An Effective Feature Selection Method for Class-Imbalance Datasets Applied to Chemical Toxicity Prediction

Aurelio Antelo Collado,[†] Ramón Carrasco Velar,[†] Nicolás García-Pedrajas,[‡] and Gonzalo Cerruela-García[*,‡]

[†]*University of Informatics Science. Cheminformatic Group. Havana, Cuba.*

[‡]*University of Córdoba. Department of Computing and Numerical Analysis. Campus de Rabanales. Albert Einstein Building. E-14071 Córdoba, Spain.*

E-mail: gcerruela@uco.es

Abstract

During the drug development process, it is common to carry out toxicity tests and adverse effects studies, which are essential to guarantee patient safety and the success of the research. The use of in silico QSAR approaches for this task involves processing a huge amount of data that in many cases has an imbalanced distribution of active and inactive samples. This is usually termed the class-imbalance problem and may have a significant negative effect on the performance of the learned models.

The performance of feature selection for QSAR models is usually damaged by the class-imbalance nature of the involved datasets. This paper proposes the use of a feature selection method focused on dealing with the class-imbalance problems. The method is based on the use of feature selection ensembles constructed by boosting and using two well-known feature selection methods, fast clustering-based feature selection and

the fast correlation-based filter. The experimental results demonstrate the efficiency of the proposal in terms of the classification performance compared to standard methods. The proposal can be extended to other feature selection methods and applied to other problems in cheminformatics.

1 Introduction

In the construction of Quantitative Structure-Activity Relationship (QSAR) models based on classification or regression techniques, preprocessing of the data is a fundamental component. One of the most widely used preprocessing tasks is feature selection, which is carried out to avoid the use of data that has an identical effect, has no effect or has a deceptive effect.[1] The objective of feature selection is to eliminate as many irrelevant and redundant features as possible to improve the performance of the prediction algorithms, reducing the problem of high dimensionality, accelerating the learning process and improving the generalization and interpretability of the models.

As a consequence of research activity in recent years, there is an enormous amount of high-throughput screening (HTS) data. Unfortunately, one of the main drawbacks of these HTS data lies in their inherent class-imbalance nature. In most cases, we find very few active compounds (positive class) among a large number of inactive compounds (negative class).[2] Feature selection as well as other machine learning tasks such as classification achieve good results when the datasets present a balanced distribution of the classes but may show poor performances with class-imbalance data.

The processing of class-imbalance datasets continues to be a challenge today, which has been explored and discussed in many works[3–6] to find new methods to avoid obscuring information and increase the ability to extract useful information from the dataset. Computational solutions for class-imbalance problems can be divided into two major groups: solutions based on developing new, more efficient algorithms and data-based solutions that modify the dataset. Within the former group in the field of cheminformatics, Li *et al.*[7]

proposed a support vector machine (SVM)-based method that selects a subset of inactive samples of interest and uses all active samples to build the SVM model. Chen *et al.*[8] proposed an algorithm based on random forests (RFs) that assigns a greater weight to the minority class. A similar idea using an SVM was developed by Chang *et al.*[9] Joachims[10] proposed another SVM modification based on the optimization of performance measures such as the area under the receiver operating characteristic (ROC) curve. The quantitative toxicity analysis is another important issue. Recently Jiang *et al.*[11] proposed a boosting tree-assisted multitask deep learning architecture that integrates gradient boosting decision trees and multitask deep learning to predict several toxicity values (oral rat $LD50$, 40h Tetrahymena pyriformis IGC_{50}, 96h fathead minnow LC_{50}, 48h Daphnia magna LC_{50}).

On the other hand, data-based methods are aimed at modifying the imbalanced distribution of class samples in the dataset to make both classes more balanced. In this group of methods, the two best-known techniques are oversampling, where repeated copies of minority class samples are added to the training set or new synthetic minority samples are created by interpolating already existing samples,[12] and undersampling, where the number of samples of the majority class is reduced.

In general, data-based methods are the most widely used in cheminformatics due to their independence from any machine learning algorithm. For example, the undersampling method has been used by Newby *et al.*[13] in the construction of QSAR models with imbalanced oral absorption datasets and by Chen *et al.*[14] in the construction of a toxicity model for tetrahymena pyriformis. To compensate for the reduction in the number of instances due to undersampling, many authors apply ensemble algorithms using multiple undersampling methods. For example, Kondratovich *et al.*[15] used undersampling ensemble methods for predicting the assignment of organic compounds to different pharmacological groups. Zakharov*et al.*[16] used a method that included undersampling approaches and cost-sensitive learning to construct QSAR models for different HTS datasets.

The oversampling approach has also been used in cheminformatics. Imrie *et al.*[17] used

convolutional neural networks and oversampling for structure-based virtual screening. Ezzat *et al.*[5] proposed an ensemble learning method that incorporates oversampling techniques to address the class-imbalance problem.

Other proposals combine undersampling and oversampling to improve classification performance, this group was known as hybrid-sampling methods.[18] In a recent work Korkmaz[19] evaluated the performance of undersampling, oversampling and hybrid methods in the performance of a classifier based on Deep Learning. Moreover, in a paper on predicting bioactivity in imbalanced data sets, Sun *et al.*[20] used the conformal prediction method to create a prediction region potentially with multiple predicted labels instead of creating single-value or single-label output predictions as occurs in regression or classification.

Feature selection methods have been adapted to address the problem of class-imbalanced datasets. Al-Shahib *et al.*[21] proposed a combination of feature selection and undersampling algorithms to predict the protein function from sequences using an SVM classifier. Maldonado *et al.*[22] proposed a backward elimination feature selection approach based on successive holdout steps applied to DNA microarray analysis. Han *et al.*[23] made an imbalanced feature selection proposal based on the passive-aggressive (PA) algorithm as a truncated gradient (TG) method.

In this paper, we propose the application of a feature selection method for the prediction of toxicity that is focused on solving class-imbalance problems. This method is based on the use of boosting feature selection ensembles constructed using two well-known feature selection methods, fast clustering-based (FAST)[24] and fast correlation-based filter (FCBF).[25] It also allows the use of other feature selection methods. Although there are feature selection methods that have been developed for class-imbalanced datasets,[22,23] the use of a general framework for enabling any feature selection method to address class-imbalance problems has the advantage of being more generally applicable.

The use of ensembles of feature selectors introduces the advantages of boosting feature selection for class-imbalanced datasets for QSAR models. The fact that boosting has been

proven an efficient way to address class-imbalanced dataset learning in classification[3] makes our approach a useful contribution. The methodology opens the possibility of extending many previously validated methods in classification to the problem of feature selection in class-imbalanced datasets, which are common in cheminformatics.

The rest of the paper is organized as follows: section 2 describes the dataset characteristics and molecular representation, the ensemble algorithm for feature selection in imbalance datasets, and the experimental setup; Section 3 describes the experimental results; and finally, Section 4 provides a summary of the conclusions of this work.

2 Material and Methods

In this section, we discuss the methdology used in this work, the dataset and the algorithms used.

2.1 Datasets Characteristics and Molecular Representation

Tox21 Data Challenge[26] helps researchers understand the chemical toxicology that can disrupt biological pathways so that may result in toxic effects. It is an open project where the challenger must predict compound interventions in biochemical pathways by using only physicochemical structure data. The active molecules (drugs) in the dataset are those that can bind to one or more biochemical pathway assays and create some toxic effects in human bodies.

The toxic effects included in the Tox21 dataset refer to the stress response (SR) and the effects of the nuclear receptor (NR). Both effects are highly relevant in human health since the activation of the stress response pathways can cause liver damage or cancer and activation of nuclear receptors can disrupt the function of the endocrine system.

Table 1 shows the datasets included in the Tox21 challenge. Of the twelve datasets, eight correspond to NR effects and four to SR effects. The information shown in the table

includes a unique identifier for each dataset, the number of total molecules, the number of active elements (positive or minority class), the percentage of elements of the minority class and a description of the molecular pathway endpoint.

Figure 1(a) shows that these datasets are highly imbalanced. The percentage of the minority class, active drug molecules, ranges from 2.6% to 12.6%. Furthermore, Figure 1 (b) shows the Similarity Cumulative Distribution Function using the pairwise Tanimoto similarity with the ECFP4 fingerprint. Ninety percent of the pairwise similarity values are less than 0.2, showing the high structural diversity of the molecular structures. In a first step, each compound in the twelve datasets was represented in two different ways: i) using basis molecular fragments (GSFrag) and ii) using the Extended-Connectivity Fingerprint (ECFP). ECFP is one of the most popular fingerprint approaches.[27] Boyle and Sayle[28] showed that extended connectivity fingerprints of diameter 4 (ECFP4) and 6 (ECFP6) are among the best performing fingerprints whether separating actives from decoys in a virtual screen or ranking diverse structures by similarity. On the other hand, GS-Frag has been widely used in challenging tasks in toxicology as the identification of a potential toxicity via high-throughput screening.[29–31]

GSFrag[32,33] considers 1138 molecular fragments (247 GSFrag + 891 GSFragl), with the fragments consisting of one or more disconnected components. Each component considers, among others, paths of length n, cycles on m vertices or paths (cycles) with a number of attached chains of unit length. ECFP4[27] is a circular fingerprint generated by exhaustively enumerating circular fragments containing all atoms within a radius of 4 from each heavy atom of the molecule and then hashing these fragments into a 1024 bitstring. Finally, the results were extended to include an additional molecular representation model based on molecular descriptors (OCHEM[34] CDK Descriptors[35] implementation).

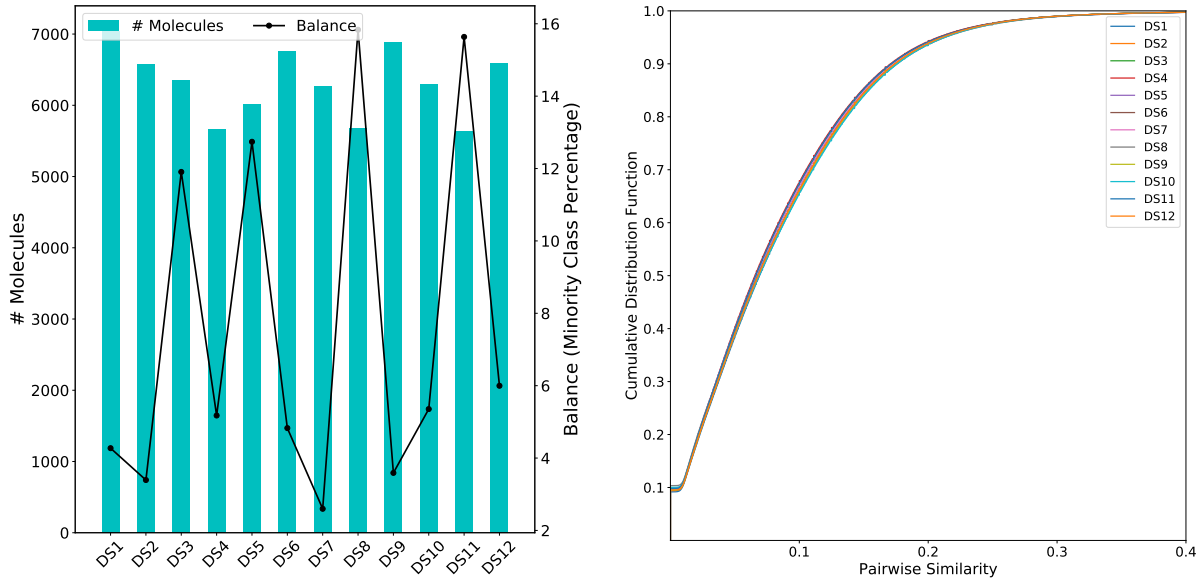
Chemaxon Standardizer[36] was used to perform a dataset curation process before calculating GSFrag or ECFP4. This step included standardization, neutralization, removing salts and cleaning the structure. For a more detailed description of these preprocessing options,

please refer to the Chemaxon Standardizer in the official Chemaxon documentation.[37] The Chemistry Development Kit (CDK) library[35] was used to calculate the ECFP4 fingerprint, and OCHEM environment[34] was used to calculate the GSFrag and CDK Descriptors. During the curation and molecular representation processes all the molecules with errors were eliminated from the dataset.

Table 1: Toxicity Datasets

Dataset	# Molecules	Class−	Class+	Balance[a]	Molecular pathway endpoint
DS1	7044	6743	301	4.3	Androgen receptor MDA-kb2 AR-luc cell line (NR-AR)
DS2	6572	6349	223	3.4	Androgen receptor GeneBLAzer AR-UAS-bla-GripTite cell line (NR-AR-LBD)
DS3	6358	5601	757	11.9	Aryl hydrocarbon receptor (NR-AhR)
DS4	5661	5368	293	5.2	Aromatase enzyme (NR-Aromatase)
DS5	6013	5247	766	12.7	Estrogen receptor alpha BG1-Luc-4E2 cell line (NR-ER)
DS6	6752	6426	326	4.8	Estrogen receptor alpha ER-alpha-UAS-bla GripTiteTM cell line (NR-ER-LBD)
DS7	6273	6110	163	2.6	Peroxisome proliferator-activated receptor gamma (NR-PPAR-gamma)
DS8	5684	4784	900	15.8	Nuclear factor (erythroid-derived 2)-like 2/antioxidant responsive element (NR-ARE) (SR-ARE)
DS9	6880	6633	247	3.6	ATAD5 receptor (SR-ATAD5)
DS10	6294	5957	337	5.4	Heat shock factor response element (SR-HSE)
DS11	5634	4753	881	15.6	Mitochondrial membrane potential (SR-MMP)
DS12	6586	6191	395	6.0	p53 signaling pathway (SR-p53)

[a] The balance has been measured as the percentage of the minority class



(a) Datasets Size and Balance

(b) Similarity Cumulative Distribution Function

Figure 1: Datasets Characteristics

2.2 Boosting-Based Ensemble Algorithm for Feature Selection for Class-Imbalance Datasets

The algorithm used in this work is based on the construction of a combination of feature selectors following a similar approach used for the construction of classifier ensembles with boosting.[38] In this way, feature selection rounds are applied in order to focus on the instances that have been considered the most problematic. Each one of the executed rounds obtains a subset of suitable features to classify the most problematic instances. As minority class instances are often ignored by the learning algorithm, boosting is an efficient method to deal with the class-imbalance problem. A voting process is applied to combine the solutions of the different rounds.[4]

Figure 2 shows the ensemble algorithm for feature selection. Initially, the weights of all instances are initialized using a uniform distribution $w_i = 1/N$, choosing the original dataset $S' = S$ as the initial sample. Then, for the fixed number of iterations, T, the following steps are performed: i) using a specific feature selection algorithm and the sampled subset $S' \subset S$, a feature selection process is carried out, ii) the obtained m_t subset of selected features is stored in vector $\mathbf{v}$, and iii) a classifier is trained using m_t, and the weights associated with each instance, w_i, and α_t are updated for this classifier. The way to update $\mathbf{w}$ and α_t depends on the boosting algorithm used.

The approach has two main elements, the boosting scheme and the feature selection (FS) algorithm. In this work, we have chosen two FS algorithms, fast clustering-based feature selection (FAST) and fast correlation-based filter (FCBF). FAST[24] operates in two steps. In the first step, the features are divided into clusters via graph-theoretic clustering methods. In the second step, the most representative feature that is strongly related to target classes is selected from each cluster to form a subset of features. Features that belong to different clusters are relatively independent; the clustering-based strategy of FAST has a high probability of producing a subset of useful and independent features. To ensure the efficiency of FAST, the efficient minimum-spanning tree (MST) clustering method is used.

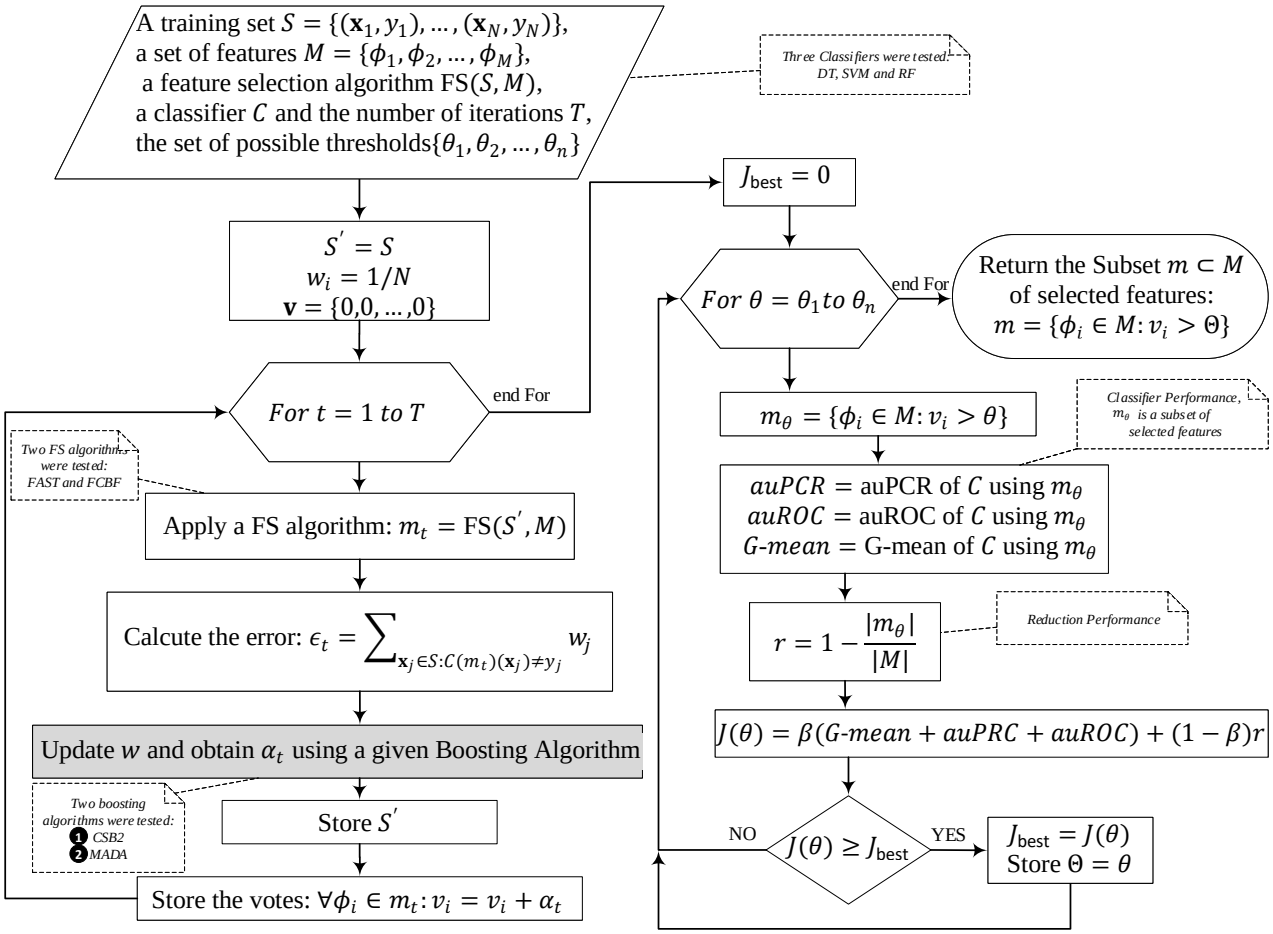
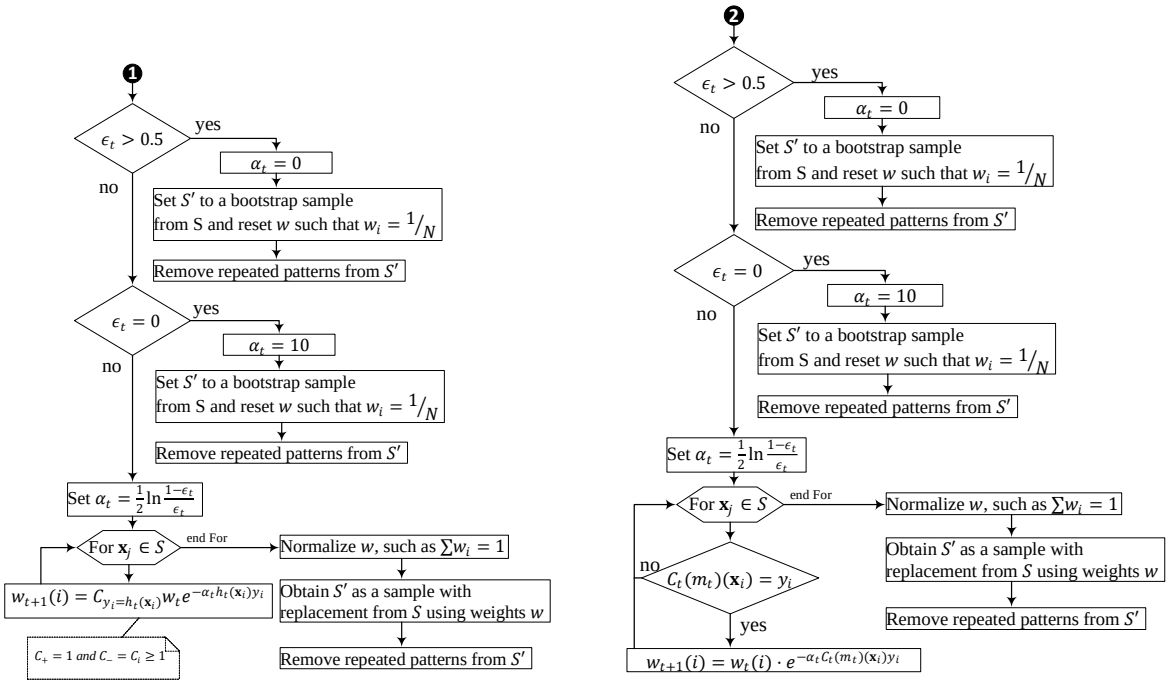


Figure 2: Boosting Ensemble Algorithm for Feature Selection

FCBF[25] is a multivariate subset selection algorithm that utilizes the concept of feature redundancy to perform explicit redundancy analysis in feature selection. FCBF authors propose a framework that decouples relevance analysis and redundancy analysis.

Different boosting algorithms can be used for the proposed algorithm, and in this work we evaluate two approaches: MadaBoost and CSB2. MadaBoost is a standard boosting algorithm, and CSB2 is a boosting algorithm specifically designed for class-imbalance datasets. MadaBoost[39] is a modification of the well-known AdaBoost[40] algorithm where the weight updating of the instances is made in a more conservative way. Although MadaBoost was not specifically developed for imbalance datasets we also consider it because, in general,



(a) CSB2 Boosting Ensemble Algorithm (b) MadaBoost Boosting Ensemble Algorithm

Figure 3: Boosting Ensembles Methods

all boosting algorithms are effective for dealing with class-imbalance problems.[3,41] Figure 3 shows these two algorithms.

CSB2 was developed to specifically address the class-imbalance problem. CSB0, CSB1 and CSB2[42] comprise a group of boosting methods developed to cope with the unbalanced distribution of positive and negative samples by modifying the AdaBoost method. The CSBX methods use two values $C_+ = 1$ and $C_- = C_i \geq 1$, which are the costs of correctly classifying an instance and misclassifying an instance, respectively. CSB0 modifies the standard AdaBoost update rule using $w_{t+1}(i) = C_{y_i=h_t(\mathbf{x}_i)} w_t$, and CSB1 uses $w_{t+1}(i) = C_{y_i=h_t(\mathbf{x}_i)} w_t e^{-h_t(\mathbf{x}_i) y_i}$. Both methods use $\alpha_t = 1$. CSB2 uses for α_t the same values as standard AdaBoost and modifies the weight update as $w_{t+1}(i) = C_{y_i=h_t(\mathbf{x}_i)} w_t e^{-\alpha_t h_t(\mathbf{x}_i) y_i}$, which is the same formula of CSB1 with the introduction of α_t.

As the result of repeatedly applying a certain feature selection algorithm using boosting, a vector of votes is obtained. For every feature, the results of each round of the feature

selection process are stored. This vector of votes will be used to obtain the final selection of features that is the algorithm outcome. In ensembles of classifiers it is common to use the majority voting approach to obtain the class of a new instance. In our case, we would select features whose count of votes indicated that they had been selected more often than not. Although this method is simple and fast, it does not achieve good results, as it is very unlikely that this static threshold would be appropriate for all datasets. Therefore, we use an adaptive model where the decision to maintain a feature is established by evaluating all the thresholds. This process has a low computational cost since the number of different thresholds is limited by the number of T rounds.

In the proposed algorithm (see Figure 2), the optimal threshold, (Θ), is obtained using the vector of votes $\mathbf{v}$, which records the votes obtained for every feature. Thus, a certain threshold, θ, selects the features whose records of votes are above this threshold: $v_i > \theta$.

To obtain the best threshold a $J(\theta)$ criterion was defined, and all of the possible thresholds were evaluated. In each threshold θ evaluation, the subset, $m_\theta \subset M$, of features using θ: $m_\theta = \{\phi_i \in M : v_i > \theta\}$ was selected, evaluating the criterion $J(\theta)$ with m_θ to learn the selected classifier.

We have defined J based on two fundamental criteria in feature selection, the classification performance and the reduction capacity. In addition, a parameter β has been added in order to weigh the relative importance of both criteria. In the experiments, a value of $\beta = 0.5$ was used. In this way J was formulated as follows: $J(\theta) = \beta(G\text{-}mean + auPRC + auROC) + (1 - \beta)r$. The terms *G-mean*, *auROC* and *auPRC* define the classification performance, *G-mean* is the geometric mean of the sensitivity (Sn) and specificity (Sp) metrics as defined in equation 1, *auPRC* is the area under the precision recall curve (PRC), and *auROC* is the area under the ROC. The reduction capacity, r (see eq. 3), is measured as the percentage of features removed by any algorithm.

2.3 Experimental Methodology

For each of the twelve datasets studied in this work, two matrices with M rows (number of molecular compounds) have been constructed, with the cardinality of the columns (features) being 1138 for the GSFrag representation and 1024 for the ECFP4 representation. For comparison against the proposed methods, the standard feature selection algorithms (FAST and FCBF) were run using random undersampling. This is a fast and simple method that generally achieves similar results to other more complex methods. FAST and FCBF were compared against our proposal using the two boosting algorithms described above: CSB2.FAST, CSB2.FCBF, MadaBoost.FAST, MadaBoost.FCBF.

For testing the classification performance of the FS methods, three different classifiers were used: a decision tree (DT), an SVM and an RF. Each model was tested following the procedure described in Figure 4. Each dataset was split into inner and outer (test) sets using a nested 5-fold cross validation split scheme. Inner cross validation sets were used to train the models based on feature selection algorithms by fixing the hyperparameters. The resulting model was validated with the test set not previously considered for model construction. The entire external validation process was repeated five times to construct and evaluate five different independent external test sets. In the experimental results, we show the average test values over the five folds.

As we mentioned, external validation was tested with three classifiers, DTs, SVMs and RFs. For SVMs, three hyperparameters were set: the kernel type, the C value, and for the Gaussian kernel, the γ value. Thus, we tested a linear kernel with $C \in \{0.1, 1, 10\}$ and a Gaussian kernel with $C \in \{0.1, 1, 10\}$ and $\gamma \in \{0.0001, 0.001, 0.01, 0.1, 1, 10\}$. All 21 possible combinations were evaluated. For random forests we used a size of 100 trees and the Gini impurity criterion to measure the quality of a split, the nodes were expanded until all leaves were pure or until all leaves contained less than two samples, and bootstrap samples were used for building the trees. DTs have no relevant hyperparameters.Table S1 of the supplementary material summarizes all hyperparameters, the source code licensed under

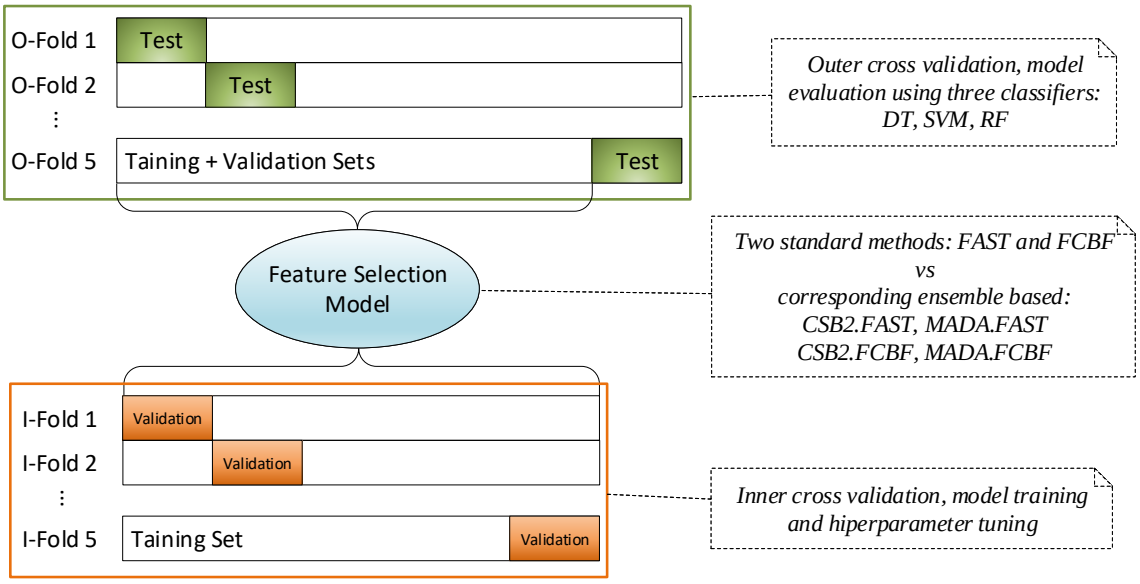


Figure 4: Experimental setup

the GNU General Public License is freely available from the authors using the following link: http://cib.uco.es/source-code/. The data sets employed in the work have been included as supplemental material (SupMaterial_File-2.xlsx)

To measure the performance achieved by classifiers, two well-known metrics were used: the geometric mean (*G-Mean*) of sensitivity and specificity and Mathews correlation coefficient (*MCC*).[43] These metrics are usually used for class-imbalance datasets because they take into account the uneven distribution of class samples.

The *G-Mean*[44] of sensitivity and specificity can be defined as follows:

$$G\text{-}Mean = \sqrt{\frac{TP}{TP+N} \times \frac{TN}{TN+FP}} \tag{1}$$

Mathews correlation coefficient (*MCC*)[44] can be defined as follows:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}. \tag{2}$$

In the equations, TP, TN, FP, and FN are the numbers of true positives, true negatives,

false positives and false negatives, respectively.

The reduction capacity r can be defined as follows:

$$r = 1 - m/|M| \tag{3}$$

Where m is the number of selected features and M the set of all of features.

2.3.1 Multiple Comparison Statistical Tests

When several algorithms are compared on different datasets, it is recommended to use statistical tests that support the conclusions. Moreover, before using the multiple comparison statistical tests it is necessary to apply the Iman–Davenport test[45] to determine whether there are significant differences among the methods. It is based on the χ_F^2 Friedman test, which compares the average ranks of different algorithms, but it is more powerful than the Friedman test.[45,46]

The Wilcoxon test is especially recommended for the pairwise comparison of algorithms.[46] Its definition is as follows: consider d_i the difference between the values of a metric of the evaluated algorithms for the *i-th* dataset. Let R^+ be the sum of the ranks for the datasets for which the second algorithm outperformed the first and R^- the sum of ranks for which the first algorithm outperformed the second. Ranks of $d_i = 0$ are split evenly among the sums:

$$R^+ = \sum_{d_i>0} rank\,(d_i) + \frac{1}{2}\sum_{d_i=0} rank\,(d_i) \tag{4}$$

$$R^- = \sum_{d_i<0} rank\,(d_i) + \frac{1}{2}\sum_{d_i=0} rank\,(d_i) \tag{5}$$

Considering T be the smaller of the two sums and N be the number of datasets, there are tables with the exact critical values for small N values. Moreover, for larger N values, the z statistic is calculated to find the corresponding probability (*p-value*) from the table of

normal distributions:[47]

$$z = \frac{T - \frac{1}{4}N\left(N+1\right)}{\sqrt{\frac{1}{24}N\left(N+1\right)\left(2N+1\right)}} \tag{6}$$

When comparing multiple models simultaneously, the Nemenyi test[46,48] was used. In this case, the performance between two algorithms to be compared is considered significantly different if the corresponding average Friedman's ranks differ by at least the critical difference (CD), calculated as follows:

$$CD = q_{\infty}\sqrt{\frac{k\left(k+1\right)}{6N}}, \tag{7}$$

where N is the number of datasets and k the number of algorithms to be compared. For all test we used a significance level $\alpha = 0.05$.

3 Results and Discussion

In this section, we present and discuss the experimental results carried out by following the experimental setup described above. First, the proposals for feature selection based on boosting (CSB2.FAST, CSB2.FCBF, MadaBoost.FAST, MadaBoost.FCBF) were compared with the corresponding FS base algorithms (FAST and FCBF) independently for each representation. Then, the boosting-based feature selection proposals were compared against the corresponding FS base algorithms considering all the representation models and using the statistical tests with the aim of studying the global behavior of the proposals. Finally, the reduction capacity of the proposals was analyzed.

The experimental results for the selected metrics are shown in Table S2 - S7 of the supplemental material (SupMaterial_File-1.pdf). In this section, we include a graphical representation of the results for easy presentation and discussion. The graphics are based on the relative movement diagrams proposed by Maudes *et al.*,[49] but in this case, instead of

the κ-error difference values, we use *G-Mean* and *MCC* to evaluate the performance results of the three classifiers tested. In a simple way, the graphs allow both metrics to be plotted at the same time, where each axis represents the difference of a given metric between two different algorithms compared for the same dataset. Every arrow starts at the coordinate origin, and the coordinates of the tip are calculated as the difference between the two chosen metrics in the two compared algorithms. In all the figures, the values of the differences are represented as percentages.

We have set a rule to assign the result of our proposal as minuend and the result of the base FS algorithm as subtrahend. In this way, taking Figure 5 (a) as a reference, the arrows pointing downward-left represent the datasets for which the base FS algorithm outperformed our proposal in both the *G-Mean* and *MCC*, the arrows pointing upward-left indicate that our proposal improved the *G-Mean* but had an inferior *MCC*, the arrows pointing upward-right shows datasets for which our proposal outperformed the base FS algorithm in both *G-Mean* and *MCC*, and arrows pointing downward-right show the datasets for which our proposal improved the *MCC* but resulted in a worse *G-mean*.

Figures 5 - 7 show the results obtained using the FAST FS base algorithm compared to those obtained by the proposed algorithms CSB2.FAST and MadaBoost.FAST. For DTs, Figure 5 shows in terms of *G-Mean* that our proposals improved the performance for all datasets. In terms of *MCC*, it occurred similarly, except for the DS1 dataset for GSFrag representation and the datasets GS8 and GS12 for ECFP4 representation. For SVM classifier (Figure 6), the results for performance for algorithms CSB2.FAST and MadaBoost.FAST were superior in most datasets, reaching increases up to 12% in terms of *G-Mean* and *MCC* compared to the use of FAST. Overall, the classifiers DTs and SVM showed the same behavior.

The results for RFs were different, with a marked dependence on the method of representation. GSFrag representation, Figure 7 (a,b), showed the same behavior as the one described above, and for all datasets the performance of CSB2.FAST and MadaBoost.FAST

was better in terms of *G-Mean* with increases up to 11% in the best case. Additionally, there were better results in terms of *MCC* except for datasets DS1 and DS2 where the opposite occurred. The ECFP4 representation, Figure 7 (c,d), showed a lower performance. There are four datasets (DS4, DS8, DS11, DS12) that appear in the lower left quadrant, showing for them a worse performance of CSB2.FAST or MadaBoost.FAST compared to FAST.

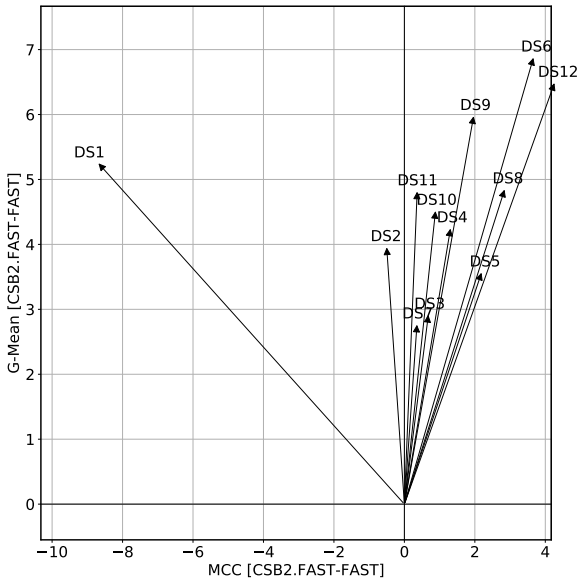
Figures 8 to 10 show the comparison for the approaches FCBF, CSB2.FCBF and MadaBoost.FCBF. For DT classifier (Figure 8), CSB2.FCBF and MadaBoost.FCBF did not outperform FCBF for the datasets, DS1, DS2, and DS5. However, taking into account the results for the rest of the datasets, we can affirm that CSB2.FCBF and MadaBoost.FCBF were better than FCBF in terms of *G-Mean* and *MCC*. Figure 9 shows results for SVM classifier and again points the superiority of CSB2.FCBF and MadaBoost.FCBF with respect to FCBF, and the datasets DS1 and DS2 had the worst performance results.

The behavior of RF classifier for FCBF was similar to that described above for FAST, with a great influence of the representation model used. Using CSB2.FCBF and MadaBoost.FCBF on the GSFrag representation increased the performance on almost all datasets compared to FCBF, while its application on ECFP4 representation showed worse results.

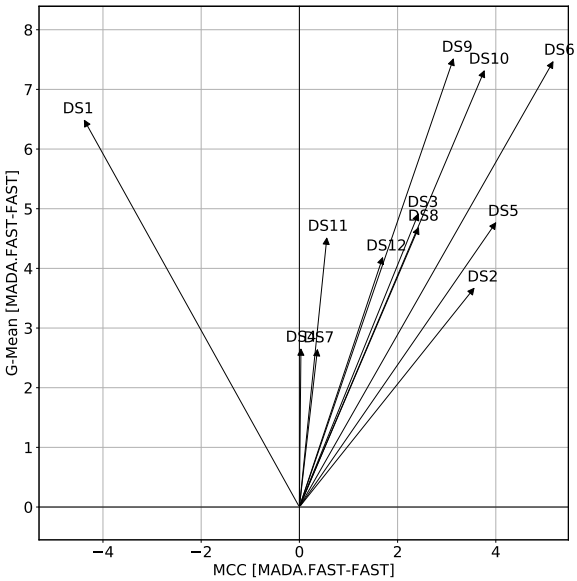
3.1 Statistical Test Analysis

To perform a robust analysis when comparing algorithms over multiple datasets we evaluate the performance of our proposal using the statistical tests described above. In this case, we considered all the datasets regardless of differences between the representation models, taking on different molecular representations on the same dataset as different datasets. Thus, in the tests $N = 24$ datasets (12 datasets $\times$ 2 representation models) were used. In accordance with Demšar's[46] recommendations, this value must be greater than 20 for it to be considered significant.

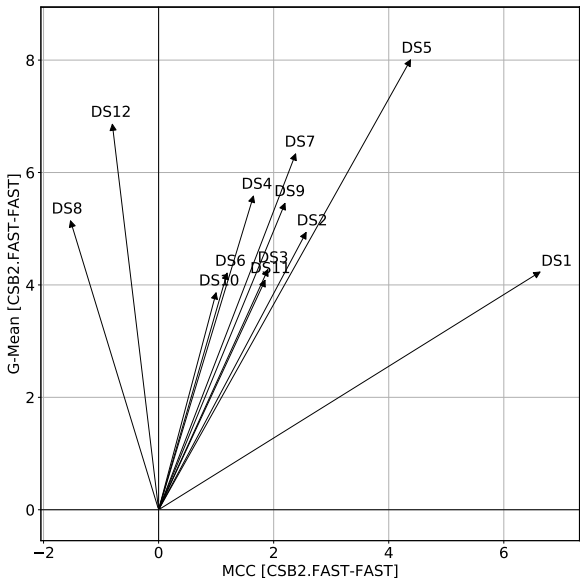
The Iman-Davenport test comparing the proposals CSB2.FAST, CSB2.FCBF, MADA.FAST, MADA.FCBF and the base methods, FAST and FCBF, using undersampling found signifi-



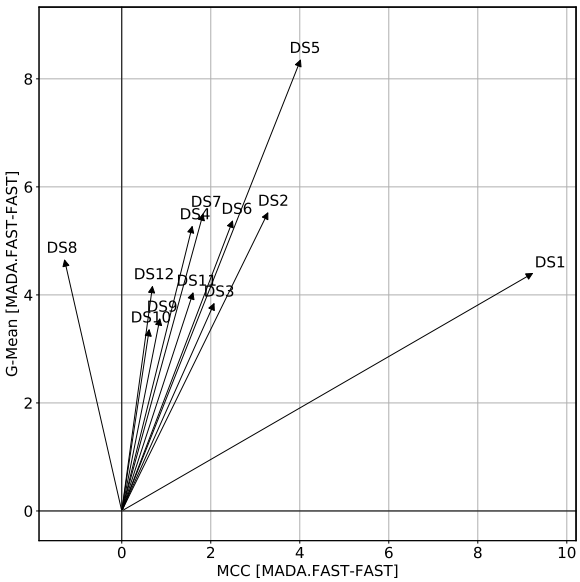
(a) CSB2.FAST vs FAST using GSFrag



(b) MadaBoost.FAST vs FAST using GSFrag

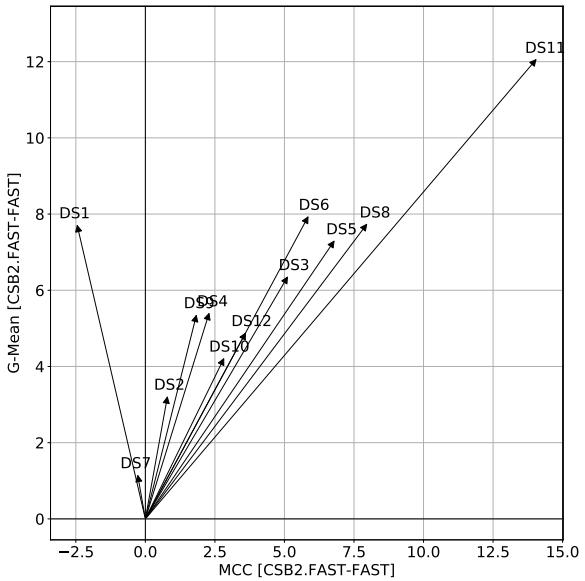


(c) CSB2.FAST vs FAST using ECFP4

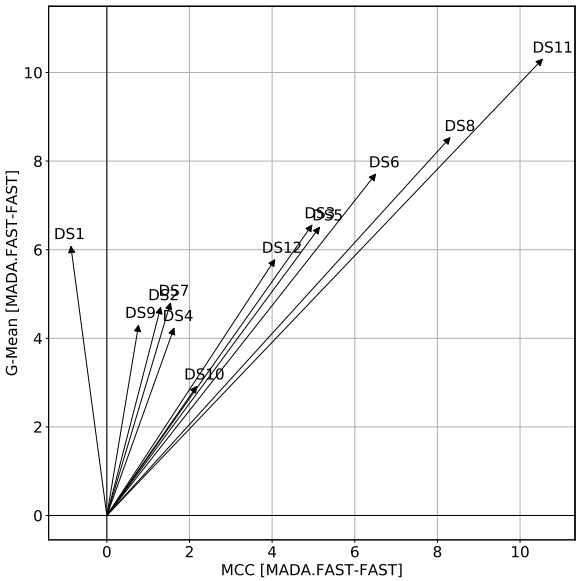


(d) MadaBoost.FAST vs FAST using ECFP4

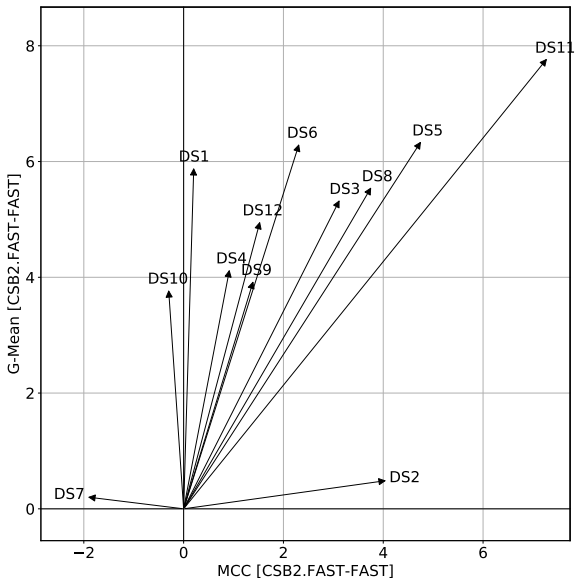
Figure 5: Performance Results for FAST FS algorithm for DT classifier



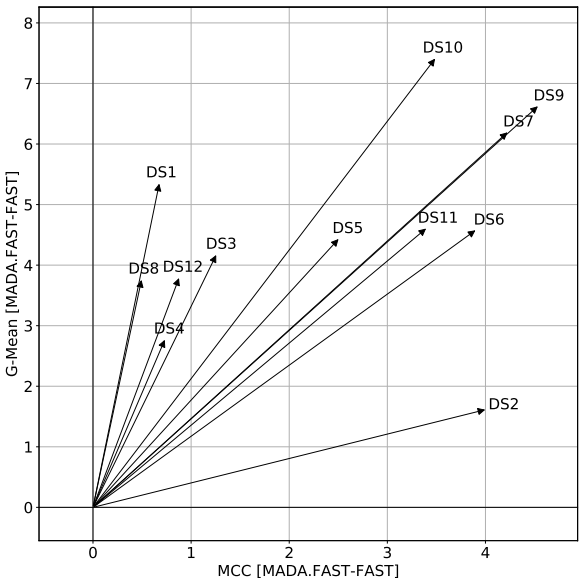
(a) CSB2.FAST vs FAST using GSFrag



(b) MadaBoost.FAST vs FAST using GSFrag

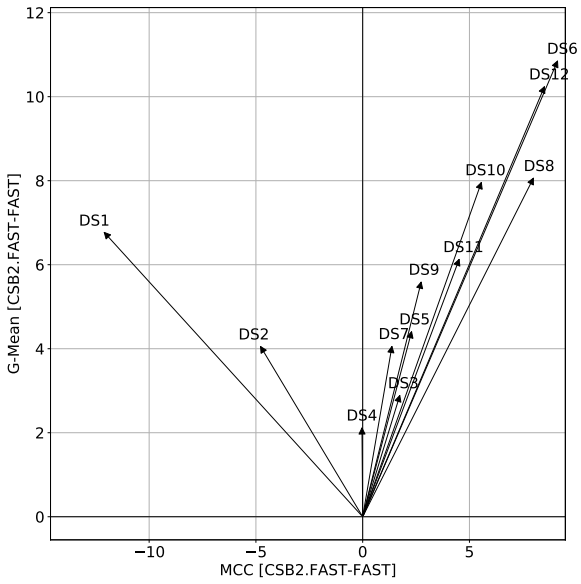


(c) CSB2.FAST vs FAST using ECFP4

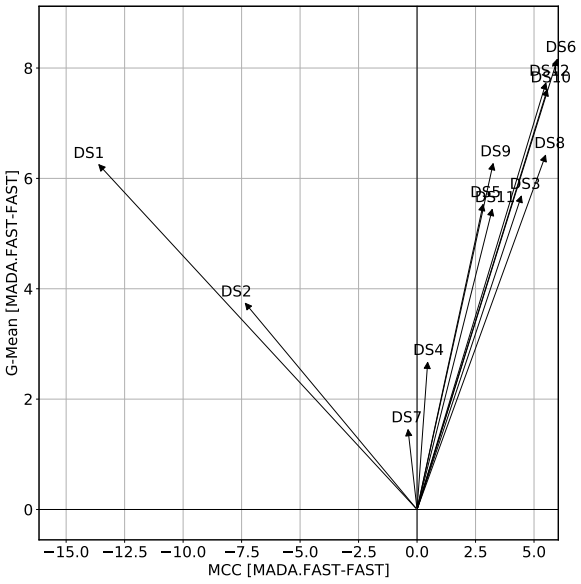


(d) MadaBoost.FAST vs FAST using ECFP4

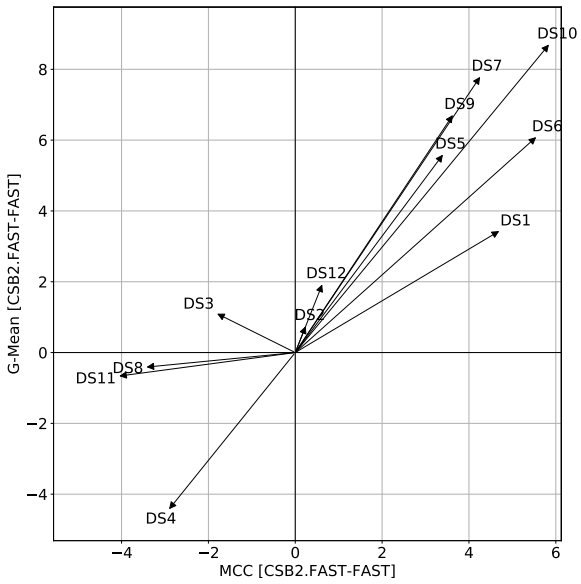
Figure 6: Performance Results for FAST FS algorithm for SVM classifier



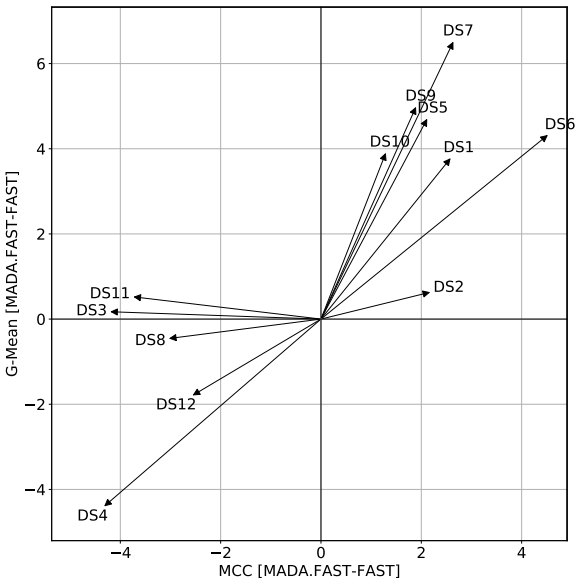
(a) CSB2.FAST vs FAST using GSFrag



(b) MadaBoost.FAST vs FAST using GSFrag

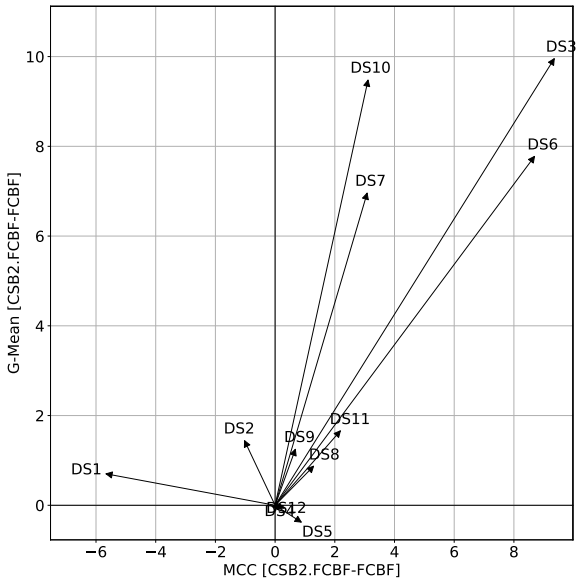


(c) CSB2.FAST vs FAST using ECFP4

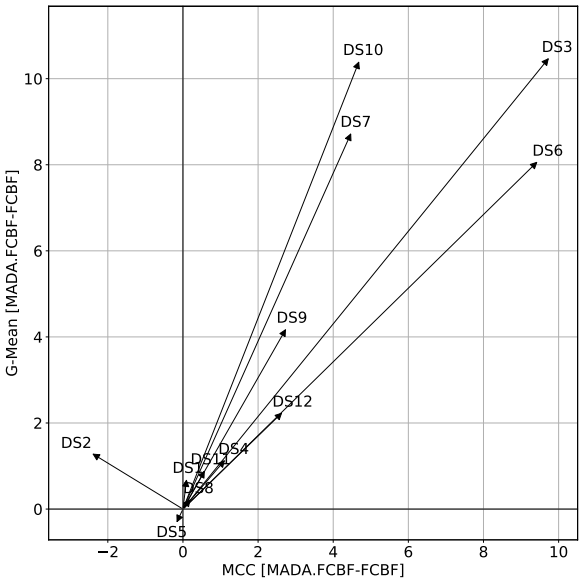


(d) MadaBoost.FAST vs FAST using ECFP4

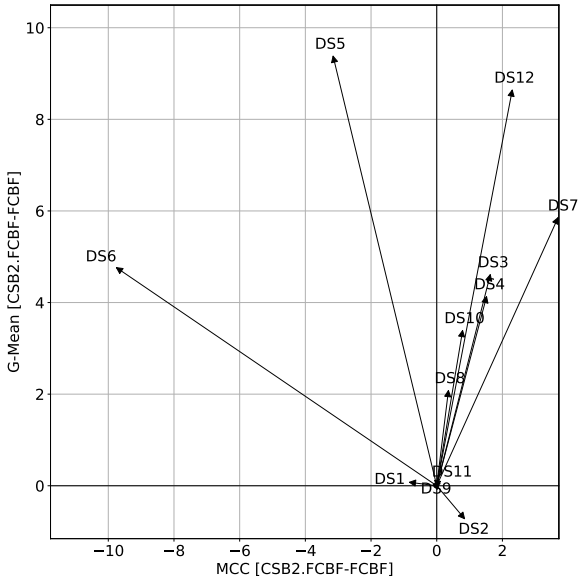
Figure 7: Performance Results for FAST FS algorithm for RF classifier



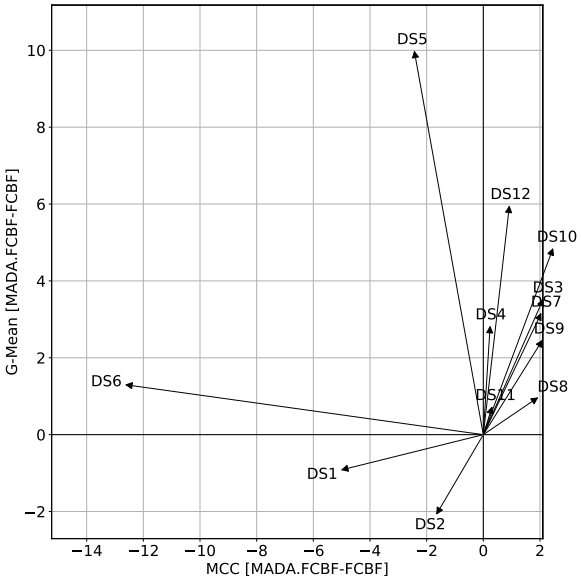
(a) CSB2.FCBF vs FCBF using GSFrag



(b) MadaBoost.FCBF vs FCBF using GSFrag

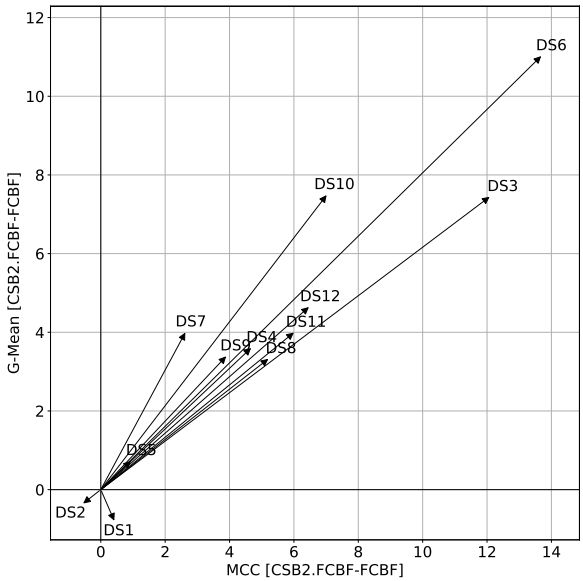


(c) CSB2.FCBF vs FCBF using ECFP4

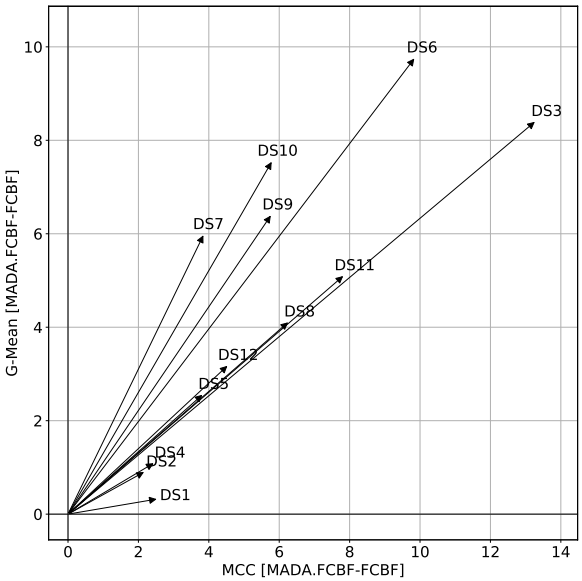


(d) MadaBoost.FCBF vs FCBF using ECFP4

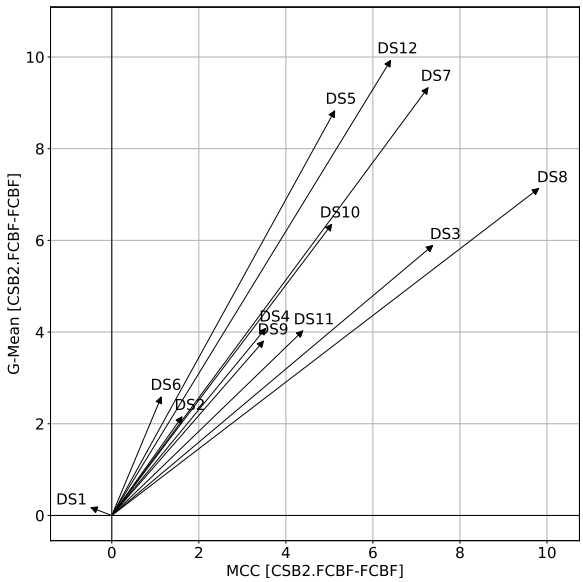
Figure 8: Performance Results for FCBF FS algorithm for DT classifier



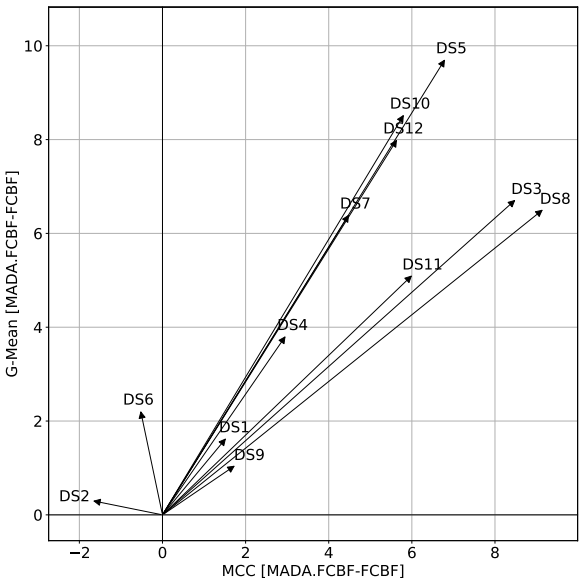
(a) CSB2.FCBF vs FCBF using GSFrag



(b) MadaBoost.FCBF vs FCBF using GSFrag

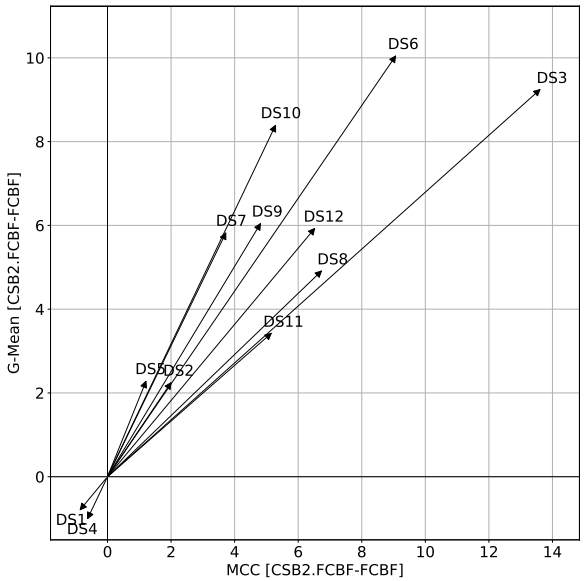


(c) CSB2.FCBF vs FCBF using ECFP4

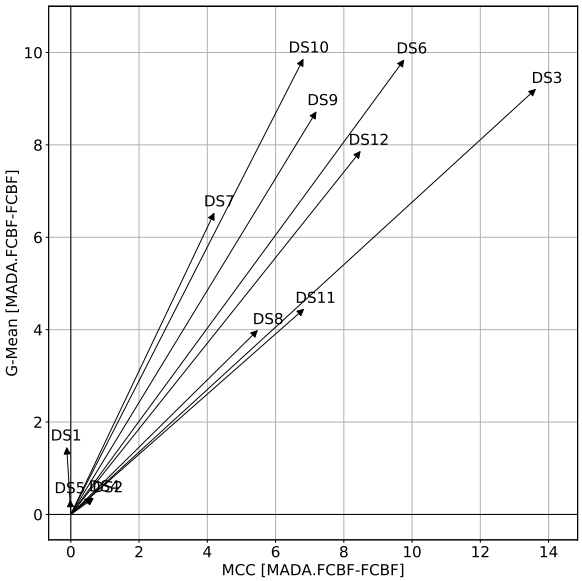


(d) MadaBoost.FCBF vs FCBF using ECFP4

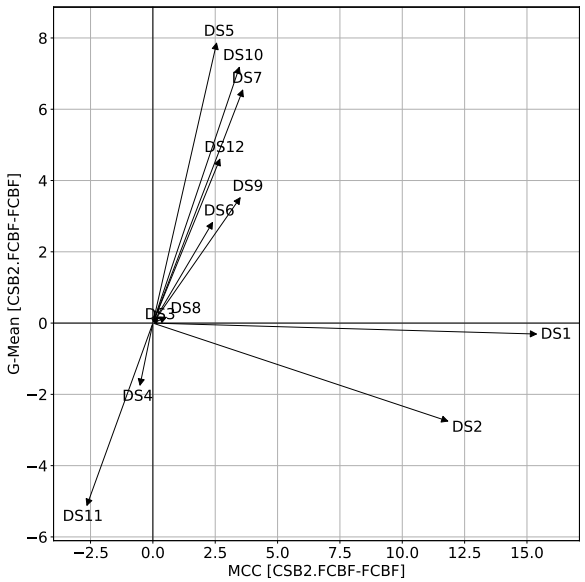
Figure 9: Performance Results for FCBF FS algorithm for SVM classifier



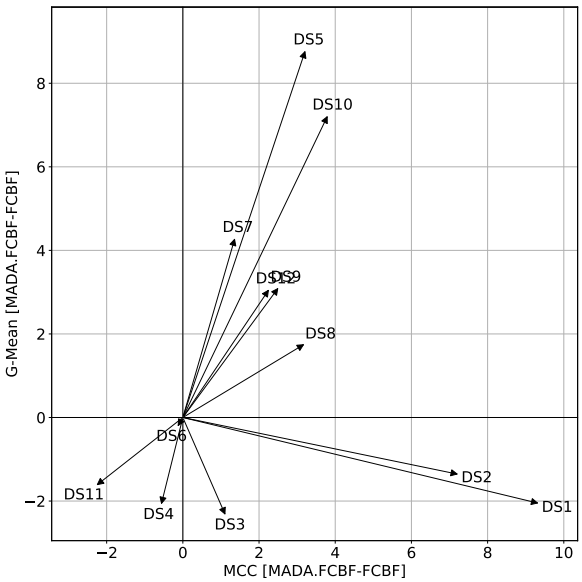
(a) CSB2 vs FCBF using GSFrag



(b) MadaBoost vs FCBF using GSFrag



(c) CSB2 vs FCBF using ECFP4



(d) MadaBoost vs FCBF using ECFP4

Figure 10: Performance Results for FCBF FS algorithm for RF classifier

cant differences for all cases. Tables 2 - 4 show the results of Wilcoxon and Nemenyi tests. To facilitate results analysis in the tables for the Wilcoxon test, we included a tag "Yes/No" to indicate if Wilcoxon test p-value is below the critical level of 0.05, which indicates that there were significant differences between the performance of both algorithms at a confidence level of 95%. Moreover, when there were significant differences, a '+' was added when the algorithm in the column was better than the algorithm in the row and a '−' when the algorithm of the column was worse.

For the Nemenyi test, a tag "Yes/No" was included to indicate if the absolute value of the range difference is greater than CD and the symbols '+' and '−' as in the previous case for the Wilcoxon test. Additionally, in order to facilitate comparisons, we use the Nemenyi graphs of results proposed by Demšar[46] (Figures 11 to 13). In the Figures, we connect with a line the groups of algorithms that were not significantly different. We also show the critical difference in the upper left corner of the graph.

The Wilcoson and Nemenyi test results for DTs are shown in Table 2. In terms of *G-Mean*, the ensemble proposals (CSB2.FAST, MadaBoost.FAST, CSB2.FCBF and MadaBoost.FCBF) were better than the corresponding FS base algorithms (FAST and FCB) according to Wilcoxon test. The Nemenyi test results were consistent with these differences; Figure 11 (b,d) graphically shows these differences, highlighting that the increase in performance for ensemble proposals is significant compared to the FS base methods and showing that all ensemble proposals have a very similar behavior as there is no significant difference between them. In terms of MCC, Figure 11 (a,c), the performance was very similar as described above, with the exception of FCBF-based algorithms where MadaBoost.FCBF performed better but the difference with others (FCBF and CSB2.FCBF) was not statistically significant.

Table 3 shows the results for the SVM classifier. The ensemble proposals were better than the corresponding FS base algorithms both in terms of *G-Mean* and *MCC* according to the Wilcoxon and Nemenyi tests. Figure 12 shows the existence of significant

differences, with better performance for the ensemble proposals. MadaBoost.FAST and MadaBoost.FCBF achieved the best performance although without significant differences compared to CSB2.FAST and CSB2.FCBF. The Wilcoson and Nemenyi test results for RF are in Table 4 and Figure 13. The results obtained were similar to the DT and SVM, and ensemble proposals were better than the corresponding FS base algorithms, with the only exception of FAST-based algorithms where there was no significant difference between the algorithms FAST, MadaBoost.FAST and CSB2.FAST in terms of *MCC*.

Table 2: Statistical test results using DT

MCC				G-Mean			
	FAST	CSB2.FAST	MadaBoost.FAST		FAST	CSB2.FAST	MadaBoost.FAST
Mean	0.1935	0.2038	0.2114	Mean	0.5982	0.6490	0.6490
Ranks	2.6250	1.8750	1.5000	Ranks	3.0000	1.4167	1.5833
Nemenyi CD	0.6767			Nemenyi CD	0.6767		
Win/draw/loss		18/0/6	21/0/3	Win/draw/loss		24/0/0	24/0/0
Wilcoxon p-value		0.0086	0.0014	Wilcoxon p-value		0.00002	0.00002
R+/R-		242.0/58.0	262.0/38.0	R+/R-		300.0/0.0	300.0/0.0
Wilcoxon test		YES(+)	YES(+)	Wilcoxon test		YES(+)	YES(+)
Ranks difference		0.7500	1.1250	Ranks difference		1.5833	1.4167
Nememyi test		YES(+)	YES(+)	Nememyi test		YES(+)	YES(+)
MCC				G-Mean			
	FCBF	CSB2.FCBF	MadaBoost.FCBF		FCBF	CSB2.FCBF	MadaBoost.FCBF
Mean	0.1981	0.2019	0.2032	Mean	0.6153	0.6500	0.6498
Ranks	2.2083	1.9583	1.8333	Ranks	2.6667	1.7500	1.5833
Nemenyi CD	0.6767			Nemenyi CD	0.6767		
Win/draw/loss		14/0/10	15/0/9	Win/draw/loss		19/0/5	21/0/3
Wilcoxon p-value		0.3037	0.3458	Wilcoxon p-value		0.0004	0.0002
R+/R-		186.0/114.0	183.0/117.0	R+/R-		274.0/26.0	279.0/21.0
Wilcoxon test		No	No	Wilcoxon test		YES(+)	YES(+)
Ranks difference		0.2500	0.3750	Ranks difference		0.9167	1.0833
Nememyi test		No	No	Nememyi test		YES(+)	YES(+)

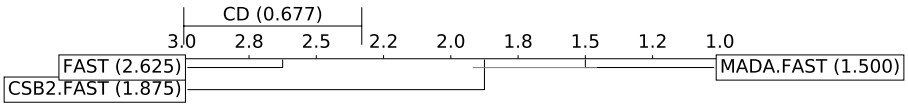
Table 3: Statistical test results using SVM

MCC				G-Mean			
	FAST	CSB2.FAST	MadaBoost.FAST		FAST	CSB2.FAST	MadaBoost.FAST
Mean	0.2104	0.2382	0.2404	Mean	0.6347	0.6894	0.6892
Ranks	2.7500	1.6667	1.5833	Ranks	3.0000	1.3750	1.6250
Nemenyi CD	0.6767			Nemenyi CD	0.6767		
Win/draw/loss		19/0/5	23/0/1	Win/draw/loss		24/0/0	24/0/0
Wilcoxon p-value		0.0012	0.00006	Wilcoxon p-value		0.00002	0.00002
R+/R-		263.0/37.0	291.0/9.0	R+/R-		300.0/0.0	300.0/0.0
Wilcoxon test		YES(+)	YES(+)	Wilcoxon test		YES(+)	YES(+)
Ranks difference		1.0833	1.1667	Ranks difference		1.6250	1.3750
Nememyi test		YES(+)	YES(+)	Nememyi test		YES(+)	YES(+)
MCC				G-Mean			
	FCBF	CSB2.FCBF	MadaBoost.FCBF		FCBF	CSB2.FCBF	MadaBoost.FCBF
Mean	0.1777	0.2240	0.2250	Mean	0.6369	0.6849	0.6860
Ranks	2.8333	1.6250	1.5417	Ranks	2.9167	1.6667	1.4167
Nemenyi CD	0.6767			Nemenyi CD	0.6767		
Win/draw/loss		22/0/2	22/0/2	Win/draw/loss		22/0/2	24/0/0
Wilcoxon p-value		0.00006	0.00004	Wilcoxon p-value		0.00004	0.00002
R+/R-		291.0/9.0	293.0/7.0	R+/R-		294.0/6.0	300.0/0.0
Wilcoxon test		YES(+)	YES(+)	Wilcoxon test		YES(+)	YES(+)
Ranks difference		1.2083	1.2917	Ranks difference		1.2500	1.5000
Nememyi test		YES(+)	YES(+)	Nememyi test		YES(+)	YES(+)

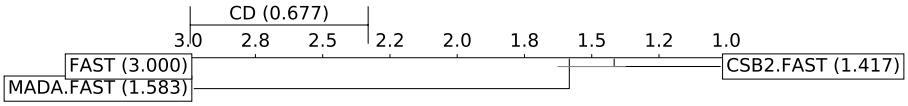
Table 4: Statistical test results using RF

MCC				G-Mean			
	FAST	CSB2.FAST	MadaBoost.FAST		FAST	CSB2.FAST	MadaBoost.FAST
Mean	0.1932	0.2057	0.1941	Mean	0.6295	0.6763	0.6675
Ranks	2.2917	1.6667	2.0417	Ranks	2.7500	1.4167	1.8333
Nemenyi CD	0.6767			Nemenyi CD	0.6767		
Win/draw/loss		16/0/8	15/0/9	Win/draw/loss		21/0/3	21/0/3
Wilcoxon p-value		0.1451	0.5677	Wilcoxon p-value		0.0001	0.0002
R+/R-		201.0/99.0	170.0/130.0	R+/R-		285.0/15.0	279.0/21.0
Wilcoxon test		No	No	Wilcoxon test		YES(+)	YES(+)
Ranks difference		0.62500	0.25000	Ranks difference		1.3333	0.9167
Nememyi test		No	No	Nememyi test		YES(+)	YES(+)
MCC				G-Mean			
	FCBF	CSB2.FCBF	MadaBoost.FCBF		FCBF	CSB2.FCBF	MadaBoost.FCBF
Mean	0.1713	0.2092	0.2079	Mean	0.6347	0.6682	0.6690
Ranks	2.5833	1.7917	1.6250	Ranks	2.5000	1.7083	1.7917
Nemenyi CD	0.6767			Nemenyi CD	0.6767		
Win/draw/loss		19/0/5	19/0/5	Win/draw/loss		18/0/6	18/0/6
Wilcoxon p-value		0.0007	0.0005	Wilcoxon p-value		0.0022	0.0033
R+/R-		269.0/31.0	271.0/29.0	R+/R-		257.0/43.0	253.0/47.0
Wilcoxon test		YES(+)	YES(+)	Wilcoxon test		YES(+)	YES(+)
Ranks difference		0.7917	0.9583	Ranks difference		0.7917	0.7083
Nememyi test		YES(+)	YES(+)	Nememyi test		YES(+)	YES(+)

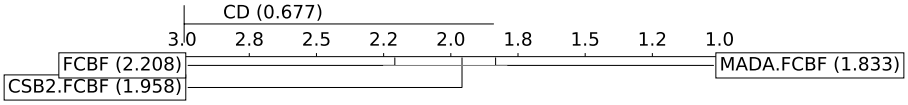
The ability to reduce features was another factor under study in the evaluation of our proposal. From this perspective we present the Nemenyi graphs (Figure 14), and the Table S8 of the supplementary material (SupMaterial_File-1.pdf) provides all the values for the Wilcoxon and Nemenyi tests. We can conclude that the impact on the ability to reduce fea-



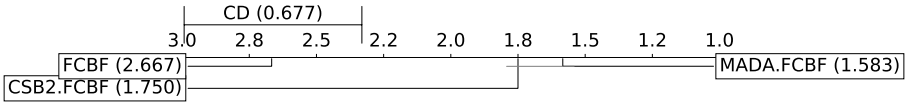
(a) FAST FS results in terms of MCC



(b) FAST FS results in terms of G-Mean

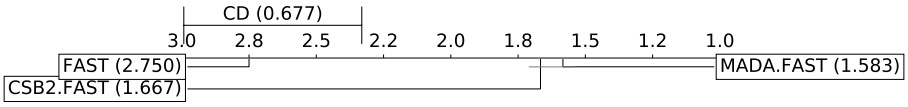


(c) FCBF FS results in terms of MCC

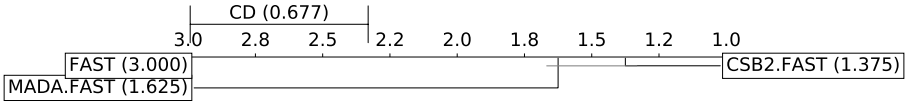


(d) FCBF FS results in terms of G-Mean

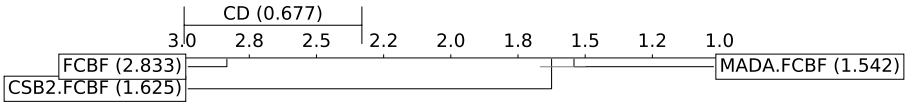
Figure 11: Nemenyi test results for all datasets and representation models using the DT classifier



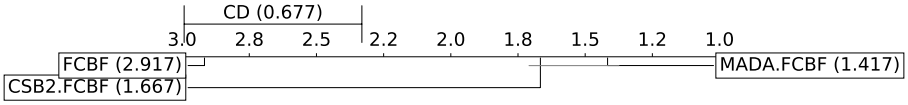
(a) FAST FS results in terms of MCC



(b) FAST FS results in terms of G-Mean

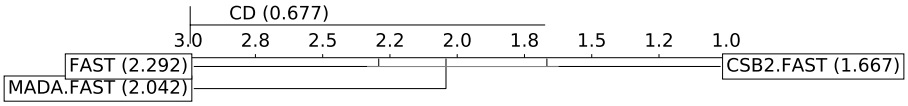


(c) FCBF FS results in terms of MCC

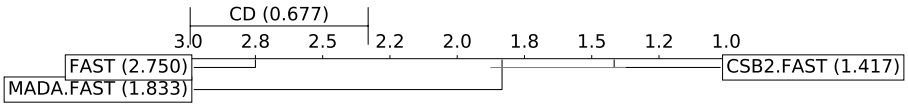


(d) FCBF FS results in terms of G-Mean

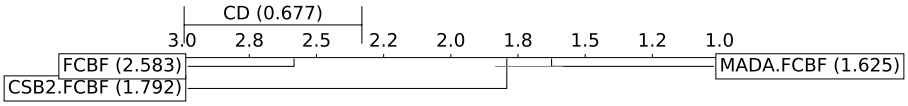
Figure 12: Nemenyi test results for all datasets and representation models using the SVM classifier



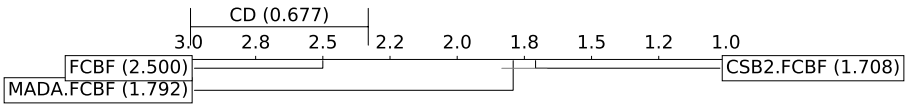
(a) FAST FS results in terms of MCC



(b) FAST FS results in terms of G-Mean



(c) FCBF FS results in terms of MCC



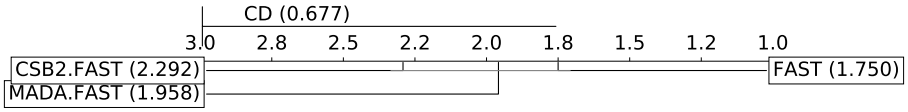
(d) FCBF FS results in terms of G-Mean

Figure 13: Nemenyi test results for all datasets and representation models using the RF classifier

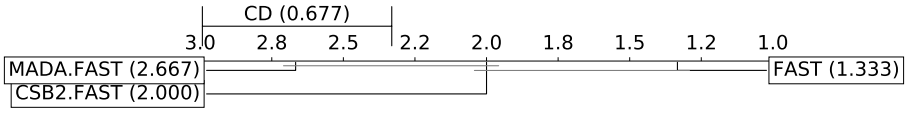
tures of the proposals depended on the FS base method chosen. As shown in Figure 14 (a,b,c) for the approaches based on FAST, a similar behavior was observed, with no significant differences between FAST, CSB2.FAST and MadaBoost.FAST in terms of reduction. However, for the FCBF-based approaches (Figure 14 (d,e,f)) a penalty in the reduction of features was observed, however, this behavior was compensated by an increase in performance for the CSB2.FCBF and MadaBoost.FCBF approaches as described above.

3.2 Comparison against SMOTE Method

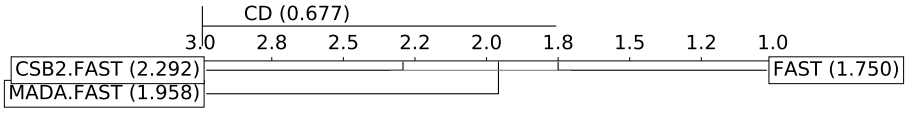
In the previous seciton our proposals CSB2.FAST, CSB2.FCBF, MADA.FAST, MADA.FCBF were compared with the respective base methods, FAST or FCBF, using undersampling to deal with the class-imbalance problem. In this section we extend the study using SMOTE (Synthetic Minority Oversampling Technique), another well-known method for class-imbalance problems, and including a new model to represent molecular structures based on molecular descriptors. To simplify the comparisons we only use *G-Mean* to evaluate the performance



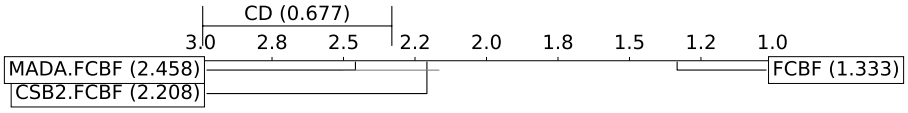
(a) Reduction of FAST based algorithms using DT



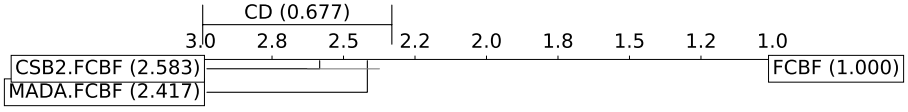
(b) Reduction of FAST based algorithms using SVM



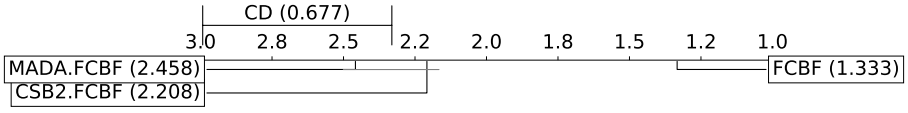
(c) Reduction of FAST based algorithms using RF



(d) Reduction of FCBF based algorithms using DT



(e) Reduction of FCBF based algorithms using SVM



(f) Reduction of FCBF based algorithms using RF

Figure 14: Nemenyi test results for all datasets and representation models in terms of reduction

of the classifiers.

Figure 15 shows the Nemenyi test for the classifier performance results when comparing the base methods FAST and FCBF using undersampling and SMOTE against the proposals CSB2.FAST, MADA.FAST, CSB2.FCBF and MADA.FCBF for the three classifiers (DT,SVM, RF), evaluating 36 datasets (12 datasets $\times$ 3 representation models, GSFrag, ECFP4, and CDK Descriptors). Tables S9 - S12 of the supplementary material (SupMaterial_File-1.pdf) provides all the results for the extended methods and the Nemenyi tests.

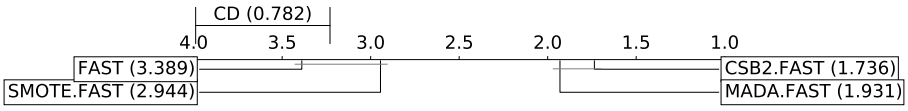
As shown in the Nemenyi graphs, the superior performance of our proposal was confirmed when compared to the base methods. The proposals based on FAST showed a statistically significant improvement for all classifiers. The same results were observer for FCBF, with the exception of the RF classifier where the results of CSB2.FCBF and MADA.FCBF were significantly better than SMOTE.FCBF but not significantly better than those obtained by FCBF.

Figure 16 shows the results of the Nemenyi test in terms of reduction. The FAST based proposals showed similar behavior for DTs and RFs. The best results were obtained with MADA.FAST and FAST, without significant differences between them.

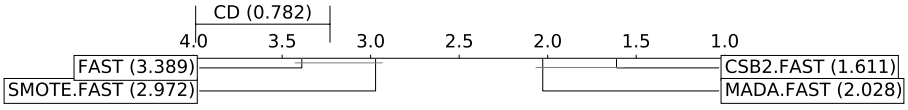
The FCBF based proposals also showed a similar behavior for the DT and RF classifiers. In this case, FCBF achieved the best results, and no significant differences were found between SMOTE.FCBF, CSB2.FCBF and MADA.FCBF.

4 Conclusions

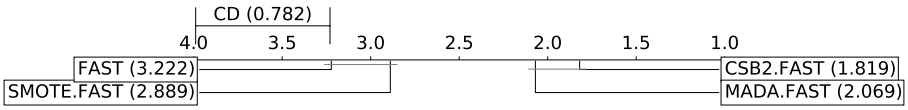
In this work, we have demonstrated the usefulness of approaching the problem of feature selection for class-imbalance datasets based on the construction of boosting feature selection ensembles. We addressed the toxicity prediction problem and achieved better results with our proposal compared with the use of standard FS methods using undersampling and SMOTE.



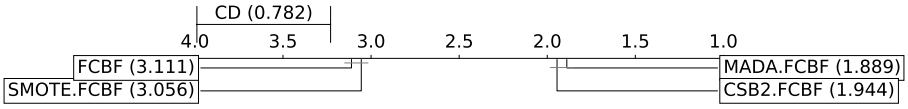
(a) Nemenyi results (FAST) for DT



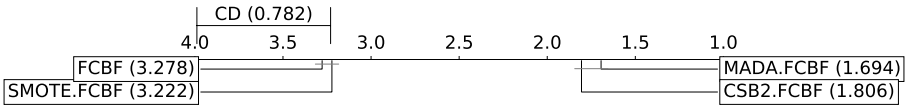
(b) Nemenyi results (FAST) for SVM



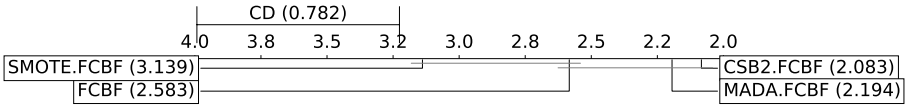
(c) Nemenyi results (FAST) for RF



(d) Nemenyi results (FCBF) for DT

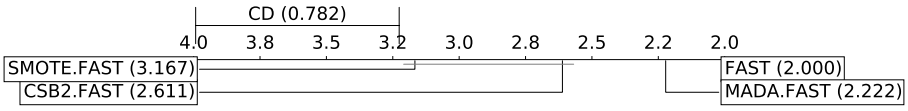


(e) Nemenyi results (FCBF) for SVM

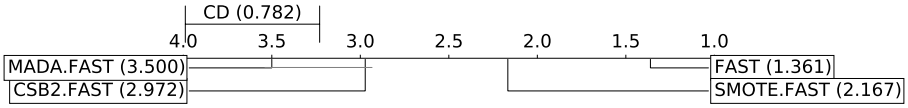


(f) Nemenyi results (FCBF) for RF

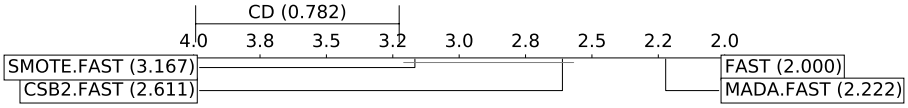
Figure 15: Nemenyi test results in terms of *G-Mean*, extended to include the CDK Descriptor representation model and the SMOTE algorithm



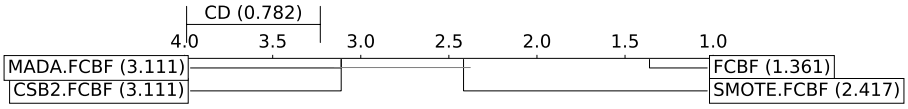
(a) Nemenyi results (FAST) for DT



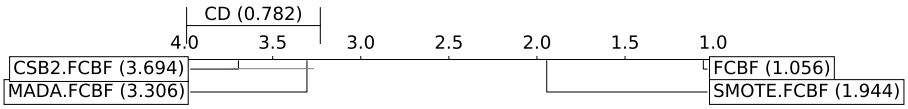
(b) Nemenyi results (FAST) for SVM



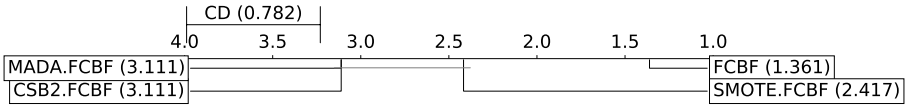
(c) Nemenyi results (FAST) for RF



(d) Nemenyi results (FCBF) for DT



(e) Nemenyi results (FCBF) for SVM



(f) Nemenyi results (FCBF) for RF

Figure 16: Nemenyi test results in terms of reduction (r), extended to include the CDK Descriptor representation model and the SMOTE algorithm

We have shown a superior performance using different feature selection methods and three models to represent the molecular structures.

An important characteristic of the proposal is its general applicability, allowing the choice of two elements, the FS method and the ensemble method. For the ensembles used (CSB2, MadaBoost), the behavior was similar, and no significant differences were detected between them. For FS methods (FAST, FCBF), the proposals (CSB2.FAST, MadaBoost.FAST, CSB2.FCBF and MadaBoost.FCBF) improved the results in terms of classifier performance. In terms of reduction, the results obtained for FAST-based approaches were superior to those obtained with FCBF. Thus, CSB2.FAST and MadaBoost.FAST achieved a significant increase in performance without damaging the ability to reduce features compared to the base method.

Although this work focuses on the application for toxicity prediction, the results are encouraging for developing this research in the direction of obtaining improvements for other problems in cheminformatics. Moreover, the proposal offers the possibility to use different feature selection algorithms and other ensemble methods.

Supporting Information Available

- Tables S1 - S12 (SupMaterial_File-1.pdf)
- Datasets in SMILE format (SupMaterial_File-2.xlsx)

Acknowledgement

This work was supported in part by grant PID2019-109481GB-I00 / AEI / 10.13039/501100011033 of the Spanish Ministry of Science and Innovation, by grant UCO-1264182 of the Junta de Andalucía Excellence in Research program, by Grant PP2019-Submod-1.2 of Cordoba University, and FEDER funds.

References

(1) Masoudi-Sobhanzadeh, Y.; Motieghader, H.; Masoudi-Nejad, A. FeatureSelect: a software for feature selection based on machine learning approaches. *BMC bioinformatics* **2019**, *20*, 170.

(2) Chang, C.-Y.; Hsu, M.-T.; Esposito, E. X.; Tseng, Y. J. Oversampling to overcome overfitting: exploring the relationship between data set composition, molecular descriptors, and predictive modeling methods. *J. Chem. Inf. Model.* **2013**, *53*, 958–971.

(3) Galar, M.; Fernandez, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Syst. Man Cybern. Part C-Appl. Rev.* **2011**, *42*, 463–484.

(4) de Haro-García, A.; Cerruela-García, G.; García-Pedrajas, N. Ensembles of Feature Selectors for dealing with Class-Imbalanced Datasets: A proposal and comparative study. *Inf. Sci.* **2020**, *540*, 89–116.

(5) Ezzat, A.; Wu, M.; Li, X.-L.; Kwoh, C.-K. Drug-target interaction prediction via class imbalance-aware ensemble learning. *BMC bioinformatics* **2016**, *17*, 267–276.

(6) Klimenko, K.; Rosenberg, S. A.; Dybdahl, M.; Wedebye, E. B.; Nikolov, N. G. QSAR modelling of a large imbalanced aryl hydrocarbon activation dataset by rational and random sampling and screening of 80,086 REACH pre-registered and/or registered substances. *Plos one* **2019**, *14*, e0213848.

(7) Li, Q.; Wang, Y.; Bryant, S. H. A novel method for mining highly imbalanced high-throughput screening data in PubChem. *Bioinformatics* **2009**, *25*, 3310–3316.

(8) Chen, C.; Liaw, A.; Breiman, L. Using random forest to learn imbalanced data, Dept. Statistics. *Univ. California, Berkeley, CA, Tech. Rep* **2004**, *666*.

(9) Chang, C.-C.; Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–27.

(10) Joachims, T. In *Advances in Kernel Methods - Support Vector Learning*; Schölkopf, B., Burges, C., Smola, A., Eds.; MIT Press, 1999.

(11) Jiang, J.; Wang, R.; Wang, M.; Gao, K.; Nguyen, D. D.; Wei, G.-W. Boosting Tree-Assisted Multitask Deep Learning for Small Scientific Datasets. *J. Chem. Inf. Model.* **2020**, *60*, 1235–1244, PMID: 31977216.

(12) Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357.

(13) Newby, D.; Freitas, A. A.; Ghafourian, T. Coping with unbalanced class data sets in oral absorption models. *J. Chem. Inf. Model.* **2013**, *53*, 461–474.

(14) Chen, J.; Tang, Y. Y.; Fang, B.; Guo, C. In silico prediction of toxic action mechanisms of phenols for imbalanced data with Random Forest learner. *J. Mol. Graphics Modell.* **2012**, *35*, 21–27.

(15) Kondratovich, E.; Zhokhova, N.; Baskin, I.; Palyulin, V.; Zefirov, N. Fragmental descriptors in (Q) SAR: Prediction of the assignment of organic compounds to pharmacological groups using the support vector machine approach. *Russ. Chem. Bull.* **2009**, *58*, 657–662.

(16) Zakharov, A. V.; Peach, M. L.; Sitzmann, M.; Nicklaus, M. C. QSAR modeling of imbalanced high-throughput screening data in PubChem. *J. Chem. Inf. Model.* **2014**, *54*, 705–712.

(17) Imrie, F.; Bradley, A. R.; van der Schaar, M.; Deane, C. M. Protein family-specific models using deep neural networks and transfer learning improve virtual screening and highlight the need for more data. *J. Chem. Inf. Model.* **2018**, *58*, 2319–2330.

(18) Batista, G. E.; Prati, R. C.; Monard, M. C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter* **2004**, *6*, 20–29.

(19) Korkmaz, S. Deep Learning-Based Imbalanced Data Classification for Drug Discovery. *J. Chem. Inf. Model.* **2020**,

(20) Sun, J.; Carlsson, L.; Ahlberg, E.; Norinder, U.; Engkvist, O.; Chen, H. Applying mondrian cross-conformal prediction to estimate prediction confidence on large imbalanced bioactivity data sets. *J. Chem. Inf. Model.* **2017**, *57*, 1591–1598.

(21) Al-Shahib, A.; Breitling, R.; Gilbert, D. Feature selection and the class imbalance problem in predicting protein function from sequence. *Appl. Bioinformatics* **2005**, *4*, 195–203.

(22) Maldonado, S.; Weber, R.; Famili, F. Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines. *Inf. Sci.* **2014**, *286*, 228–246.

(23) Han, C.; Tan, Y.-K.; Zhu, J.-H.; Guo, Y.; Chen, J.; Wu, Q.-Y. Online feature selection of class imbalance via pa algorithm. *J. Comput. Sci. Technol.* **2016**, *31*, 673–682.

(24) Song, Q.; Ni, J.; Wang, G. A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Trans. Knowl. Data Eng.* **2011**, *25*, 1–14.

(25) Hall, M. Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. ICML. 2000.

(26) for Advancing Translational Sciences (NCATS), N. C. Tox21 Data Challenge. `https://tripod.nih.gov/tox21/challenge/`, (accessed Jun 20, 2020).

(27) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754, PMID: 20426451.

(28) O'Boyle, N. M.; Sayle, R. A. Comparing structural fingerprints using a literature-based similarity benchmark. *J. Cheminformatics* **2016**, *8*, 1–14.

(29) Antczak, P.; Ortega, F.; Chipman, J. K.; Falciani, F. Mapping drug physico-chemical features to pathway activity reveals molecular networks linked to toxicity outcome. *PLoS One* **2010**, *5*, e12385.

(30) Abdelaziz, A.; Spahn-Langguth, H.; Schramm, K.-W.; Tetko, I. V. Consensus modeling for HTS assays using in silico descriptors calculates the best balanced accuracy in Tox21 challenge. *Front. Environ. Sci.* **2016**, *4*, 2.

(31) Sosnin, S.; Karlov, D.; Tetko, I. V.; Fedorov, M. V. Comparative study of multitask toxicity modeling on a broad chemical space. *J. Chem. Inf. Model.* **2018**, *59*, 1062–1072.

(32) Skvortsova, M.; Baskin, I.; Skvortsov, L.; Palyulin, V.; Zefirov, N.; Stankevich, I. Chemical graphs and their basis invariants. *J. Mol. Struct.* **1999**, *466*, 211–217.

(33) Skvortsova, M.; Fedyaev, K.; Baskin, I.; Palyulin, V.; Zefirov, N. A new technique for coding chemical structures based on basis fragments. Doklady Chemistry. 2002; pp 33–36.

(34) Sushko, I.; Novotarskyi, S.; Körner, R.; Pandey, A. K.; Rupp, M.; Teetz, W.; Brandmaier, S.; Abdelaziz, A.; Prokopenko, V. V.; Tanchuk, V. Y., et al. Online chemical modeling environment (OCHEM): web platform for data storage, model development and publishing of chemical information. *J. Comput.-Aided Mol. Des.* **2011**, *25*, 533–554.

(35) Willighagen, E. L.; Mayfield, J. W.; Alvarsson, J.; Berg, A.; Carlsson, L.; Jeliazkova, N.; Kuhn, S.; Pluskal, T.; Rojas-Chertó, M.; Spjuth, O., et al. The Chemistry Development Kit (CDK) v2. 0: atom typing, depiction, molecular formulas, and substructure searching. *J. Cheminformatics* **2017**, *9*, 33.

(36) Chemaxon, ChemAxon products. `https://chemaxon.com`, (accessed Jun 20, 2020).

(37) Chemaxon, ChemAxon documentation. `https://docs.chemaxon.com/display/Documentation.html`, (accessed Jun 20, 2020).

(38) Pérez-Rodríguez, J.; deHaro García, A.; del Castillo, J. A.; García-Pedrajas, N. A general framework for boosting feature subset selection algorithms. *Inf. Fusion* **2018**, *44*, 147–175.

(39) Domingo, C.; Watanabe, O. MadaBoost: A Modification of AdaBoost. Proceedings of the 13th Annual Conference on Computational Learning Theory. 2000; p 180–189.

(40) Webb, G. I. MultiBoosting: A Technique for Combining Boosting and Wagging. *Mach. Learn.* **2000**, *40*, 159–196.

(41) Nikolaou, N.; Edakunni, N.; Kull, M.; Flach, P.; Brown, G. Cost-sensitive boosting algorithms: Do we really need them? *Mach. Learn.* **2016**, *104*, 359–384.

(42) Ting, K. M. A comparative study of cost-sensitive boosting algorithms. Proceedings of the 17th International Conference on Machine Learning. San Francisco, USA, 2000; p 983–990.

(43) Delgado, R.; Tibau, X.-A. Why Cohen's Kappa should be avoided as performance measure in classification. *PloS one* **2019**, *14*.

(44) Tharwat, A. Classification assessment methods. *Appl. Comp. Inform.* **2018**,

(45) Iman, R. L.; Davenport, J. M. Approximations of the critical region of the fbietkan statistic. *Communications in Statistics-Theory and Methods* **1980**, *9*, 571–595.

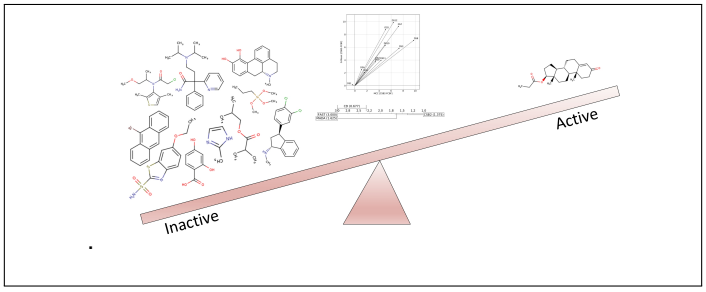
(46) Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.

(47) Sheskin, D. J. *Handbook of parametric and nonparametric statistical procedures*; crc Press, 2020.

(48) Nemenyi, P. Distribution-free multiple comparisons (doctoral dissertation, princeton university, 1963). *Dissertation Abstracts International* **1963**, *25*, 1233.

(49) Maudes, J.; Rodríguez, J. J.; García-Osorio, C.; Pardo, C. Random projections for linear SVM ensembles. *Appl. Intell.* **2011**, *34*, 347–359.

Graphical TOC Entry



.