

## **RANDOM FORESTS SUPPLEMENTED BY PARALLEL FENCES PLOTS: A STUDY IN SPIROMETRIC DIAGNOSTICS**

**Anna M. Bartkowiak<sup>1</sup>, Jerzy Liebhart<sup>2</sup>**

<sup>1</sup>Inst. of Computer Science, Wrocław University  
e-mail: aba@cs.unu.wroc.pl

<sup>2</sup>Dept. and Clinic of Internal Diseases, Geriatrics and Allergology  
Medical Academy of Wrocław

### **Summary**

We try to assess, what truly can be obtained when using only the spirometric measurements for the diagnosis of pulmonary disorders. Using spirometric data recorded for 202 patients and the Random Forests methodology, we obtained an algorithm permitting for a 100% correct classification of the learning sample, and for a 94% correct classification of the 125 'not-healthy' patients included into the validation sample.

We propose, as a routine continuation, to verify the results - especially those for misclassified patients - by an independent multivariate graphical procedure called by us the parallel fences plot. By inspecting the patient's data vectors displayed in parallel fences plots it becomes evident if they are typical or not-typical for the 'healthy' or 'not-healthy' state of the patient.

The conclusion is, that the spirogram can be used for a relatively safe diagnosis of majority of patients. Only those that do not pass the parallel fences plot test should undergo further pulmonary scrutinizing test.

**Keywords and phrases:** medical diagnosis, random forests, parallel fences plot

**Classification AMS 2010:** 62-07, 62P10, 62H30

## 1. Introduction

We are concerned with medical diagnosis of possible pulmonary disorders. Spirometry tests are a basic tool commonly used to assess lung function. Their interpretation is usually done in clinical practice by means of arbitrarily chosen parameters and arbitrarily set criteria, like those described in guidelines by Boros et al., 2006. This laborious paper in 23 pages including 56 references, considers lung impairment diagnosis using confidence intervals derived as percentiles of a normal distribution.

Generally, it is believed, that diagnosis made on spirometric measurement alone is unsure. It is not known, what is the percentage of correct (erroneous) diagnoses obtained that way.

This article presents a set of statistical methods that may allow objective differentiation between the in-norm and impaired lung ventilation. It shows also how obtain a rigorous estimate on the percentage of erroneous diagnoses performed on the basis of spirometric measurements.

Our considerations are based on a sample data matrix  $\mathbf{X}_{n \times d}$ , containing  $n$   $d$ -variate data vectors denoting values of  $d$  spirometric variables recorded in  $n$  adult patients. Each patient has its medical diagnosis memorized in a vector  $\mathbf{y}_{n \times 1}$ , taking only values 'yes' (the patients is in norm), and 'no' (the patients is beyond the norm). Our goal is to construct a prediction algorithm permitting to perform a machine diagnosis of the given patient.

The prediction formula may be based on many methods. Bartkowiak and Liebhart (2018) have made preliminary evaluations considering the following methods: Binary decision trees, Random Forests, Neural Networks, Logistic Regression, Linear Discriminant function, Quadratic Discriminant function. The results were not dramatically different.

Here we concentrate on one algorithm: the Random Forest (RF).

This algorithm exploits the principles of ensemble learning (Li, Wu & Ngom, 2018), is conceptually simple, does not need any probabilistic assumptions, is largely robust against outliers, uses simple calculations that prevent from overfitting or ill-conditioning.

## 2. Analyzed data

The data were gathered in the Department and Clinic of Internal Medicine and Allergology, Wrocław Medical University. The analyzed here data matrix  $\mathbf{X}$

is of size  $n \times d = 202 \times 12$ , with rows corresponding to patients, and columns to variables (attributes) characterizing the patients.

The variables, their shorts and their meanings are given in the list below.

Variables no.s 3-12 were computed by the spirometric device on the base of the flow curve(s) of exhalation. Variable  $X_4$ , named here  $VC.$ , is known also as  $VC\%$  or percentage of  $VC_{predicted}$ , was computed from an equation predicting the  $VC$  of a given patient as function of age, height, sex and potentially some other parameters, see guidelines in (Boros et al., 2006). The value of  $X_5 = VC.$  for a given patient is defined as the percentage of the observed  $VC.$  in relation to the  $VC_{predicted}$  for that patient.

$X_1$ ; *Age*: Age;

$X_2$ ; *Height*: Height;

$X_3$ ; *VC.*: Vital Capacity;

$X_4$ ; *VC.*:  $VC\%$ , observed  $VC.$  as percentage of predicted  $VC.$ ;

$X_5$ ; *FEV 1*:  $FEV_1$ , Forced Expiratory Volume in one second;

$X_6$ ; *Tiff*: Ratio  $FEV_1/VC.$   $\times 100$ ;

$X_7$ ; *FEF*:  $FEF_{0.2-1.2}$ , Forced Expiratory Flow at level 0.2–1.2 dm<sup>3</sup>;

$X_8$ ; *MMFR*:  $MMFR$ , Maximal Mid–expiratory Flow Rate;

$X_9$ ; *MMFT*: Maximal Mid–expiratory Flow Time;

$X_{10}$ ; *FR.FT*: Ratio  $MMFR/MMFT$ , calculated as  $X_8/X_9$ ;

$X_{11}$ ; *FEV.VC*: Ratio  $FEF_{0.2-1.2}/VC.$  calculated as  $1000 \times X_7/X_3$ ;

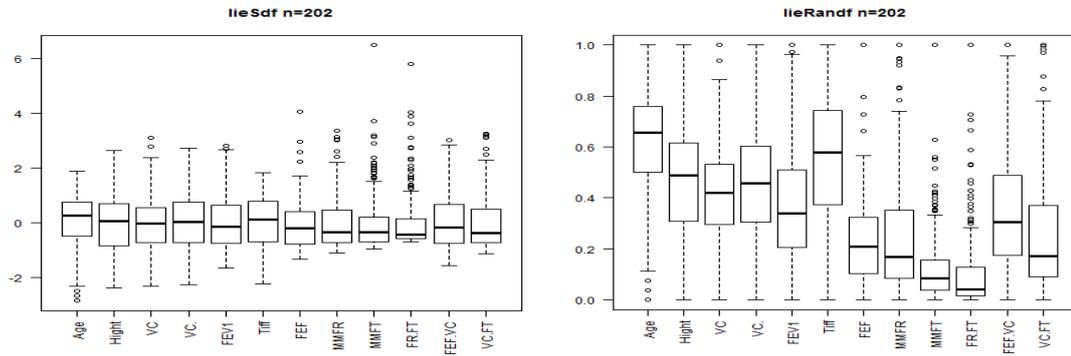
$X_{12}$ ; *VC.FT*: Ratio  $VC./MMFT$ .

The twelve variables indicated above will be hereafter called 'spirometric variables'. Ten of them are strictly spirometric. The first two (*Age* and *Height*) were added, because the essential parameters:  $X_3 = VC.$ , volume of breathed in air, and  $X_4 = VC\%$ , percentage of predicted volume, are considered as depending from both age and height of the patient.

The variables were recorded for  $n = 202$  adult patients, among them 28 pulmonary 'innorm', and 174 pulmonary 'not in norm'. The 12 variables are differentiated to their magnitudes. Before further elaboration presented in this paper, each of the variables was standardized in two ways:

- *statistically (S)*, that is to have mean = 0 and variance equal = 1,
- *to range [0,1] (Ran)*, that is to have values belonging to the interval [0,1].

The boxplots of the subsequent variables are shown in Fig. 1.



**Fig. 1.** Boxplots of 12 variables constituting the spirometric data set. Left: for data standardized (S), that is to have mean =0 and variance = 1. Right: for data standardized (Ran), that is to be contained in the range [0,1]. Tukey's fences are marked for each variable in both plots by short horizontal bars. Data values outside the fences are deemed to be outliers.

Looking at the graphs in Fig. 1 one may state that the asymmetry of the standardized variables is not so dramatic. For sure, the distributions are not normal (Gaussian). Some of the distributions are heavy-tailed (platykurtic). And some of them contain not so few outliers. Thus, the usual statistical assumption on multivariate normality is not met here. This was confirmed by applying the classical Shapiro-Wilk test. It appeared, that only 3 variables had the P-value slightly above 0.05; these were:  $X_2$ ;  $X_3$ ;  $X_4$  with their P-s equal 0.058, 0.09, 0.058 appropriately. The remaining P-s were extremely small indicating for a high deviation from normality.

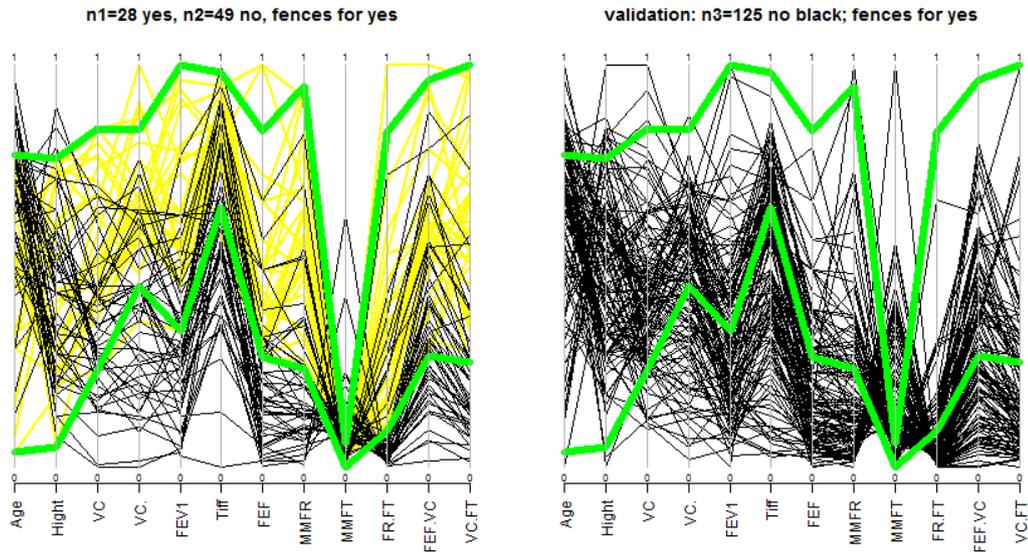
The recorded data were subdivided into two parts: the training and the validation sets. Unfortunately, we had only 28 'healthy' patients. We assigned them all to the training sample, to get firm and stable prediction.

It was accepted that

- *The train sample* contains 77 (=49+28) patients: 49 'no' patients with impaired lung ventilation (28 with obturation and 21 with restriction), and 28 'yes' patients, that is, with their lung function in norm.
- *The validation sample* contains 125 (=28+97) 'no' patients: 28 of them with pulmonary disorders of mixed type, and the remaining 97 ones with obturative disorders.

Each patient is represented by a data vector  $\mathbf{x}$  containing values of the 12 considered variables.

The obtained train and validation data are illustrated in Fig. 2 by a parallel fences plot. Both plots contain thick green line segments meaning fences derived from the training data set for the 'yes' patients. We did not need to create any test sample, because the RF algorithm creates so called OOB (Out-Of Bag) data set which serves as test set (see Section 'Methods').



**Fig. 2.** Parallel Fences plots with fences derived from the 'yes' data set, that is from the 'in-norm' data vectors; appears both in the left and in the right graph. Left: The training data: the 'yes' data vectors  $n1=28$  (yellow), and the  $n2=49$  'no' data vectors (black line segments). Right: The  $n3=125$  'no' data vectors from the validation data (in black).

The upper and lower fences were obtained for each variable  $j$ ;  $j = 1, \dots, 12$  as (Tukey 1977)

$$f_{up} = q3 + \min(\max j, 1.5 \times (q3 - q1)),$$

$$f_{dw} = q1 - \max(\min j, 1.5 \times (q3 - q1)),$$

where  $\max j$  and  $\min j$  are the  $\max$  and  $\min$  of the  $j$ -th variable, and  $q3$  and  $q1$  are its 3rd and 1st quartiles;  $f_{up}$  and  $f_{dw}$  denote the upwards and downwards fences respectively.

Both the left and right graphs in Fig. 2 are displaying, apart from the green fences, also additional data vectors: The left graph shows data vectors of the 28 'yes' patients as yellow horizontal line segments, and 47 'no' patents from the train

data set - as line segments in black. The right graph shows in black 125 'no' data vectors from the validation data set, put in the framework the same green fences as the left graph.

Looking at the left graph one sees that the black lines are partially mixed with the yellow lines. Both in the left and right graphs one may see that a considerable amount of the black line segments are invading the area delimited, by the upper and lower fences of the 'yes' patients. This means difficulties with assigning the category 'yes' or 'no' (meaning 'healthy' or 'not-healthy') for some data vectors looking similar. It is hard to believe that from these data it is possible to build an algorithm which yields about 90% and 94% correct classifications for the train and validation data shown in Fig. 2 above.

### 3. Methods

We have data points that belong to two classes, labeled 'yes' (class 1, healthy) and 'no' (class 2, not-healthy). Our main goal is to subdivide the entire set of data points into two more homogeneous subgroups that are relatively pure, that is, they contain a larger proportion of one class of the points each. This is measured by the Gini criterion, defined for each subgroup as:

$$G = p(1 - p),$$

with  $p$  denoting the fraction of the majority class points in the subgroup. If this subgroup is composed only of points belonging to one category, then the *Gini* index  $G$  is equal to 0.

Alternatively, the quality of the division into two subgroups may be measured by *accuracy*, *misclassification error*, *confusion matrix*, *deviance*, *cross-entropy*, and other criteria (see, e.g., Kuhn & Johnson, 2013; James et al., 2014).

The basic method used in our elaboration is the Random Forest (RF) method. However, because the RF method uses essentially the Binary Decision Tree (BDT) methodology, we introduce the BDT firstly. Next we show, how from single trees the Random Forest is constructed and emphasize its specific properties.

#### Binary Decision Tree (BDT) algorithm

The detailed description of the method may be found in (James et al., 2014; Kuhn & Johnson, 2013). The data vectors representing the patients are considered as data points located in the  $d$ -dimensional data space.

The BDT algorithm splits the current data space hierarchically, into subsequent nonoverlapping regions, possibly pure in the meaning of the accepted criterion, for example, of the Gini criterion. The resulting regions are becoming with each split more and more pure.

In each iteration we find that variable, which provides the 'best' subdivision according to the accepted criterion; the found variable (no.  $J$ ) is next used for subdividing the current data space into two mutually exclusive regions (subspaces), using the inequalities:

$$X_J < x_0(J), \quad \text{or} \quad X_J \geq x_0(J)$$

where  $x_0(J)$  is a point located on the  $X_J$  axis.

The process is iterated so long, till the created regions are pure, or till they contain less than a declared number *minSize* of data points of mixed category (usual *minSize*=10 ).

The entire iterative procedure is usually illustrated by a graph called *tree*. In such a graph the splitting points are represented as nodes, and the created regions by branches. See section 'Results' and figure 3 for a tree obtained from our training data, and for seeing how a tree may be used for classification of a (new) data vector  $\mathbf{x}$ .

### The Random Forest (RF) algorithm

The random Forest method ( Breiman 2001; Breiman & Cutler, 2003) relies heavily on decision trees: instead using only one tree, it uses the methodology of ensemble learning (Li, Wu, Ngom, 2018) and constructs a large random forest of them. The trees are obtained from  $B$  independent bootstrap samples derived directly from the training data set and are grown deep, without any cross-validation or pruning. The algorithm for constructing the subsequent trees is very fast, therefore we may use a large number of the trees (we have used the default  $B = 500$ ).

To obtain the classification of a data vector  $\mathbf{x}$  from the test sample, we put the vector  $\mathbf{x}$  down each tree. The label of the final branch (leaf) attained by the given  $\mathbf{x}$  is indicated as the predicted category ('yes' or 'no') for that  $\mathbf{x}$ . In such a way each  $\mathbf{x}$  obtains one vote from each tree constituting the forest. The final prediction for the given  $\mathbf{x}$  is that category, which was indicated most frequently (principle: *majority voting*).

Such algorithm is known as *bagging* (bootstrap aggregating). It has been demonstrated to give impressive improvements in accuracy just by combining together hundreds or even thousands of trees into a single procedure, see (James et. al, 2014), p.317.

Breiman (2001) has modified the above procedure, making it more efficient. Namely, when building the trees, at each split a smaller number of variables (so called  $mtry < d$ ) are used. In our data with  $d=12$  we tried  $mtry=3, 4$ , and  $5$ . The number  $mtry$  is at each node established as a random sub-sample of the integers  $1, 2, \dots, d$  - to give to all variables the chance to appear as classifiers. This reduces the expected classification error even more - see Fig. 8.8 in (James et al., 2014).

Breiman (2001a) and Breiman & Cutler (2003) have also introduced the concept of *out of bag* (OOB) sample, which permits to obtain a better estimate of the classification error. The OOB sample is composed from small portions of data vectors that did not appear in subsequent bootstrap samples serving for construction of the forest. These left out data vectors constitute an independent sample and may be used for estimating an independent estimate of the classification error.

The OOB sample may serve also for assessing the *importance* of variables in the performed classification task. This is a tricky approach that works as follows: To assess the importance of the  $j$ -th variable ( $j = 1, \dots, d$ ), we permute randomly in the OOB sample the  $j$ -th column. Next we put the modified OOB sample through the trees of the random Forest. In each node, where the  $j$ -th variable has appeared, we record its effect on the decrease in accuracy of classification or the increase of the *Gini* index. This is compared with analogous results obtained with the normal results, that is obtained without the introduced permutation. The difference serves as an index of importance of the  $j$ -th variable in the classification task.

#### 4. Results of calculations

All the calculations were obtained using free software developed in R see: (The R Project for Statistical Computing, 2018). In particular, the BDT results and the RF results were obtained using the R-packages 'tree' and 'randomForest' appropriately. Many indications how to use this software are included in the books (James et al. 2014; Kuhn & Johnson, 2013).

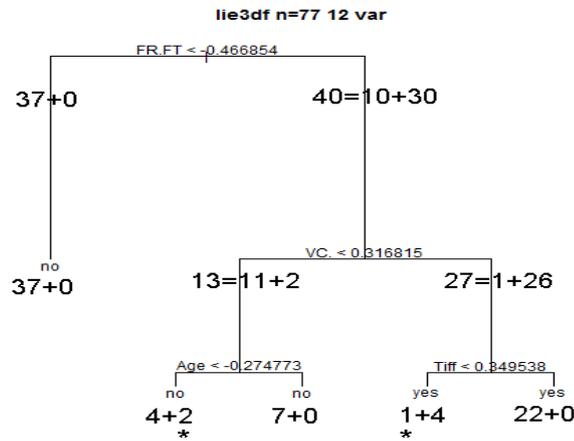
##### Results by Binary Decision Tree

We show here only one tree obtained from the training data set counting  $n=77$  data vectors and called in our computations `lie3df`. The tree was obtained by spitting the entire space into subsequent non-overlapping regions defined by the following inequalities:

**Table 1.** Variables and their values providing splitting inequalities

split level	inequality $X_j < x_0(J)$	inequality $X_j \geq x_0(J)$
1st split	FR.FT < -0.466854	FR.FT $\geq$ -0.466854
2nd split	VC. < 0.316815	VC. $\geq$ 0.316815
3rd split, left branch	Age < -0.274773	Age $\geq$ -0.274773
3rd split, right branch	Tiff < 0.349538	Tiff $\geq$ 0.349538

The tree, obtained as the result of subsequent splits, is shown in Fig. 3. The consecutive splitting points  $x_0(J)$  constitute nodes of the tree. Each branch, originating in a node, is represented by two perpendicular line segments: the first segment is horizontal, and the second is vertical downwards. The inscriptions  $37+0$  over the left branch and  $40=10+30$  over the right branch indicate the number and categories of data points contained in the regions obtained by the split in the given node.  $37+0$  means 37 'no's and 0 'yes'-s; and  $40=10+30$  means a total of 40 data points of mixed category: 10 of them 'no's and 30 'yes'-es.



**Fig. 3.** Binary tree constructed from  $n_1=77$  data vectors constituting the train data set; 'lie3df' in the title indicates data file serving for calculations. Stars in final nodes (leafs) indicate number of misclassified data vectors

The tree grows down, till the created regions become final, that is indivisible. The final branches are labeled by the category of majority of data points associated with them (via the corresponding regions). In particular, in the first split using variable  $J$  labelled FR.FT and splitting point  $x_0(J) = 0.466854$ , resulted in two (sub) regions; the first of them (associated with the left branch of the tree) contained 37 data points of the category 'no' and 0 points of the category 'ones', and the other one (associated with the right branch of the tree) contained 10 data points of the category 'no' and 30 points of the category 'yes'.

Looking at this graph, one is surprised to see, that only four variables were used for providing the subsequent splits of the data space and obtain only 3 misclassified data vectors.

The root of the tree starts at the top and the tree grows downwards. The first *split* provided by the variable FR.FT, has subdivided the entire space into two subspaces (regions), containing 37 and 40 data points appropriately. They are indicated as the left and right branch of the tree. It happened, that the *left one* was already a *final* one, because the respective subspace contained only data points labeled 'no', and as such it was a pure one. The subspace, designated by the *right branch* of the first split, was mixed (10 'no's and 30 'yes'es), and as such was eligible for further splitting.

The next splits indicated by the splitting inequalities in Table 1 proceeded in a similar way.

The final branches are called also called 'leaves'. In Fig. 3 we have five such leaves. They are labelled by the majority category of data points in their regions. Additionally, below each leaf there is information on the category of points contained in the respective regions. Stars indicate number of misclassified points from the data set lie3df obtained when using the ad hoc constructed tree algorithm.

### **Results by Random Forests**

The random forest was constructed from 500 trees evaluated on 500 bootstrap samples drawn from the train data set.

#### **Classification and error rates.**

When putting down through the 500 trees a data vector  $\mathbf{x}$ , one obtains 500 votes for the category of that  $\mathbf{x}$ . The predicted category is established by the principle maximum number of votes; each of the  $B=500$  trees has given one vote. The predictions for the train data, the OOB test data, and the validation data are shown in Tab. 2.

It may be seen, that - both when using 12 and 6 TOP SIX variables- the train data set was classified perfectly in 100%. This happens often in supervised learning and means simply that the algorithm has learned perfectly what it was told to learn. More important are the results obtained from new data, like the test or validation data sets, that were not used for learning. As was said already, in RFs the role of a test set is played by the OOB sample, constructed during learning. Thus, the results from the OOB test sample, shown in Tab. 1, are more interesting: here we observe the overall error rate of 10.39 % for the 12-variables data and the overall error rate of 11.69 % for the 6-variables data. Practically there

are jointly 8 and 7 misclassified data vectors in the 12-variables and 6 variables data set. The difference between the error rates in the full and reduced data set is due to one wrongly classified patient in the reduced data set.

**Table 2.** Confusion matrices from Random Forests when applied to the train, OOB train, and validation data and using all 12 or TOP SIX variables

train set n=77				OOB test set n=77				validation set n=125			
USING 12 VARIABLES											
	No	Yes	ErrRate		No	Yes	ErrRate		No	Yes	ErrRate
No	49	0	0.0	No	45	4	8.16%	No	118	7	5.6%
Yes	0	28	0.0	Yes	4	24	14.29%	Yes	-	-	-
overall			0.0	overall			10.39%	overall			
USING TOP SIX VARIABLES											
	No	Yes	ErrRate		No	Yes	ErrRate		No	Yes	ErrRate
No	49	0	0.0	No	46	3	6.1%	No	117	8	6.4%
Yes	0	28	0.0	Yes	6	22	21.4%	Yes	-	-	-
overall			0.0	overall			11.69%	overall			

### Reduced number of variables

The RFs has provided also a list of ordered importance of variables, evaluated on the basis of the OOB sample using as criterion the mean decrease in Accuracy of classification and the mean increase of the Gini index. The 6 top ranked variables (TOP SIX) appeared to be:  $X_4$ ,  $X_{10}$ ,  $X_{12}$ ,  $X_6$ ,  $X_8$ ,  $X_5$ , that is the variables labelled as *VC.*, *FR.FT*, *VC.MMFT*, *Tiff*, *MMFr*, *FEV1*. However, for simplicity avoiding possible confusion, in the following these TOP SIX variables will be considered in that order as they appear in the list presented in Section 2, that is as shown in the table below:

**Table 3.** Six highly ranked variables

no. of variable	$X_4$	$X_5$	$X_6$	$X_8$	$X_{10}$	$X_{12}$
labeled as	<i>VC.</i>	<i>FEV1</i>	<i>Tiff</i>	<i>MMFR</i>	<i>FR.FT</i>	<i>VC.MMFT</i>

The classification error rates for the reduced 6-variables data are shown in Tab. 2. It is seen that the basic RF with 500 trees yielded a 100% of correct classifications for the  $n=77$  train data, and 117 correct classification in the  $n=125$  validation data, which means a 93.6% of correct classifications. However, the OOB test set yielded 9 misclassifications.

Figure 4 shows misclassified data vectors (from the reduced 6-variables data) in the framework of parallel fences.

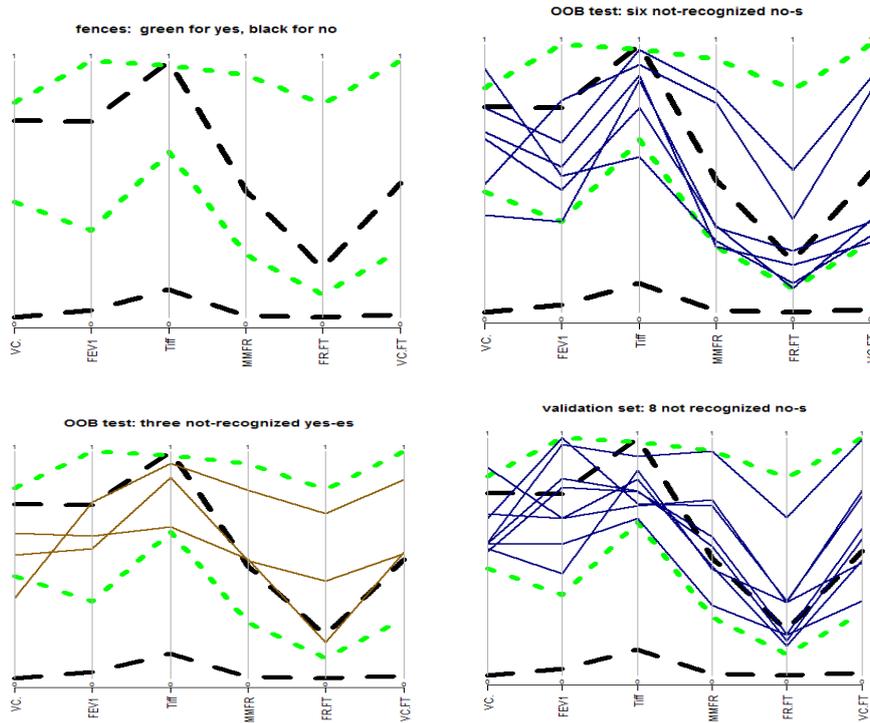


Fig. 4. Parallel fences plots illustrating misclassified data vectors.

Figure 4 is composed from 4 charts illustrating Tukey's fences and misclassified data vectors from the validation set. In particular:

- (a) top row, left chart shows two pairs of fences (upper and lower) derived from the train data. The thick green dotted line segments denote fences for the 'yes' patients, while thick black dashed line segments denote fences for 'no' patients from the same set. The green fences are a subset of the green fences shown in fig. 2.
- (b) top row, right chart shows the parallel fences plot containing the 8 not-recognized 'no's from the validation set in the framework of the same fences as in (a). The line segments of the questioned 'no's are in navy color. Looking at the chart, it is seen, that 7 out of the 8 misclassified 'no's are located in the overlapping area designated both by the green and black fences. Only one item has 4 values of its variables located outside the black fences. Thus this is the case of indistinguishability between the two categories clearly seen already in Fig. 2, right graph. It is surprising, that

only 8 items were misclassified in the case of such big overlap, as shown in Fig. 2.

- (c) bottom row, left chart shows the 3 not recognized 'yes'-es from the OOB test data set - in the framework of the same fences as in (a). Here the 3 not-recognized 'yes'es from the OOB test set are shown in dark orange (a reminiscence of yellow). All 3 of them are located partially in the overlapping area of the green and black fences.
- (d) bottom row, right chart shows the 6 not recognized 'no's from the OOB test data set - in the framework of the same fences as in (a). The respective line segments are in navy (as in chart (b)), and the situation is as in (b) and (c): the misclassified 'no's belong partially both to area delimited by the green and the black fences.

When applied to the internal OOB test data, the RF's made the proper diagnosis for 93.9% of the 'no's (46 out of 49) and 78.6% of the 'yes's (22 out of 28). The 3 not-recognized 'no's and the 6 not-recognized 'yes'es are shown in Fig. 4, bottom row. Again, looking at these plots, one is not surprised that the RFs could not correctly recognize the category label of these data vectors.

## 5. Summary and final conclusions

We have considered 202 data vectors containing values of 12 spirometric variables. Each data vector represents spirometric measurements for one patient. Our aim was to detect if the patient's pulmonary ventilation system is in 'in norm'.

We found that the Random Forest is able to make the proper diagnosis in 92% of the 'no' and 86% of the 'yes' patients. These are results from an internal RF test set built from the OOB data vectors. In a validation sample of 125 new patients in the 'no' category, 118 patients were properly recognized as 'no', which means an accuracy 94.4%.

The RF algorithm provided also the ranking of the used 12 variables with respect of their importance in the constructed diagnosis. Using this result, we repeated the analysis using only the top-ranked 6 variables. The Random Forests, when applied to the *internal OOB test data*, made the proper diagnosis in 93.9% of the 'no' and 78.6% of the 'yes' patients. In the *validation sample* of 125 new patients in the 'no' category, 117 patients were properly recognized as 'no's, which means an accuracy 93.6%.

Using combined parallel coordinates and parallel fences plots we have shown that fences delimiting the plausible area of the 'yes' and 'no' data vectors have a

remarkable overlap. In such a situation the proper diagnosis is impossible for some data vectors. We have shown that the misclassified vectors are located at least partially in such overlapping areas.

The constructed plots show clearly, that the model training data for the 'yes' and 'no' categories are overlapping, thus for some part of the data it is not possible to make the right diagnosis.

The final conclusions are: In the present state of art, and using the methodology of RFs and parallel fences plots, we are able to get a high percentage (higher than 80%) of the right diagnoses, and we are able to sort out the patients that got the right diagnosis. Only for the remaining, say 20 percentage of patients, it is necessary to undergo further examinations.

### Acknowledgements

The authors express their thanks to the secretary of CB and two anonymous reviewers for their suggestions and comments, which helped improving the quality of this paper.

### References

- Bartkowiak A.M., Liebhart J. (2018). Assessing importance of variables in a classification problem using collective intelligence methods. Slides 1-25. Presentation at: 48th Int. Biometrical Colloquium Szamotuly 9-13 Sept. 2018, <http://www1.up.poznan.pl/cb48/en/presentations-2/>
- Boros M., Franczuk M., Wesołowski S. (2006). Guidelines of Polish Respiratory Society concerning the performance of spirometric tests (in Polish). *Pneumonol. Alergol. Pol.* 2006; 74 (Suppl 1). pp. 1-23
- Breiman L. (2001). Statistical Modeling: The two cultures. *Statistical Science*, Vol. 16(3), 199-231
- Breiman L. (2001a). Random Forests, *Machine Learning* 2001, 45(1), 5-32
- Breiman L., Cutler A. (2003). *Random Forest Manual* v. 4.0, Technical Report UC Berkeley 2003
- Inselberg, A. (2009). *Parallel coordinates: Visual multidimensional geometry and its applications*. (Textbook 554 pages). New York: Springer.
- James G., Witten D., Hastie T., Tibshirani R. (2014). *An Introduction to Statistical Learning, with Applications in R. Book*. Springer Science + Business Media New York 2013 (corrected at 4 printing 2014). DOI 10. 1007/978-1-4614-7138-7
- Kuhn M., Johnson K. (2013). *Applied Predictive Modelling*, Corrected at 5th printing 2016, DOI 10.1007/978-1-4514-6849-3, Springer Science + Business Media New York
- Li Y., Wu F-X, Ngom A. (2018). A review on machine learning principles for multi-view biological data integration. *Briefings in Bioinformatics*, 19 (2), 325-340. doi: 10.1093/bib/bbw113
- The R Project for Statistical Computing. Packages 'MASS' (for 'parcoord'), 'tree', 'randomForest'. <https://www.r-project.org>, {viewed 24. Oct. 2018}
- Tukey, J.W. (1977). *Exploratory Data Analysis*. Addison-Wesley. ISBN 0-201-07616-0