

Population-based detection of children ASD/ADHD comorbidity from atypical sensory processing

Supplementary material

Manuel Fernández-Delgado, Sara Cruz, Eva Cernadas,
Heba Alateyat, María Tubío-Fungueiriño, Adriana Sampaio,
Angel Carracedo, Montse Fernández-Prieto

A Classifiers list

Table 1 lists the 42 standard and 8 proposed classifiers, alongside with the implementations, used in the experimental work.

B Hyper-parameter tuning of standard classifiers

Table 2 reports the values used for the hyper-parameter tuning in the standard classifiers. The values for those classifiers executed in R were selected using the `getModelInfo` function of the `caret` R package [1]. The notation `n:p:m` means values from `n` to `m` with step `p`, with `p=1` when missing.

C Pseudo-code of population-based classifiers

Algorithm 1, continued in algorithm 2, describes the proposed feature population classifiers `fpv`, `fps`, `mrf`, `mrf5`, `fpr` and `fnf`. Note that $|\mathcal{A}|$ is the cardinal or number of elements in set $\mathcal{A}$.

D Feature selection methods

Table 3 lists the 22 methods of feature selection that were combined to the best classifiers in Table 1 for the experimental work.

Table 1 List of 42 standard and 8 proposed (identified by *) classifiers and languages (M=Matlab, O=Octave, P=Python) used for ADHD detection.

Name	Description	Language and package
adaboost	AdaboostM1 ensemble	M,R(adabag)
awnb	Attrib. weight. naive Bayes	R(bncclassify)
bagging	Bagging ensemble	M,R(caret)
ctree	Classification tree	M,O,P,R(ipred)
dlda	Diag. linear discr. anal.	M
elm	Extreme learning machine	O,R(elmNNRcpp)
earth	Multivariate adaptive regression spline	R(earth)
fda	Flexible discr. anal.	R(mda)
fnn*	Feature nearest neighbors	O
fps*	Feature population sum	O
fpv*	Feature popul. voting	O
frp*	Feature relevance popul.	O
gbm	Gradient boosting machine	P,R(gbm)
gpr	Gaussian process regression	R(kernlab)
kfd	Kernel Fisher discriminant	P(Kernel FDA)
knn	K-nearest neighbor classifier	M,R(class)
lda	Linear discriminant analysis	P,R(MASS)
lvq	Learning vector quantization	R(class)
manb	Model averaged naive Bayes	R(bncclassify)
mda	Mixture discr. anal.	R(mda)
mlp	Multi-layer perceptron	M,P,R(nnet,neuralnet)
mrf*	Most relevant features	O
mrf5*	Most relevant 5 features	O
nb	Naive Bayes	R(klaR)
nbDiscrete	Naive Bayes discrete features	R(bncclassify)
nbSearch	Semi-naive structure learner wrapper	R(bncclassify)
nn*	Nearest neighbor classifier	O
nnc*	Class nearest neighbors	O
pnn	Probabilistic neural network	M
qda	Quadratic discr. anal.	M
rforest	Random forest ensemble	M,P,R(random Forest)
rpart	Recursive partitioning tree	R(rpart)
svm	Support vector machine	M,O(LibSVM),P,R(kernlab)

Table 2 Hyper-parameter tuning of standard classifiers.

Classifier	Hyper-parameter tuning details
adaboost	No. estimators=5:5:31
bagging	No. estimators=5:5:31
elm	No. hidden neurons=3:2: N (N =No. training vectors)
earth	Max. terms in pruned model=2:2:18
gbm	Max. depth=50:50:500, No. trees=1:10
kfd	Inverse of Gaussian kernel spread $\gamma = 1/(2\sigma^2)=2^{(-20:20)}$
knn	No. neighbors=5:2:23
lvq	No. neighbors=1:5:46
manb	Smoothing (α) for Bayesian parameter estimation=1:9 Prior probability for arc class-feature: 0.1:0.09:0.9
mlp	No. hidden neurons=1:20
nbDiscrete	Smoothing (same as in manb)=0:9
pnn	$\sigma=\{0.001,0.005,0.007,0.008,0.01,0.05,0.1,0.2,0.3,0.4,0.5,0.6, 0.7\}$
rforest	No. estimators=5:5:31, Max. features=3: I (I =No. features)
rpart	Complexity parameter= $\{0,0.03,0.06,0.09,0.12,0.15,0.19,0.22,0.25, 0.28\}$
svm	Regularization $\lambda=2^{(-5:2:15)}$ Inverse of Gaussian kernel spread $\gamma = 1/(2\sigma^2)=2^{(-20:2:20)}$

Algorithm 1: Feature population classifiers.

```
1 Algorithm:  $\{Y_r\}_{r=1}^6 = \text{FPC}(\{\mathbf{x}_n\}_{n=1}^N, \{c_n\}_{n=1}^N, \mathbf{x})$   
   Data:  $\{\mathbf{x}_n\}_{n=1}^N$ : training patterns:  $\mathbf{x}_n = (j_{n1}, \dots, j_{nI})$ ;  $\{c_n\}_{n=1}^N$ : true class  
           labels  $c_n \in \{1, \dots, C\}$ ;  $\mathbf{x} = (j_1, \dots, j_I)$ : test pattern;  $j_{ni}, j_i \in \{1, \dots, J\}$ .  
   Result:  $\{Y_r\}_{r=1}^6$ : predicted class for  $\mathbf{x}$  and  $r$ -th classifier.  
2 for  $k \leftarrow 1, C$  do  
3    $N_k \leftarrow |\{n = 1, \dots, N : c_n = k\}|$   
   for  $i \leftarrow 1, I$  do  
4     for  $j \leftarrow 1, J$  do  
5        $\mathcal{D}_{ijk} \leftarrow \{n = 1, \dots, N : c_n = k, j_{ni} = j\}$   
        $N_{ijk} \leftarrow |\mathcal{D}_{ijk}|$ ;  $P_{ijk} \leftarrow \frac{N_{ijk}}{N_k}$   
6     end  
7      $P_{ik}(\mathbf{x}) \leftarrow P_{ijik}$  //  $P_{ik}(\mathbf{x}) = [P_{ijk}]_{j=j_i}$   
      $y_i(\mathbf{x}) \leftarrow \arg \max_{k=1, \dots, C} \{P_{ik}(\mathbf{x})\}$  // Prediction of feature  $i$   
8   end  
9    $\mathcal{A}_k(\mathbf{x}) \leftarrow \{i = 1, \dots, I : y_i(\mathbf{x}) = k\}$   
10 end  
11 for  $i \leftarrow 1, I$  do // Calculation of feature relevances  
12   for  $n \leftarrow 1, N$  do //  $j_{ni}$  is the  $i$ -th feature of  $\mathbf{x}_n$   
13     for  $k \leftarrow 1, C$  do  
14        $P_{ik}(\mathbf{x}_n) \leftarrow P_{ij_{ni}k}$  //  $P_{ij_{ni}k} = [P_{ijk}]_{j=j_{ni}}$   
15     end  
16      $y_i(\mathbf{x}_n) \leftarrow \arg \max_{k=1, \dots, C} \{P_{ik}(\mathbf{x}_n)\}$   
17   end  
18    $R_i \leftarrow \frac{1}{N} \sum_{n=1}^N \delta(c_n, y_i(\mathbf{x}_n))$  // Relevance of feature  $i$   
19 end
```

E Performance metrics for binary classification

Eqs. 1-10 report the metrics that we used for classification into classes ASD (negative class) and ASD/ADHD (positive class), in terms of the numbers of true negatives (TN), false positives (FP), false negatives (FN) and true positives (TP).

Recall (Re), also known as true positive rate (TPR) and sensitivity, is the percentage of TP with respect to the number of positives according to the true labeling (sum of FN and TP, eq. 1).

$$Re = TPR = Se = \frac{100 \cdot TP}{FN + TP} \quad (1)$$

Algorithm 2: Continuation of algorithm 1.

```

1  $Y_1(\mathbf{x}) \leftarrow \arg \max_{k=1,\dots,C} \left\{ \sum_{i=1}^I \delta(y_i(\mathbf{x}), k) \right\}$  ; // FPV

2 In case of tie:  $Y_1(\mathbf{x}) = \arg \max_{k=1,\dots,C} \left\{ \sum_{i \in \mathcal{A}_k(\mathbf{x})} R_i \right\}$ 


---


3  $Y_2(\mathbf{x}) \leftarrow \arg \max_{k=1,\dots,C} \left\{ \sum_{i \in \mathcal{A}_k(\mathbf{x})} P_{ik}(\mathbf{x}) \right\}$  ; // FPS


---


4  $Y_3(\mathbf{x}) \leftarrow \arg \max_{k=1,\dots,C} \left\{ \max_{i \in \mathcal{A}_k(\mathbf{x})} \{R_i\} \right\}$  ; // MRF


---


5  $\{l_i\}_{i=1}^I \leftarrow \text{argsort}(\{R_i\}_{i=1}^I, \text{descending})$ ;  $\mathcal{B} \leftarrow \{l_i\}_{i=1}^5$ 
    $Y_4(\mathbf{x}) \leftarrow \arg \max_{k=1,\dots,C} \left\{ \max_{i \in \mathcal{B}} \{\delta(k, y_i(\mathbf{x}))\} \right\}$  ; // MRF5


---


6  $Y_5(\mathbf{x}) \leftarrow \arg \max_{k=1,\dots,C} \left\{ \sum_{i \in \mathcal{A}_k(\mathbf{x})} R_i P_{ik}(\mathbf{x}) \right\}$  ; // FRP


---


7 for  $i \leftarrow 1, I$  do
8   for  $k \leftarrow 1, C$  do
9      $d_{ki}(\mathbf{x}) \leftarrow \min_{n \in \mathcal{D}_{ij_k}} \{d_n(\mathbf{x})\}$  //  $d_n$  defined in section 2.3
10  end
11   $y_i(\mathbf{x}) = \arg \min_{k=1,\dots,C} \{d_{ki}(\mathbf{x})\}$ 
12 end
13  $Y_6(\mathbf{x}) = \arg \max_{k=1,\dots,C} \left\{ \sum_{i=1}^I \delta(k, y_i(\mathbf{x})) \right\}$  ; // FNN

```

Precision (Prec), a.k.a. predictive positivity value (PPV), is the percentage of TP with respect to the total number of positives according to the prediction (sum of FP and TP, eq. 2).

$$Prec = PPV = \frac{100 \cdot TP}{FP + TP} \quad (2)$$

Specificity (Sp) is the percentage of TN with respect to number of negatives according to the true labeling (sum of TN and FP, eq. 3).

$$Sp = \frac{100 \cdot TN}{TN + FP} \quad (3)$$

Balanced accuracy (Bal.Acc) is the average of sensitivity and specificity (eq 4).

Table 3 Feature selection methods, gruped by families, combined with the best classifiers.

Family	Method	Description, language, function and package
Octave		
Univariate	rel	Feature relevance according to eq. 6
	fss	Feature subset selection, fss , package nan
Matlab, Statistics and Machine Learning Toolbox		
Univariate	chi2	Chi-square test feature ranking, fsschi2
	mrmr	Minimum redundancy maximum relevance, fscmr [2]
	relief	ReliefF, relieff [3]
Sequential	fsfs	Forward sequential feature selection, sequentialfs
	bsfs	Backward sequential feature selection, sequentialfs
Model based	fsbag	Bagging out-of-bag ranking, oobPermuted- PredictorImportance
	fstree	Ctree feature ranking, predictorImportance
Python, module sklearn.feature_selection		
Low variance feat. removal	vt	Variance thresholding, VarianceThreshold
Univariate	skbc2	Select K best chi-square score, SelectKBest
	skbfc	SelectKBest with f_classif
	skbmi	SelectKBest with mutual_info_classif
	sp	Select scoring percentile, SelectPercentile
	sfprc2	Select false positive rate (SelectFpr) with chi-square score
	sfprfc	SelectFpr with f_classif
	sfdrc2	Select false discovery rate (SelectFdr), chi-square score
	sfdrfc	SelectFdr with f_classif
	sfwec2	Select family wise error (SelectFwe) with chi-square
	sfwefc	SelectFwe with f_classif
Recursive	rfe	Recursive feature elimination, RFE
Model based	sfm	Select from model, SelectFromModel

$$Bal.Acc = \frac{Se + Sp}{2} \quad (4)$$

False positive rate (FPR) is the percentage of FP with respect to the number of negatives according to the true labeling (sum of TN and FP, eq. 5).

$$FPR = \frac{100 \cdot FP}{TN + FP} \quad (5)$$

False negative rate (FNR) is the percentage of FN with respect to the number of positives according to the true labeling (sum of FN and TP, eq. 6).

$$FNR = \frac{100 \cdot FN}{FN + TP} \quad (6)$$

The F-score is a combination of precision and recall weighted by a parameter β : a value $\beta > 1$ weights more recall, while $\beta < 1$ weights more precision.

$$F\text{-score} = (1 + \beta^2) \frac{Prec \cdot Re}{\beta^2 Prec + Re} = \frac{(1 + \beta^2) TP}{(1 + \beta^2) TP + \beta^2 FN + FP} \quad (7)$$

When $\beta=1$, both are equally weighted, and the F-score is named as F1 (eq. 8):

$$F1 = \frac{2 \cdot TP}{2TP + FN + FP} \quad (8)$$

The Youden index Y is defined as the sum of recall and specificity (not as percentages) minus one, and it can be calculated from the confusion matrix elements as in eq. 9:

$$Y = Re + Sp - 1 = \frac{TP}{FN + FP} + \frac{TN}{TN + FP} - 1 \quad (9)$$

Finally, the Matthews correlation coefficient (MCC), a.k.a. Yule ϕ , measures the correlation of the binary classifications given by the true and predicted class labelings (eq. 10):

$$MCC = \phi = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (10)$$

F Feature ranking

We calculated the feature relevance R_i defined by eq. 8 as follows. In the n -th LOO trial, with $n = 1, \dots, N$, the best classifier (fps) was trained using the whole dataset excepting the n -th pattern, and tested on the n -th pattern using the i most relevant features, varying $i = 1, \dots, I$. The process was repeated for the N trials. Figure 1 plots the kappa against the number of features used by fps and dataset **avoiding**. While using only the most relevant feature kappa is about 25% (left end of the plot), adding the following features reduces the performance, that only recovers 30% using the 5 most relevant features. Finally, all the 20 features are required to achieve the highest kappa (60.5%, right end of the plot). Therefore, to use only the most relevant feature does not guarantee a good performance. In fact, the proposed classifiers that are based on feature relevance (mrf and mrf5) achieve kappa of 43.9% and 37.9%, respectively, in dataset **sensitivity**. This is much lower than fps using all the features on dataset **avoiding** (60.5%).

Table 4 reports the ranking of features in dataset **avoiding** that achieves the best results, according to fpv and to other six well-known methods in the literature: feature subset selection (FSS) based on distance discriminant (FSDD), Pearson correlation and Rank correlation [4], implemented in Octave by package **nan**; random forest importance in Python and R, using packages **scikit-learn** and **randomForest**, respectively;

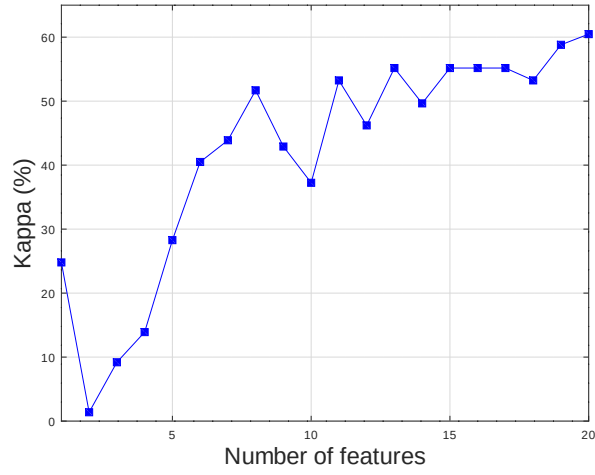


Fig. 1 Kappa (in %) achieved by classifier fps using the i most relevant features, according to eq. 8, for $i = 1, \dots, I$ for dataset **avoiding**, where $I=20$.

Table 4 Position achieved by each feature in dataset **avoiding** using: relevance with fps (eq. 8); importance estimated using FSS-DD, FSS-Pearson, FSS-Rank, random forest in Python and R, and 1D-sensitivity analysis with radial basis function kernel SVM classification. The last column reports the median position of each feature over the 7 methods.

Feature	FPV	Feature subset selection			Random forest		1DSA	Median
		DD	Pearson	Rank	Python	R	KSVM	
SP66	1	1	1	19	2	1	1	1
SP68	2	2	5	6	5	2	4	4
SP65	7	5	14	7	4	3	5	5
SP5	6	4	11	2	1	11	8	6
SP64	11	6	15	16	3	6	3	6
SP71	5	3	16	13	6	7	11	7
SP75	20	7	6	4	11	8	2	7
SP70	8	13	9	1	7	5	9	8
SP2	9	10	12	9	12	4	20	10
SP81	16	20	2	3	16	10	10	10
SP61	15	19	3	11	10	18	6	11
SP15	3	16	4	12	9	19	19	12
SP58	10	8	10	15	17	12	15	12
SP72	12	15	13	17	13	14	12	13
SP59	13	12	17	14	8	15	14	14
SP67	4	11	20	18	15	9	18	15
SP18	19	18	7	20	14	16	7	16
SP74	17	17	8	5	20	13	16	16
SP1	18	14	19	8	18	17	13	17
SP63	14	9	18	10	19	20	17	17

and 1-dimensional sensitivity analysis (1D-SA), implemented in R by package **Rminer**

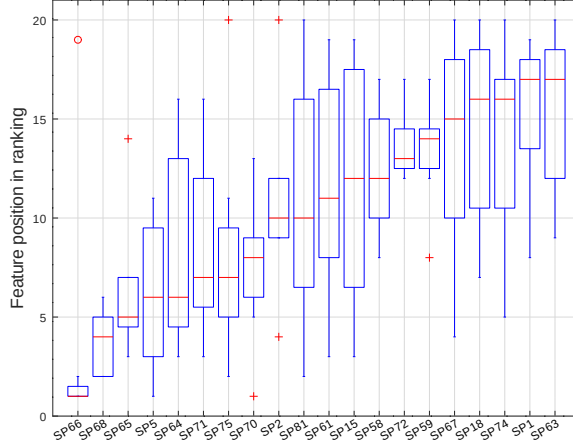


Fig. 2 Box plot with positions of features in dataset **avoiding** over the different methods in Table 4, sorted by increasing median position.

[5]. This method uses a SVM classifier (KSVM) with radial basis function kernel provided by package `kernlab` [6]. The features in Table 4 are sorted by increasing median position over the 7 methods, reported in the last column. Rankings associated to these methods share some similarities. The feature SP66 is the first in 5 of 7 methods, being the 2nd in other method (Random Forest in Python), so its median position is 1. This means that SP66 is clearly the most important feature independently on the method used to evaluate its relevance. In fact, SP66 is the most relevant feature in 49 of $N=52$ (94.2%) trials of the LOO procedure (not showed in Table 4) that calculates relevance R_i , with an average position of 1.1 over the N trials. Feature SP68 achieves the 2nd position in three methods, 4th position in 1 method, 5th position in 2 methods and 6th position in other method, being in median the 4th best feature. As well, SP68 is the second in 37 LOO trials (71.1%). The following features are SP65 and SP5, with lower positions (5 and 6, respectively).

Figure 2 shows the box plots with the feature positions over the seven methods, also sorted by increasing median. The box of feature SP66 is much below the remaining boxes, and its median (red line) is in the lower end of the box, so SP66 achieve the first position in the vast majority of the methods. Feature SP68 is almost always within the first five positions, and SP65 achieves a median fifth position. Features SP5, SP64, SP71, SP75 and SP70 have median positions between 5th and 10th positions, but the remaining features are above 10th position.

Figure 3 plots the box plots with the kappa values achieved by the 7 selected classifiers without and with the feature selection method that provided the best performance for each classifier (e.g. the box plot labeled “fps” is without feature selection, and the box plot labeled “fps-bsfs” is with bsfs feature selection method). Each box plot includes the kappa achieved over the 6 datasets. Boxes are sorted by decreasing median of the kappa, for each classifier, over the 6 datasets. We can see that use of

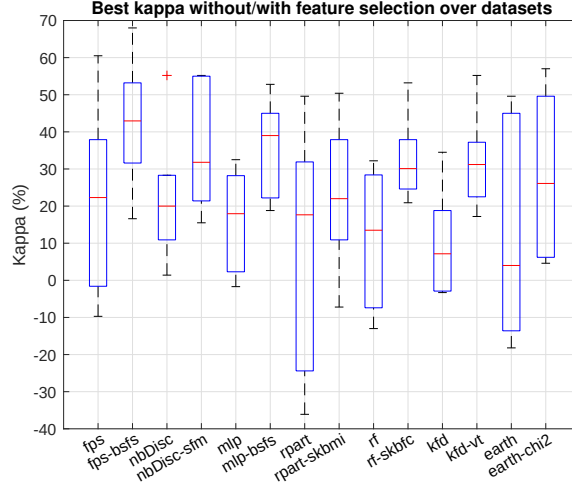


Fig. 3 Box plot of the kappa (in %) achieved by the 7 selected classifiers without feature selection and with the best feature selection method.

Table 5 Classifiers and sampling methods (used with the 7 selected classifiers) for unbalanced classification.

Method	Description and details
Octave, package Multi-Imbalance [7]	
adaboostM1	Adaptive boosting M1
samme	Stagewise additive modeling multi-class exponential loss
adaboostNC	Adaboost with negative correlations
Matlab, Statistics and Machine Learning Toolbox	
rusboost	Random undersampling boost <code>fitcensemble(RUSBoost)</code>
fps-smote	fps and synthetic minority oversampling technique, SMOTE [8]
Python, package imbalanced-learn	
fps-smote	fps and SMOTE
fps-smote-enn	fps and SMOTE with edited-nearest neighbors
fps-svmsmote	fps and SMOTE selecting samples with SVM
wsvm	SVM with class weights
R, package ROSE : fps combined with:	
fps-os	oversampling, <code>ovun.sample(over)</code>
fps-us	undersampling, <code>ovun.sample(under)</code>
fps-ous	over/undersampling, <code>ovun.sample(both)</code>
fps-rose	random over sampling examples, ROSE
fps-smote	SMOTE, <code>smotefamily</code>
fps-adasyn	adaptive synthetic sampling, ADASYN
fps-blsmote	borderline SMOTE, BLSMOTE
fps-dbsmote	density based SMOTE, DBSMOTE
fps-sls	safe-level SMOTE, SLS
fps-rsls	relocation safe-level SMOTE, RSLs

feature selection raises the box plots for all the classifiers. The highest box, with the globally highest kappa and the highest median corresponds to fps. The median values of the other classifier are below 40%. Note also the relatively high maximum values

achieved by nbDiscrete and earth with feature selection, nbDisc-sfm and earth-chi2, respectively.

G Unbalanced classification methods

Table 5 reports the list of 19 unbalanced classifiers and sampling methods combined to the selected best classifiers in the experimental work.

H Detailed results of the experimental work

Table 6 reports the whole results for all the standard and proposed classifiers and all the datasets. In some classifiers several implementations were used, in Matlab (M), Octave (O), Python (P) and/or R.

Table 6: Value of kappa (in %) of classifiers for each dataset.

Classifier	Avoiding	Regist.	Seeking	Sensit.	Total	Touch
aboost-M	0.0	10.1	-7.4	-7.2	0.0	21.8
aboost-R	6.4	18.1	-5.0	0.4	-10.3	0.4
awnb	10.9	-4.7	-12.6	22.2	10.9	-27.1
bag-M	31.9	-4.2	-1.7	3.3	3.3	-7.3
bag-R	31.9	12.9	16.5	14.1	20.9	0.4
ctree-O	18.8	25.0	25.2	18.1	-7.5	-40.3
ctree-M	-7.2	-7.2	-19.3	14.1	-16.5	-3.7
ctree-P	22.2	9.2	3.9	15.8	-23.6	21.8
ctree-R	-1.0	-3.7	0.0	18.1	-10.3	0.0
dlda	31.3	18.8	15.0	28.3	18.8	-7.5
elm-O	-12.6	-29.0	-19.5	28.4	-17.3	10.9
elm-R	9.2	13.0	-5.5	32.2	7.8	21.8
earth	45.0	-18.2	-13.6	49.6	0.0	8.0
fda	-5.0	7.8	12.0	12.0	0.0	21.8
fnn	54.8	-1.0	28.3	-17.3	20.9	-4.2
fps	60.5	25.3	-9.7	19.3	37.9	-1.6
fpv	57.0	28.3	-17.1	7.8	34.5	-2.0
frp	55.2	28.3	-3.3	18.8	37.9	-1.6
gbm-P	-9.9	-7.4	-15.0	12.0	-10.1	10.9
gbm-R	7.9	14.1	-0.4	14.1	1.3	-5.0
gpr	-1.0	0.0	0.0	6.2	0.0	0.0
kfd	34.5	13.9	-3.3	18.8	0.4	-2.9
knn-M	-1.7	24.6	-13.0	-10.5	2.5	24.6
knn-R	-16.5	28.4	6.2	0.0	2.5	-1.0
lda-P	-5.0	7.8	5.0	12.0	-9.9	25.0
lda-R	-5.0	7.8	5.0	12.0	-9.9	25.0
lvq	15.8	2.3	-10.5	1.3	7.9	-4.2
manb	12.0	-24.4	17.4	31.9	12.0	2.5

mda	-2.4	-5.5	-2.4	7.9	0.0	-0.4
mlp-M	12.0	-5.3	-2.9	21.8	14.1	9.6
mlp-P	19.3	16.6	32.5	-1.7	28.2	2.3
mrf	31.9	41.4	6.4	43.9	-9.9	-46.1
mrf5	28.3	1.3	32.2	37.9	24.8	15.8
nn	31.6	8.0	47.4	12.9	9.6	0.0
nb	41.4	21.4	-4.7	21.8	18.1	-4.7
nbDisc	55.2	10.9	15.0	25.0	28.3	1.4
nbSearch	21.4	-15.7	-19.3	-15.0	-36.1	-16.5
nnc	34.1	11.7	22.2	-19.6	13.0	-18.9
neuralnet-R	19.8	9.2	-5.0	9.2	22.2	9.2
nnet-R	6.5	31.3	2.9	25.0	22.2	7.9
pnn	-13.0	-10.3	-26.7	-1.0	-7.2	-2.4
qda	1.3	-1.0	-1.0	-10.1	-7.3	-1.0
rf-M	32.2	-7.4	17.4	28.4	9.6	-13.0
rf-P	12.9	-13.0	-1.0	7.9	1.3	-16.5
rf-R	28.4	-4.2	16.5	4.6	-13.0	-13.6
rpart	31.9	-36.1	-24.4	49.6	18.8	16.5
svm-M	-13.0	8.0	-7.3	24.8	-7.3	16.5
svm-O	3.3	18.1	9.6	9.6	-13.6	17.4
svm-P	3.3	21.4	9.6	9.6	-13.6	14.1
svm-R	-7.4	-4.2	24.3	-1.7	-16.5	-10.3
Best kappa	60.5	41.4	47.4	49.6	37.9	25.0
Classifier	fps	mrf	nn	earth	fps	lda1

Table 7 reports the results achieved by each combination of classifier and feature selection method for each of the six datasets. Both the Matlab (M) and Python (P) implementations of bsfs were used.

Table 7: Value of kappa (in %) of classifiers with feature selection for each dataset.

Classif.-Feat.Sel.	Avoiding	Regist.	Seeking	Sensit.	Total	Touch
fps-rel	60.5	41.4	36.2	43.9	41.4	28.3
fps-fss	66.5	34.5	15.0	49.6	37.9	22.8
fps-bsfs-M	68.0	43.9	16.6	31.6	53.2	42.0
fps-fsfs-M	62.6	47.4	32.5	43.9	41.4	25.6
fps-bagging	60.5	34.5	12.3	28.3	37.9	20.2
fps-chi2	60.5	57.0	17.4	43.9	41.4	33.4
fps-mrmr	60.5	34.5	13.0	49.6	41.4	7.7
fps-ctree	60.5	37.9	11.4	49.6	43.9	22.8
fps-relieff	60.5	28.3	15.8	22.2	48.8	23.3
fps-bsfs-P	64.1	43.9	18.8	46.2	47.4	30.8
fps-vt	60.5	25.3	0.0	19.3	37.9	0.0
fps-skbc2	53.6	40.5	13.0	34.1	45.0	31.1

fps-skbfc	53.6	37.9	10.3	49.6	46.2	28.2
fps-skbmi	57.0	34.5	8.9	42.9	47.4	22.5
fps-sp	60.5	40.5	13.0	34.1	43.9	31.1
fps-sfprc2	49.6	33.8	13.0	25.3	34.5	23.3
fps-sfprfc	46.2	10.9	7.8	15.8	25.2	7.7
fps-sfdrc2	45.0	16.0	10.3	25.3	45.0	10.9
fps-sfdrfc	37.9	30.8	-1.0	46.2	28.3	18.3
fps-sfwec2	45.0	6.6	9.2	15.0	45.0	13.1
fps-sfwefc	45.0	7.6	6.5	49.6	28.3	18.3
fps-rfe	62.6	31.9	38.6	43.5	45.0	22.2
fps-sfm	55.2	41.4	36.2	43.9	35.0	22.2
nbDisc-chi2	48.8	28.4	32.5	28.4	35.0	-10.5
nbDisc-mrmr	37.9	17.4	32.2	37.9	43.5	9.6
bnDisc-relieff	37.9	20.9	21.8	43.5	35.0	3.3
nbDisc-fsfs	46.3	7.9	9.6	47.8	38.6	7.9
nbDisc-bsfs	34.5	47.8	14.1	39.4	43.5	7.9
nbDisc-fsbag	46.2	25.0	31.3	31.6	28.4	-0.4
nbDisc-fstree	42.3	28.4	31.6	35.0	52.8	21.4
nDisc-vt	31.9	24.8	24.6	35.5	28.4	18.1
nbDisc-skb2	41.4	8.0	24.8	35.0	31.3	-0.4
nbDisc-skbfc	47.4	16.5	28.3	41.4	37.9	5.0
nbDisc-skbmi	46.2	24.3	10.9	24.8	31.9	5.0
nbDisc-sp	41.4	8.0	24.8	35.0	31.3	-0.4
nbDisc-sfprc2	41.4	8.0	24.8	35.0	31.3	-0.4
nbDisc-sfdrc2	41.4	8.0	24.8	35.0	31.3	-0.4
nbDisc-sfdrfc	47.4	16.5	28.3	41.4	37.9	5.0
nbDisc-sfwec2	41.4	8.0	24.8	35.0	31.3	-0.4
nbDisc-sfwefc	47.4	16.5	28.3	41.4	37.9	5.0
nbDisc-rfe	40.5	21.4	38.6	32.2	38.6	9.2
nbDisc-sfm	55.2	55.0	25.0	21.4	38.6	15.5
earth-rel	21.4	-1.0	2.5	49.6	47.4	11.7
earth-fss	17.4	6.2	15.5	49.6	35.0	11.7
earth-chi2	57.0	4.6	6.2	49.6	31.9	20.3
earth-mrmr	37.9	-13.0	6.2	49.6	42.3	20.3
earth-relieff	42.3	-7.2	16.5	45.0	36.2	20.3
earth-fsfs	14.1	16.5	2.5	37.9	21.4	11.7
earth-bsfs	39.4	8.0	19.6	42.3	31.6	8.0
earth-fsbag	20.9	10.1	4.6	42.3	37.9	2.5
earth-fstree	28.4	28.5	6.2	49.6	22.2	15.5
earth-vt	25.0	20.3	11.7	35.5	40.5	8.0
earth-skb2	35.5	-7.3	11.7	31.6	31.9	11.7
earth-skbfc	21.4	6.2	6.4	49.6	45.0	11.7
earth-skbmi	35.5	4.6	11.7	41.4	21.4	15.5
earth-sp	35.5	-7.3	11.7	31.6	31.9	11.7
earth-sfprc2	35.5	-7.3	11.7	31.6	31.9	11.7

earth-sfdrc2	35.5	-7.3	11.7	31.6	31.9	11.7
earth-sfdrfc	21.4	6.2	6.4	49.6	45.0	11.7
earth-sfwec2	35.5	-7.3	11.7	31.6	31.9	11.7
earth-sfwefc	21.4	6.2	6.4	49.6	45.0	11.7
earth-rfe	24.6	-1.0	19.6	34.5	28.4	15.5
earth-sfm	38.6	2.5	6.2	48.8	38.6	6.2
rpart-rel	35.5	39.4	-7.2	49.6	45.0	14.1
rpart-fss	17.4	14.1	-3.7	49.6	31.6	16.5
rpart-chi2	50.4	21.4	-7.2	49.6	25.0	16.5
rpart-mrmr	46.2	2.3	-3.7	49.6	31.9	24.3
rpart-relieff	48.8	25.0	-10.5	37.9	39.4	24.6
rpart-fsfs	17.4	13.0	-7.2	41.4	28.4	16.5
rpart-bsfs	45.0	31.9	-10.5	38.6	31.6	9.6
rpart-fsbagging	24.8	25.3	-3.7	38.6	22.2	6.2
rpart-fstree	15.0	16.5	-13.0	49.6	35.0	24.6
rpart-vt	41.4	32.2	-7.2	28.4	21.8	20.9
rpart-skbc2	31.6	21.8	-10.5	31.6	28.4	16.5
rpart-skbfc	28.4	7.9	-10.5	49.6	41.4	16.5
rpart-skbmi	50.4	18.8	-7.2	37.9	25.2	10.9
rpart-sp	31.6	21.8	-10.5	31.6	28.4	16.5
rpart-sfprc2	31.6	21.8	-10.5	31.6	28.4	16.5
rpart-sfdrc2	31.6	21.8	-10.5	31.6	28.4	16.5
rpart-sfdrfc	28.4	7.9	-10.5	49.6	41.4	16.5
rpart-sfwec2	31.6	21.8	-10.5	31.6	28.4	16.5
rpart-sfwefc	28.4	7.9	-10.5	49.6	41.4	16.5
rpart-rfe	31.9	28.4	-3.7	37.9	18.1	28.4
rpart-sfm	35.5	35.5	0.0	45.0	25.0	20.3
kfd-rel	34.5	25.3	13.9	42.0	34.1	19.3
kfd-fss	53.2	28.2	6.5	35.4	31.6	13.0
kfd-chi2	47.4	17.2	19.8	31.3	31.1	13.0
kfd-mrmr	37.9	22.5	12.3	42.9	28.3	18.8
kfd-relieff	48.3	25.2	13.9	31.3	40.5	16.6
kfd-fsfs	53.2	23.1	10.9	25.3	31.6	31.3
kfd-bsfs	49.6	22.5	16.6	22.5	37.9	15.8
kfd-fsbagging	37.2	19.3	12.0	37.9	41.4	13.9
kfd-fstree	37.9	31.1	15.0	37.9	31.6	7.8
kfd-vt	55.2	30.8	17.2	31.6	37.2	22.5
kfd-skbc2	40.5	39.1	28.3	34.1	31.6	25.2
kfd-skbfc	40.5	39.1	28.3	42.9	34.1	25.2
kfd-skbmi	47.4	28.2	25.3	42.9	28.3	15.8
kfd-sp	40.5	39.1	28.3	34.1	31.6	25.2
kfd-sfprc2	40.5	39.1	28.3	34.1	31.6	25.2
kfd-sfdrc2	40.5	39.1	28.3	34.1	31.6	25.2
kfd-sfdrfc	40.5	39.1	28.3	42.9	34.1	25.2
kfd-sfwec2	40.5	39.1	28.3	34.1	31.6	25.2

kfd-sfwefc	40.5	39.1	28.3	42.9	34.1	25.2
kfd-rfe	37.9	22.5	1.4	31.1	31.9	19.8
kfd-sfm	40.5	30.8	0.0	31.3	34.5	19.8
mlp-rel	28.3	32.2	25.0	43.5	32.2	38.6
mlp-fss	31.9	31.3	24.6	32.2	47.4	24.8
mlp-chi2	24.6	28.4	20.3	28.4	36.2	20.9
mlp-mrmr	35.0	25.0	16.5	31.9	41.4	17.4
mlp-relieff	31.9	35.5	9.6	35.0	43.9	35.0
mlp-fsfs	41.4	35.5	12.9	35.5	42.3	10.9
mlp-bsfs	45.0	22.2	43.5	52.8	34.5	18.8
mlp-fsboosting	31.3	25.2	18.1	34.5	48.3	31.3
mlp-fstree	25.0	28.4	24.3	40.5	38.6	21.4
mlp-vt	38.6	25.0	28.4	43.5	45.0	38.6
mlp-skbc2	37.9	32.2	5.0	28.4	48.8	28.4
mlp-skbcfc	37.2	28.4	21.8	21.4	51.7	24.8
mlp-skbbmi	42.3	31.6	18.1	47.4	43.9	17.4
mlp-sp	37.9	32.2	5.0	28.4	48.8	28.4
mlp-sfprc2	37.9	32.2	5.0	28.4	48.8	28.4
mlp-sfdrc2	37.9	32.2	5.0	28.4	48.8	28.4
mlp-sfdrfc	37.2	28.4	21.8	21.4	51.7	24.8
mlp-sfwec2	37.9	32.2	5.0	28.4	48.8	28.4
mlp-sfwefc	37.2	28.4	21.8	21.4	51.7	24.8
mlp-rfe	24.6	33.8	35.5	28.4	39.4	35.0
mlp-sfm	31.3	35.0	17.4	51.1	51.1	35.0
rf-rel	39.4	20.9	28.4	47.4	32.2	12.9
rf-fss	38.6	18.8	36.2	42.3	32.2	8.0
rf-chi2	28.4	21.4	20.9	41.4	35.0	17.4
rf-mrmr	32.2	24.8	28.4	47.4	31.9	9.6
rf-relieff	38.6	18.1	32.2	24.8	28.4	24.6
rf-fsfs	31.9	28.4	18.1	16.5	25.0	9.6
rf-bsfs	47.8	24.6	28.4	28.4	32.2	20.9
rf-fsboosting	39.4	16.5	16.5	35.5	42.3	3.3
rf-fstree	21.8	21.4	12.9	47.4	31.9	3.3
rf-vt	39.4	35.5	35.5	38.6	48.8	9.6
rf-skbc2	31.9	21.8	28.4	37.9	31.9	21.4
rf-skbcfc	35.0	25.2	20.9	53.2	37.9	24.6
rf-skbbmi	38.6	24.8	24.6	31.6	21.8	10.9
rf-sp	31.9	21.8	28.4	37.9	31.9	21.4
rf-sfprc2	31.9	21.8	28.4	37.9	31.9	21.4
rf-sfdrc2	31.9	21.8	28.4	37.9	31.9	21.4
rf-sfdrfc	35.0	25.2	20.9	53.2	37.9	24.6
rf-sfwec2	31.9	21.8	28.4	37.9	31.9	21.4
rf-sfwefc	35.0	25.2	20.9	53.2	37.9	24.6
rf-rfe	21.4	35.0	20.9	42.3	38.6	14.1
rf-sfm	35.0	21.4	28.4	14.1	35.0	24.8

Best kappa	68.0	57.0	43.5	53.2	53.2	42.0
Classif+FS	fps-bsfs-M	fps-chi2	mlp-bsfs	rf-skbfc	fps-bsfs-M	fps-bsfs-M

Table 8 reports the results achieved by each unbalanced classifier or undersampling/oversampling method for each dataset. The SMOTE method was used in its Matlab (M), Python (R) and R implementations.

Table 8: Value of kappa (in %) achieved by each unbalanced classification method for each dataset.

Classifier	Avoiding	Regist.	Seeking	Sensit.	Total	Touch
adaboostM1	14.1	3.3	-7.3	-9.9	-15.0	28.3
samme	14.1	3.3	-7.3	-9.9	-15.0	28.3
adac2M1	21.4	3.3	-13.0	-9.9	-15.0	28.3
adaboostNC	14.1	3.3	-7.3	-9.9	-15.0	28.3
rusboost	17.2	6.6	4.6	6.5	-3.7	12.3
fps-smote-M	16.5	0.0	-7.2	6.2	0.0	-3.7
fps-smote-P	47.2	25.3	-19.6	5.3	17.8	-5.9
fps-smote-enm	32.9	-4.3	-3.4	7.7	0.2	21.6
fps-svmsmote	45.1	19.3	-7.6	7.8	37.2	-2.0
fps-os	46.2	22.2	5.3	25.2	34.5	12.3
fps-us	47.2	17.8	7.7	7.7	39.1	-4.0
fps-ous	60.5	31.3	7.8	19.3	31.6	13.1
fps-rose	42.9	4.3	-1.0	17.8	18.3	7.8
fps-smote	28.4	10.1	20.3	11.7	10.1	-1.0
fps-adasyn	12.9	0.0	2.5	10.1	0.0	-3.7
fps-blsmote	20.9	0.0	-3.7	2.5	6.2	6.2
fps-dbsmote	16.5	10.1	11.7	11.7	0.0	-3.7
fps-rsls	35.5	10.1	6.4	28.4	8.0	15.5
fps-sls	35.5	10.1	6.4	28.4	8.0	8.0
rf-rose	13.9	5.3	-46.0	28.2	5.0	0.0
rf-smote	21.4	6.4	3.3	9.6	-10.1	14.1
nbDisc-smote-M	28.2	12.3	-5.3	21.8	6.6	-15.9
nbDisc-smote-P	39.7	5.0	-11.9	16.6	3.9	19.8
nbDisc-smtenn	42.0	-7.7	-11.4	6.6	6.6	14.7
nbDisc-svmsmote	39.7	18.8	-14.0	13.9	25.5	-7.6
nbDisc-ous	25.0	3.3	3.3	6.5	6.4	1.4
nbDisc-os	33.8	4.3	-0.6	2.1	30.9	-7.6
nbDisc-us	36.2	1.3	-2.4	6.4	6.4	-1.0
nbDisc-rose	33.8	1.4	5.7	-7.7	25.6	-7.7
nbDisc-adasyn	39.1	19.8	-18.9	16.6	31.1	8.9
nbDisc-smote-R	28.2	25.0	-3.3	25.3	25.3	8.9
nbDisc-blsmote	39.7	28.3	-1.0	22.2	10.3	4.3

nbDisc-dbsmote	46.2	5.0	-9.8	25.3	15.8	14.7
nbDisc-sls	40.5	-4.7	6.5	22.2	28.3	10.3
nbDisc-rsls	37.2	18.1	6.5	22.2	19.3	5.3
earth-smote-M	24.8	-12.6	-16.5	49.6	12.0	-4.2
earth-smote-P	5.5	20.6	-11.9	41.4	6.6	13.0
earth-smtenn	42.0	-7.7	-11.4	6.6	6.6	14.7
earth-svmsmote	39.7	18.8	-14.0	13.9	25.5	-7.6
earth-ous	12.3	7.7	-18.2	7.7	22.5	5.5
earth-os	-7.6	18.3	0.4	11.7	10.9	-25.0
earth-us	7.8	-11.4	-11.9	13.1	-2.0	-4.0
earth-rose	25.3	19.3	2.9	9.9	40.6	2.9
earth-adasynt	17.8	7.7	-38.5	23.1	8.9	5.5
earth-smote-R	22.2	33.4	10.3	16.6	13.0	-9.8
earth-blsmote	10.3	8.9	-9.7	13.9	2.9	6.6
earth-dbsmote	34.5	17.8	-27.4	40.5	11.4	0.6
earth-sls	18.8	16.0	16.6	25.3	-7.6	-9.9
earth-rsls	13.0	20.6	13.0	19.3	12.0	-3.3
rpart-smote-M	42.3	-5.3	-12.6	49.6	18.8	9.6
rpart-smote-P	-9.8	8.9	-36.1	31.6	17.8	21.8
rpart-smtenn	18.3	8.5	4.6	-2.8	22.8	11.4
rpart-svmsmote	13.0	1.4	-18.6	2.3	10.3	-2.4
rpart-os	6.6	7.8	-18.6	8.9	-1.0	1.4
rpart-us	7.8	7.7	10.3	19.3	-11.7	3.4
rpart-ous	13.0	6.5	-3.3	8.9	-2.9	-7.6
rpart-rose	28.3	42.9	20.2	14.7	7.7	-5.5
rpart-adasynt	8.9	-5.5	-18.6	8.9	28.3	-14.0
rpart-blsmote	25.3	-14.4	18.1	15.0	25.2	6.5
rpart-smote	21.4	-7.6	25.3	7.9	9.2	17.2
rpart-dbsmote	28.4	-1.0	12.0	18.8	18.1	-2.4
rpart-sls	31.9	-7.5	-4.2	3.3	14.1	3.9
rpart-rsls	21.4	-7.5	-13.6	24.6	17.4	13.0
wsvm	39.7	2.3	-2.9	7.9	21.8	-11.9
kfd-smote-M	6.5	15.8	-7.6	15.8	-7.4	7.8
kfd-smote-P	-4.7	-2.9	-15.0	10.9	-12.6	-20.5
kfd-smtenn	25.8	-3.4	-9.4	-9.7	25.7	4.6
kfd-svmsmote	-17.3	5.0	-17.3	9.6	-4.7	0.6
kfd-os	-4.2	31.9	-10.5	-13.0	0.4	-1.0
kfd-us	23.1	12.3	-4.3	16.6	-5.7	8.7
kfd-ous	-1.0	9.6	-1.0	21.4	-7.3	24.6
kfd-rose	10.3	-9.7	-11.9	21.4	-7.5	12.3
kfd-adasynt	-18.2	28.4	-19.3	7.9	1.3	-7.3
kfd-smote-R	-17.3	1.4	-33.1	-5.0	-10.1	-9.9
kfd-blsmote	-1.7	-4.7	-10.3	-0.4	-20.6	10.9
kfd-dbsmote	-10.3	14.1	-10.3	31.9	28.4	3.3
kfd-sls	-1.7	13.9	-24.4	14.1	-28.9	-13.0

kfd-rsls	-4.7	11.4	-26.7	17.4	-24.4	35.5
mlp-smote-M	12.0	16.6	6.4	0.4	19.3	9.2
mlp-smote-P	7.8	17.8	13.9	-2.9	-5.7	-3.7
mlp-smtenn	19.2	2.0	-11.1	-1.6	8.7	5.5
mlp-svmsmote	14.7	16.6	6.5	6.5	22.5	-16.6
mlp-os	9.2	31.6	-10.1	12.0	5.0	-0.4
mlp-us	2.7	-2.0	-2.0	-1.0	17.8	5.5
mlp-ous	34.5	1.4	3.3	19.3	18.1	16.6
mlp-rose	19.3	19.8	14.7	5.3	25.5	11.4
mlp-adasyn	34.1	34.5	18.1	2.3	15.8	-7.5
mlp-smote-R	13.0	-1.0	-2.4	15.0	-1.6	-25.9
mlp-blsmote	-0.4	25.2	3.3	18.8	19.3	12.0
mlp-dbsmote	19.3	-0.4	15.0	7.8	16.6	9.2
mlp-sls	16.6	12.0	-7.4	10.9	2.3	-5.0
mlp-rsls	10.3	15.8	-7.4	25.0	-0.4	-15.7
rf-smote-M	31.9	12.9	8.0	17.4	14.1	6.4
rf-smote-P	-9.8	17.4	-15.0	7.9	10.9	28.3
rf-smtenn	8.5	-7.8	-6.3	3.4	23.3	15.4
rf-svmsmote	9.2	0.4	-17.3	9.2	14.1	14.1
rf-os	3.9	3.9	7.9	2.3	-4.7	3.9
rf-us	20.6	2.9	-22.1	14.7	19.8	-21.4
rf-ous	15.0	-22.8	-4.7	2.3	7.9	2.3
rf-rose	22.2	25.2	5.3	1.4	-5.5	16.6
rf-adasyn	37.2	-2.9	-21.6	12.0	25.0	6.5
rf-smote-R	15.8	10.9	10.9	2.3	-10.1	3.9
rf-blsmote	25.2	11.4	4.6	18.1	-5.0	15.8
rf-dbsmote	43.9	25.0	-15.0	21.4	25.0	18.1
rf-sls	18.1	-15.7	-1.7	-10.1	18.1	6.5
rf-rsls	18.1	10.3	0.4	18.1	-4.7	18.1
Best kappa	60.5	42.9	20.3	49.6	40.6	35.5
Classifier	fps-ous	rpart- rose	fps- smote	earth- smote1	earth- rose	kfd- rsls

References

- [1] Kuhn, M.: Caret: Classification and Regression Training. (2016). R package. <http://topepo.github.io/caret/train-models-by-tag.html>
- [2] Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. J Bioinf Comput Biol **3**(2), 185–205 (2005) <https://doi.org/10.1142/s0219720005001004>
- [3] Robnik-Sikonja, M., Kononenko, I.: Theoretical and empirical analysis of ReliefF and RReliefF. Mach Learn **53**, 23–69 (2003) <https://doi.org/10.1023/A:1025667309714>

- [4] Peng, H.C., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE T Pattern Anal* **27**(8), 1226–1238 (2005) <https://doi.org/10.1109/TPAMI.2005.159>
- [5] Cortez, P., Embrechts, M.J.: Using sensitivity analysis and visualization techniques to open black box data mining models. *Inf Sci* **225**, 1–17 (2013) <https://doi.org/10.1016/j.ins.2012.10.039>
- [6] Karatzoglou, A., Smola, A., Hornik, K., Zeileis, A.: Kernlab – an S4 package for kernel methods in R. *J Stat Softw* **11**(9), 1–20 (2004) <https://doi.org/10.18637/jss.v011.i09>
- [7] Zhang, C., Bi, J., Xu, S., Ramentol, E., Fan, G., Qiao, B., Fujita, H.: Multi-imbalance: An open-source software for multi-class imbalance learning. *Knowl-based Syst* **174**, 137–143 (2019) <https://doi.org/10.1016/j.knosys.2019.03.001>
- [8] Larsen, B.S.: Synthetic Minority Over-sampling Technique (SMOTE). https://github.com/dkbsl/matlab_smote (2022). <https://doi.org/10.1613/jair.953>