artif. Intell. 137 (1–2) (2002) 239–263.
[76] R.E. Banfield, L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer, Ensemble diversity measures and their application to thinning, Inf. Fus. 6 (1) (2005) 49–62.
[77] L. Kuncheva, Fuzzy Classifier Design, vol. 49, Springer Science & Business Media, 2000.
[78] R.P. Duin, D.M. Tax, Classifier conditional posterior probabilities, in: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Springer, 1998, pp. 611–619.
[79] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 226–239.
[80] T.K. Ho, J.J. Hull, S.N. Srihari, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1) (1994) 66–75.
[81] Y.S. Huang, C.Y. Suen, A method of combining multiple experts for the recognition of unconstrained handwritten numerals, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 90–94.
[82] L.I. Kuncheva, J.C. Bezdek, R.P.W. Duin, Decision templates for multiple classifier fusion: an experimental comparison, Pattern Recognit. 34 (2001) 299–314.
[83] Y. Lu, Knowledge integration in a multiple classifier system, Appl. Intell. 6 (2) (1996) 75–86.
[84] G.L. Rogova, Combining the results of several neural network classifiers, Neural Netw. 7 (5) (1994) 777–781.
[85] D.M.J. Tax, M. van Breukelen, R.P.W. Duin, J. Kittler, Combining multiple classifiers by averaging or by multiplying? Pattern Recognit. 33 (9) (2000) 1475–1485.
[86] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.
[87] L. Lam, C.Y. Suen, Optimal combinations of pattern classifiers, Pattern Recognit. Lett. 16 (9) (1995) 945–954.
[88] D.H. Wolpert, Stacked generalization, Neural Netw. 5 (1992) 241–259.
[89] Š. Raudys, Trainable fusion rules. ii. small sample-size effects, Neural Netw. 19 (10) (2006) 1517–1527.
[90] Š. Raudys, Trainable fusion rules. i. large sample size case, Neural Netw. 19 (10) (2006) 1506–1516.
[91] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive mixtures of local experts, Neural Comput 3 (1991) 79–87.
[92] S. Masoudnia, R. Ebrahimpour, Mixture of experts: a literature survey, Artif. Intell. Rev. (2014) 1–19.
[93] S.E. Yuksel, J.N. Wilson, P.D. Gader, Twenty years of mixture of experts, IEEE Trans. Neural Netw. Learn. Syst. 23 (8) (2012) 1177–1193.
[94] H. Cevikalp, R. Polikar, Local classifier weighting by quadratic programming, IEEE Trans. Neural Netw. 19 (10) (2008) 1832–1838.
[95] D. Jiménez, Dynamically weighted ensemble neural networks for classification, in: Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on, 1, IEEE, 1998, pp. 753–756.
[96] D. Štefka, M. Holeňa, Dynamic classifier aggregation using interaction-sensitive fuzzy measures, Fuzzy Sets Syst. 270 (2015) 25–52.
[97] R.M.O. Cruz, R. Sabourin, G.D.C. Cavalcanti, META-DES.H: a dynamic ensemble selection technique using meta-learning and a dynamic weighting approach, in: International Joint Conference on Neural Networks, 2015, pp. 1–8.
[98] L.M. Vriesmann, A.S. Britto, L.S. Oliveira, A.L. Koerich, R. Sabourin, Combining overall and local class accuracies in an oracle-based method for dynamic ensemble selection, in: Neural Networks (IJCNN), 2015 International Joint Conference on, IEEE, 2015, pp. 1–7.
[99] R.M.O. Cruz, R. Sabourin, G.D.C. Cavalcanti, Meta-des. Oracle: meta-learning and feature selection for dynamic ensemble selection, Inf. Fus. 38 (2017) 84–103.
[100] M. Wozniak, M. Zmyslony, Designing fusers on the basis of discriminants–evolutionary and neural methods of training, in: International Conference on Hybrid Artificial Intelligence Systems, Springer, 2010, pp. 590–597.
[101] L. Didaci, G. Giacinto, F. Roli, G.L. Marcialis, A study on the performances of dynamic classifier selection based on local accuracy estimation, Pattern Recognit. 38 (11) (2005) 2188–2191.
[102] R.M. O. Cruz, G.D. C. Cavalcanti, T.I. Ren, A method for dynamic ensemble selection based on a filter and an adaptive distance to improve the quality of the regions of competence, Proceedings of the International Joint Conference on Neural Networks (2011) 1126–1133.
[103] L. Didaci, G. Giacinto, Dynamic classifier selection by adaptive k-nearest-neighbourhood rule, in: International Workshop on Multiple Classifier Systems, Springer, 2004, pp. 174–183.
[104] R.M.O. Cruz, R. Sabourin, G.D.C. Cavalcanti, Analyzing different prototype selection techniques for dynamic classifier and ensemble selection, in: International Joint Conference on Neural Networks (IJCNN), 2017, pp. 3959–3966.
[105] T. Woloszynski, M. Kurzynski, A measure of competence based on randomized reference classifier for dynamic ensemble selection, in: International Conference on Pattern Recognition (ICPR), 2010, pp. 4194–4197.
[106] P.R. Cavalin, R. Sabourin, C.Y. Suen, Dynamic selection approaches for multiple classifier systems, Neural Comput. Appl. 22 (3–4) (2013) 673–688.
[107] R.M.O. Cruz, R. Sabourin, G.D. Cavalcanti, Prototype selection for dynamic classifier and ensemble selection, Neural Comput. Appl. (2016) 1–11.
[108] R.M.O. Cruz, R. Sabourin, G.D.C. Cavalcanti, A DEEP analysis of the META-DES framework for dynamic selection of ensemble of classifiers, CoRR (2015). abs/1509.00825.
[109] D.V. Oliveira, G.D. Cavalcanti, R. Sabourin, Online pruning of base classifiers for dynamic ensemble selection, Pattern Recognit. (2017), doi:10.1016/j.patcog.2017.06.030.
[110] T.P.F. de Lima, A.T. Sergio, T.B. Ludermir, Improving classifiers and regions of competence in dynamic ensemble selection, in: Intelligent Systems (BRACIS), 2014 Brazilian Conference on, IEEE, 2014, pp. 13–18.
[111] T.P.F. De Lima, T.B. Ludermir, Optimizing dynamic ensemble selection procedure by evolutionary extreme learning machines and a noise reduction filter, in: Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on, IEEE, 2013, pp. 546–552.
[112] G. Giacinto, F. Roli, Dynamic classifier selection based on multiple classifier behaviour, Pattern Recognit. 34 (2001) 1879–1881.
[113] P.R. Cavalin, R. Sabourin, C.Y. Suen, Logid: an adaptive framework combining local and global incremental learning for dynamic selection of ensembles of HMMs, Pattern Recognit. 45 (9) (2012) 3544–3556.
[114] L. Rastrigin, R. Erenstein, Method of Collective Recognition, vol. 595, 1981. (in Russian).
[115] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, Q. Zou, Libd3c: ensemble classifiers with a clustering and dynamic selection strategy, Neurocomputing 123 (2014) 424–435.
[116] M.C. de Souto, R.G. Soares, A. Santana, A.M. Canuto, Empirical comparison of dynamic classifier selection methods based on diversity and accuracy for building ensembles, in: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, IEEE, 2008, pp. 1480–1487.
[117] J. Wang, P. Neskovic, L.N. Cooper, Improving nearest neighbor rule with a simple adaptive distance measure, Pattern Recognit. Lett. 28 (2007) 207–213.
[118] J. Wang, P. Neskovic, L.N. Cooper, Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence, Pattern Recognit. 39 (2006) 417–423.
[119] B. Sierra, E. Lazkano, I. Irigoien, E. Jauregi, I. Mendialdua, K nearest neighbor equality: giving equal chance to all existing classes, Inf. Sci. 181 (23) (2011) 5158–5168.
[120] T. Woloszynski, M. Kurzynski, On a new measure of classifier competence applied to the design of multiclassifier systems, in: International Conference on Image Analysis and Processing (ICIAP), 2009, pp. 995–1004.
[121] L. Batista, E. Granger, R. Sabourin, Dynamic ensemble selection for off-line signature verification, in: International Workshop on Multiple Classifier Systems, 2011, pp. 157–166.
[122] K. M., W. T., R. Lysiak, On two measures of classifier competence for dynamic ensemble selection - experimental comparative analysis, in: International Symposium on Communications and Information Technologies, 2010, pp. 1108–1113.
[123] A.L. Brun, A.S.B. Jr., L.S. Oliveira, F. Enembreck, R. Sabourin, Contribution of data complexity features on dynamic classifier selection, in: International Joint Conference on Neural Networks (IJCNN), 2016, pp. 4396–4403.
[124] R.M.O. Cruz, R. Sabourin, G.D.C. Cavalcanti, On meta-learning for dynamic ensemble selection, in: 22nd International Conference on Pattern Recognition (ICPR), 2014, pp. 1230–1235.
[125] F. Pinto, C. Soares, J. Mendes-Moreira, Chade: metalearning with classifier chains for dynamic combination of classifiers, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2016, pp. 410–425.
[126] J. Xiao, L. Xie, C. He, X. Jiang, Dynamic classifier ensemble model for customer classification with imbalanced class distribution, Expert Syst. Appl. 39 (2012) 3668–3675.
[127] E.M. Dos Santos, R. Sabourin, P. Maupin, A dynamic overproduce-and-choose strategy for the selection of classifier ensembles, Pattern Recognit. 41 (2008) 2993–3009.
[128] M. Wozniak, Hybrid Classifiers: Methods of Data, Knowledge, and Classifier Combination, Springer, 2013.
[129] G. Giacinto, F. Roli, Methods for dynamic classifier selection, in: Image Analysis and Processing, 1999. Proceedings. International Conference on, IEEE, 1999, pp. 659–664.
[130] R.M.O. Cruz, Dynamic Selection of Ensemble of Classifiers Using Meta-Learning, Ph.D. thesis, École de Technologie Supérieure, 2016.
[131] C.A. Shipp, L.I. Kuncheva, Relationships between combination methods and measures of diversity in combining classifiers, Inf. Fus. 3 (2002) 135–148.
[132] T.K. Ho, M. Basu, Complexity measures of supervised classification problems, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 289–300.
[133] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes, Pattern Recognit. 44 (8) (2011) 1761–1776.
[134] M. Galar, A. Fernández, E. Barrenechea, F. Herrera, Drcw-ovo: distance-based relative competence weighting combination for one-vs-one strategy in multi-class problems, Pattern Recognit. 48 (1) (2015) 28–42.
[135] D.M.J. Tax, One-class classification: Concept Learning in the Absence of Counter-Examples, Ph.D. thesis, Technische Universiteit Delft, 2001.

[136] B. Antosik, M. Kurzynski, New measures of classifier competence – heuristics and application to the design of multiple classifier systems., in: Computer Recognition Systems, vol. 4, 2011, pp. 197–206.
[137] J. Martins, L.S. Oliveira, A. Britto, R. Sabourin, Forest species recognition based on dynamic classifier selection and dissimilarity feature vector representation, Mach Vis Appl 26 (2–3) (2015) 279–293.
[138] A.T. Sergio, T.P. de Lima, T.B. Ludermir, Dynamic selection of forecast combiners, Neurocomputing 218 (2016) 37–50.
[139] M. Kurzynski, A. Wolczowski, Dynamic selection of classifiers ensemble applied to the recognition of emg signal for the control of bioprosthetic hand, in: Control, Automation and Systems (ICCAS), 2011 11th International Conference on, IEEE, 2011, pp. 382–386.
[140] M. Kurzynski, M. Krysmann, P. Trajdos, A. Wolczowski, Multiclassifier system with hybrid learning applied to the control of bioprosthetic hand, Comput. Biol. Med. 69 (2016) 286–297.
[141] M. Kurzynski, M. Krysmann, P. Trajdos, A. Wolczowski, Two-stage multiclassifier system with correction of competence of base classifiers applied to the control of bioprosthetic hand, in: Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on, IEEE, 2014, pp. 620–626.
[142] D.J. Hand, W.E. Henley, Statistical classification methods in consumer credit scoring: a review, J. R. Statist. Soc. 160 (3) (1997) 523–541.
[143] C.N. Silla Jr, A.L. Koerich, C.A. Kaestner, The latin music database., in: ISMIR, 2008, pp. 451–456.
[144] J. Milgram, M. Cheriet, R. Sabourin, Estimating accurate multi-class probabilities with support vector machines, in: Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on, vol. 3, IEEE, 2005, pp. 1906–1911.
[145] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen, Automatic recognition of handwritten numerical strings: a recognition and verification strategy, IEEE Trans. Pattern Anal. Mach. Intell. 24 (11) (2002) 1438–1454.
[146] F. Vargas, M. Ferrer, C. Travieso, J. Alonso, Off-line handwritten signature gpds-960 corpus, in: Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on, vol. 2, IEEE, 2007, pp. 764–768.
[147] A. Wolczowski, M. Kurzynski, Human-machine interface in bioprosthesis control using EMG signal classification, Expert Syst. 27 (1) (2010) 53–70.
[148] K. Bache, M. Lichman, UCI Machine Learning Repository, 2013.
[149] R.D. King, C. Feng, A. Sutherland, Statlog: Comparison of Classification Algorithms on Large Real-World Problems, 1995.
[150] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, Mult. Val. Log. Soft Comput. 17 (2–3) (2011) 255–287.
[151] L. Kuncheva, Ludmila kuncheva collection LKC, 2004.
[152] R.P.W. Duin, P. Juszczak, D. de Ridder, P. Paclik, E. Pekalska, D.M. Tax, Prtools, a matlab toolbox for pattern recognition, 2004.
[153] S. Mirjalili, A. Lewis, S-shaped versus v-shaped transfer functions for binary particle swarm optimization, Swarm Evol. Comput. 9 (2013) 1–14.
[154] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (200) (1937) 675–701.
[155] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[156] Ş.Y. Sağlam, W.N. Street, Distant diversity in dynamic class prediction, Ann. Oper. Res. (2016) 1–15.
[157] A. Benavoli, G. Corani, F. Mangili, Should we really use post-hoc tests based on mean-ranks, J. Mach. Learn. Res. 17 (5) (2016) 1–10.
[158] B. Bergmann, G. Hommel, Improvements of general multiple test procedures for redundant systems of hypotheses, in: Multiple Hypothesenprüfung/Multiple Hypotheses Testing, Springer, 1988, pp. 100–115.
[159] S. Garcia, F. Herrera, An extension on"statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons, J. Mach. Learn. Res. 9 (Dec) (2008) 2677–2694.
[160] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[161] J. Zhu, H. Zou, S. Rosset, T. Hastie, Multi-class adaboost, Stat. Interface 2 (3) (2009) 349–360.
[162] S. Bernard, L. Heutte, S. Adam, Influence of hyperparameters on random forest accuracy, in: International Workshop on Multiple Classifier Systems, Springer, 2009, pp. 171–180.
[163] S. Bernard, L. Heutte, S. Adam, Forest-rk: a new random forest induction method, in: International Conference on Intelligent Computing, Springer, 2008, pp. 430–437.
[164] M.T. Hagan, M.B. Menhaj, Training feedforward networks with the marquardt algorithm, IEEE Trans. Neural Netw. 5 (6) (1994) 989–993.
[165] A. Roy, R.M.O. Cruz, R. Sabourin, G.D. Cavalcanti, Meta-regression based pool size prediction scheme for dynamic selection of classifiers, in: 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 216–221.
[166] M.A. Souza, G.D. Cavalcanti, R.M.O. Cruz, R. Sabourin, On the characterization of the oracle for dynamic classifier selection, in: International Joint Conference on Neural Networks (IJCNN), 2017, pp. 332–339.
[167] E.M. dos Santos, R. Sabourin, P. Maupin, A dynamic overproduce-and-choose strategy for the selection of classifier ensembles, Pattern Recognit. 41 (10) (2008) 2993–3009.
[168] A. Roy, R.M.O. Cruz, R. Sabourin, G.D.C. Cavalcanti, Meta-learning recommendation of default size of classifier pool for META-DES, vol. 216, 2016, pp. 351–362.
[169] S. Garcia, J. Derrac, J. Cano, F. Herrera, Prototype selection for nearest neighbor classification: taxonomy and empirical study, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 417–435.
[170] I. Triguero, J. Derrac, S. García, F. Herrera, A taxonomy and experimental study on prototype generation for nearest neighbor classification, IEEE Trans. Syst. Man Cybern. Part C 42 (1) (2012) 86–100.
[171] R. Lysiak, M. Kurzynski, T. Woloszynski, Probabilistic approach to the dynamic ensemble selection using measures of competence and diversity of base classifiers, in: International Conference on Hybrid Artificial Intelligence Systems, 2011, pp. 229–236.
[172] T.K. Ho, Complexity of classification problems and comparative advantages of combined classifiers, in: International Workshop on Multiple Classifier Systems, Springer, 2000, pp. 97–106.
[173] B. Krawczyk, M. Woźniak, G. Schaefer, Cost-sensitive decision tree ensembles for effective imbalanced classification, Appl. Soft Comput. 14 (2014) 554–562.
[174] Y. Sun, M.S. Kamel, A.K. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognit. 40 (12) (2007) 3358–3378.
[175] S. Bernard, C. Chatelain, S. Adam, R. Sabourin, The multiclass roc front method for cost-sensitive classification, Pattern Recognit. 52 (2016) 46–60.
[176] C. Dubos, S. Bernard, S. Adam, R. Sabourin, Roc-based cost-sensitive classification with a reject option, in: International Conference on Pattern Recognition (ICPR), 2016, pp. 3320–3325.
[177] J. Díez-Pastor, J.J. Rodríguez, C. García-Osorio, L.I. Kuncheva, Diversity techniques improve the performance of the best imbalance learning ensembles, Inf. Sci. 325 (2015) 98–117.