FIRE-DES++: Enhanced online pruning of base classifiers for dynamic ensemble selection

Rafael M.O. Cruz\textsuperscript{a,}\textsuperscript{*}, Dayvid V.R. Oliveira\textsuperscript{a}, George D.C. Cavalcanti\textsuperscript{a}, Robert Sabourin\textsuperscript{b}

\textsuperscript{a}Centro de Informática, Universidade Federal de Pernambuco, Brazil
\textsuperscript{b}École de Technologie Supérieure, Université du Québec, Brazil

\textbf{A R T I C L E   I N F O}

Article history:
Received 11 January 2018
Revised 27 June 2018
Accepted 31 July 2018
Available online 4 August 2018

Keywords:
Ensemble of classifiers
Dynamic ensemble selection
Classifier competence
Prototype selection

\textbf{A B S T R A C T}

Dynamic Ensemble Selection (DES) techniques aim to select one or more competent classifiers for the classification of each new test sample. Most DES techniques estimate the competence of classifiers using a given criterion over the region of competence of the test sample, usually defined as the set of nearest neighbors of the test sample in the validation set. Despite being very effective in several classification tasks, DES techniques can select classifiers that classify all samples in the region of competence as belonging from the same class. The Frienemey Indecision Region DES (FIRE-DES) tackles this problem by pre-selecting classifiers that correctly classify at least one pair of samples from different classes in the region of competence of the test sample. However, FIRE-DES applies the pre-selection for the classification of a test sample if and only if its region of competence is composed of samples from different classes (indecision region), even though this criterion is not reliable for determining if a test sample is located close to the borders of classes (true indecision region) when the region of competence is obtained using classical nearest neighbors approach. Because of that, FIRE-DES makes noisy regions for true indecision regions, leading to the pre-selection of incompetent classifiers, and mistakes true indecision regions for safe regions, leaving samples in such regions without any pre-selection. To tackle these issues, we propose the FIRE-DES++, an enhanced FIRE-DES that removes noise and reduces the overlap of classes in the validation set; and defines the region of competence using an equal number of samples of each class, avoiding selecting a region of competence with samples of a single class. Experiments are conducted using FIRE-DES++ with 8 different dynamic selection techniques on 64 classification datasets. Experimental results show that FIRE-DES++ increases the classification performance of all DES techniques considered in this work, outperforming FIRE-DES with 7 out of the 8 DES techniques, and outperforming state-of-the-art DES frameworks.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Dynamic Ensemble Selection (DES) has become an important research topic in the last few years [1]. Given a test sample and a pool of classifiers, DES techniques select one or more competent classifiers for the classification of that test sample. The most important part in DES techniques is how to evaluate the competence level of each base classifier for the classification of a given test sample [2]. In general, DES techniques evaluate the competence level of base classifiers for the classification of a test sample, \(x_{\text{query}}\), based on the performance of the base classifier in a local region surrounding the test sample, named region of competence. Most DES techniques define the region of competence of test samples using the K-Nearest Neighbors of the test sample in the validation set, we refer to this validation set as the dynamic selection dataset \(\text{DES} \) [3].

Despite being very effective in several classification tasks, DES techniques can select classifiers that classify all samples in the region of competence of a test sample to the same class, even when the test sample is located close to a decision border, having neighbors belonging to different classes (indecision region) [4].

Fig. 1 represents a query sample, \(x_{\text{query}}\), located in a indecision region. In this example, the decision boundary of classifier c1 crosses the region of competence of \(x_{\text{query}}\), and it predicts different class labels for the samples belonging to this region. It also correctly classifies at least one sample from each class. On the other hand, c2 does not cross the region of competence of \(x_{\text{query}}\). However, since it correctly classifies the same number of samples as

\(\text{https://doi.org/10.1016/j.patcog.2018.07.037}\)

0031-3203/© 2018 Elsevier Ltd. All rights reserved.
c1, a DES algorithm could select c2 as a local competent classifier, instead of c1, misclassifying the query.

To deal with this issue, Oliveira et al. [4] proposed the Frienemy Indecision Region Dynamic Ensemble Selection (FIRE-DES), a DES framework that pre-selects classifiers with decision boundaries crossing the region of competence when the test sample is located in an indecision region. Given a test sample \( x_{\text{query}} \), FIRE-DES decides if it is located in an indecision region. If so, it uses the Dynamic Frienemy Pruning (DFP) to pre-select classifiers with decision boundaries crossing the region of competence of \( x_{\text{query}} \). Then, only the pre-selected pool is passed down to a DES technique to select the final ensemble of classifiers.

However, the FIRE-DES does not consider whether or not the region of competence is a good representation of the type of region in which the test sample is located. For instance, the FIRE-DES can mistake a safe region as being an indecision region due to the presence of noise in \( D_{\text{AS}} \). In this case, the DFP can remove local competent classifiers from the pool as they do not correctly classify the noise instance, leaving only the base classifiers that modeled the noise in the local region for the DES step.

In addition, when dealing with small sized datasets, some regions of the feature space may not be well populated. In such cases, the region of competence of \( x_{\text{query}} \) can contain samples belonging to a single class (safe region) even though \( x_{\text{query}} \) may be located close to the class borders (true indecision region). In such cases, the FIRE-DES algorithm will mistake that \( x_{\text{query}} \) is located in a safe region. Hence, the DFP algorithm will not be employed to remove incompetent classifiers. However, the query is located in a true indecision region since it is close to the decision border of classes, regardless of the classes represented in its region of competence.

In this paper, we propose the FIRE-DES++, an enhanced FIRE-DES framework that tackles the noise sensitivity and indecision region restriction drawbacks of the previous framework. The main differences between the FIRE-DES++ to the original version are: (1) The FIRE-DES++ applies a prototype selection (PS) technique in order to remove noise from the validation set \( D_{\text{AS}} \). Hence, the FIRE framework will not mistake a noisy region for an indecision region when estimating the regions of competence. (2) During the test phase, the FIRE-DES++ employs a K-Nearest Neighbors Equality (KNNE) [5] to define the region of competence. The KNNE is a variation of the KNN technique which selects the same amount of samples from each class. By using the KNNE, test instances that are located close to the decision borders (in a true indecision region) will never be mistaken as belonging to a safe region since its region of competence will always be composed of samples from different classes. Thus, solving the indecision region restriction drawback of the FIRE-DES framework. Like FIRE-DES, FIRE-DES++ can be used with any dynamic selection technique based on the nearest neighbors to estimate the competence level of base classifiers.

The experiments were conducted over 64 datasets from the Knowledge Extraction based on Evolutionary Learning (KEEL) repository [6]. We evaluated FIRE-DES++ on 8 dynamic selection techniques: Overall Local Accuracy (OLA) [7], Local Class Accuracy (LCA) [7], A Priori selection [8], A Posteriori selection [8], Multiple Classifier Behavior (MCB) [9], Dynamic Selection KNN [10] and the K-Nearest Oracles Union (KNU) and Eliminate (KNE) [11]. We also compared FIRE-DES++ with the better performing dynamic selection technique according to a recent survey [1]: Randomized Reference Classifier (RRC) [12], META-DES [13], and META-DES.Oracle [14] as well as several static ensemble approaches.

This paper is organized as follows: Section 2 presents the problem statement, Section 3 presents the proposed framework, Section 4 presents the experimental study, and Section 5 concludes the paper.

2. Problem statement

2.1. FIRE-DES

The Frienemy Indecision Region Dynamic Ensemble Selection (FIRE-DES) framework works as an online pruning mechanism to pre-select base classifiers before applying the dynamic ensemble selection techniques. Given a new input query to the system, \( x_{\text{query}} \), the FIRE-DES framework analyze its region of competence to decide whether or not it is located in an indecision region (region of competence with samples from different classes). If the sample is located in a safe region, i.e., the whole region of competence is composed of samples belonging to the same class, all base classifiers are passed down to the dynamic selection technique. However, when the query is located on an indecision region, the framework applies the Dynamic Frienemy Pruning (DFP) technique to pre-select base classifiers that are able to correctly classify at least a pair of samples belonging to different classes in the region of competence. This pair of samples is called frienemy. Two instances \( x_b \) and \( x_q \) are considered frienemies if they are located in the region of competence of \( x_{\text{query}} \), and have different class labels.

Ideally, a local competent classifier would be able to distinguish all frienemies pair in the region of competence, thus being able to separate between the two classes locally. The DFP is applied to pre-select only the base classifiers that correctly classify at least one pair of frienemies. Then, only the pre-selected base classifiers are passed down to the DES algorithm for the competence estimation and classification. In the example presented in Fig. 1, the DFP would remove c2 since it does not correctly classify a single pair of frienemies. That way, although c1 and c2 may have the same local competence level, c2 would not be taken into consideration by the DS algorithm. Hence, the c1 would be selected predicting the correct label of the query. In a case where no base classifier correctly classifies a single pair of frienemies, all base classifiers are considered for competence estimation.

Although the FIRE-DES framework can be used to significantly improve the performance of several DES techniques [4], it suffers from two main drawbacks: the noise sensitivity, and indecision region restriction.
correctly classify 3 samples in $\Psi$ (A, B, and C), the classifier $c_2$ correctly classify 2 samples in $\Psi$ (B, and C), and the classifier $c_3$ correctly classify 4 samples in $\Psi$ (A, B, and C).

In Fig. 2(b), OLA selects the classifier that correctly classify the most samples in $\Psi$, that is, $c_3$, even though $c_3$ classify all samples in the region of competence of the test sample as being from the same class $\alpha$, misclassifying the test sample.

In the example from Fig. 2(b), the FIRE-DES does not apply the DFP because it considers $x_{\text{query}}$ as being located in a safe region, even though it is located in a true indecision region. Therefore, FIRE-DES with OLA also misclassifies the test sample as being from the class $\alpha$. This scenario is very likely to happen when dealing with small sized as well as imbalanced datasets, in which one of the classes may not contain enough examples in the local region.

3. The proposed framework

In this section, we propose an enhanced Frienemy Indecision Region Dynamic Ensemble Selection (FIRE-DES++). FIRE-DES++ is divided into four phases (Fig. 3): overproduction, filtering, region of competence definition and selection. The main differences between the original FIRE-DES framework and the proposed FIRE-DES++ are the addition of the filtering phase to deal with the noise sensitivity drawback, and the region of competence definition phase, in which the KNN-Equality is applied to guarantee that all classes are represented in the region of competence. Algorithms 1 and 2 present the training and test stages of the FIRE-DES++ framework, respectively.

1. Overproduction phase. where the pool of classifiers $C$ is generated using the training set ($T$). The overproduction phase is performed only once in the training stage.

2. Filtering phase, where a Prototype Selection (PS) [16] technique is applied to the validation set $D_{\text{SEL}}$, removing noise and outliers, and reducing the level of overlap between classes in $D_{\text{SEL}}$. The improved validation set is named $D_{\text{SEL}}'$. The filtering phase is performed only once in the training stage.

3. Region of competence definition (RoCD) phase, there the framework defines the region of competence ($\Psi$) using the K-Nearest Neighbors Equality (KNNE) [5] to select samples from...
Fig. 3. Overview of FIRE-DES++, where $\mathcal{G}$ is the test set, $x_{query}$ is the test sample, $\mathcal{T}$ is the training set, Generation is a ensemble generation process (i.e. Bagging) used to generate the pool of classifiers $C$, $\mathcal{D}_{SEL}$ is the validation set, Filtering is the process of filtering $\mathcal{D}_{SEL}$ using a prototype selection algorithm which results in the improved validation set $\mathcal{D}'_{SEL}$. Region of competence definition (RoCD) is the process of selecting the region of competence $\Psi$ of $x_{query}$ with size $K$, Dynamic Frienemy Pruning is the Dynamic Pruning step, $C_{\text{pruned}}$ is the pre-selected ensemble of classifiers, Dynamic Selection is the Dynamic Selection step; $C'$ is the ensemble of selected classifiers, Combination is the process of combining the prediction of the classifiers in $C'$, and $\text{class}(x_{query})$ is the final prediction of $x_{query}$.

The phases of FIRE-DES++ complement each other as the filtering phase tackles the noise sensitivity drawback, removing noise and reducing the level of overlap between classes; the region of competence definition phase tackles the indecision region restriction drawback, as it ensures that all classes are represented in the region of competence of the test sample; and, finally, the selection phase pre-selects classifiers with decision boundaries crossing the region of competence, without having to consider the effect of noise (since noise is removed in the filtering phase), or deciding if a test sample is located in an indecision region or not (as the region of competence definition phase always selects regions of competence composed of samples of different classes). The phases of FIRE-DES++ are detailed in the following subsections.

3.1. Overproduction

The overproduction phase uses any ensemble generation technique to generate the pool of classifiers $C$ trained with the training set $\mathcal{T}$. Since the focus of this work is on dynamic selection, the Bagging technique [17,18] is used to generate the pool of classifiers, following the approach used in [4].
3.2. Filtering phase

The filtering phase tackles the noise sensitivity drawback (Section 2.2), as removing noise from $D'_{SEL}$ prevents FIRE-DES from estimating the competence level of base classifiers using noisy data. This step is conducted by applying a PS technique to the validation set ($D'_{SEL}$), resulting in an improved validation set ($D''_{SEL}$) with less noise, and less overlap between classes.

Garcia et al. [16] presented a taxonomy of prototype selection, classifying prototype selection techniques into three categories: (1) Condensation techniques, that remove samples in the center of classes, maintaining the borderline samples. (2) Edition techniques, that remove sample in the borders of classes, maintaining safe samples (located in the center of classes). (3) Hybrid techniques, that combine condensation and edition approaches.

We expect the filtering phase to cause a high performance gain to the FIRE-DES++ framework, as in [2]. Cruz et al. show that state-of-the-art techniques fail to obtain a good approximation of the decision boundaries of classes when noise is added to $D_{SEL}$, and also demonstrate that using PS increases the classification performance of DES techniques.

Two PS techniques are considered: the Relative Neighborhood Graph (RNG) [19] and the Edited Nearest Neighborhood (ENN) [20]. These two PS techniques were the best approaches for dynamic selection purposes according to Cruz et al. [21]. Furthermore, since our experimental study is focused on small datasets with different levels of class imbalance, only samples of the majority class are removed from the validation set. Therefore, they also help to alleviate class imbalance problems when performing dynamic selection [22].

3.2.1. Relative neighborhood graph (RNG)

The RNG technique uses the concept of Proximity Graph (PG) to select prototypes. RNG builds a PG, $G = (V,E)$, in which the vertices are samples ($V = D_{SEL}$) and the set of edges $E$ contains an edge connecting two samples $(x_i, x_j)$ if and only if $(x_i, x_j)$ satisfy the neighborhood criterion in Eq. (1):

$$ (x_i, x_j) \in E \iff \text{dist}(x_i, x_j) \leq \max\{\text{dist}(x_i, x_k), \text{dist}(x_j, x_k)\}$$

$$ \forall x_k \in X, k = i, j $$

where $\text{dist}$ is the Euclidean distance between two samples, and $X$ is the validation set $D_{SEL}$. The corresponding geometric is defined as the disjoint intersection between two hyperspheres centered in $x_i$ and $x_j$, and radius equal to $\text{dist}(x_i, x_j)$. Two samples are relative neighbors if and only if this intersection does not contain any other sample from $D_{SEL}$. The relative neighborhood of a sample is the set of all its relative neighbors. After building the PG and defining all graph neighbors, all samples with class label different from the majority of their respective relative neighbors are removed from $D_{SEL}$.

Algorithm 3 presents the pseudo-code of the RNG technique used in this work. Given the validation set $D_{SEL}$, all samples are added in the filtered validation set $D'_{SEL}$ (Line 1), and the proximity graph of the samples in $D_{SEL}$ are stored in PG (Line 2). Now, for each sample $x_i \in D_{SEL}$, the relative neighbors (RN) of $x_i$ are selected, and, if the most common class label in RN is different from the class label of $x_i$, and $x_i$ is not from the minority class, $x_i$ is removed from the filtered validation set $D''_{SEL}$ (Line 3–10). Finally, the filtered validation set $D''_{SEL}$ is returned (Line 11).

3.2.2. Edited nearest neighbors (ENN)

Algorithm 4 presents the pseudo-code of the ENN technique used in this work. Given the validation set $D_{SEL}$, all samples are added in the filtered validation set $D'_{SEL}$ (Line 1), and for each sample $x_i \in D_{SEL}$, if $x_i$ is misclassified by its K nearest neighbors in $D'_{SEL}$, $x_i$ is removed from the filtered validation set $D''_{SEL}$ (Line 2–8). Finally, the filtered validation set $D''_{SEL}$ is returned (Line 9).

3.3. Region of competence definition phase

In order to solve the indecision region drawback (Section 2.3), the FIRE-DES++ employs the K-Nearest Neighbors Equality (KNE) instead of the traditional KNN algorithm in order to define the region of competence, $\Psi$, for each new query, $x_{query}$. The KNNE is a variation of the KNN technique which selects the same amount of samples from each class [5].

The advantage of using the ENN instead of the original KNN method employed by the previous FIRE-DES algorithm is that we ensure all classes are represented in the region of competence. Thus, test instances that are located close to the decision borders (i.e., in a true indecision region) will never be mistaken as belonging to a safe region. Moreover, the use of KNNE complements the filtering stage of the FIRE-DES++ framework. By reducing the overlap between the classes, the filtering phase may remove important samples that are close to the class borders [2,16], which could make indecision regions being mistaken as safe regions. By using the KNNE, the FIRE-DES++ framework guarantees that the DFP mechanism will be employed in such scenarios.

The region of competence, $\Psi$, is then passed down to the selection phase.

3.4. Selection phase

In the selection phase, first, the framework pre-selects classifiers using the DFP. Next, a dynamic selection technique is employed, over the pre-selected pool, to select the final ensemble $C'$, that is used for the classification of $x_{query}$.
3.4.1. Dynamic frienemy pruning

The Dynamic Frienemy Pruning (DFP) [4] aims to pre-select competent classifiers (classifiers with decision boundaries crossing the region of competence) for the classification of each new test sample, before the final selection of classifiers. The DFP algorithm uses the frienemy samples concept: Given a test sample \( x_{\text{query}} \) and its region of competence \( \Psi \), two samples \( \Psi_a \) and \( \Psi_b \) are frienemy samples in regards to \( x_{\text{query}} \) if, \( \Psi_a \) is in \( \Psi \), \( \Psi_b \) is in \( \Psi \), and \( \Psi_a \) and \( \Psi_b \) are from different classes. Fig. 4 shows a test sample \( \bullet \) and its region of competence (samples A, B, C, D, and E). In this example, the frienemy samples are the pairs of samples of opposite classes \( \circ, \bullet \), named \( (A, C), (A, D), (A, E), (B, C), (B, D), (B, E) \).

For each new test sample, if the test sample is located in an indecision region, the DFP algorithm pre-selects classifiers with decision boundaries crossing the region of competence. That is, if the test sample have samples of different classes in the region of competence, DFP pre-selects classifiers that correctly classify at least one pair of frienemy samples (if such classifier exists).

Algorithm 5 presents the DFP pseudo-code. Given the region of competence \( \Psi \) of the test sample, and the pool of classifiers \( C \), DFP creates an empty list \( C_{\text{pruned}} \) in which the pre-selected classifiers will be stored (Line 1), finds the pairs of frienemy samples \( \mathcal{F} \) in \( \Psi \) (Line 2), and, for each classifier \( c_i \) in \( C \), \( c_i \) is included in \( C_{\text{pruned}} \) if \( c_i \) correctly classify at least one pair of friemies (Lines 3 - 8). If no classifier is pre-selected, DFP includes all classifiers in \( C \) into \( C_{\text{pruned}} \) (Lines 9 - 11). Finally, \( C_{\text{pruned}} \) is returned (Line 12).

**Algorithm 5 Dynamic frienemy pruning.**

**Require:** \( \Psi \): region of competence of the test sample

**Require:** \( C \): pool of classifiers

1. \( C_{\text{pruned}} \leftarrow \) empty ensemble of classifiers
2. \( \mathcal{F} \leftarrow \) all pair of friemies in \( \Psi \)
3. for \( c_i \) in \( C \) do
4. \( \mathcal{F}_i \leftarrow \) pairs of samples in \( \mathcal{F} \) correctly classified by \( c_i \),
5. if \( \mathcal{F}_i \) is not empty then
6. \( C_{\text{pruned}} \leftarrow C_{\text{pruned}} \cup c_i \)
7. end if
8. end for
9. if \( C_{\text{pruned}} \) is empty then
10. \( C_{\text{pruned}} \leftarrow C \)
11. end if
12. return \( C_{\text{pruned}} \)

3.5. Dynamic selection

In this step, the pruned pool \( C_{\text{pruned}} \) and the region of competence, \( \Psi \), are passed down to a DES technique which selects an ensemble \( E \), from \( C_{\text{pruned}} \), containing the most competence base classifiers for the classification of \( x_{\text{query}} \).

**Fig. 5.** DES applied to the classification of a test sample \( \bullet \) of class \( \star \). The continuous straight lines are the decision boundaries of classifiers \( c_1, c_2, \) and \( c_3 \), the markers \( \circ, \bullet, \) and \( \bullet \) are samples of different classes, and samples connected to the test sample by a dotted line \( (A, B, E, \) and \( F) \) define the region of competence of the test sample. First, the FIRE-DES++ removes noise from the validation set (the example from Fig. 2(a) is turned into the example from Fig. 2(b)), tackling the noise sensitivity drawback of FIRE-DES. Then, the framework uses the KNNE to define the region of competence, selecting an equal amount of samples from different classes (the example from Fig. 2(b) is turned into the example from Fig. 5), tackling the indecision region restriction drawback of FIRE-DES. The region of competence now is composed of the samples A, B, E, and F (instead of A, B, C, F) due to the use of KNNE.

In this example, the classifier \( c_1 \) now correctly classifies 2 samples in \( \Psi \), the classifier \( c_2 \) now correctly classifies 3 samples in \( \Psi \), and the classifier \( c_3 \) now correctly classifies 2 samples in \( \Psi \). The OLA technique now selects \( c_2 \), correctly classifying the test sample.

By applying the DFP in this example (after the PS technique and the KNNE), FIRE-DES++ pre-selects the classifier \( c_2 \) as it is the only classifier that correctly classifies at least one pair of friemies, correctly classifying the test sample as being from the class \( \star \). In this example, FIRE-DES++ performed optimal classification for OLA and the same concept can be extended to other DES techniques.

4. Experiments

In this section, we evaluate FIRE-DES++ using different dynamic selection techniques. We evaluate the impact of the filtering phase using the PS techniques, the region of competence definition phase using the K-Nearest Neighbors Equality (KNNE), and the selection phase, using the Dynamic Frienemy Pruning (DFP). We also compare the filtering phase using the ENN and RNG.

4.1. Dynamic selection techniques

We used 8 dynamic classifier selection techniques from the literature (Table 1): Overall Local Accuracy (OLA), Local Class Accuracy (LCA), A Priori (APRI), A Posteriori (APOS), Multiple Classifier Behavior (MCB), Dynamic Selection KNN (DSKNN), K-Nearest Oracles Union (KNU), and K-Nearest Oracles Eliminate (KNE). These eight techniques were selected since they are the most well-known dynamic selection techniques, having the highest number of citations according to Google Scholar. Moreover, they are all based on the KNNE to estimate the region of competence. So they are suitable to be used in the FIRE-DES++ framework. A step-by-step explanation of such techniques can be found in the following surveys [1,3].

In addition, we compare the proposed FIRE-DES++ with the three dynamic ensemble selection frameworks that achieved the
best classification performance in [1]: Randomized Reference Classifier (RRC) [12], META-DES [13], and META-DES.Oracle [14]. They are briefly described below:

- **RRC**: Instead of estimating the competence of the base classifiers in the neighborhood of the query, this method uses all samples in $D_{set}$, and weights the influence of each example using a Gaussian potential function so that samples closer to the query have a higher influence in the competence estimation than the more distant ones. The source of competence is estimated based on the concept of randomized reference classifier (RRC) proposed in [12]. The base classifiers that presented a competence level higher than the random classifier are selected to compose the ensemble for an input $x_{query}$.

- **META-DES**: The META-DES is a dynamic ensemble selection framework that model the competence estimation as a meta-problem. Each measure used to estimate the local competence of a base classifier is encoded as a meta-feature. Five sets of meta-features for the estimation of the classifier competence are considered. Then, a meta-classifier is trained, based on the training data, to predict whether or not a base classifier is competent enough for the classification of a new input $x_{query}$.

- **META-DES.Oracle**: The META-DES.Oracle is an extension of the META-DES framework based on the concept of Oracle, that is an ideal dynamic selection scheme which always selects the classifiers that predict the correct label for the current sample if such classifier exists [25]. In this case, the Oracle definition is used in an optimization scheme, so that the meta-classifier can achieve results that are closer to the Oracle, improving the dynamic selection of base classifiers.

These state-of-the-art frameworks are not based exclusively on the KNN for the competence level estimation. Hence, neither the KNN nor the DFP can be applied to these techniques.

The experiments were conducted using the Python 3.5 language with the scikit-learn library [26] for the training of the base classifiers. The dynamic ensemble selection techniques were evaluated using the DESLib library [27], which contains fast implementation of all dynamic ensemble selection techniques evaluated in this work. The library is publicly available on GitHub: https://github.com/Menelau/DESLib.

The size of the region of competence (neighborhood size) $K$ was equally set to 7 for all dynamic selection technique (as suggested in [1]). This is the only hyper-parameter required for the majority of dynamic selection methods. The only exception is the DS-KNN technique, which requires to predefine the number of selected base classifiers. In this case, the number of base classifiers selected using accuracy ($N$) and diversity ($J$) was set to 30% of the whole pool as suggested in [10].

For the state-of-the-art techniques, the RRC has no hyper-parameter to set. The META-DES framework has two additional hyper-parameters: The number of samples selected using output profiles $K_p$ and the sample selection threshold $h_c$. The values of the hyper-parameters $K_p$ and $h_c$ for the META-DES framework were set to 5 and 80% according to the results presented in [13,14].

### 4.2. Datasets

We conducted the experiments on 64 datasets from the Knowledge Extraction based on Evolutionary Learning (KEEL) repository [6]. This experimental study is focused on small datasets with different levels of class imbalance. So, the framework is evaluated under a diverse set of classification problems. Table 1 shows the characteristics of the datasets used in this experiment: label, name, number of features, number of samples and the Imbalanced Ratio (IR). The IR is a common metric used by Branco and Co-workers [28,29] to characterize the imbalanced level of a distribution. It is calculated by the number of instances of the majority class per instance of the minority class.

### 4.3. Evaluation

For each dataset, the experiments were carried out using a stratified 5-fold cross validation (1 fold for test and 4 folds for training). For the sake of simplicity, we use the 5-fold partitions provided in the KEEL website. Thus, making it easier to replicate the results of this paper. The process of creating the dynamic selection dataset (DSEL) was guided by the experiments conducted in [22]. Due to the low sample size, the whole training set is used for the generation of DSEL. There is an overlap between the training bootstraps and DSEL. However, due to the randomized nature of the Bagging technique as well as the application of the PS techniques its distribution is not exactly the same. Moreover, as reported by Dietrich et al. [30] a small overlap between both datasets can be suitable for dealing with small sized datasets.

Similar to our previous works [4], the pool of classifiers $C$ was composed of 100 Perceptrons generated using the Bagging technique [17]. The training process was conducted using the scikit-learn library [26]. The learning rate and number of iterations used for the training were set to $\alpha = 0.001$ and $n_{iter} = 100$. The activation function is the Heaviside function, which predicts 0 if the sample is on one side of the hyperplane and 1 otherwise. Moreover, each Perceptron was calibrated to estimate posterior probabilities using Platt’s sigmoid model [31] provided in the scikit-learn library through the CalibratedClassifierCV class.

For evaluation metric, we used the Area Under the ROC Curve (AUC) [32]. We used the AUC because this metric has been widely

### Table 1 Dynamic selection techniques considered in the experiments.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Category</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCS</td>
<td>Overall Local Accuracy (OLA)</td>
<td>Accuracy Woods et al. [7]</td>
</tr>
<tr>
<td></td>
<td>Local Class Accuracy (LCA)</td>
<td>Accuracy Woods et al. [7]</td>
</tr>
<tr>
<td></td>
<td>A Priori (APri)</td>
<td>Probabilistic Giacinto and Roli [8]</td>
</tr>
<tr>
<td></td>
<td>A Posteriori (APos)</td>
<td>Probabilistic Giacinto and Roli [8]</td>
</tr>
<tr>
<td></td>
<td>Multiple Classifier Behavior (MCB)</td>
<td>Behavior Giacinto and Roli [9]</td>
</tr>
<tr>
<td>DES</td>
<td>Dynamic Selection KNN (DSKNN)</td>
<td>Diversity Santana et al. [10]</td>
</tr>
<tr>
<td></td>
<td>K-Nearests Oracles Union (KNU)</td>
<td>Oracle Ko et al. [11]</td>
</tr>
<tr>
<td></td>
<td>K-Nearests Oracles Eliminate (KNE)</td>
<td>Oracle Ko et al. [11]</td>
</tr>
<tr>
<td></td>
<td>META-DES</td>
<td>Meta-learning Cruz et al. [13]</td>
</tr>
<tr>
<td></td>
<td>META-DES.Oracle</td>
<td>Meta-learning Cruz et al. [14]</td>
</tr>
</tbody>
</table>
used to evaluate the performance of classifiers on imbalanced data
[33].

Furthermore, we used the Wilcoxon Signed Rank Test [34] and the Sign Test [35] to conduct a pairwise comparison between techniques over all datasets. These methods were used since they were suggested by Demšar and Co-workers [36,37]. The Wilcoxon Signed Rank Test is a non-parametric alternative to the paired t-test. The Sign test works upon the number of wins, ties and losses obtained by an algorithm over the baseline. The algorithm is deemed statistically better if its number of wins plus half of the number of ties is higher than a critical value.

Comparison between multiple datasets over all datasets is conducted using the Friedman test with the Bonferroni–Dunn post-hoc test as suggested by Demšar [36]. The Friedman test is a non-parametric equivalent of the repeated-measures ANOVA. It ranks the algorithms for each data set separately, the best one getting the rank of 1, the second best rank 2 and so on. In case of a tie, i.e., two methods presented the same classification accuracy for the dataset, their average ranks were summed and divided by two. However, the Friedman test only tells that there is a difference between the classifiers, but does not present which methods differ. For this reason, the Bonferroni–Dunn post-hoc test is employed to find out which techniques actually differs.

4.4. Filtering phase: RND vs. ENN

In this section, we evaluate FIRE-DES++ using RND and ENN for the filtering phase. Both techniques follow the same approach of maintaining all samples of the minority class. In other words, a sample is only considered a noise and removed if it belongs to the majority class. This comparison is important for verifying whether the FIRE-DES++ is sensitive to changes in PS techniques in the filtering phase, and also for finding the PS technique that causes the highest classification performance gain in FIRE-DES++.  

Fig. 6 shows the scatter plot of average AUC of FIRE-DES++ using the ENN (vertical axis) and the RND (horizontal axis). In this figure, all markers are above the diagonal line, meaning that using the ENN was, on average, better than using the RND for all DE techniques in the proposed framework.

Using the Wilcoxon Signed Rank Test ($\alpha = 0.05$), we can confirm that using the proposed framework with the ENN is statistically better than RND for the majority of DE techniques: OLA ($p$-value = 0.0121), LCA ($p$-value = 0.0011), APR1 ($p$-value = 0.0040), MCB ($p$-value = 0.0007), DSKNN ($p$-value = 0.0002), KNU ($p$-value = 0.0001).
4.5. Comparison among different scenarios

In this section, we analyze eight different scenarios for the dynamic selection techniques (Table 3). Each Scenario corresponds to a different combination of the three modules present in the FIRE-DES++ framework: DFP, ENN, and KNNE. Scenario I corresponds to the original dynamic selection techniques (i.e., no additional step is performed). Scenario IV corresponds to the FIRE-DES framework, in which only the DFP method is applied without using the modifications proposed in this paper (ENN and KNNE). Scenario VIII corresponds to the FIRE-DES++, in which the DFP, ENN and KNNE are all employed in the framework.

For each scenario, we evaluated the classification performance of each DES technique over the 64 datasets, a total of 512 experiments (64 datasets × 8 DS techniques) per scenario. We performed the Friedman test to have a comparison between the eight scenarios considering all datasets. For each dataset and dynamic selection technique, we ranked each scenario from rank 1 to rank 8 (rank 1 being the best), and used the Friedman test to calculate their average rank (Table 4). The result of the Friedman test was 
\[ p-value = 2.39 \times e^{-70} \]
indicating that there is statistical difference between the scenarios. In order to know where the difference lies, the Bonferroni–Dunn post-hoc test is conducted. The result of the post-hoc analysis is presented using a critical difference diagram (Fig. 7). Scenarios significantly different have a difference in ranking higher than the critical difference (\( CD = 0.3750 \)).

Fig. 7 shows that FIRE-DES++ (Scenario VIII) achieved the lowest average ranking (3.75), statistically outperforming Scenarios I, II, IV, V, and VI. Scenarios VI (DFP+KNNE) and VII (DFP+ENN) obtained lower average rank when compared to scenario IV (DFP alone). The reason for Scenario IV obtaining the highest average rank in this analysis is due to the fact that it never obtained the best result (lowest rank) for any combination of 64 datasets × 8 DES techniques. There is always a better alternative either by using DFP+ENN to solve the noise sensitivity drawback (Section 2.2), DFP+KNNE to solve the indecision region definition drawback (Section 2.3) or using them all together. Thus, we can conclude the addition of ENN and KNNE really helps in improving the performance of the FIRE-DES framework.

Fig. 8 shows the performance gain (AUC) obtained by adding each step of the proposed FIRE-DES++ framework in relation to the regular DES techniques. The regular DES techniques correspond to Scenario I (Table 3), while the DFP, DFP+KNNE, DFP+ENN, and DFP+KNNE+ENN corresponds to Scenarios IV, VI, VII, and VIII respectively. This figure shows that the three phases combined (DFP, KNNE, and ENN) causes the highest performance gain (2.34), followed by DFP and ENN combined (2.09), DFP and KNNE combined (0.96), and finally DFP alone (0.57). These results indicate that the filtering and the region of competence definition phases in the FIRE-DES++ framework cause performance gain over FIRE-DES, with the performance best being the use of both the ENN and KNNE combined.

Thus, we can conclude that all steps of FIRE-DES++ are important. Each step helps in improving the performance of the DES techniques. Furthermore, using all three combined leads to the highest overall improvement in classification performance.

4.6. Comparison with FIRE-DES

In this section, we compare FIRE-DES++ and FIRE-DES for each DES technique considered in this work. The goal of this analysis is to investigate whether FIRE-DES++ significantly improves the performance of FIRE-DES as well as to identify which DES techniques are more benefited from the proposed framework.

The average rank and AUC for each DES techniques is shown on Table 5. Fig. 9 presents the CD diagram comparing FIRE-DES++ (FOLA++, FLCA++, FAPRI++, FAPOS++, FMBC++, FDSKKNN++, FKNU++, and FKNE++) with FIRE-DES (FOLA, FLCA, FAPRI, FAPOS, FMBC, FDSKKNN, and FKNE) using the Bonferroni–Dunn post-hoc test. We can see that FIRE-DES++ outperformed FIRE-DES for 7 out
of 8 DES techniques. The only exception was for the LCA method, in which the FLCA and FLCA++ had statistically equivalent results.

In addition, Fig. 10 presents a pairwise comparison of FIRE-DES++ and FIRE-DES for each DES technique. This comparison used the Sign test calculated on the computed wins, ties and losses of FIRE-DES++. The null hypothesis $H_0$ was that using the FIRE-DES++ did not make any difference compared to FIRE-DES, and a rejection of $H_0$ meant that FIRE-DES++ significantly outperformed FIRE-DES. In this evaluation, we considered three levels of significance $\alpha = \{0.10, 0.05, 0.01\}$. To reject $H_0$, the number of wins plus half of the number of ties needs to be greater or equal to a critical value $n_c$ (Eq. (2)):

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

(2)

where $n_{exp} = 64$ (the number of experiments), $n_c = \{37.12, 38.58, 41.30\}$ is the critical value for each significance level $\alpha = \{0.10, 0.05, 0.01\}$, respectively.

Fig. 10 shows that FIRE-DES++ caused a significant performance gain over FIRE-DES based on the Sign test. For a confidence level $\alpha = 0.10, 0.05$ (first 2 lines left to right), FIRE-DES++ significantly improved the performance of 7 out of 8 techniques. In addition, with a more restrict confidence level $\alpha = 0.01$, the proposed FIRE-DES++ presented statistically better results for the A Posteriori, A Posteriori, MCB, DSKNN and KNE. Only for the LCA technique the FIRE-DES++ did not significantly improve over the FIRE-DES framework. However, the FLCA++ still obtained a higher number of wins (35) than losses (29). Thus, we can conclude that by the addition of ENN filter and the KNNE, the FIRE-DES++ can significantly improve the performance of a diverse set of dynamic selection techniques.

In addition, we measured the processing time of the original FIRE-DES framework and the proposed FIRE-DES++ framework. The processing time was calculated by computing the average processing time of the 64 datasets. The average running time of the proposed FIRE-DES++ framework was about 10% slower than the original FIRE-DES framework. Therefore, we can conclude that the FIRE-DES++ significantly improves the performance of DES techniques with a minimal increase in the computational time.

4.7. Comparison with state-of-the-art

In this section we compare the results of the FIRE-DES++ with the state-of-the-art dynamic ensemble selection frameworks (Table 1) as well as static ensemble methods. The following static ensemble methods were considered: Bagging [17], AdaBoost [38], Random Forests [39], Extremely Randomized Forest [40], Gradient Boosted Trees [41] and Random Balance ensembles [42]. Each technique was evaluated with a total of 100 base classifiers. The hyper-parameters of such techniques were set with the values suggested in [43].

For the sake of simplicity, only the FKNE++ was considered in this analysis since it performed better in the previous experiments. Table 6 presents the average AUC and ranking of FKNE++, the state-of-the-art DES frameworks and the static ensemble methods. The FKNE++ obtained the lowest average rank (2.84), and the

Table 5 Overall results considering the 64 datasets. The average ranks and AUC for each algorithm is presented. The algorithms are ordered according to their performance.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg. Rank</th>
<th>Algorithm</th>
<th>Mean AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FKNE++</td>
<td>4.71</td>
<td>FD$k$KNN++</td>
<td>5.15</td>
</tr>
<tr>
<td>FKNU++</td>
<td>5.67</td>
<td>FAPRI++</td>
<td>6.59</td>
</tr>
<tr>
<td>FOLAN++</td>
<td>6.67</td>
<td>FAPOS++</td>
<td>6.78</td>
</tr>
<tr>
<td>FM$C$B+</td>
<td>7.43</td>
<td>FKNU</td>
<td>8.32</td>
</tr>
<tr>
<td>FLCA++</td>
<td>13.70</td>
<td>FICA++</td>
<td>13.18</td>
</tr>
<tr>
<td>FICA</td>
<td>10.87</td>
<td>FAPRI</td>
<td>10.65</td>
</tr>
<tr>
<td>FOLAN</td>
<td>9.71</td>
<td>FM$C$B</td>
<td>8.92</td>
</tr>
<tr>
<td>DSKNN</td>
<td>8.79</td>
<td>FAPOS</td>
<td>8.78</td>
</tr>
<tr>
<td>KNE</td>
<td>8.78</td>
<td>FKNU</td>
<td>8.32</td>
</tr>
<tr>
<td>FLCA</td>
<td>8.78</td>
<td>FICA</td>
<td>7.75</td>
</tr>
</tbody>
</table>

Fig. 9. CD diagram of Bonferroni-Dunn post-hoc test considering all dynamic selection approaches. CD = 1.0608.

Fig. 10. Performance of FIRE-DES++ compared with FIRE-DES using different DES techniques in terms of wins, ties and losses considering the average AUC over the 64 datasets. Each line illustrates the critical values $n_c = \{37.12, 38.58, 41.30\}$ considering significance levels of $\alpha = \{0.10, 0.05, 0.01\}$. respectively.

Table 6

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Avg. Rank</th>
<th>Algorithm</th>
<th>Mean AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FKNE++</td>
<td>2.84</td>
<td>Random Balance</td>
<td>85.37</td>
</tr>
<tr>
<td>Random Balance</td>
<td>3.32</td>
<td>FKNE++</td>
<td>85.17</td>
</tr>
<tr>
<td>META-DES</td>
<td>4.84</td>
<td>META-DES.O</td>
<td>82.56</td>
</tr>
<tr>
<td>Boosting</td>
<td>5.09</td>
<td>META-DES</td>
<td>82.18</td>
</tr>
<tr>
<td>META-DES</td>
<td>5.20</td>
<td>Gradient Boosting</td>
<td>81.00</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>5.43</td>
<td>Boosting</td>
<td>80.76</td>
</tr>
<tr>
<td>RRC</td>
<td>5.48</td>
<td>RRC</td>
<td>80.50</td>
</tr>
<tr>
<td>Bagging</td>
<td>5.93</td>
<td>Bagging</td>
<td>78.41</td>
</tr>
<tr>
<td>Extreme Forest</td>
<td>6.68</td>
<td>Extreme Forest</td>
<td>78.00</td>
</tr>
</tbody>
</table>

The second best average AUC, 85.17 vs 85.37 obtained by the Random Balance ensemble.

Moreover, Fig. 11 presents the results of the rank analysis using critical difference diagram. The critical value was computed using the Bonferroni–Dunn test with a confidence level \( \alpha = 0.05 \) (CD = 1.2028). We can see that the FKNE++ statistically outperformed all state-of-the-art DES framework based on the rank analysis. Using the Wilcoxon Signed Rank Test (\( \alpha = 0.05 \)) for a more robust pairwise analysis, we also observed that FKNE++ statistically outperformed all three state-of-the-art DES frameworks: META-DES (\( p\text{-value} = 1.29 \times e^{-6} \)), META-DES.Oracle (\( p\text{-value} = 2.95 \times e^{-5} \)) and RRC (\( p\text{-value} = 2.33 \times e^{-6} \)). Thus, we can conclude the proposed FIRE-DES++ presents a significant performance gain over the state-of-the-art DES frameworks for these datasets.

The FKNE++ also statistically outperformed the majority of static ensemble combination methods. The only exception being the Random Balance technique. This could be explained by the fact the Random Balance was proposed to deal specifically with small sized and imbalanced data [42], which comprises the 64 datasets in this study. Moreover, this technique achieved the state-of-the-art performance for such datasets in several comparative studies [22,29]. Hence, the FKNE++ is competitive with the state-of-the-art methods for dealing with small sized and imbalanced datasets.

5. Conclusion

In this paper, we presented 2 drawbacks of the Frienemey Indecision Région Dynamic Ensemble Selection (FIRE-DES) framework: (1) noise sensitivity drawback: the classification performance of FIRE-DES is strongly affected by noise, as it mistakes noisy regions for indecision regions and applies the pre-selection of classifiers. (2) indecision region restriction drawback: FIRE-DES uses the region of competence to decide if a test sample is located in an indecision region, and only pre-selects classifiers when the region of competence of the test sample is composed of samples from different classes, restricting the number of test samples in which the pre-selection is applied for its classification.

To tackle these drawbacks of FIRE-DES, we use the Edited Nearest Neighbors (ENN) [20] to remove noise from the validation set (tackling the noise sensitivity drawback), and we use the K-Nearest Neighbors Equality (KNNE) [5] to define the region of competence selecting the nearest neighbors from each class (tackling the indecision region restriction drawback). We named this new framework FIRE-DES++.

We compared the results FIRE-DES++ with DES and FIRE-DES with 8 dynamic selection techniques over 64 datasets. The experimental results show that the FIRE-DES++ significantly outperforms FIRE-DES for 7 out of 8 DES techniques. Moreover, results also show that each individual phase of the new framework, filtering and region of competence definition, helps in significantly improving generalization performance of DES techniques.

We also compared the performance of the FIRE-DES++ with the state-of-the-art DES frameworks and ensemble methods. The results showed that the proposed framework significantly outperformed all three state-of-the-art DES frameworks with statistical confidence as well as the majority of the state-of-the-art ensemble methods. Furthermore, the FIRE-DES++ is equivalent to the Random Balance method which is considered the state-of-the-art ensemble algorithm for dealing with the KEEL imbalanced datasets according to Diez-Pastor et al. [29].

Future works on this topic will involve extending the FIRE-DES++ framework for handling multi-class classification problems; evaluating the use of different types of base classifier as well as other ensemble generation methods in the framework, and performing a complete study on the FIRE-DES++ together with data preprocessing techniques for dealing with imbalanced classification problems.

Acknowledgments

The authors would like to thank CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, in portuguese), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico, in portuguese) and FACEPE (Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco, in portuguese).

References


Rafael M.O. Cruz obtained a Ph.D. in Engineering from the École de Technologie Supérieure (ETS), Université du Québec in 2016. Currently he is a post-doctoral researcher at LIVIA (Laboratoire d’imagerie, de vision et d’intelligence artificielle). His main research interests are ensemble of classifiers, dynamic ensemble selection, meta-learning, prototype selection and imbalanced learning.

Dayvid V. R. Oliveira is a Ph.D. student at Center for Informatics, Federal University of Pernambuco, Brazil. His main research interests are on ensemble of classifiers, imbalanced learning, and noisy data.

George B. C. Cavalcanti received his Ph.D in computer science from Center for Informatics, Federal University of Pernambuco, Brazil. He is currently an associate professor with the Center for Informatics, Federal University of Pernambuco, Brazil. His research interests include ensemble of classifiers, meta-learning, imbalanced learning, and instance reduction.

Robert Sabourin joined the physics department of the Montreal University in 1977 where his main contribution was the design and the implementation of a microprocessor-based fine tracking system combined with a low-light level CCD detector. In 1983, he joined the staff of the École de Technologie Supérieure, Université du Québec, in Montreal, where he co-founded the Dept. of Automated Manufacturing Engineering where he is currently Full Professor and teaches Pattern Recognition, Evolutionary Algorithms, Neural Networks and Fuzzy Systems. In 1992, he joined also the Computer Science Department of the Pontificia Universidade Católica do Paraná (Curitiba, Brazil). Since 1996, he is a senior member of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI, Concordia University). Since 2012, he is the Research Chair holder specializing in Adaptive Surveillance Systems in Dynamic Environments. Dr. Sabourin is the author (and co-author) of more than 400 scientific publications including journals and conference proceedings. He was co-chair of the program committee of CIFED98, in Quebec, Canada and IWFHR04 in Tokyo, Japan. He was nominated as Conference co-chair of ICDAR07 that has been held in Curitiba, Brazil in 2007. His research interests are in the areas of adaptive biometric systems, adaptive surveillance systems in dynamic environments, intelligent watermarking systems, evolutionary computation and bio-cryptography.