

# On Meta-Learning for Dynamic Ensemble Selection

Rafael M. O. Cruz and Robert Sabourin

École de technologie supérieure - Université du Québec  
Email: rafaelmenelau@gmail.com, robert.sabourin@livia.etsmtl.ca

George D. C. Cavalcanti

Centro de Informática - Universidade Federal de Pernambuco  
Email: gdcc@cin.ufpe.br

**Abstract**—In this paper, we propose a novel dynamic ensemble selection framework using meta-learning. The framework is divided into three steps. In the first step, the pool of classifiers is generated from the training data. The second phase is responsible to extract the meta-features and train the meta-classifier. Five distinct sets of meta-features are proposed, each one corresponding to a different criterion to measure the level of competence of a classifier for the classification of a given query sample. The meta-features are computed using the training data and used to train a meta-classifier that is able to predict whether or not a base classifier from the pool is competent enough to classify an input instance. Three different training scenarios for the training of the meta-classifier are considered: problem-dependent, problem-independent and hybrid. Experimental results show that the problem-dependent scenario provides the best result. In addition, the performance of the problem-dependent scenario is strongly correlated with the recognition rate of the system. A comparison with state-of-the-art techniques shows that the proposed-dependent approach outperforms current dynamic ensemble selection techniques.

**Keywords**—Ensemble of classifiers; dynamic ensemble selection; meta-Learning.

## I. INTRODUCTION

Ensembles of Classifiers (EoC) have been widely studied in the past years as an alternative to increase efficiency and accuracy in many pattern recognition [1], [2]. There are many examples in the literature that show the efficiency of an ensemble of classifiers in various tasks, such as signature verification [3], handwritten recognition [4], [5] and image labeling [6]. Classifiers ensembles involve two basic approaches, namely classifier fusion and dynamic ensemble selection. With classifier fusion approaches, every classifier in the ensemble is used and their outputs are aggregated to give the final prediction. However, such techniques [1], [7], [8], [4] presents two main problems: they are based on the assumption that the base classifiers commit independent errors, which is difficult to find in real pattern recognition applications. Moreover, not every classifier in the pool of classifiers is an expert for every test pattern. Different patterns are associated with distinct degrees of difficulties. It is therefore reasonable to assume that only a few base classifiers can achieve the correct prediction.

On the other hand, dynamic ensemble selection (DES) techniques work by estimating the level of competence of a classifier for each query sample separately. Then, only the most competent classifiers in relation to the input sample are selected to form the ensemble. Thus, the key point in DES techniques is to define a criterion to measure the level of competence of a base classifier for the classification of the given query sample. In the literature, we can observe several criteria based on estimates of the classifier accuracy in local

regions of the feature space surrounding the query sample [9], [10], [11], [12], [13], extent of consensus [14] and decision templates [15], [16], [17], [18]. However, in our previous works [10], we demonstrate that using only one criterion to measure the level of competence of a base classifier is very error-prone.

In this paper, we propose a novel dynamic ensemble selection framework using meta-learning. The framework is divided into three steps: (1) overproduction, where the pool of classifiers is generated, (2) Meta-training where the meta-features are extracted, using the training data, and used as inputs to train a meta-classifier that works as a classifier selector. Five sets of meta-features are proposed in this work. Each set of meta-features correspond to a different criteria used to measure the level of competence of a base classifier such as the confidence of the base classifier for the classification of the input sample, and its performance in predefined regions of the feature space. (3) Generalization phase, in which the meta-features are extracted from each query sample and used as input to the meta-classifier to perform the ensemble selection. Thus, based on the proposed framework we integrate multiple dynamic selection criteria in order to achieve a more robust dynamic selection technique.

Three different training scenarios for the meta-classifier are investigated: (1) The meta-classifier is trained using data from one classification problem, and is used as the classifier selector<sup>1</sup> on the same problem; (2) The meta-classifier is trained using one classification problem, and is used as the classifier selector on a different one; (3) A single meta-classifier is trained using the data of all classification problems considered in this work, and is used as the classifier selector for all classification problems.

Based on these three scenarios, we aim to answer three research questions: (1) Can the use of meta-features lead to a more robust dynamic selection technique? (2) Is the training of the meta-classifier problem-dependent? (3) Can we improve the performance of the meta-classifier using knowledge from different classification problems? Experiments conducted over eleven classification datasets demonstrate that the proposed technique outperforms current dynamic selection techniques. Furthermore, the accuracy of the DES system is correlated to the performance of the meta-classifier.

This paper is organized as follows: In Section II we introduce the notion of classifier competence for dynamic selection. The architecture of the proposed system is presented in Section III. Experimental results are given in Section IV. Finally, a conclusion is presented in the last section.

<sup>1</sup>In this paper, we use the terms meta-classifier and classifier selector interchangeably

## II. CLASSIFIER COMPETENCE

The level of competence of a classifier defines how much we trust an expert, given a classification task. It is used as a way of selecting, from a pool of classifiers  $C$ , the one(s) that best fit(s) a given test pattern  $\mathbf{x}_j$ . Thus, in dynamic selection, the level of competence is measured on-the-fly according to some criteria applied for each input instance separately. There are three categories present in the literature [19]: the classifier accuracy over a local region, i.e., in a region close to the test pattern; decision templates, and the extent of consensus.

### A. Classifier accuracy over a local region

Classifier accuracy is the most commonly used criterion for dynamic classifier and ensemble selection techniques [9], [11], [10], [20], [21], [12], [22]. Techniques that are based on this paradigm first define a local region around the test instance, called the region of competence. This region is computed using either the K-NN algorithm [11], [9], [10] or by Clustering techniques [20], [23]. For example, the OLA technique [9] selects the classifier that obtains the highest accuracy rate in the region of competence. The Local classifier accuracy (LCA) [9] selects the classifier with the highest accuracy in relation to a specific class label and the K-Nearest Oracle (KNORA) technique [11] selects all classifiers that achieve a perfect accuracy in the region of competence. The drawback of these techniques is that their performance ends up limited by the algorithm that defines the region of competence [10].

### B. Decision Templates

In this class of methods, the goal is also to select patterns that are close to the test sample  $\mathbf{x}_j$ . However, the similarity is computed in the decision space through the concept of decision templates [24]. This is performed by transforming both the test instance  $\mathbf{x}_j$  and the validation data into output profiles using the transformation  $T$ , ( $T : \mathbf{x}_j \Rightarrow \tilde{\mathbf{x}}_j$ ), where  $\mathbf{x}_j \in \mathbb{R}^D$  and  $\tilde{\mathbf{x}}_j \in Z^M$  [25], [18] ( $M$  is the pool size). The output profile of a pattern  $\mathbf{x}_j$  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 classifier  $c_i$  for  $\mathbf{x}_j$ . Based on the information extracted from the decision space, the K-Nearest Output Profile (KNOP) [25] 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 Multiple Classifier Behaviour (MCB) technique [15] selects the classifiers that achieve a performance higher than a given threshold. The problem with using such information lies in the fact it neglects the local performance of the base classifiers.

### C. Extent of Consensus or confidence

In this class of techniques, the first step is to generate a population of an ensemble of classifiers (EoC),  $C^* = \{C'_1, C'_2, \dots, C'_{M'}\}$  ( $M'$  is the number of EoC generated) using an optimization algorithm such as a genetic algorithms or greedy search [26], [27]. Then, for each new query instance  $\mathbf{x}_j$ , the level of competence of each EoC is computed using techniques such as the Ambiguity-guided dynamic selection (ADS), Margin-based dynamic selection (MDS) and Class-strength dynamic selection (CSDS) [14], [17]. The drawback of these techniques is that they require the pre-computation

of EoC, which increases the computational complexity. In addition, the pre-computation of EoC also reduces the level of diversity and the Oracle performance (the Oracle performance is the upper limit performance of an EoC [2]) of the pool [14].

## III. PROPOSED DYNAMIC ENSEMBLE SELECTOR

A general overview of the proposed framework is depicted in Figure 1. It is divided into three phases: Overproduction, Meta-training and Generalization.

### A. Overproduction

In this step, the pool of classifiers  $C = \{c_1, \dots, c_M\}$ , where  $M$  is the pool size, is generated using the training dataset  $\mathcal{T}$ . The Bagging technique [28] is used in this work in order to build a diverse pool of classifiers.

### B. Meta-Training

In this phase, the meta-features are computed and used to train the meta-classifier  $\lambda$ . We select five subset of meta-features derived from the three categories presented in Section II. As shown in Figure 1, the meta-training stage consists of three steps: sample selection, meta-features extraction process and meta-training. A different dataset  $\mathcal{T}_\lambda$  is used in this phase to prevent overfitting.

1) *Sample selection*: We focus the training of  $\lambda$  on cases in which the extent of consensus of the pool is low. Thus, we employ a sample selection mechanism based on a threshold  $h_C$ , called the consensus threshold. For each  $\mathbf{x}_{j,train_\lambda} \in \mathcal{T}_\lambda$ , the degree of consensus of the pool, denoted by  $H(\mathbf{x}_{j,train_\lambda}, C)$ , is computed. If  $H(\mathbf{x}_{j,train_\lambda}, C)$  falls below the threshold/ $h_C$ ,  $\mathbf{x}_{j,train_\lambda}$  is passed down to the meta-features extraction process.

2) *Meta-feature extraction*: The first step in extracting the meta-features is to compute the region of competence of  $\mathbf{x}_{j,train_\lambda}$ , denoted by  $\theta_j = \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$ . The region of competence is defined in the  $\mathcal{T}_\lambda$  set using the K-Nearest Neighbor algorithm. Then,  $\mathbf{x}_j$  is transformed into an output profile,  $\tilde{\mathbf{x}}_j$  by applying the transformation  $T$  (Section II-B). The similarity between  $\tilde{\mathbf{x}}_j$  and the output profiles of the instances in  $\mathcal{T}_\lambda$  is obtained through the Manhattan distance. The most similar output profiles are selected to form the set  $\phi_j = \{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_{K_p}\}$ , where each output profile  $\tilde{\mathbf{x}}_k$  is associated with a label  $w_{1,k}$ . Next, for each base classifier  $c_i \in C$ , five sets of meta-features are calculated:

- **$f_1$  - Neighbors' hard classification**: First, a vector with  $K$  elements is created. For each pattern  $\mathbf{x}_k$ , 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 pattern  $\mathbf{x}_k$ , belonging to the region of competence  $\theta_j$ , the posterior probability of  $c_i$ ,  $P(w_1 | \mathbf{x}_k)$  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  $f_3$ .

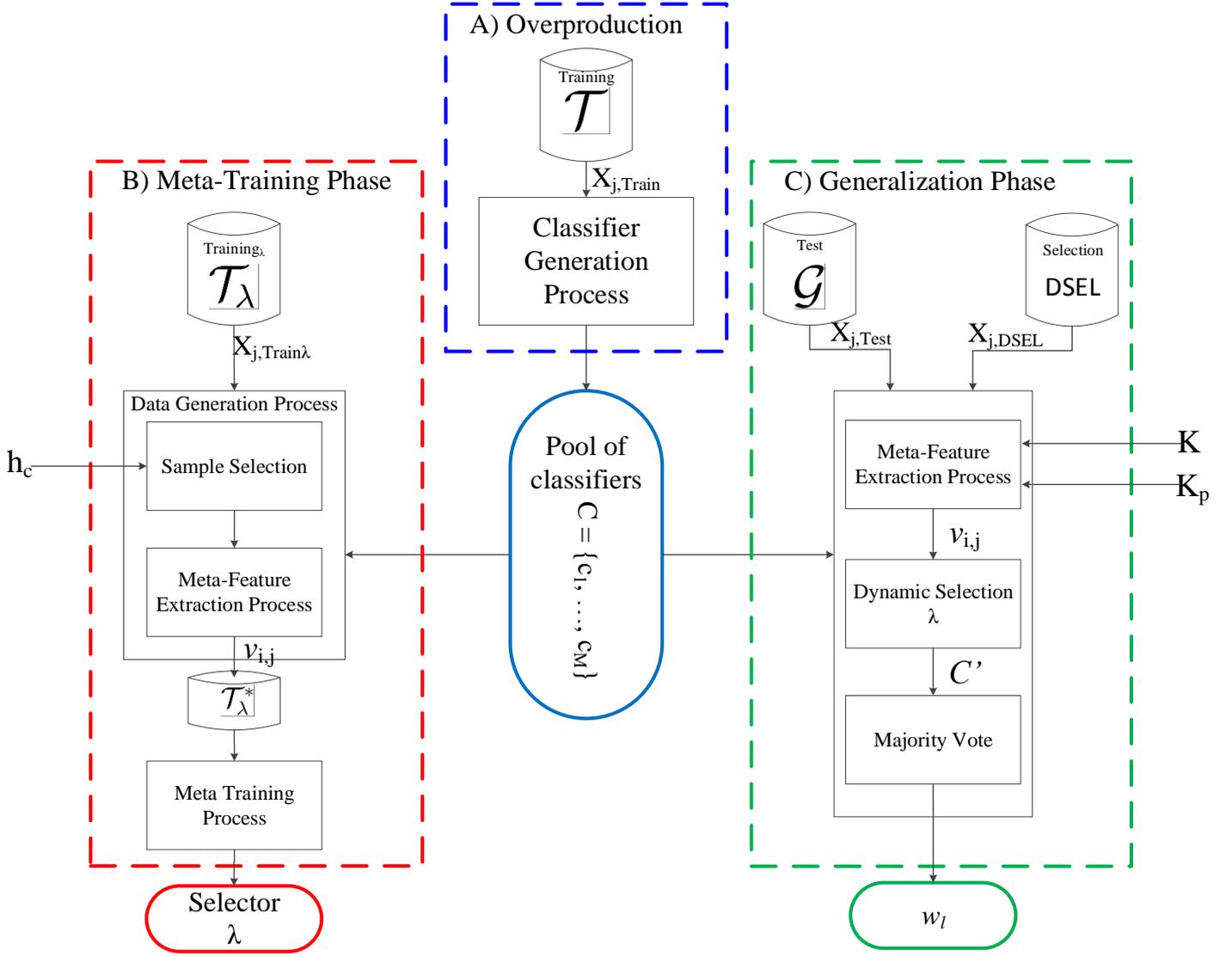


Fig. 1. Overview of the proposed framework. It is divided into three steps 1) Overproduction, where the pool of classifiers  $C = \{c_1, \dots, c_M\}$  is generated, 2) The training of the selector  $\lambda$  (meta-classifier), and 3) The generalization phase where an ensemble  $C'$  is dynamically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

- **$f_4$  - Outputs' profile classification:** First, a vector with  $K_p$  elements is generated. Then, for each member  $\tilde{x}_k$  belonging to the set of output profiles  $\phi_j$ , if the label produced by  $c_i$  for  $x_k$  is equal to the label  $w_{l,k}$  of  $\tilde{x}_k$ , the  $k$ -th position of the vector is set to 1, otherwise it is set to 0. A total of  $K_p$  meta-features are extracted using output profiles.
- **$f_5$  - Classifier's confidence:** The perpendicular distance between  $x_j$  and the decision boundary of the base classifier  $c_i$  is calculated and encoded as  $f_5$ .

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. If  $c_i$  correctly classifies  $x_j$ , 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  $\mathcal{T}_\lambda^*$ .

3) **Training:** The last step of the meta-training phase is the training of  $\lambda$ . 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 with 10 neurons in the hidden layer is used as the meta-classifier  $\lambda$ . The training process is stopped if its performance on the validation set decreases or fails to improve for five consecutive epochs.

### C. Generalization

Given an input test sample  $x_{j,test}$  from the generalization dataset  $\mathcal{G}$ , first, the region of competence  $\theta_j$  and the set of output profiles  $\phi_j$ , are calculated using the samples from the dynamic selection dataset  $D_{SEL}$ . For each classifier  $c_i \in C$ , the meta-features are extracted (Section III-B2), returning the meta-features vector  $v_{i,j}$ .

Next,  $v_{i,j}$  is passed down as input to the meta-classifier

$\lambda$ , which decides whether  $c_i$  is competent enough to classify  $\mathbf{x}_{j,test}$ . If  $c_i$  is considered competent, it is inserted into the ensemble  $C'$ . After each classifier of the pool is evaluated, the majority vote rule [2] is applied over the ensemble  $C'$ , giving the label  $w_l$  of  $\mathbf{x}_{j,test}$ . Tie-breaking is handled by choosing the class with the highest a posteriori probability.

#### IV. EXPERIMENTS

We evaluated the generalization performance of the proposed technique using eleven classification datasets, nine from the UCI machine learning repository, and two, artificially generated using the Matlab PRTOOLS toolbox<sup>2</sup>. The experiment was conducted using 20 replications. For each replication, the datasets were randomly divided on the basis of 25% for training ( $\mathcal{T}$ ), 25% for meta-training  $\mathcal{T}_\lambda$ , 25% for the dynamic selection dataset ( $D_{SEL}$ ) and 25% for generalization ( $\mathcal{G}$ ). The divisions were performed maintaining the prior probability of each class. The pool of classifiers was composed of 10 Perceptrons. The value of the hyper-parameters  $K$ ,  $K_p$  and  $h_c$  were 7, 5 and 70% respectively. They were selected empirically based on previous results [10].

We evaluate three different scenarios for the training of the meta-classifier  $\lambda$ . For the following definitions, let  $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_{11}\}$  be the eleven classification problems considered in this paper, and  $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{11}\}$  a set of meta-classifiers trained using the meta-training dataset,  $\mathcal{T}_{\lambda,i}^*$  related to a classification problem  $\mathcal{D}_i$ .

- 1) *Scenario I -  $\lambda$  dependent ( $\lambda_D$ ):* The selector  $\lambda_i$  is trained using the meta-training data  $\mathcal{T}_{\lambda,i}^*$ , and is used as the classifier selector for the same classification problem  $\mathcal{D}_i$ . This scenario is performed in order to answer the first research question of this paper: Can the use of meta-features lead to a more robust dynamic selection technique?
- 2) *Scenario II -  $\lambda$  independent ( $\lambda_I$ ):* The selector  $\lambda_i$  is trained using the meta-training data  $\mathcal{T}_{\lambda,i}^*$ , and is used as the classifier selector for a different classification problem  $\mathcal{D}_j \mid i \neq j$ . The objective of this scenario is to answer the second question posed in this work: Is the training of the meta-classifier application independent?
- 3) *Scenario III -  $\lambda_{ALL}$ :* Here, we train a single meta-classifier  $\lambda_{ALL}$  using the meta-training data derived from all classification problems  $\mathcal{D}_i \in \mathcal{D}$ ,  $\mathcal{T}_{\lambda,ALL}^* = \{\mathcal{T}_{\lambda,1}^* \cup \mathcal{T}_{\lambda,2}^* \cup \dots \cup \mathcal{T}_{\lambda,11}^*\}$ . The objective of this scenario is to answer the third question posed in this paper: Can we improve the performance of the meta-classifier using knowledge from different classification problems?

For the rest of this paper, we refer to each scenario as  $\lambda_D$ ,  $\lambda_I$  and  $\lambda_{ALL}$ . We refer to  $DES_D$ ,  $DES_I$  and  $DES_{ALL}$ , the DES system created using each training scenario, respectively.

##### A. Results

Table I shows a comparison of the results achieved related to scenarios I, II and III. Both the DES performance and

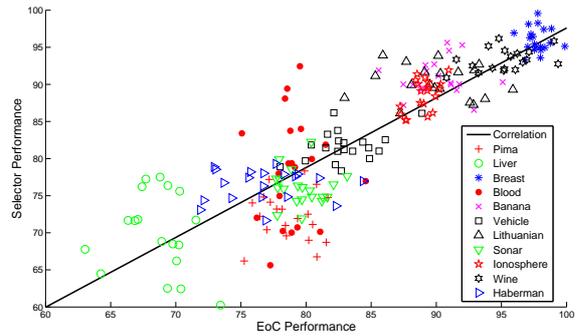


Fig. 2. Correlation between the performances of the proposed  $DES_D$  and  $\lambda_D$ .  $\rho = 0.88$ .

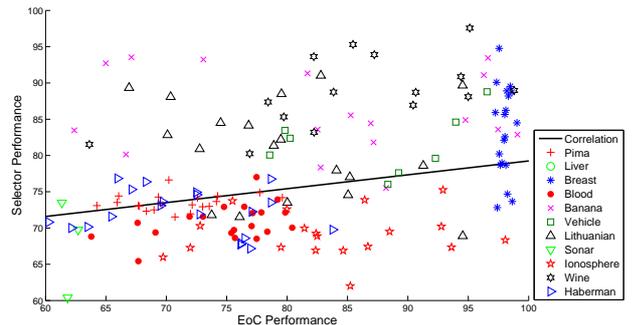


Fig. 3. Correlation between the performances of the proposed  $DES_I$  and  $\lambda_I$ .  $\rho = 0.42$ .

the meta-classifier performance are presented. We compare each pair of results using the Kruskal-Wallis non-parametric statistical test with a 95% confidence interval. Results that improved the accuracy significantly are underlined.

The  $\lambda$ -dependent scenario ( $DES_D$ ) obtained the best results. The only exception is for the Vehicle problem, where the  $\lambda_{ALL}$  achieved the best result. Furthermore, when the performance of the meta-classifier is significantly better, the accuracy of the DES system is also significantly better. This finding shows how the performance of the meta-classifier is correlated with the accuracy of its corresponding DES system. The independent scenario,  $\lambda_I$ , presented the lowest results for both the DES system ( $DES_I$ ) and meta-classifier ( $\lambda_I$ ) in all cases. The accuracies of  $\lambda_I$  and  $DES_I$  are also significantly worse when compared to the other two scenarios.

We also study the correlation between the accuracy of the DES system and the performance of the meta-classifier for the three scenarios. Figures 2, 3 and 4 show the correlation between the accuracy of the proposed DES system and the performance of the meta-classifier for the  $\lambda_D$ ,  $\lambda_I$  and  $\lambda_{ALL}$  scenarios, respectively. We compute the correlation coefficient,  $\rho$ , using the Pearson's Product-Moment.

Scenario I achieved the highest correlation coefficient  $\rho = 0.88$ , while Scenario III  $\lambda_{ALL}$  presented a slightly lower coefficient,  $\rho = 0.76$ . Thus, the use of knowledge from a different classification problem also reduced the correlation

<sup>2</sup>www.prtools.org

TABLE I. MEAN AND STANDARD DEVIATION RESULTS OF THE ACCURACY FOR THE THREE SCENARIOS. THE BEST RESULTS ARE IN BOLD. RESULTS THAT ARE SIGNIFICANTLY BETTER ( $p < 0.05$ ) ARE UNDERLINED.

Datasets	DES <sub>D</sub>	DES <sub>I</sub>	DES <sub>ALL</sub>	$\lambda_D$	$\lambda_I$	$\lambda_{ALL}$
Pima	<u>77.74(2.34)</u>	72.14(3.69)	77.18(2.99)	<b>73.20 (3.48)</b>	68.53(1.79)	72.57(2.12)
Liver	<b>68.83(5.57)</b>	59.22(3.64)	65.53(3.20)	<b>68.92(2.22)</b>	52.90(3.66)	62.29(3.14)
Breast	<u>97.41(1.07)</u>	96.99(3.64)	96.96(1.00)	<u>97.54(1.04)</u>	85.66(6.84)	96.97(1.15)
Blood	<u>79.14(1.88)</u>	75.39(5.55)	75.79(2.62)	<b>82.83(5.57)</b>	69.32(2.90)	74.28(2.87)
Banana	<u>90.16(2.09)</u>	82.52(13.24)	85.98(1.73)	<u>91.14(3.09)</u>	83.58(6.09)	80.21(8.97)
Vehicle	82.50(2.07)	80.25(3.73)	<b>83.53(1.26)</b>	82.38(2.34)	73.70(3.85)	<b>88.67(3.15)</b>
Lithuanian	<u>90.26(2.78)</u>	79.48(13.56)	87.40(1.87)	<b>89.42(3.41)</b>	82.20(6.31)	81.70(3.97)
Sonar	79.72(1.86)	53.14(6.66)	<b>80.38(4.32)</b>	<b>76.15(2.43)</b>	60.70(7.34)	75.42(2.91)
Ionosphere	<b>89.31(0.95)</b>	86.69(6.94)	88.97(2.51)	89.18(2.31)	67.44(3.42)	<b>89.52(3.72)</b>
Wine	<u>96.94(3.12)</u>	94.39(10.91)	95.11(6.69)	<b>93.33(1.56)</b>	90.86(4.49)	78.11(6.69)
Haberman	76.71(3.52)	72.77(6.34)	<b>77.63(2.55)</b>	<b>76.31(2.35)</b>	71.88(2.72)	76.23(4.91)

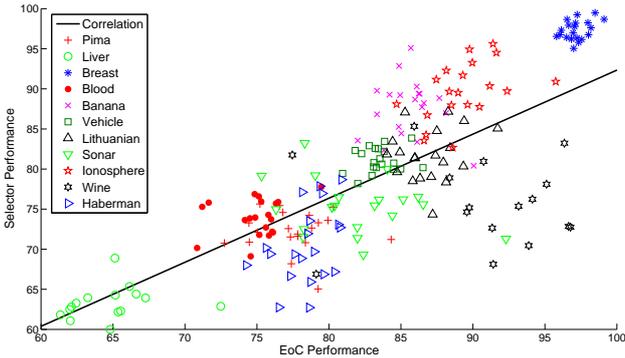


Fig. 4. Correlation between the performances of the proposed DES<sub>ALL</sub> and  $\lambda_{ALL}$ .  $\rho = 0.76$ .

between the meta-classifier and the accuracy of the DES system. The correlation between  $\lambda_I$  and DES<sub>I</sub> was  $\rho = 0.42$ , which is significantly lower than Scenarios I and III.

Therefore, experimental results indicate that the training of the meta-classifier is problem-dependent. The behavior of a competent classifier differs according to each classification problem. Furthermore, as the  $\lambda_{ALL}$  selector performed worse than the  $\lambda_D$ , we failed to improve the performance of the meta-classifier and DES system by adding knowledge derived from other classification problems. However, the loss in accuracy might be explained by the use of classification problems with completely different distributions and data complexities [29].

### B. Comparison with the state-of-the-art

In Table II, we compare the recognition rates obtained by the proposed DES<sub>D</sub> against dynamic selection techniques in the literature (KNORA-Eliminate [11], KNORA-Union [11], DES-FA [10], LCA [9], OLA [9] and KNOP [17]). We compare each pair of results using the Kruskal-Wallis non-parametric statistical test with a 95% confidence interval. The results of the proposed DES<sub>D</sub> over the Pima, Liver Disorders, Blood Transfusion, Vehicle, Sonar and Ionosphere datasets are statistically superior to the result of the best DES from the literature. For the other datasets, Breast, Banana and Lithuanian, the results are statistically equivalent.

We can thus answer the first question posed in this paper: Can the use of meta-features lead to a more robust dynamic

selection technique? As the result of the proposed DES<sub>D</sub> is significantly better in eight datasets, the use of meta-learning indeed leads to a more robust dynamic ensemble selection technique.

## V. CONCLUSION

In this paper, we present a novel DES framework using meta-learning. Different properties of the behavior of a base classifier are extracted from the training data and encoded as meta-features. These meta-features are used to train a meta-classifier that can estimate whether a base classifier is competent enough to classify a given input sample. Based on the proposed framework, we perform three experiments considering three different scenarios for the training of the meta-classifier.

Experimental results show that the training of the proposed meta-classifier is problem-dependent as the dependent scenario,  $\lambda_D$ , outperformed both  $\lambda_I$  and  $\lambda_{ALL}$ . In addition, the correlation between the performances of  $\lambda_D$  and the accuracy of the corresponding DES<sub>D</sub> is also higher than that of the other two scenarios.

A comparison with the state-of-the-art dynamic ensemble selection techniques shows that the proposed DES<sub>D</sub> outperforms current techniques. Moreover, the gain in accuracy observed with our system is also statistically significant. Thus, we can conclude that the use of multiple properties of the behavior of a base classifier in the classification environment indeed leads to a more robust DES system.

Future works on this topic will involve:

- 1) The evaluation of a different training scenario using only classification problems with similar data complexity for the training of the meta-classifier.
- 2) the design of new meta-features in order to improve the performance of the meta-classifier, and consequently, the DES system.

## ACKNOWLEDGEMENT

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).

TABLE II. MEAN AND STANDARD DEVIATION RESULTS OF THE ACCURACY OBTAINED FOR THE PROPOSED  $DES_D$  AND THE DES SYSTEMS IN THE LITERATURE. THE BEST RESULTS ARE IN BOLD. RESULTS THAT ARE SIGNIFICANTLY BETTER ( $p < 0.05$ ) ARE UNDERLINED.

Database	$DES_D$	KNORA-E [11]	KNORA-U [11]	DES-FA [10]	LCA [9]	OLA [9]	KNOP [25]
Pima	<u>77.74(2.34)</u>	73.16(1.86)	74.62(2.18)	76.04(1.61)	72.86(2.98)	73.14(2.56)	73.42(2.11)
Liver Disorders	<u>68.92(2.22)</u>	63.86(3.28)	64.41(3.76)	65.72(3.81)	62.24(4.01)	62.05(3.27)	65.23(2.29)
Breast (WDBC)	<u>97.54(1.04)</u>	96.93(1.10)	96.35(1.02)	97.18(1.13)	97.15(1.58)	96.85(1.32)	95.42(0.89)
Blood Transfusion	<u>79.14(1.88)</u>	74.59(2.62)	75.50(2.36)	76.42(1.16)	72.20(2.87)	72.33(2.36)	77.54(2.03)
Banana	<u>90.16(2.09)</u>	88.83(1.67)	89.03(2.87)	<b>90.16(3.18)</b>	89.28(1.89)	89.40(2.15)	85.73(10.65)
Vehicle	<u>82.5(2.07)</u>	81.19(1.54)	82.08(1.70)	80.20(4.05)	80.33(1.84)	81.50(3.24)	80.09(1.47)
Lithuanian Classes	90.26(2.78)	88.83(2.50)	87.95(2.64)	<u>92.23(2.46)</u>	88.10(2.20)	87.95(1.85)	89.33(2.29)
Sonar	<u>79.72(1.86)</u>	74.95(2.79)	76.69(1.94)	<u>77.52(1.86)</u>	76.51(2.06)	74.52(1.54)	75.72(2.82)
Ionosphere	<u>89.31(0.95)</u>	87.37(3.07)	86.22(1.67)	86.33(2.12)	86.56(1.98)	86.56(1.98)	85.71(5.52)
Wine	<u>96.94(3.12)</u>	95.00(1.53)	96.13(1.62)	95.45(1.77)	95.85(2.25)	96.16(3.02)	95.00(4.14)
Haberman	<u>76.71(3.52)</u>	71.23(4.16)	74.40(2.27)	74.47(2.41)	70.16(3.56)	72.26(4.17)	75.00(3.40)

## REFERENCES

- [1] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.
- [2] Ludmila I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, 2004.
- [3] Luana Batista, Eric Granger, and Robert Sabourin, "Improving performance of hmm-based off-line signature verification systems through a multi-hypothesis approach," *International Journal of Document Analysis and Recognition*, vol. 13, pp. 33–47, 2010.
- [4] Rafael M.O. Cruz, George D.C. Cavalcanti, Ing Ren Tsang, and Robert Sabourin, "Feature representation selection based on classifier projection space and oracle analysis," *Expert Systems with Applications*, vol. 40, no. 9, pp. 3813 – 3827, 2013.
- [5] Rafael M. O. Cruz, George D. C. Cavalcanti, and Tsang Ing Ren, "An ensemble classifier for offline cursive character recognition using multiple feature extraction techniques," *Proceedings of the International Joint Conference on Neural Networks*, pp. 744–751, 2010.
- [6] S. Singh and M. Singh, "A dynamic classifier selection and combination approach to image region labelling," *Signal Processing:Image Communication*, vol. 20, no. 3, pp. 219–231, March 2005.
- [7] David M. J. Tax, Martijn van Breukelen, Robert P. W. Duin, and Josef Kittler, "Combining multiple classifiers by averaging or by multiplying?," *Pattern Recognition*, vol. 33, no. 9, pp. 1475–1485, 2000.
- [8] Rafael M. O. Cruz, George D. C. Cavalcanti, and Tsang Ing Ren, "Handwritten digit recognition using multiple feature extraction techniques and classifier ensemble," *17th International Conference on Systems, Signals and Image Processing*, pp. 215–218, 2010.
- [9] Kevin Woods, W. Philip Kegelmeyer, Jr., and Kevin Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 405–410, 1997.
- [10] Rafael M. O. Cruz, George D. C. Cavalcanti, and Tsang Ing Ren, "A method for dynamic ensemble selection based on a filter and an adaptive distance to improve the quality of the regions of competence," *International Joint Conference on Neural Networks*, pp. 1126–1133, 2011.
- [11] Albert H. R. Ko, Robert Sabourin, and Alceu Souza Britto, Jr., "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recognition*, vol. 41, pp. 1735–1748, 2008.
- [12] Paul C Smits, "Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 4, pp. 801–813, 2002.
- [13] Tomasz Woloszynski and Marek Kurzynski, "A probabilistic model of classifier competence for dynamic ensemble selection," *Pattern Recognition*, vol. 44, pp. 2656–2668, October 2011.
- [14] Eulanda M. Dos Santos, Robert Sabourin, and Patrick Maupin, "A dynamic overproduce-and-choose strategy for the selection of classifier ensembles," *Pattern Recognition*, vol. 41, pp. 2993–3009, 2008.
- [15] Giorgio Giacinto and Fabio Roli, "Dynamic classifier selection based on multiple classifier behaviour," *Pattern Recognition*, vol. 34, pp. 1879–1881, 2001.
- [16] Y. S. Huang and C. Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 90–94, 1995.
- [17] Paulo R. Cavalin, Robert Sabourin, and Ching Y. Suen, "Dynamic selection approaches for multiple classifier systems," *Neural Computing and Applications*, pp. 673–688, 2013.
- [18] Paulo Rodrigo Cavalin, Robert Sabourin, and Ching Y. Suen, "Dynamic selection of ensembles of classifiers using contextual information," in *MCS*, 2010, pp. 145–154.
- [19] Alceu de Souza Britto, Jr., Robert Sabourin, and Luiz Oliveira, "Dynamic selection of classifiers - a comprehensive review," May 2014, Accepted for *Pattern Recognition*.
- [20] Ludmila 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.
- [21] M. Sabourin, A. Mitiche, D. Thomas, and G. Nagy, "Classifier combination for handprinted digit recognition," in *ICDAR*, 1993, pp. 163–166.
- [22] Jin Xiao, Ling Xie, Changzheng He, and Xiaoyi Jiang, "Dynamic classifier ensemble model for customer classification with imbalanced class distribution," *Expert Systems with Applications*, vol. 39, pp. 3668–3675, 2012.
- [23] Rodrigo G. F. Soares, Alixandre Santana, Anne M. P. Canuto, and Marcílio Carlos Pereira 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.
- [24] Ludmila I. Kuncheva, James C. Bezdek, and Robert P. W. Duin, "Decision templates for multiple classifier fusion: an experimental comparison," *Pattern Recognition*, vol. 34, pp. 299–314, 2001.
- [25] Paulo R. Cavalin, Robert Sabourin, and Ching Y. Suen, "Logid: An adaptive framework for combining local and global incremental learning for dynamic selection of ensembles of hmms," *Pattern Recognition*, 2013.
- [26] Ioannis Partalas, Grigorios Tsoumakas, and Ioannis Vlahavas, "Focused ensemble selection: A diversity-based method for greedy ensemble selection," in *Proceeding of the 18th European Conference on Artificial Intelligence*, 2008, pp. 117–121.
- [27] Eulanda Miranda dos Santos, Robert Sabourin, and Patrick Maupin, "Single and multi-objective genetic algorithms for the selection of ensemble of classifiers," in *Proceedings of the International Joint Conference on Neural Networks*, 2006, pp. 3070–3077.
- [28] Leo Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.
- [29] Núria Macià, Ester Bernadó-Mansilla, Albert Orrriols-Puig, and Tin Kam Ho, "Learner excellence biased by data set selection: A case for data characterisation and artificial data sets," *Pattern Recognition*, vol. 46, no. 3, pp. 1054–1066, 2013.