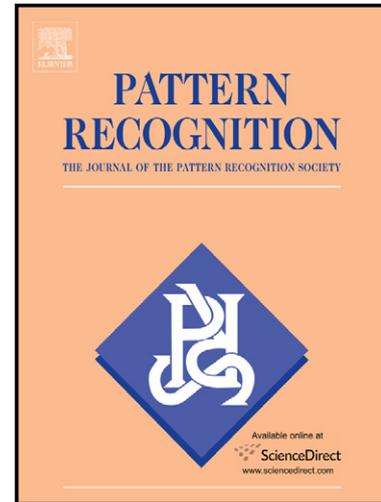


# Author's Accepted Manuscript

META-DES: A Dynamic Ensemble Selection Framework using Meta-Learning

Rafael M.O. Cruz, Robert Sabourin, George D.C. Cavalcanti, Tsang Ing Ren



[www.elsevier.com/locate/pr](http://www.elsevier.com/locate/pr)

PII: S0031-3203(14)00491-9  
DOI: <http://dx.doi.org/10.1016/j.patcog.2014.12.003>  
Reference: PR5293

To appear in: *Pattern Recognition*

Received date: 9 April 2014  
Revised date: 24 October 2014  
Accepted date: 2 December 2014

Cite this article as: Rafael M.O. Cruz, Robert Sabourin, George D.C. Cavalcanti, Tsang Ing Ren, META-DES: A Dynamic Ensemble Selection Framework using Meta-Learning, *Pattern Recognition*, <http://dx.doi.org/10.1016/j.patcog.2014.12.003>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting galley proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

# META-DES: A Dynamic Ensemble Selection Framework using Meta-Learning

Rafael M. O. Cruz<sup>a,\*</sup>, Robert Sabourin<sup>a</sup>, George D. C. Cavalcanti<sup>b</sup>, Tsang Ing Ren<sup>b</sup>

<sup>a</sup>LIVIA, École de Technologie Supérieure, University of Quebec, Montreal, Que., Canada - [www.livia.etsmtl.ca](http://www.livia.etsmtl.ca)

<sup>b</sup>Centro de Informática, Universidade Federal de Pernambuco, Recife, PE, Brazil - [www.cin.ufpe.br/~viisar](http://www.cin.ufpe.br/~viisar)

---

## Abstract

Dynamic ensemble selection systems work by estimating the level of competence of each classifier from a pool of classifiers. Only the most competent ones are selected to classify a given test sample. This is achieved by defining a criterion to measure the level of competence of a base classifier, such as, its accuracy in local regions of the feature space around the query instance. However, using only one criterion about the behavior of a base classifier is not sufficient to accurately estimate its level of competence. In this paper, we present a novel dynamic ensemble selection framework using meta-learning. We propose five distinct sets of meta-features, each one corresponding to a different criterion to measure the level of competence of a classifier for the classification of input samples. The meta-features are extracted from the training data and used to train a meta-classifier to predict whether or not a base classifier is competent enough to classify an input instance. During the generalization phase, the meta-features are extracted from the query instance and passed down as input to the meta-classifier. The meta-classifier estimates, whether a base classifier is competent enough to be added to the ensemble. Experiments are conducted over several small sample size classification problems, i.e., problems with a high degree of uncertainty due to the lack of training data. Experimental results show the proposed meta-learning framework greatly improves classification accuracy when compared against current state-of-the-art dynamic ensemble selection techniques.

*Keywords:*

Ensemble of Classifiers, Dynamic Ensemble Selection, Meta-Learning, Classifier competence

---

## 1. Introduction

Multiple Classifier Systems (MCS) aim to combine classifiers to increase the recognition accuracy in pattern recognition systems [1, 2]. MCS are composed of three phases [3]: (1) Generation, (2) Selection and (3) Integration. In the first phase, a pool of classifiers is generated. In the second phase, a single classifier or a subset having the best classifiers of the pool is(are) selected. We refer to the subset of classifiers as Ensemble of Classifiers (EoC). The last phase is the integration, and the predictions of the selected classifiers are combined to obtain the final decision [1].

For the second phase, there are two types of selection approaches: static and dynamic. In static approaches, the selection is performed during the training stage of the system. Then, the selected classifier or EoC is used for the classification of all unseen test samples. In contrast, dynamic ensemble selection approaches (DES) [4, 5, 6, 7, 8, 9, 10, 11, 12, 13] select a different classifier or a different EoC for each new test sample. DES techniques rely on the assumption that each base classifier is an expert in a different local region of the feature space [14]. So, given a new test sample, DES techniques aim to select the most competent classifiers for the local region in the feature space where the test sample is located. Only the classifiers that attain a certain competence level, according to a selection criterion, are selected. Recent work in the dynamic selection literature demonstrates that dynamic selection techniques is an effective tool for classification problems that are ill-defined, i.e., for problems where the size of the training data is small and there are not enough data available to model the classifiers [6, 7].

The key issue in DES is to define a criterion to measure the level of competence of a base classifier. Most DES techniques [4, 12, 11, 10, 15, 16, 17, 18] use estimates of the classifiers' local accuracy in small regions of the feature space surrounding the query instance as a search criterion to perform the ensemble selection. However, in our previous work [10], we demonstrated that the use of local accuracy estimates alone is insufficient to achieve results close to the Oracle performance. The Oracle is an abstract model defined in [19] which always selects the classifier that predicted the correct label, for the given query sample, if such classifier exists. In other

---

\*Corresponding author. Email Address: rafaelmenelau@gmail.com.

*Email addresses:* rmoc@cin.ufpe.br (Rafael M. O. Cruz), robert.sabourin@etsmt1.ca (Robert Sabourin), gdcc@cin.ufpe.br (George D. C. Cavalcanti), tir@cin.ufpe.br (Tsang Ing Ren)

words, it represents the ideal classifier selection scheme. In addition, as reported by Ko et al. [4], addressing the behavior of the Oracle is much more complex than applying a simple neighborhood approach.

On the other hand, DES techniques based on other criteria, such as the degree of consensus of the ensemble classifiers [5, 6], encounter some problems when the search cannot find a consensus among the ensembles. In addition, they neglect the local performance of the base classifiers. As stated by the “No Free Lunch” theorem [20], no algorithm is better than any other over all possible classes of problems. Using a single criterion to measure the level of competence of a base classifier is very error-prone. Thus, we believe that multiple criteria to measure the competence of a base classifier should be taken into account in order to achieve a more robust dynamic ensemble selection technique.

In this paper, we propose a novel dynamic ensemble selection framework using meta-learning. From the meta-learning perspective, the dynamic ensemble selection problem is considered as another classification problem, called meta-problem. The meta-features of the meta-problem are the different criteria used to measure the level of competence of the base classifier. We propose five sets of meta-features in this paper. Each set captures a different property about the behavior of the base classifier, and can be seen as a different dynamic selection criterion such as, the classification performance in a local region of the feature space and the classifier confidence for the classification of the input sample. Using five distinct sets of meta-features, even though one criterion might fail due to problems in the local regions of the feature space [10] or due to low confidence results [21], the system can still achieve a good performance as other meta-features are also considered by the selection scheme. Furthermore, in a recent analysis [22] we compared the criteria used to measure the competence of base classifiers embedded in different DES techniques. The result demonstrates that, given the same query sample, distinct DES criteria select a different base classifier as the most competent one. Thus, they are not fully correlated. Hence, we believe that a more robust dynamic ensemble selection technique is achieved using five sets of meta-features rather than only one.

The meta-features are used as input to a meta-classifier that decides whether or not a base classifier is competent enough for the classification of an input sample based on the meta-features. The use of meta-learning has recently been proposed in [23] as an alternative for performing

classifier selection in static scenarios. We believe that we can carry this further, and extend the use of meta-learning to dynamically estimate the level of competence of a base classifier.

The proposed framework is divided into three phases: overproduction, meta-training and generalization. In the overproduction stage, a pool of classifiers is generated using the training data. In the meta-training stage, the five sets of meta-features are extracted from the training data, and are used to train the meta-classifier that works as the classifier selector. During the generalization phase, the meta-features are extracted from the query instance and passed down as inputs to the meta-classifier. The meta-classifier estimates whether a base classifier is competent enough to classify the given test instance. Thus, the proposed system differs from the current state-of-the-art dynamic selection techniques not only because it uses multiple criteria to perform the classifier selection, but also because the classifier selection rule is learned by the meta-classifier using the training data.

The generalization performance of the system is evaluated over 30 classification problems. We compare the proposed framework against eight state-of-the-art dynamic selection techniques as well as static combination methods. The evaluation is focused on small size dataset, since DES techniques has shown to be an effective tool for problems where the level of uncertainty for recognition is high due to few training samples [6]. However, a few larger datasets were also considered in order to evaluate the performance of the proposed framework under different conditions. The goal of the experiments is to answer the following research questions: (1) Can the use of multiple DES criteria, as meta-features, lead to a more robust dynamic selection technique? (2) Does the proposed framework outperform current DES techniques for ill-defined problems?

This paper is organized as follows: Section 2 introduces the notion of classifier competence, and the state-of-the-art techniques for dynamically measuring the classifiers' competence are presented. The proposed framework is presented in Section 3. The experimental study is conducted in Section 4. Finally, our conclusion is presented in the last section.

## **2. Classifier competence for dynamic selection**

Classifier competence defines how much we trust an expert, given a classification task. The notion of competence used is extensively in the field of machine learning as a way of selecting,

from the plethora of different classification models, the one that best fits the given problem. Let  $C = \{c_1, \dots, c_M\}$  ( $M$  is the size of the pool of classifiers) be the pool of classifiers and  $c_i$  a base classifier belonging to the pool  $C$ . The goal of dynamic selection is to find an ensemble of classifiers  $C' \subset C$  that has the best classifiers to classify a given test sample  $\mathbf{x}_j$ . This is different from static selection, where the ensemble of classifiers  $C'$  is selected during the training phase, and considering the global performance of the base classifiers over a validation dataset [24, 25, 26, 27].

Nevertheless, the key issue in dynamic selection is how to measure the competence of a base classifier  $c_i$  for the classification of a given query sample  $\mathbf{x}_j$ . In the literature, we can observe three categories: the classifier accuracy over a local region, i.e., in a region of the feature space surrounding the query instance  $\mathbf{x}_j$ , decision templates [28], which are techniques that work in the decision space (i.e, a space defined by the outputs of the base classifiers) and the extent of consensus or confidence. The three categories are described in the following subsections.

### 2.1. Classifier accuracy over a local region

Classifier accuracy is the most commonly used criterion for dynamic classifier and ensemble selection techniques [12, 4, 10, 17, 13, 15, 29, 16, 9]. Techniques that are based on local accuracy first define a small region in the feature space surrounding a given test instance  $\mathbf{x}_j$ , called the region of competence. This region is computed using either the K-NN algorithm [4, 12, 10] or by Clustering techniques [17, 13], and can be defined either in the training set [12] or in the validation set, such as in the KNORA techniques [4].

Based on the samples belonging to the region of competence, a criterion is applied in order to measure the level of competence of a base classifier. For example, the Overall Local Accuracy (OLA) [12] technique uses the accuracy of the base classifier in the whole region of competence as a criterion to measure its level of competence. The classifier that obtains the highest accuracy rate is considered the most competent one. The Local Classifier Accuracy (LCA) [12] computes the performance of the base classifier in relation to a specific class label using a posteriori information [30]. The Modified Local Accuracy [16] works similarly to the LCA technique, with the only difference being that each sample belonging to the region of competence is weighted by its Euclidean distance to the query instance. That way, instances from the region of competence that

are closer to the test sample have a higher influence when computing the performance of the base classifier. The classifier rank method [29] uses the number of consecutive correctly classified samples as a criterion to measure the level of competence. The classifier that correctly classifies the most consecutive samples coming from the region of competence is considered to have the highest competence level or “rank”.

Ko et al. [4] proposed the K-Nearest Oracles (KNORA) family of techniques, inspired by the Oracle concept. Four techniques are proposed: the KNORA-Eliminate (KNORA-E) which, considers that a base classifier  $c_i$  is competent for the classification of the query instance  $\mathbf{x}_j$  if  $c_i$  achieves a perfect accuracy for the whole region of competence. Only the base classifiers with a perfect accuracy are used during the voting scheme. In The KNORA-Union (KNORA-U) technique, the level of competence of a base classifier  $c_i$  is measured by the number of correctly classified samples in the defined region of competence. In this case, every classifier that correctly classified at least one sample can submit a vote. In addition, two weighted versions, KNORA-E-W and KNORA-U-W were also proposed, in which the influence of each sample belonging to the region of competence was weighted based on its Euclidean distance to the query sample  $\mathbf{x}_j$ . Lastly, Xiao et al. [9] proposed the Dynamic Classifier Ensemble for Imbalanced Data (DCEID), which is based on the same principles as the LCA technique. However, this technique also takes into account each class prior probability when computing the performance of the base classifier for the defined region of competence in order to deal with imbalanced distributions.

The difference between these techniques lies in how they utilize the local accuracy information in order to measure the level of competence of a base classifier. The main issue with the techniques arises from the fact that they depend on the performance of the techniques that define the region of competence such as K-NN or clustering techniques. In our previous work [10], we demonstrated that the effectiveness of dynamic selection techniques is limited by the performance of the algorithm that defines the region of competence. The dynamic selection technique is likely to commit errors when outlier instances (i.e., mislabelled samples) exists around the query sample in the feature space [10]. Using the local accuracy information alone is not sufficient to achieve results close to the Oracle. Moreover, any difference between the distribution of validation and test datasets may negatively affect the system performance. Consequently, we believe that additional

information should also be considered.

## 2.2. Decision Templates

In this class of methods, the goal is also to select samples that are close to the query instance  $\mathbf{x}_j$ . However, the similarity is computed over the decision space through the concept of decision templates [28]. This is performed by transforming both the test instance  $\mathbf{x}_j$  and the validation data into output profiles. The output profile of an instance  $\mathbf{x}_j$  is denoted by  $\tilde{\mathbf{x}}_j = \{\tilde{\mathbf{x}}_{j,1}, \tilde{\mathbf{x}}_{j,2}, \dots, \tilde{\mathbf{x}}_{j,M}\}$ , where each  $\tilde{\mathbf{x}}_{j,i}$  is the decision yielded by the base classifier  $c_i$  for the sample  $\mathbf{x}_j$ .

Based on the information extracted from the decision space, the K-Nearest Output Profile (KNOP) [7] is similar to the KNORA technique, with the difference being that the KNORA works in the feature space, while the KNOP works in the decision space. The KNOP technique first defines a set with the samples that are most similar to the output profile of the input sample,  $\tilde{\mathbf{x}}_j$  in the decision space, called the output profiles set. The validation set is used for this purpose. Then, similarly to the KNORA-E technique, only the base classifiers that achieve a perfect recognition accuracy for the samples belonging to the output profiles set are used during the voting scheme. The Multiple Classifier Behaviour (MCB) technique [11] also defines a set with the most similar output profiles to the input sample using the decision space. Here, the selection criterion is based on a threshold. The base classifiers that achieve a performance higher than the predefined threshold are considered competent and are selected to form the ensemble.

The advantage of this class of methods is that they are not limited by the quality of the region of competence defined in the feature space, with the similarity computed based on the decision space rather than the feature space. However, the disadvantage with this comes from the fact that only global information is considered, while the local expertise of each base classifier is neglected.

## 2.3. Extent of Consensus or confidence

Different from other methods, techniques that are based on the extent of consensus work by considering a pool of ensemble of classifiers (EoC) rather than a pool of classifiers. Hence, the first step is to generate a population of EoC,  $C^* = \{C'_1, C'_2, \dots, C'_{M'}\}$  ( $M'$  is the number of EoC generated) using an optimization algorithm such as genetic algorithms or greedy search [26, 5, 27].

Then, for each new query instance  $x_j$ , the level of competence of an ensemble of classifiers  $C'_i$  is equal to the extent of consensus among its base classifiers.

Several criterion based on this paradigm was proposed: the Margin-based Dynamic Selection (MDS) [5], where the criterion is the margin between the most voted class and the second most voted class. The margin is computed simply by considering the difference between the number of votes received by the most voted class and those received by the second most voted class. Two variations of the MDS were proposed in [5], the Class-Strength Dynamic Selection (CSDS), which includes the ensemble decision in the computation of the MDS, and the GSDS, where the global performance of each EoC is also taken into account [6]. Another technique from this paradigm is the Ambiguity-guided Dynamic Selection (ADS) [5], which uses the ambiguity among the base classifiers of an EoC as the criterion for measuring the competence level of an EoC. The ambiguity is calculated by the number of base classifiers of an ensemble that disagrees with the ensemble decision. The lower the number of classifiers that disagree with the ensemble decision, the higher the level of competence of the EoC.

The greatest advantage of this class of methods stems from the fact that it does not require information from the region of competence. Thus, it does not suffer from the limitations of the algorithm that defines the region of competence. However, these techniques present the following disadvantages: In many cases, the search cannot find an EoC with an acceptable confidence level. There is a tie between different members of the pool, and the systems end up performing a random decision [6]. In addition, some classifiers are more overtrained than others. In this case, they end up dominating the outcome even though they do not present better recognition performance [31]. The pre-computation of ensembles also greatly increases the overall system complexity as we are dealing with a pool of EoC rather than a pool of classifiers.

### 3. The Proposed Framework: META-DES

#### 3.1. Problem definition

From the meta-learning perspective, the dynamic selection problem can be seen as another classification problem, called the meta-problem. This meta-problem uses different criteria regarding the behavior of a base classifier in order to decide whether it is competent enough to classify

a given sample  $\mathbf{x}_j$ . Thus, a dynamic selection system can be defined based on two environments. A classification environment in which the input features are mapped into a set of class labels  $w = \{w_1, w_2, \dots, w_L\}$  and a meta-classification environment in which information about the behavior of the base classifier is extracted from the classification environment and used to decide whether a base classifier  $c_i$  is competent enough to classify  $\mathbf{x}_j$ .

To keep with the conventions of the meta-learning literature, we define the proposed dynamic ensemble selection in a meta-learning framework as follows:

- The **meta-problem** consists in defining whether a base classifier  $c_i$  is competent enough to classify  $\mathbf{x}_j$ .
- The **meta-classes** of this meta-problem are either “competent” or “incompetent” to classify  $\mathbf{x}_j$ .
- Each **meta-feature**  $f_i$  corresponds to a different criterion to measure the level of competence of a base classifier.
- The meta-features are encoded into a **meta-features vector**  $v_{i,j}$  which contains the information about the behavior of a base classifier  $c_i$  in relation to the input instance  $\mathbf{x}_j$ .
- A **meta-classifier**  $\lambda$  is trained based on the meta-features  $v_{i,j}$  to predict whether or not  $c_i$  will achieve the correct prediction for  $\mathbf{x}_j$ .

In other words, a meta-classifier  $\lambda$  is trained, based on  $v_{i,j}$ , to predict whether a base classifier  $c_i$  is competent enough to classify given a test sample  $\mathbf{x}_j$ . Thus, the proposed system differs from the current state-of-the-art dynamic selection techniques not only because it uses multiple criteria, but also because the selection rule is learned by the meta-classifier  $\lambda$  using the training data.

### 3.2. The proposed META-DES

The META-DES framework is divided into three phases (Figure 1):

1. The overproduction phase, where the pool of classifiers  $C = \{c_1, \dots, c_M\}$ , composed of  $M$  classifiers, is generated using the training instances  $\mathbf{x}_{j,train}$  from the dataset  $\mathcal{T}$ .

2. The meta-training stage, in which samples  $\mathbf{x}_{j,train_\lambda}$  from the meta-training dataset  $\mathcal{T}_\lambda$  are used to extract the meta-features. A different dataset  $\mathcal{T}_\lambda$  is used in this phase in order to prevent overfitting. The meta-feature vectors  $v_{i,j}$  are stored in the set  $\mathcal{T}_\lambda^*$  that is later used to train the meta-classifier  $\lambda$ .
3. The generalization phase, given a test sample  $\mathbf{x}_{j,test}$  resulting from the generalization data  $\mathcal{G}$ ; its region of competence is extracted using the samples from the dynamic selection dataset  $D_{SEL}$  in order to compute the meta-features. The meta-feature vector  $v_{i,j}$  is then passed to the selector  $\lambda$ , which decides whether  $c_i$  is competent enough to classify  $\mathbf{x}_{j,test}$  and should be added to the ensemble,  $C'$ . The majority vote rule is applied over the ensemble  $C'$ , giving the classification  $w_l$  of  $\mathbf{x}_{j,test}$ .

### 3.2.1. Overproduction

In this work, the Overproduction phase is performed using the Bagging technique [32, 33]. Bagging is an acronym for Bootstrap AGGREGatING. The idea behind this technique is to build a diverse ensemble of classifiers by randomly selecting different subsets of the training data. Each subset is used to train one individual classifier  $c_i$ . As the focus of the paper is on classifier selection, and not on classifier generation methods, only the bagging technique is considered.

### 3.2.2. Meta-training

As shown in Figure 1, the meta-training stage consists of three steps: the sample selection process, the meta-features extraction process, and the training of the meta-classifier  $\lambda$ . For every sample  $\mathbf{x}_{j,train_\lambda} \in \mathcal{T}_\lambda$ , the first step is to apply the sample selection mechanism in order to know whether or not  $\mathbf{x}_{j,train_\lambda}$  should be used for the training of the meta-classifier  $\lambda$ . The whole Meta-training phase is formalized in Algorithm 1.

#### 3.2.2.1 Sample Selection

As demonstrated by Dos Santos et al. [5] and Cavalin et al. [6], one of the main issues in dynamic ensemble selection arises when classifying testing instances where the degree of consensus among the pool of classifier is low, i.e., when the number of votes from the winning class is close or even

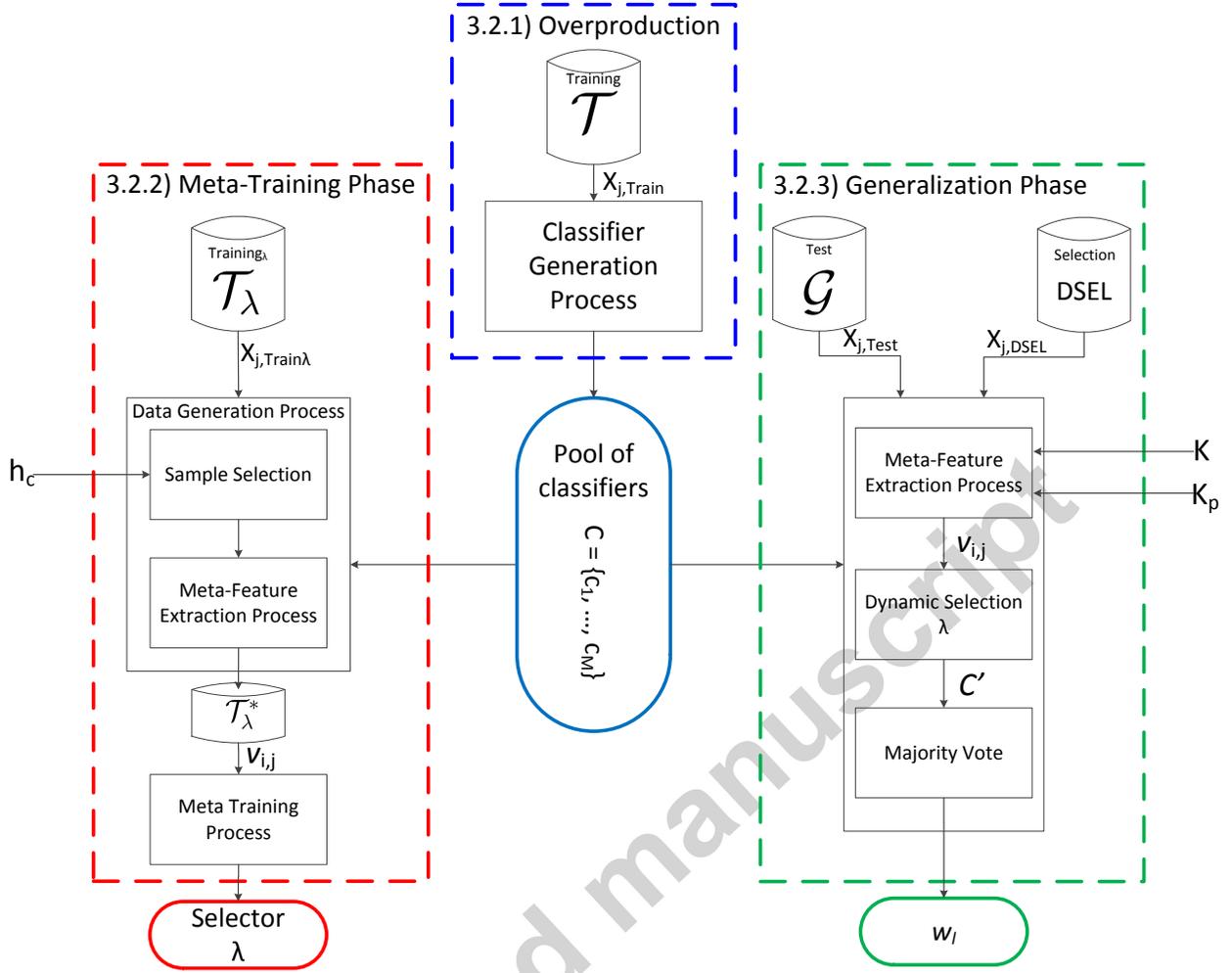


Figure 1: Overview of the proposed META-DES framework. It is divided into three steps 1) Overproduction, where the pool of classifiers  $C = \{c_1, \dots, c_M\}$  is generated, 2) The training of the meta-classifier  $\lambda$ , and 3) The generalization phase where an ensemble  $C'$  is dynamically defined based on the meta-information extracted from  $\mathbf{x}_{j,test}$  and the pool  $C = \{c_1, \dots, c_M\}$ . The generalization phase returns the label  $w_l$  of  $\mathbf{x}_{j,test}$ .  $h_C$ ,  $K$  and  $K_p$  are the hyper-parameters required by the proposed system.

equal to the number of votes from the second class. To tackle this issue, we decided to focus the training of the meta-classifier  $\lambda$  to specifically deal with cases where the extent of consensus among the pool is low. This step is conducted using a threshold  $h_C$ , called the consensus threshold. Each instance  $\mathbf{x}_{j,train_\lambda}$  is first evaluated by the whole pool of classifiers in order to compute the degree of consensus among the pool, denoted by  $H(\mathbf{x}_{j,train_\lambda}, C)$ . If the consensus  $H(\mathbf{x}_{j,train_\lambda}, C)$  falls below the consensus threshold  $h_C$ , the instance  $\mathbf{x}_{j,train_\lambda}$  is used to compute the meta-features.

**Algorithm 1** The Meta-Training Phase

---

**Input:** Training data  $\mathcal{T}_\lambda$   
**Input:** Pool of classifiers  $C = \{c_1, \dots, c_M\}$

- 1:  $\mathcal{T}_\lambda^* = \emptyset$
- 2: **for all**  $\mathbf{x}_{j,train_\lambda} \in \mathcal{T}_\lambda$  **do**
- 3:   Compute the consensus of the pool  $H(\mathbf{x}_{j,train_\lambda}, C)$
- 4:   **if**  $H(\mathbf{x}_{j,train_\lambda}, C) < h_C$  **then**
- 5:     Find the region of competence  $\theta_j$  of  $\mathbf{x}_{j,train_\lambda}$  using  $\mathcal{T}_\lambda$ .
- 6:     Compute the output profile  $\tilde{\mathbf{x}}_{j,train_\lambda}$  of  $\mathbf{x}_{j,train_\lambda}$ .
- 7:     Find the  $K_p$  similar output profiles  $\phi_j$  of  $\tilde{\mathbf{x}}_{j,train_\lambda}$  using  $\tilde{\mathcal{T}}_\lambda$ .
- 8:     **for all**  $c_i \in C$  **do**
- 9:        $v_{i,j} = MetaFeatureExtraction(\theta_j, \phi_j, c_i, \mathbf{x}_{j,train_\lambda})$
- 10:       **if**  $c_i$  correctly classifies  $\mathbf{x}_{j,train_\lambda}$  **then**
- 11:          $\alpha_{i,j} = 1$  “ $c_i$  is competent for  $\mathbf{x}_{j,train_\lambda}$ ”
- 12:       **else**
- 13:          $\alpha_{i,j} = 0$  “ $c_i$  is incompetent for  $\mathbf{x}_{j,train_\lambda}$ ”
- 14:       **end if**
- 15:        $\mathcal{T}_\lambda^* = \mathcal{T}_\lambda^* \cup \{v_{i,j}\}$
- 16:     **end for**
- 17:   **end if**
- 18: **end for**
- 19: Divide  $\mathcal{T}_\lambda^*$  into 25% for validation and 75% for training.
- 20: Train  $\lambda$  using the Levenberg-Marquadt algorithm.
- 21: **return** The meta-classifier  $\lambda$ .

---

Before extracting the meta-features, the region of competence of the instance  $\mathbf{x}_{j,train_\lambda}$ , denoted by  $\theta_j = \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$ , must first be computed. The region of competence  $\theta_j$  is defined in the  $\mathcal{T}_\lambda$  set, using the K-Nearest Neighbor algorithm (line 5). Then,  $\mathbf{x}_{j,train_\lambda}$  is transformed into an output profile. The output profile of the instance  $\mathbf{x}_{j,train_\lambda}$  is denoted by  $\tilde{\mathbf{x}}_{j,train_\lambda} = \{\tilde{\mathbf{x}}_{j,train_\lambda,1}, \tilde{\mathbf{x}}_{j,train_\lambda,2}, \dots, \tilde{\mathbf{x}}_{j,train_\lambda,M}\}$ , where each  $\tilde{\mathbf{x}}_{j,train_\lambda,i}$  is the decision yielded by the base classifier  $c_i$  for the sample  $\mathbf{x}_{j,train_\lambda}$  [6, 7].

Next, with the region of competence  $\theta_j$  and the set with the most similar output profiles  $\phi_j$  computed, for each base classifier  $c_i$  belonging to the pool of classifiers  $C$ , one meta-feature vector  $v_{i,j}$  is extracted (lines 8 to 14). Each  $v_{i,j}$  contains five sets of meta-features:

### 3.2.2.2 Meta-feature extraction process

Five different sets of meta-features are proposed in this work. Each feature set  $f_i$ , corresponds to a different criterion for measuring the level of competence of a base classifier. Each set captures a different property about the behavior of the base classifier, and can be seen as a different criterion to dynamically estimate the level of competence of base classifier such as, the classification performance estimated in a local region of the feature space and the classifier confidence for the classification of the input sample. Using five distinct sets of meta-features, even though one criterion might fail due to imprecisions in the local regions of the feature space or due to low confidence results, the system can still achieve a good performance as other meta-features are considered by the selection scheme. Table 1 shows the criterion used by each  $f_i$  and its relationship with one dynamic ensemble selection paradigm presented in Section 2.

Table 1: Relationship between each meta-features and different paradigms to compute the level of competence of a base classifier.

Meta-Feature	Criterion	Paradigm
$f_1$	Local accuracy in the region of competence	Classifier Accuracy over a local region
$f_2$	Extent of consensus in the region of competence	Classifier consensus
$f_3$	Overall accuracy in the region of competence	Accuracy over a local region
$f_4$	Accuracy in the decision space	Decision Templates
$f_5$	Degree of confidence for the input sample	Classifier confidence

Three meta-features,  $f_1$ ,  $f_2$  and  $f_3$ , are computed using information extracted from the region of competence  $\theta_j$ .  $f_4$  uses information extracted from the set of output profiles  $\phi_j$ .  $f_5$  is calculated directly from the input sample  $\mathbf{x}_{j,train_\lambda}$ , and corresponds to the level of confidence of  $c_i$  for the classification of  $\mathbf{x}_{j,train_\lambda}$ .

**$f_1$  - Neighbors' hard classification:** First, a vector with  $K$  elements is created. For each instance  $\mathbf{x}_k$ , belonging to the region of competence  $\theta_j$ , if  $c_i$  correctly classifies  $\mathbf{x}_k$ , the  $k$ -th position of the vector is set to 1, otherwise it is 0. Thus,  $K$  meta-features are computed.

**$f_2$  - Posterior probability:** First, a vector with  $K$  elements is created. Then, for each instance

$\mathbf{x}_k$ , belonging to the region of competence  $\theta_j$ , the posterior probability of  $c_i$ ,  $P(w_l | \mathbf{x}_k)$  is computed and inserted into the  $k$ -th position of the vector. Consequently,  $K$  meta-features are computed.

$f_3$  - **Overall Local accuracy:** The accuracy of  $c_i$  over the whole region of competence  $\theta_j$  is computed and encoded as  $f_3$ .

$f_4$  - **Output profiles classification:** First, a vector with  $K_p$  elements is generated. Then, for each member  $\tilde{\mathbf{x}}_k$  belonging to the set of output profiles  $\phi_j$ , if the label produced by  $c_i$  for  $\mathbf{x}_k$  is equal to the label  $w_{l,k}$  of  $\tilde{\mathbf{x}}_k$ , the  $k$ -th position of the vector is set to 1, otherwise it is 0. A total of  $K_p$  meta-features are extracted using output profiles.

$f_5$  - **Classifier's Confidence:** The perpendicular distance between the input sample  $\mathbf{x}_{j,train_\lambda}$  and the decision boundary of the base classifier  $c_i$  is calculated and encoded as  $f_5$ .  $f_5$  is normalized to a  $[0 - 1]$  range using the Min-max normalization.

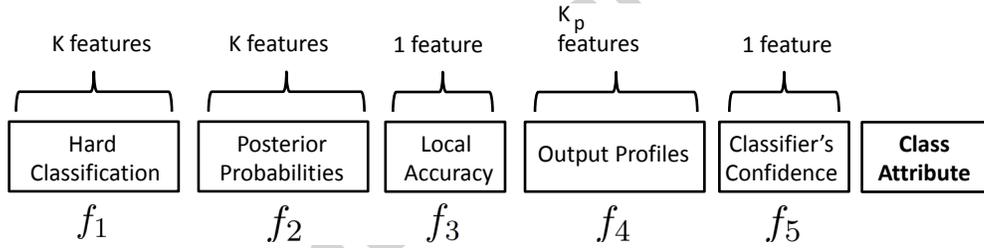


Figure 2: Feature Vector containing the meta-information about the behavior of a base classifier. A total of 5 different meta-features are considered. The size of the feature vector is  $(2 \times K) + K_p + 2$ . The class attribute indicates whether or not  $c_i$  correctly classified the input sample.

A vector  $v_{i,j} = \{f_1 \cup f_2 \cup f_3 \cup f_4 \cup f_5\}$  is obtained at the end of the process (Figure 2). If  $c_i$  correctly classifies  $\mathbf{x}_{j,train_\lambda}$ , the class attribute of  $v_{i,j}$ ,  $\alpha_{i,j} = 1$  (i.e.,  $v_{i,j}$  corresponds to the behavior of a competent classifier), otherwise  $\alpha_{i,j} = 0$ .  $v_{i,j}$  is stored in the meta-features dataset  $\mathcal{T}_\lambda^*$  (lines 10 to 16).

For each sample  $\mathbf{x}_{j,train_\lambda}$  used in the meta-training stage, a total of  $M$  ( $M$  is the size of the pool of classifiers  $C$ ) meta-feature vectors  $v_{i,j}$  are extracted, each one corresponding to one classifier from the pool  $C$ . In this way, the size of the meta-training dataset  $\mathcal{T}_\lambda^*$  is the pool size  $M \times$  number

of training samples  $N$ . For instance, consider that 200 training samples are available for the meta-training stage ( $N = 200$ ), if the pool  $C$  is composed of 100 weak classifiers ( $M = 100$ ), the meta-training dataset is the number of training samples  $N \times$  the number classifiers in the pool  $M$ ,  $N * M = 20.000$ . Hence, even though the classification problem may be ill-defined due to the size of the training set, we can overcome this limitation in the meta-problem by increasing the size of the pool of classifiers.

### 3.2.2.3 Training

The last step of the meta-training phase is the training of the meta-classifier  $\lambda$ . The dataset  $\mathcal{T}_\lambda^*$  is divided on the basis of 75% for training and 25% for validation. A Multi-Layer Perceptron (MLP) neural network is considered as the selector  $\lambda$ . The validation data was used to select the number of nodes in the hidden layer. We use a configuration of 10 neurons in the hidden layer since there were no improvement in results with more than 10 neurons. The training process for  $\lambda$  is performed using the Levenberg-Marquadt algorithm. In addition, the training process is stopped if its performance on the validation set decreases or fails to improve for five consecutive epochs.

### 3.2.3. Generalization Phase

The generalization procedure is formalized by Algorithm 2. Given the query sample  $\mathbf{x}_{j,test}$ , in this phase, the region of competence  $\theta_j$  is computed using the samples from the dynamic selection dataset  $D_{SEL}$  (line 2). Following that, the output profiles  $\tilde{\mathbf{x}}_{j,test}$  of the test sample,  $\mathbf{x}_{j,test}$ , are calculated. The set with  $K_p$  similar output profiles  $\phi_j$ , of the query sample  $\mathbf{x}_{j,test}$ , is obtained through the Euclidean distance applied over the output profiles of the dynamic selection dataset,  $\tilde{D}_{SEL}$ .

Next, for each classifier  $c_i$  belonging to the pool of classifiers  $C$ , the meta-feature extraction process is called (Section 3.2.2.2), returning the meta-features vector  $v_{i,j}$  (lines 5 and 6). Then,  $v_{i,j}$  is used as input to the meta-classifier  $\lambda$ . If the output of  $\lambda$  is 1 (i.e., competent),  $c_i$  is included in the ensemble  $C'$  (lines 8 to 10). After every base classifier,  $c_i$ , is evaluated, the ensemble  $C'$  is obtained. The base classifiers in  $C'$  are combined through the Majority Vote rule [1], giving the label  $w_l$  of  $x_{j,test}$  (line 12 and 13). The majority vote rule is used to combine the selected

---

**Algorithm 2** Classification steps using the selector  $\lambda$

---

**Input:** Query sample  $\mathbf{x}_{j,test}$

**Input:** Pool of classifiers  $C = \{c_1, \dots, c_M\}$

**Input:** dynamic selection dataset  $D_{SEL}$

```

1:  $C' = \emptyset$ 
2: Find the region of competence  $\theta_j$  of  $\mathbf{x}_{j,test}$  using  $D_{SEL}$ .
3: Compute the output profile  $\tilde{\mathbf{x}}_{j,test}$  of  $\mathbf{x}_{j,test}$ .
4: Find the  $K_p$  similar output profiles  $\phi_j$  of  $\tilde{\mathbf{x}}_{j,test}$  using  $\tilde{D}_{SEL}$ .
5: for all  $c_i \in C$  do
6:    $v_{i,j} = FeatureExtraction(\theta_j, \phi_j, c_i, \mathbf{x}_{j,test})$ 
7:   input  $v_{i,j}$  to  $\lambda$ 
8:   if  $\alpha_{i,j} = 1$  “ $c_i$  is competent for  $\mathbf{x}_{j,test}$ ” then
9:      $C' = C' \cup \{c_i\}$ 
10:  end if
11: end for
12:  $w_l = MajorityVote(\mathbf{x}_{j,test}, C')$ 
13: return  $w_l$ 

```

---

classifiers since it has been successfully used by other DES techniques [3]. Tie-breaking is handled by choosing the class with the highest a posteriori probability.

## 4. Experiments

### 4.1. Datasets

A total of 30 datasets are used in the comparative experiments. sixteen coming from the UCI machine learning repository [34], four from the STATLOG project [35], four from the Knowledge Extraction based on Evolutionary Learning (KEEL) repository [36], four from the Ludmila Kuncheva Collection of real medical data [37], and two artificial datasets generated with the Matlab PRTOOLS toolbox [38]. We consider both ill-defined problems, such as, Heart and Liver Disorders as well as larger databases, such as, Adult, Magic Gamma Telescope, Phoneme and WDG V1. The key features of each dataset are shown in Table 2.

### 4.2. Experimental Protocol

The experiments were conducted using 20 replications. For each replication, the datasets were randomly divided on the basis 50% for training, 25% for the dynamic selection dataset ( $D_{SEL}$ ),

Table 2: Key Features of the datasets used in the experiments.

Database	No. of Instances	Dimensionality	No. of Classes	Source
Pima	768	8	2	UCI
Liver Disorders	345	6	2	UCI
Breast (WDBC)	568	30	2	UCI
Blood transfusion	748	4	2	UCI
Banana	1000	2	2	PRTOOLS
Vehicle	846	18	4	STATLOG
Lithuanian	1000	2	2	PRTOOLS
Sonar	208	60	2	UCI
Ionosphere	315	34	2	UCI
Wine	178	13	3	UCI
Haberman's Survival	306	3	2	UCI
Cardiotocography (CTG)	2126	21	3	UCI
Vertebral Column	310	6	2	UCI
Steel Plate Faults	1941	27	7	UCI
WDG V1	50000	21	3	UCI
Ecoli	336	7	8	UCI
Glass	214	9	6	UCI
ILPD	214	9	6	UCI
Adult	48842	14	2	UCI
Weaning	302	17	2	LKC
Laryngeal1	213	16	2	LKC
Laryngeal3	353	16	3	LKC
Thyroid	215	5	3	LKC
German credit	1000	20	2	STATLOG
Heart	270	13	2	STATLOG
Satimage	6435	19	7	STATLOG
Phoneme	5404	6	2	ELENA
Monk2	4322	6	2	KEEL
Mammographic	961	5	2	KEEL
MAGIC Gamma Telescope	19020	10	2	KEEL

and 25% for the test set ( $\mathcal{G}$ ). The divisions were performed maintaining the priors probabilities of each class. For the proposed META-DES, 50% of the training data was used in the meta-training process  $\mathcal{T}_\lambda$  and 50% for the generation of the pool of classifiers ( $\mathcal{T}$ ).

For the two-class classification problems, the pool of classifiers was composed of 100 Perceptrons generated using the bagging technique [32]. For the multi-class problems, the pool of classifiers was composed of 100 multi-class perceptron classifier. The use of Perceptron as base classifier comes from the following observations based on past works in the literature:

- The use of weak classifiers can show more differences between the DES schemes [4]. Thus, making it a better option for comparing different DES techniques.
- Past works in the DES literature demonstrate that the use of weak models as base classifier achieve better results [5, 6, 39, 40, 10], where the use of decision trees or Perceptrons outperform strong classification models such as KNN classifiers.
- As reported by Leo Breiman [32, 33], the bagging technique achieves better results when weak and unstable base classifiers are used.

#### 4.3. Parameters Setting

The performance of the proposed selection scheme depends on three parameters: the neighborhood size,  $K$ , the number of similar patterns using output profiles  $K_p$  and the consensus threshold  $h_C$ . The dynamic selection dataset  $D_{SEL}$  was used for the analysis. The following methodology is used:

- For the sake of simplicity, we selected the parameters that performed best.
- The value of the parameter  $K$  was selected based on the results of our previous paper [10]. In this case,  $K = 7$  showed the best overall results, considering several dynamic selection techniques.
- The Kruskal-Wallis statistical test with a 95% confidence interval was used to determine whether the difference in results was statistically significant. If two configurations yielded

similar results, we selected the one with the smaller parameter value as it leads to a smaller meta-features vector.

- The parameter  $h_C$  was evaluated with  $K_p$  initially set at 1.
- The best value of  $h_c$  was used in the evaluation of the best value for  $K_p$ .
- Only a subset with eleven of the thirty datasets are used for parameters setting procedure: Pima, Liver, Breast, Blood Transfusion, Banana, Vehicle, Lithuanian, Sonar, Ionosphere, Wine, Haberman's Survival.

#### 4.3.1. The effect of the parameter $h_C$

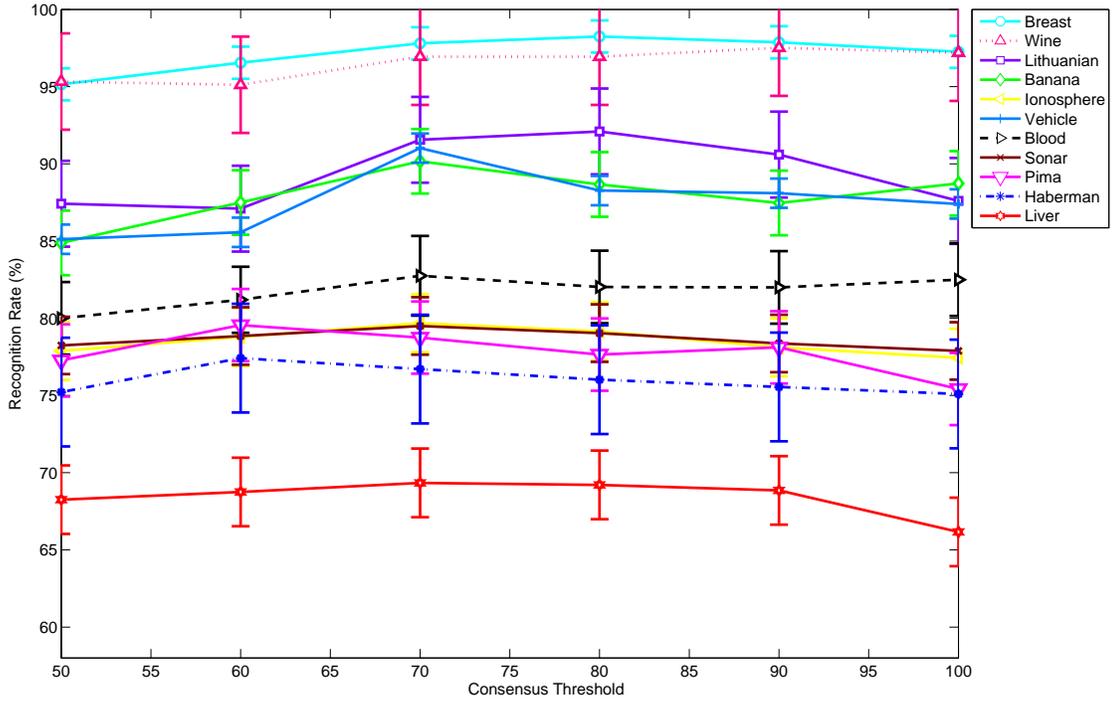


Figure 3: Performance of the proposed system based on the parameter  $h_C$  on the dynamic selection dataset,  $D_{SEL}$ .  $K = 7$  and  $K_p = 1$ .

We varied the parameter  $h_c$  from 50% to 100% at 10 percentile point interval. Figure 3 shows the mean performance and standard deviation for each  $h_C$  value. We compared each pair of results

using the Kruskal-Wallis non-parametric statistical test with a 95% confidence interval. For 6 out of 11 datasets (Vehicle, Lithuanian, Banana, Blood transfusion, Ionosphere and Sonar)  $h_C = 70\%$  presented a value that was statistically superior to the others. Hence,  $h_C = 70\%$  was selected.

#### 4.3.2. The effect of the parameter $K_p$

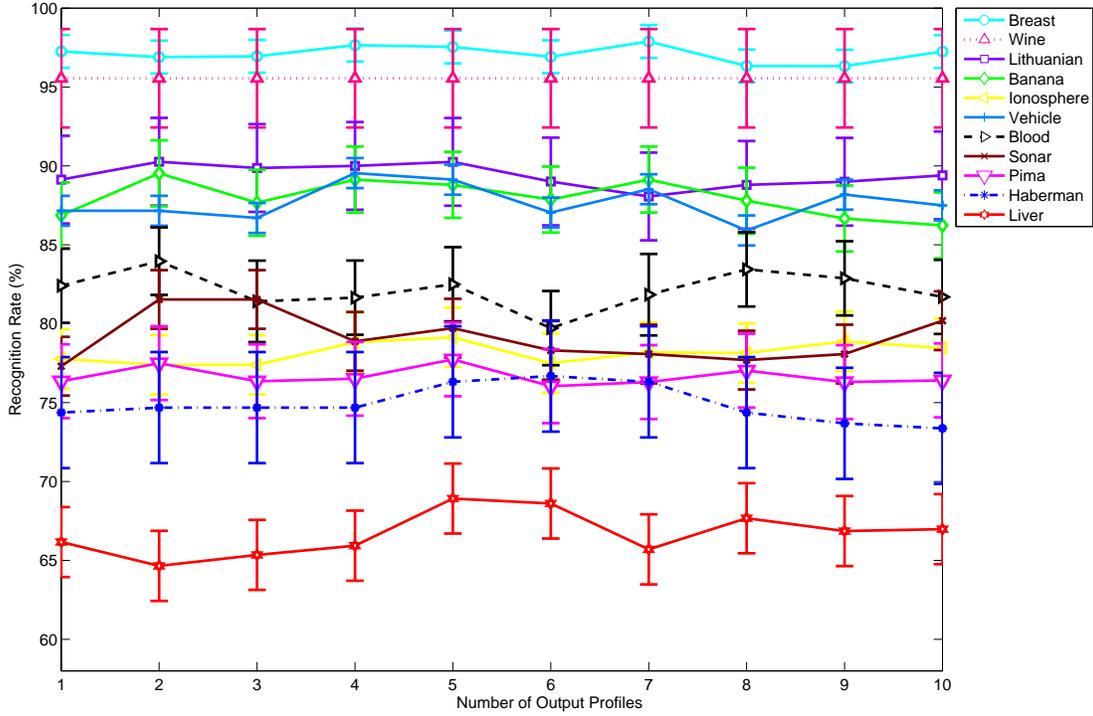


Figure 4: The performance of the system varying the parameter  $K_p$  from 1 to 10 on the dynamic selection dataset,  $D_{SEL}$ .  $h_c = 70\%$  and  $K = 7$

Figure 4 shows the impact of the value of the parameter  $K_p$  in an 1-to-10 range. Once again, we compared each pair of results using the Kruskal-Wallis non-parametric statistical test, with a 95% confidence. The results were statistically different only for the Sonar, Ionosphere and liver disorders datasets, where the value of  $K_p = 5$  showed the best results. Hence,  $K_p$  was set at 5.

#### 4.4. Comparison with the state-of-the-art dynamic selection techniques

In this section we compare the recognition rates obtained by the proposed META-DES, against eight dynamic selection techniques found in the literature [3]. The objective of this comparative study is to answer the following research question: (1) Can the use of multiple DES criteria as meta-features lead to a more robust dynamic selection technique? (2) Does the proposed framework outperform current DES techniques for ill-defined problems?

The eight state-of-the-art DES techniques used in this study are: the KNORA-ELIMINATE [4], KNORA-UNION [4], DES-FA [10], Local Classifier Accuracy (LCA) [12], Overall Local Accuracy (OLA) [12], Modified Local Accuracy (MLA) [16], Multiple Classifier Behaviour (MCB) [11] and K-Nearests Output Profiles (KNOP) [7, 6]. These techniques were selected because they presented the very best results in the dynamic selection literature according to a recent survey on this topic [3]. In addition, we also compare the performance of the proposed META-DES with static combination methods (Adaboost and Bagging), the classifier with the highest accuracy in the validation data (Single Best), static ensemble selection based on the majority voting error [41] and the abstract model (Oracle) [19]. The Oracle represents the ideal classifier selection scheme. It always selects the classifier that predicted the correct label, for any given query sample, if such classifier exists. For the static ensemble selection method, 50% of the classifiers of the pool are selected. The comparison against static methods is used since it is suggested the DES literature that the minimum requirement for a DES method is to surpass the performance of static selection and combination methods in the same pool [3].

For all techniques, the pool of classifiers  $C$  is composed of 100 Perceptrons as base classifier ( $M = 100$ ). For the state-of-the-art DES techniques (KNORA-E, KNORA-U, DES-FA, LCA, OLA, MLA, MCB and KNOP), the size of the region of competence (neighborhood size),  $K$  is set to 7, since it achieved the best result on previous publications [3, 10]. The size of the region of competence  $K$  is the only hyper-parameter required for the eight DES techniques. For the Adaboost and Bagging technique 100 iterations are used (i.e., 100 base classifier are generated).

We split the results in two tables: Table 3 shows a comparison with the proposed META-DES against the eight state-of-the-art dynamic selection techniques considered. A comparison of the

Table 3: Mean and standard deviation results of the accuracy obtained for the proposed DES<sub>D</sub> and the DES systems in the literature. A pool of 100 Perceptrons as base classifiers is used for all techniques. The best results are in bold. Results that are significantly better ( $p < 0.05$ ) are marked with a ●.

Database	META-DES	KNORA-E [4]	KNORA-U [4]	DES-FA [10]	LCA [12]	OLA [12]	MLA [16]	MCB [11]	KNOP [6]
Pima	<b>79.03(2.24) ●</b>	73.79(1.86)	76.60(2.18)	73.95(1.61)	73.95(2.98)	73.95(2.56)	77.08(4.56)	76.56(3.71)	73.42(2.11)
Liver Disorders	<b>70.08(3.49) ●</b>	56.65(3.28)	56.97(3.76)	61.62(3.81)	58.13(4.01)	58.13(3.27)	58.00(4.25)	58.00(4.25)	65.23(2.29)
Breast (WDBC)	97.41(1.07)	97.59(1.10)	97.18(1.02)	<b>97.88(0.78)</b>	97.88(1.58)	97.88(1.58)	95.77(2.38)	97.18(1.38)	95.42(0.89)
Blood Transfusion	<b>79.14(1.03) ●</b>	77.65(3.62)	77.12(3.36)	73.40(1.16)	75.00(2.87)	75.00(2.36)	76.06(2.68)	73.40(4.19)	77.54(2.03)
Banana	91.78(2.68)	93.08(1.67)	92.28(2.87)	<b>95.21(3.18)</b>	95.21(2.15)	95.21(2.15)	80.31(7.20)	88.29(3.38)	90.73(3.45)
Vehicle	82.75(1.70)	83.01(1.54)	82.54(1.70)	82.54(4.05)	80.33(1.84)	81.50(3.24)	74.05(6.65)	<b>84.90(2.01)</b>	80.09(1.47)
Lithuanian Classes	93.18(1.32)	93.33(2.50)	95.33(2.64)	98.00(2.46)	85.71(2.20)	<b>98.66(3.85)</b>	88.33(3.89)	86.00(3.33)	89.33(2.29)
Sonar	<b>80.55(5.39)</b>	74.95(2.79)	76.69(1.94)	78.52(3.86)	76.51(2.06)	74.52(1.54)	76.91(3.20)	76.56(2.58)	75.72(2.82)
Ionosphere	<b>89.94(1.96)</b>	89.77(3.07)	87.50(1.67)	88.63(2.12)	88.00(1.98)	88.63(1.98)	81.81(2.52)	87.50(2.15)	85.71(5.52)
Wine	<b>99.25(1.11) ●</b>	97.77(1.53)	97.77(1.62)	95.55(1.77)	85.71(2.25)	88.88(3.02)	88.88(3.02)	97.77(1.62)	95.50(4.14)
Haberman	<b>76.71(1.86)</b>	71.23(4.16)	73.68(2.27)	72.36(2.41)	70.16(3.56)	69.73(4.17)	73.68(3.61)	67.10(7.65)	75.00(3.40)
Cardiotocography (CTG)	84.62(1.08)	<b>86.27(1.57)</b>	85.71(2.20)	<b>86.27(1.57)</b>	86.65(2.35)	86.65(2.35)	86.27(1.78)	85.71(2.21)	86.02(3.04)
Vertebral Column	86.89(2.46)	85.89(2.27)	<b>87.17(2.24)</b>	82.05(3.20)	85.00(3.25)	85.89(3.74)	77.94(5.80)	84.61(3.95)	86.98(3.21)
Steel Plate Faults	67.21(1.20)	67.35(2.01)	67.96(1.98)	68.17(1.59)	66.00(1.69)	66.52(1.65)	67.76(1.54)	68.17(1.59)	<b>68.57(1.85)</b>
WDG V1	<b>84.56(0.36)</b>	84.01(1.10)	84.01(1.10)	84.01(1.10)	80.50(0.56)	80.50(0.56)	79.95(0.85)	78.75(1.35)	84.21(0.45)
Ecoli	77.25(3.52)	76.47(2.76)	75.29(3.41)	75.29(3.41)	75.29(3.41)	75.29(3.41)	76.47(3.06)	76.47(3.06)	<b>80.00(4.25) ●</b>
Glass	66.87(2.99)	57.65(5.85)	61.00(2.88)	55.32(4.98)	59.45(2.65)	57.60(3.65)	57.60(3.65)	<b>67.92(3.24)</b>	62.45(3.65)
ILPD	69.40(1.64)	67.12(2.35)	69.17(1.58)	67.12(2.35)	<b>69.86(2.20)</b>	<b>69.86(2.20)</b>	<b>69.86(2.20)</b>	68.49(3.27)	68.49(3.27)
Adult	<b>87.15(2.43) ●</b>	80.34(1.57)	79.76(2.26)	80.34(1.57)	83.58(2.32)	82.08(2.42)	80.34(1.32)	78.61(3.32)	79.76(2.26)
Weaning	<b>87.15(2.43) ●</b>	78.94(1.25)	81.57(3.65)	82.89(3.52)	77.63(2.35)	77.63(2.35)	80.26(1.52)	81.57(2.86)	82.57(3.33)
Laryngeal1	<b>79.67(3.78) ●</b>	77.35(4.45)	77.35(4.45)	77.35(4.45)	77.35(4.45)	77.35(4.45)	75.47(5.55)	77.35(4.45)	77.35(4.45)
Laryngeal3	72.65(2.17)	70.78(3.68)	72.03(1.89)	72.03(1.89)	72.90(2.30)	71.91(1.01)	61.79(7.80)	71.91(1.01)	<b>73.03(1.89)</b>
Thyroid	<b>96.78(0.87)</b>	95.95(1.25)	95.95(1.25)	95.37(2.02)	95.95(1.25)	95.95(1.25)	94.79(2.30)	95.95(1.25)	95.95(1.25)
German credit	<b>75.55(1.31) ●</b>	72.80(1.95)	72.40(1.80)	74.00(3.30)	73.33(2.85)	71.20(2.52)	71.20(2.52)	73.60(3.30)	73.60(3.30)
Heart	84.80(3.36)	83.82(4.05)	83.82(4.05)	83.82(4.05)	85.29(3.69)	85.29(3.69)	<b>86.76(5.50)</b>	83.82(4.05)	83.82(4.05)
Satimage	<b>96.21(0.87)</b>	95.35(1.23)	95.86(1.07)	93.00(2.90)	95.00(1.40)	94.14(1.07)	93.28(2.10)	95.86(1.07)	95.86(1.07)
Phoneme	<b>80.35(2.58)</b>	79.06(2.50)	78.92(3.33)	79.06(2.50)	78.84(2.53)	78.84(2.53)	64.94(7.75)	73.37(5.55)	78.92(3.33)
Monk2	<b>83.24(2.19) ●</b>	80.55(3.32)	77.77(4.25)	75.92(4.25)	74.07(6.60)	74.07(6.60)	75.92(5.65)	74.07(6.60)	80.55(3.32)
Mammographic	<b>84.82(1.55) ●</b>	82.21(2.27)	82.21(2.27)	80.28(3.02)	82.21(2.27)	82.21(2.27)	75.55(5.50)	81.25(2.07)	82.21(2.27)
MAGIC Gamma Telescope	<b>84.35(3.27) ●</b>	80.03(3.25)	79.99(3.55)	81.73(3.27)	81.53(3.35)	81.16(3.00)	73.13(6.35)	75.91(5.35)	80.03(3.25)

META-DES against static combination rules is shown in Table 4. Each pair of results is compared using the Kruskal-Wallis non-parametric statistical test, with a 95% confidence interval. The best results are in bold. Results that are significantly better ( $p < 0.05$ ) are marked with a ●.

We can see in Table 3 the proposed META-DES achieves results that are either superior or equivalent to the state-of-the-art DES techniques in 25 datasets (84% of the datasets). In addition, the META-DES achieved the highest recognition performance for 18 datasets, which corresponds to 60% of the datasets considered. Only for the Ecoli, Heart, Vehicle, Banana and Lithuanian datasets (16% of the datasets) the recognition rates of the proposed META-DES framework presented is statistically inferior to the best result achieved by state-of-the-art DES techniques.

Table 4: Mean and standard deviation results of the accuracy obtained for the proposed DES<sub>D</sub> and static ensemble combination. A pool of 100 Perceptrons as base classifier is used for all techniques. The best results are in bold. Results that are significantly better ( $p < 0.05$ ) are marked with a ●.

Database	META-DES	Single Best [3]	Bagging [32]	AdaBoost [42]	Static Selection [41]	Oracle [19]
Pima	<b>79.03(2.24) ●</b>	73.57(1.49)	73.28(2.08)	72.52(2.48)	72.86(4.78)	95.10(1.19)
Liver Disorders	<b>70.08(3.49) ●</b>	65.38(3.47)	62.76(4.81)	64.65(3.26)	59.18(7.02)	93.07(2.41)
Breast (WDBC)	97.41(1.07)	97.04(0.74)	96.35(1.14)	<b>98.24(0.89)</b>	96.83(1.00)	99.13(0.52)
Blood Transfusion	<b>79.14(1.03) ●</b>	75.07(1.83)	75.24(1.67)	75.18(2.08)	75.74(2.23)	94.20(2.08)
Banana	<b>91.78(2.68)</b>	84.07(2.22)	81.43(3.92)	81.61(2.42)	81.35(4.28)	94.75(2.09)
Vehicle	<b>82.75(1.70)</b>	81.87(1.47)	82.18(1.31)	80.56(4.51)	81.65(1.48)	96.80(0.94)
Lithuanian Classes	<b>93.18(1.32) ●</b>	84.35(2.04)	82.33(4.81)	82.70(4.55)	82.66(2.45)	98.35 (0.57)
Sonar	<b>80.55(5.39)</b>	78.21(2.36)	76.66(2.36)	74.95(5.21)	79.03(6.50)	94.46(1.63)
Ionosphere	<b>89.94(1.96)</b>	87.29(2.28)	86.75(2.75)	86.75(2.34)	87.50(2.23)	96.20(1.72)
Wine	<b>99.25(1.11)</b>	96.70(1.46)	95.56(1.96)	<b>99.20(0.76)</b>	96.88(1.80)	100.00(0.01)
Haberman	<b>76.71(1.86)</b>	75.65(2.68)	72.63(3.45)	75.26(3.38)	73.15(3.68)	97.36(3.34)
Cardiotocography (CTG)	<b>84.62(1.08)</b>	84.21(1.10)	<b>84.54(1.46)</b>	83.06(1.23)	84.04(2.02)	93.08(1.46)
Vertebral Column	<b>86.89(2.46)</b>	82.04(2.17)	85.89(3.47)	83.22(3.59)	84.27(3.24)	97.40(0.54)
Steel Plate Faults	<b>67.21(1.20)</b>	66.05(1.98)	67.02(1.98)	66.57(1.06)	67.22(1.64)	88.72(1.89)
WDG V1	<b>84.56(0.36)</b>	83.17(0.76)	84.36(0.56)	84.04(0.37)	84.23(0.53)	97.82(0.54)
Ecoli	<b>77.25(3.52) ●</b>	69.35(2.68)	72.22(3.65)	70.32(3.65)	67.80(4.60)	91.54(1.55)
Glass	<b>66.87(2.99) ●</b>	52.92(4.53)	62.64(5.61)	55.89(3.25)	57.16(4.17)	90.65(0.00)
ILPD	<b>69.40(1.64)</b>	67.53(2.83)	67.20(2.35)	<b>69.38(4.28)</b>	67.26(1.04)	99.10(0.72)
Adult	<b>87.15(2.43) ●</b>	83.64(3.34)	85.60(2.27)	83.58(2.91)	84.37(2.79)	95.59(0.39)
Weaning	<b>79.67(3.78) ●</b>	74.86(4.78)	76.31(4.06)	74.47(3.68)	76.89(3.15)	92.10(0.92)
Laryngeal1	<b>83.43(4.50)</b>	80.18(5.51)	81.32(3.82)	79.81(3.88)	80.75(4.93)	98.86(0.98)
Laryngeal3	<b>72.65(2.17)</b>	68.42(3.24)	67.13(2.47)	62.32(2.57)	71.23(3.18)	100(0.00)
Thyroid	<b>96.78(0.87)</b>	95.15(1.74)	95.25(1.11)	96.01(0.74)	96.24(1.25)	99.88(0.36)
German credit	<b>75.55(2.31)</b>	71.16(2.39)	74.76(2.73)	72.96(1.25)	73.60(2.69)	99.12(0.70)
Heart	<b>84.80(3.36)</b>	80.26(3.58)	82.50(4.60)	81.61(5.01)	82.05(3.72)	95.90(1.02)
Satimage	<b>96.21(0.87)</b>	94.52(0.96)	95.23(0.87)	95.43(0.92)	95.31(0.92)	98.69(0.87)
Phoneme	<b>80.35(2.58) ●</b>	75.87(1.33)	72.60(2.33)	75.90(1.06)	72.70(2.32)	99.34(0.24)
Monk2	<b>83.24(2.19)</b>	79.25(3.78)	79.18(2.57)	80.27(2.76)	80.55(3.59)	98.98(1.19)
Mammographic	84.82(1.55)	83.60(1.85)	<b>85.27(1.85)</b>	83.07(3.03)	84.23(2.14)	99.59(0.15)
MAGIC Gamma Telescope	84.35(3.27)	80.27(3.50)	81.24(2.22)	<b>87.35(1.45) ●</b>	85.25(3.25)	95.35(0.68)

For the 12 datasets where the proposed META-DES did not achieved the highest recognition rate (WDBC, Banana, Vehicle, Lithuanian, Cardiotocography, Vertebral column, Steel plate faults, Ecoli, Glass, ILPD, Laryngeal3 and Heart) we can see that each DES technique presented the best accuracy for different datasets (as shown in Figure 5). The KNOP achieves the best results for three datasets (Ecoli, Steel plate faults and Laryngeal3), the MCB for two datasets (Vehicle and Glass), the DES-FA for 3 datasets (Banana, Breast cancer and Cardiotocography) and so forth. This can be explained by the "no free lunch" theorem. There is no criterion to estimate the competence of base classifiers that dominates all other when compared with several classification problems. Since the proposed META-DES uses a combination of five different criteria as meta-features, even though one criterion might fail, the system can still achieve a good performance as other meta-features are also considered by the selection scheme. In this way, a more robust DES technique is achieved.

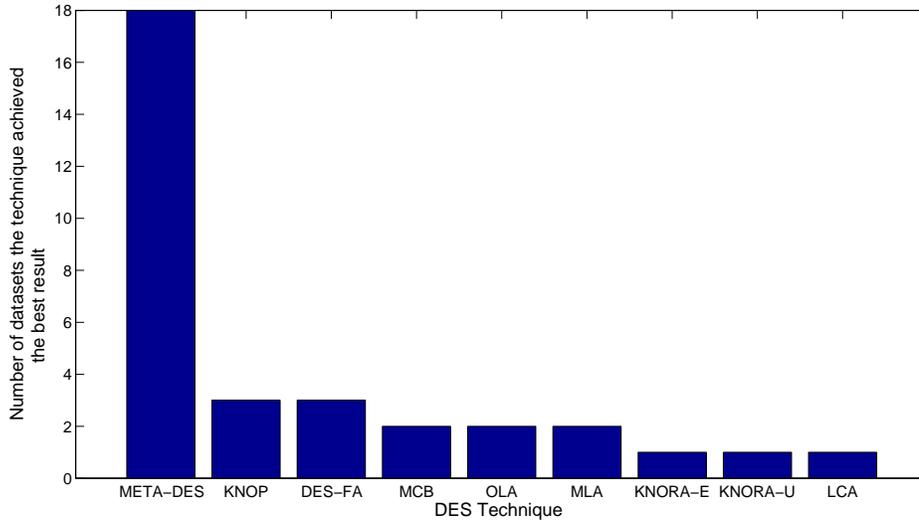


Figure 5: Bar plot showing the number of datasets that each DES technique presented the highest recognition accuracy.

Moreover, another advantage of the proposed META-DES framework comes from the fact that several meta-feature vectors are generated for each training sample in the meta-training phase (Section 3.2.2). For instance, consider that 200 training samples are available for the meta-training stage ( $N = 200$ ), if the pool  $C$  is composed of 100 weak classifiers ( $M = 100$ ), the meta-training dataset is the number of training samples  $N \times$  the number classifiers in the pool  $M$ ,  $N \times M = 20.000$ . Hence, there is more data to train the meta-classifier  $\lambda$  than for the generation of the pool of classifiers  $C$  itself. Even though the classification problem may be ill-defined, due to the size of the training set, using the proposed framework we can overcome this limitation since the size of the meta-problem is up to 100 times bigger than the classification problem. So, our proposed framework has more data to estimate the level of competence of base classifiers than the other DES methods, where only the training or validation data is available. This fact can be observed by the results obtained for datasets with less than 500 samples for training, such as, Liver Disorders, Sonar, Weaning and Ionosphere where recognition accuracy of the META-DES is statistically superior for those small size problems.

When compared against static ensemble techniques Table 4, the proposed META-DES achieves the highest recognition accuracy for 24 out of 30 datasets. This can be explained by the fact that

the majority of datasets considered are ill-defined. Hence, the results found in this paper also support the claim made by Cavalin et al. [6] that DES techniques outperform static methods for ill-defined problems.

We can thus answer the research question posed in this paper: Can the use of meta-features lead to a more robust dynamic selection technique? As the proposed system achieved better recognition rates in the majority of datasets the use of multiple properties from the classification environment as meta-features indeed leads to a more robust dynamic ensemble selection technique.

## 5. Conclusion

In this paper, we presented a novel DES technique in a meta-learning framework. The framework is based on two environments: the classification environment, in which the input features are mapped into a set of class labels, and the meta-classification environment, in which different properties from the classification environment, such as the classifier accuracy in the feature space or the consensus in the decision space, are extracted from the training data and encoded as meta-features. Five sets of meta-features are proposed. Each set corresponding to a different dynamic selection criterion. These meta-features are used to train a meta-classifier which can estimate whether a base classifier is competent enough to classify a given input sample. With the arrival of new test data, the meta-features are extracted using the test data as reference, and used as input to the meta-classifier. The meta-classifier decides whether the base classifier is competent enough to classify the test sample.

Experiments were conducted using 30 classification datasets coming from five different data repositories (UCI, KEEL, STATLOG, LKC and ELENA) and compared against eight state-of-the-art dynamic selection techniques (each technique based on a single criterion to measure the level of competence of a base classifier), as well as five classical static combination methods. Experimental results show the proposed META-DES achieved the highest classification accuracy in the majority of datasets, which can be explained by the fact that the proposed META-DES framework is based on five different DES criteria. Even though one criterion might fail, the system can still achieve a good performance as other criteria are also considered in order to perform the ensemble selection. In this way, a more robust DES technique is achieved.

In addition, we observed a significant improvement in performance for datasets with critical training size samples. This gain in accuracy can be explained by the fact that during the Meta-Training phase of the framework, each training sample generates several meta-feature vectors for the training of the meta-classifier. Hence, the proposed framework has more data to train the meta-classifier and consequently to estimate the level of competence of base classifiers than the current state-of-the-art DES methods, where only the training or validation data is available.

Future works on this topic will involve:

1. The definition of new sets of meta-features to better estimate the level of competence of the base classifiers.
2. The selection of meta-features based on optimization algorithms in order to improve the performance of the meta-classifier, and consequently, the accuracy of the DES system.
3. The evaluation of different training scenarios for the meta-classifier.

## 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] J. Kittler, M. Hatef, R. P. W. Duin, J. Matas, On combining classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (1998) 226–239.
- [2] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, 2004.
- [3] A. S. B. Jr., R. Sabourin, L. E. S. de Oliveira, Dynamic selection of classifiers - A comprehensive review, *Pattern Recognition* 47 (11) (2014) 3665–3680.
- [4] A. H. R. Ko, R. Sabourin, A. S. Britto, Jr., From dynamic classifier selection to dynamic ensemble selection, *Pattern Recognition* 41 (2008) 1735–1748.
- [5] E. M. Dos Santos, R. Sabourin, P. Maupin, A dynamic overproduce-and-choose strategy for the selection of classifier ensembles, *Pattern Recognition* 41 (2008) 2993–3009.
- [6] P. R. Cavalin, R. Sabourin, C. Y. Suen, Dynamic selection approaches for multiple classifier systems, *Neural Computing and Applications* 22 (3-4) (2013) 673–688.

- [7] 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 Recognition* 45 (9) (2012) 3544–3556.
- [8] T. Woloszynski, M. Kurzynski, A probabilistic model of classifier competence for dynamic ensemble selection, *Pattern Recognition* 44 (2011) 2656–2668.
- [9] J. Xiao, L. Xie, C. He, X. Jiang, Dynamic classifier ensemble model for customer classification with imbalanced class distribution, *Expert Systems with Applications* 39 (2012) 3668–3675.
- [10] 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.
- [11] G. Giacinto, F. Roli, Dynamic classifier selection based on multiple classifier behaviour, *Pattern Recognition* 34 (2001) 1879–1881.
- [12] K. Woods, W. P. Kegelmeyer, Jr., K. Bowyer, Combination of multiple classifiers using local accuracy estimates, *IEEE Transactions on Pattern Analysis Machine Intelligence* 19 (1997) 405–410.
- [13] L. Kuncheva, Switching between selection and fusion in combining classifiers: An experiment, *IEEE Transactions on System, Man and Cybernetics* 32 (2) (2002) 146–156.
- [14] X. Zhu, X. Wu, Y. Yang, Dynamic classifier selection for effective mining from noisy data streams, *Proceedings of the 4th IEEE International Conference on Data Mining* (2004) 305–312.
- [15] S. Singh, M. Singh, A dynamic classifier selection and combination approach to image region labelling, *Signal Processing:Image Communication* 20 (3) (2005) 219–231.
- [16] P. C. Smits, Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection, *IEEE Transactions on Geoscience and Remote Sensing* 40 (4) (2002) 801–813.
- [17] L. I. Kuncheva, Clustering-and-selection model for classifier combination, *Fourth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies* (2000) 185–188.
- [18] 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, *Proceedings of the International Joint Conference on Neural Networks* (2006) 1310–1316.
- [19] L. I. Kuncheva, A theoretical study on six classifier fusion strategies, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2) (2002) 281–286.
- [20] D. W. Corne, J. D. Knowles, No free lunch and free leftovers theorems for multiobjective optimisation problems, *Evolutionary Multi-Criterion Optimization (EMO 2003) Second International Conference* (2003) 327–341.
- [21] E. M. dos Santos, R. Sabourin, P. Maupin, A dynamic overproduce-and-choose strategy for the selection of classifier ensembles, *Pattern Recognition* 41 (10) (2008) 2993–3009.
- [22] R. M. O. Cruz, R. Sabourin, G. D. C. Cavalcanti, Analyzing dynamic ensemble selection techniques using dissimilarity analysis, in: *Artificial Neural Networks in Pattern Recognition ANNPR, 2014*, pp. 59–70.
- [23] J. H. Krijthe, T. K. Ho, M. Loog, Improving cross-validation based classifier selection using meta-learning,

- International Conference on Pattern Recognition (2012) 2873–2876.
- [24] G. Giacinto, F. Roli, Design of effective neural network ensembles for image classification purposes, *Image Vision Comput.* 19 (9-10) (2001) 699–707.
- [25] R. M. Cruz, G. D. Cavalcanti, I. R. Tsang, R. Sabourin, Feature representation selection based on classifier projection space and oracle analysis, *Expert Systems with Applications* 40 (9) (2013) 3813 – 3827.
- [26] E. M. dos Santos, R. Sabourin, P. Maupin, Single and multi-objective genetic algorithms for the selection of ensemble of classifiers, *Proceedings of the International Joint Conference on Neural Networks* (2006) 3070–3077.
- [27] I. Partalas, G. Tsoumakas, I. Vlahavas, Focused ensemble selection: A diversity-based method for greedy ensemble selection, *Proceeding of the 18th European Conference on Artificial Intelligence* (2008) 117–121.
- [28] L. I. Kuncheva, J. C. Bezdek, R. P. W. Duin, Decision templates for multiple classifier fusion: an experimental comparison, *Pattern Recognition* 34 (2001) 299–314.
- [29] M. Sabourin, A. Mitiche, D. Thomas, G. Nagy, Classifier combination for handprinted digit recognition, *Proceedings of the Second International Conference on Document Analysis and Recognition* (1993) 163–166.
- [30] L. Didaci, G. Giacinto, F. Roli, G. L. Marcialis, A study on the performances of dynamic classifier selection based on local accuracy estimation, *Pattern Recognition* 38 (11) (2005) 2188–2191.
- [31] 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.
- [32] L. Breiman, Bagging predictors, *Machine Learning* 24 (1996) 123–140.
- [33] M. Skurichina, R. P. W. Duin, Bagging for linear classifiers, *Pattern Recognition* 31 (1998) 909–930.
- [34] K. Bache, M. Lichman, UCI machine learning repository (2013).  
URL <http://archive.ics.uci.edu/ml>
- [35] R. D. King, C. Feng, A. Sutherland, Statlog: Comparison of classification algorithms on large real-world problems (1995).
- [36] 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, *Multiple-Valued Logic and Soft Computing* 17 (2-3) (2011) 255–287.
- [37] L. Kuncheva, Ludmila kuncheva collection (2004).  
URL [http://pages.bangor.ac.uk/~mas00a/activities/real\\_data.htm](http://pages.bangor.ac.uk/~mas00a/activities/real_data.htm)
- [38] R. P. W. Duin, P. Juszczak, D. de Ridder, P. Paclik, E. Pekalska, D. M. Tax, Prtools, a matlab toolbox for pattern recognition (2004).  
URL <http://www.prtools.org>
- [39] P. R. Cavalin, R. Sabourin, C. Y. Suen, Dynamic selection of ensembles of classifiers using contextual information, *Multiple Classifier Systems* (2010) 145–154.

- [40] E. M. dos Santos, R. Sabourin, Classifier ensembles optimization guided by population oracle (2011) 693–698.
- [41] D. Ruta, B. Gabrys, Classifier selection for majority voting, *Information Fusion* 6 (1) (2005) 63–81.
- [42] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *Proceedings of the Second European Conference on Computational Learning Theory* (1995) 23–37.

Accepted manuscript