Automatic learning for the classification of primary frequency control behaviour

Bertrand Cornélusse*, Claude Wéra† and Louis Wehenkel*
*Department of Electrical Engineering and Computer Science, University of Liège, Belgium
†Elia System Operator, Ancillaries services and balancing, Brussels, Belgium.

Abstract— **In this paper we propose a methodology based on supervised automatic learning in order to classify the behaviour of generators in terms of their performance in providing primary frequency control ancillary services. The problem is posed as a time-series classification problem, and handled by using state-of-the-art supervised learning methods such as ensembles of decision trees and support-vector machines combined with several pre-processing techniques. The method was designed in the context of the Belgian system and is validated on real-life data composed of more than 600 time-series recorded on this system.**

I. Introduction

In Europe and in many other places around the world, the system operator (SO) is responsible for operating the high voltage grid, so as to ensure transparent and efficient market operation and to maintain the system security at an appropriate level. Within this context, an important task of the SO is to determine the needs for various ancillary services, such as primary and secondary voltage and frequency control and to make sure that these services are provided. The methodology of most SOs to this end consists in establishing direct contracts or using ancillary service markets in order to purchase from the generators connected to the system the required reserves needed for the provision of ancillary services. An important task in this context consists in verifying, after the fact, whether or not the purchased ancillary services have indeed been provided, so as to define payments or penalties, depending on the particular regulatory or contractual framework adopted.

In this paper, we consider the verification of primary frequency control services within the Belgian power system. While currently this verification is carried out in a more or less manual fashion by experts looking at power/frequency curves of all generators, the goal of our work is to develop an automatic and objective method based on the a posteriori classification of power/frequency curves recorded in real-time, into a number of classes reflecting the type of observed satisfactory or unsatisfactory performance. Making this classification automatic has several advantages. Indeed, an automatic procedure can be applied systematically and more frequently than a manual expert driven approach, and it is free from subjective judgements and therefore more easily reproducible and more acceptable than a manual approach.

The overall methodology envisaged consists in three main steps: (i) identification of significant events in terms of frequency variations which should lead to the reaction of primary frequency controllers; (ii) extraction of the power and frequency signals from real-time measurements of each generator; (iii) classification of the power/frequency curves in a number of classes corresponding to different types of expected or unexpected behaviours.

The proposed approach is based on automatic learning which can be used in principle to solve these three steps. The paper however focuses only on the third step, which is the most difficult one. Practically, this step is a time-series classification problem, which can be handled by supervised automatic learning algorithms combined with various preprocessing steps. The training sets needed to solve this problem are provided by a number of power/frequency curves preclassified by experts into a given number of performance classes. Supervised learning is then applied in order to derive from this information a classifier able to automatically determine a performance class based on the power/frequency curves.

The rest of the paper is organised as follows. Section II provides the basic principles of supervised learning and a cursory description of the algorithms that were applied in the context of this research as well as the methods used to assess the accuracy of the derived classifiers. Section III describes the principles of applying supervised learning to the verification of primary frequency control performance and Section IV presents the main results that we have obtained on a set of 652 scenarios corresponding to different frequency events and a set of Belgian generators by combining the different algorithms with different pre-processing techniques. Section V provides conclusions and consideration for further research.

II. Supervised automatic learning algorithms

A. Notations and problem formulation

A supervised learning problem is defined by

- a universe of objects: $\mathcal{U} = \{o_1, o_2, ...\}$,
- objects are described by several inputs: $\mathbf{x}(\cdot) : \mathcal{U} \mapsto X$, $\mathbf{x}(o_i) = [x_1(o_i), x_2(o_i), ..., x_n(o_i)] = \mathbf{x}_i$,
- objects are characterised by one output: $y(\cdot) : \mathcal{U} \mapsto Y$, $y(o_i) = y_i$.

Given a learning sample $LS = (o_1, o_2, ..., o_N) \in \mathcal{U}^N$, for which we know the values of both $\mathbf{x}(\cdot)$ and $y(\cdot)$, we want to compute a function $h(\cdot) : X \mapsto Y$ minimising an error measure over $\mathcal{U}$.

In the following subsections we will focus on classification supervised learning methods, i.e. where Y is a finite set of class labels. In classification problems, supervised learning methods often compute the function h by first building from the learning sample an approximation $\hat{P}(y|\mathbf{x})$ of the conditional probability distribution $P(y|\mathbf{x})$ and then deriving from this latter a classifier, for example by choosing for a given input $\mathbf{x}$ the class label which maximises $\hat{P}(y|\mathbf{x})$.

B. Tree-based methods

Tree-based supervised learning is well known for its computational efficiency, interpretability, robustness to outliers, and its capability to cope with high-dimensional problems with a large number of irrelevant input variables [7]. However, it has also been shown that tree-based methods have a high variance [3], which implies that they are often suboptimal in terms of accuracy, specially on problems where the information is spread among a large number of equally relevant variables.

Therefore, tree based ensemble methods have been introduced to decrease variance and to allow them to cope with very complex tasks such as image, text and time-series classification. The general idea behind these methods is to avoid giving a single tree the capability of modelling the whole learning set. This can be achieved either by perturbing the learning set, either by perturbing the construction algorithm, in order to build from a learning set an (often) very large set of different trees, and by deriving the prediction h by aggregating in some fashion (e.g. by voting or by averaging) the predictions derived from each tree in the ensemble.

Below we introduce the main ideas behind the two complementary tree-based ensemble methods used in our application, and explain how to derive from them an estimation of the importance of input variables for a given problem.

1) The Extra-Trees *– Ext(remely) Ra(ndomized) Trees:* A major cause of trees variance is the sensitivity of test nodes thresholds, or cut-points, to the content of the learning set. The main aim of the Extra-Trees [4] is to mitigate this behaviour by randomly perturbing the structure of the trees, thus decreasing their dependence on the learning set.

During the construction phase of a single tree in this method, the search for the best attribute and the best threshold at each node is somewhat randomised. The level of randomisation is related to the size of the subset of input variables which are considered in the search of the best split according to a given score measure. This is controlled through the parameter K. In addition, for each attribute of the subset the threshold is also randomly chosen in its variation interval.

Except for the above, each of the T trees is built on the whole learning set using a classical top down induction algorithm, without pruning, an additional parameter $n_{\min}$ representing the number of objects at a node under which this latter is not split any further. Because all the trees are built independently and because the induction procedure is simplified, this algorithm is computationally very efficient.

The prediction of the ensemble is obtained in two steps: first each tree computes an estimate $\hat{P}(y|\mathbf{x})$ and these quantities are averaged over the ensemble of trees by giving each tree an equal weight; second a class is derived from the resulting average $\bar{P}(y|\mathbf{x})$.

The improvement brought by this method is not only computational. With respect to classical single decision trees, the accuracy is in general dramatically increased, as well as the estimation of the importance of input variables.

2) Tree Boosting: Tree boosting [2] is another ensemble method for supervised learning. It aims at building an accurate predictor from a series of "weak learners", i.e. predictors which are slightly better than a random predictor. During the construction of the ensemble, each tree focuses on the objects of the learning set which were misclassified by the previous trees. This behaviour is achieved by assigning initially equal weights to all objects, and by amplifying the weights, after the creation of each tree, of those objects that are misclassified at the previous step. The ensemble construction is terminated if the prediction error of the current tree exceeds a given threshold, e.g. $1/2$ if the objects can take only two discrete output values.

This method produces a sequence of trees, where the trees progressively focus more and more on the objects that are close to the class-boundaries. It tends to decrease both the bias and the variance relatively to a single tree.

As for Extra-Trees, the output value is predicted after averaging the conditional probability estimates over all the trees. However, in the boosting method the contribution of each tree is down-weighted as a function of its prediction error on the learning set.

When the number of input variables n is large, this method is significantly less efficient than Extra-Trees, but it is sometimes more accurate.

3) Attributes importance: The evaluation of attribute importance aims at identifying among the input variables used to build a classifier those that carry the actual information about the output variable. This evaluation allows to better understand the problem under consideration (like in a sensitivity analysis) and to select a subset of relevant attributes for the problem under consideration.

With tree-based supervised learning methods, a classical way [7] to estimate the importance of an attribute consists in computing from each tree the total information provided by this attribute and by averaging these quantities over all the trees in the ensemble. Notice that in the extra-tree method, the contributions of all trees are equally weighted in this computation while in the boosting method they are weighted in the same fashion as for prediction.

C. Support Vector Machines

Support vector machines based classification [5], [6] is highly related to the notions of kernel and separating hyperplane.

A kernel is a function which maps two objects described by their attributes to a real number, that is

$$k(\cdot,\cdot) : X \times X \mapsto \mathbb{R},$$

allowing us to measure objects similarity. The kernel's action can be understood as the embedding of the objects into a dot product space $\mathcal{V}$, a Euclidean space for example, by a vectorisation function $\phi(\cdot) : X \mapsto \mathcal{V}$. In $\mathcal{V}$ objects are represented by vectors. The value of the kernel function is obtained as the dot product between the two vectors:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j).$$

By mapping objects of $\mathcal{U}$ into a dot product space, the kernel defines the notions of norm and distance over $\mathcal{U}$. This is of interest even if objects attributes already lie in a dot product space because it allows us to design a wide variety of similarity measures, possibly based on nonlinear mappings (polynomial kernels, Gaussian kernels, ...). A kernel k such as described above must be a positive kernel in order to ensure the existence of a proper vectorisation function.

Definition 1: Let $\mathcal{U}$ be a nonempty set of objects, if for all $N \in \mathbb{N}$ and all $o_1, o_{2,}, \ldots, o_N \in \mathcal{U}$ the $N \times N$ matrix

$$K : K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$$

is symmetric and positive (semi)definite, then k is a positive kernel over $\mathcal{U}$.

Assume a classification problem with only two classes, in which the inputs are already in a Euclidean space and the classes are linearly separable. In this case we could simply compute from learning data the equation of a hyper-plane separating objects with distinct classes into different regions. With the convention that the two class-labels are respectively -1 and +1, this would result in a classifier

$$h(\mathbf{x}) = \operatorname{sgn}\left(\mathbf{w}^T \mathbf{x} + b\right). \qquad (1)$$

Notice that the set of separating hyperplanes can be characterised by the following set of constraints

$$\forall i = 1, \ldots, N : y_i \left(\mathbf{w}^T \mathbf{x}_i + b\right) \geq 0. \qquad (2)$$

To find one particular separating hyper-plane we could formulate the problem as a maximisation of the minimum distance from the plane to the points representing the learning samples:

$$\max_{\mathbf{w},b} \min_{i=1,\ldots,N} \min_{\mathbf{x}:\mathbf{w}^T\mathbf{x}+b=0} \{\|\mathbf{x} - \mathbf{x}_i\|^2\}, s.t.(2).$$

Reformulating this problem as a linearly constrained quadratic maximisation minimisation problem, constructing its Lagrangian and verifying the Karush-Khun-Tucker optimality conditions, yields a dual maximisation problem to be solved in practice. The expression (1) of the classifier becomes

$$h(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i \mathbf{x}_i^T \mathbf{x} + b\right), \qquad (3)$$

where the α_i and b are obtained from the dual maximisation problem. One can show that only those α_i values corresponding to points which are at minimum distance from the optimal hyperplane are different from zero. These points are called the support vectors, since the optimal hyperplane (defined by $\mathbf{w}^* = \sum_{i=1}^{N} y_i \alpha_i \mathbf{x}_i$) could be computed only from these samples. In classification problems, typically the number of support vectors is much smaller than the original number of learning samples. which means that the quadratic optimisation problem may be solved efficiently by constraint relaxation techniques.

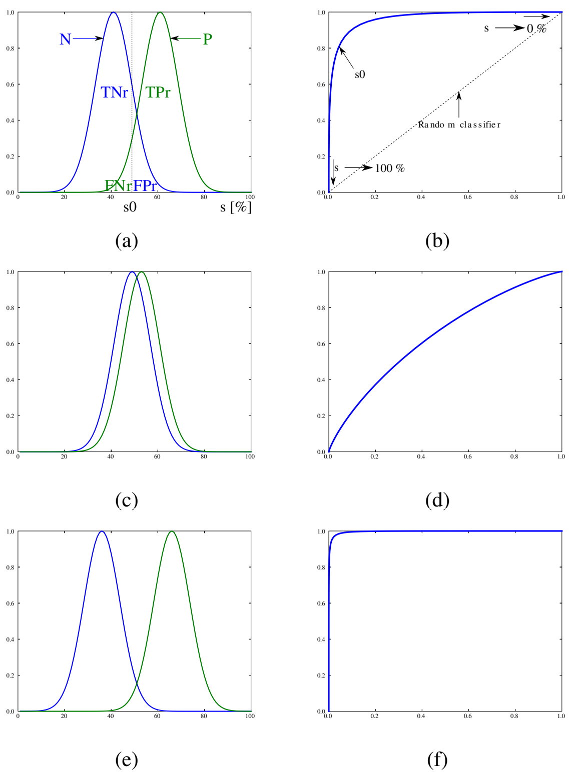
This hyper-plane classifier can be directly extended to the case where a kernel is applied to the input space X. To compute the hyper-plane in the feature space ($\mathcal{V}$), we just have to replace $\mathbf{x}_i^T \mathbf{x}$ in (3) (respectively $\mathbf{x}_i^T \mathbf{x}_j$, in the optimisation process) by $k(\mathbf{x}_i, \mathbf{x})$ (respectively $K_{i,j}$).

D. Assessing classifier performance

1) Evaluation of accuracy: Once a classifier has been obtained by applying a supervised learning algorithm to a training set, it is necessary to evaluate its performance in terms of its ability to classify correctly unseen cases. When the number of available samples is large enough, a convenient way is to split the available data into a training set and a test set. When the number of samples is deemed too small, an alternative is to use the so-called ten-fold cross-validation approach. **This is the approach which is used in this paper. It consists in splitting the dataset into ten subsets of (approximately) equal size and training ten classifiers by using all but one of the subsets. Each classifier is then tested on the subset not contained in its learning set, yielding the total number of mis-classified cases for all ten classifiers.** This approach provides a reliable estimate of accuracy at the price of some computational investment and is generally used in practice to assess the accuracy of supervised learning methods when the value of N is small.

2) Receiver operating characteristic (ROC) curves: ROC curves allow to analyse the false-alarm vs non-detection compromise of a classifier [1].

Assume a problem containing two classes, positive (P) and negative (N). Let s be a classification threshold (figure 1(a)). For example, s is the minimum percentage of P votes in an ensemble tree method to declare an object as P. As the decision taken by a classifier is either good or not, we can define the four following quantities: TPr, FPr, TNr, FNr. The true positives rate (TPr) is equal to the number of objects the classifier declares P and whose output value is P divided by the total number of P objects. The false positives rate (FPr) is equal to the number of objects the classifier declares P and whose output value is N divided by the total number of N objects. The true negatives (TNr), and the false negatives (FNr) are defined in a similar fashion. A ROC graph is simply a plot of the TPr versus the FPr parametrised by the decision threshold s (figure 1(b)). Near the top left corner of a curve ($s \rightarrow 100\%$) nearly all of the votes must be P to declare an object P. In consequence, the FPr is very small and, depending on the classifier performance, the TPr is more or less high. On the other hand, if $s \rightarrow 0\%$ then all the objects are declared P, the TPr as well as the FPr tend to 100%. A perfect classifier's ROC curve would be a broken line passing through the top left corner of the plot, meaning there exists a threshold value leading to a null FPr and a 100% TPr. Hence, if the tree rows of figure 1 correspond to distinct classifiers on the same dataset, then the classifier of the third row is the best and the classifier of the second row is the worst.



(a) (b)

(c) (d)

(e) (f)

Fig. 1. ROC curves illustration

III. Primary frequency control verification

The primary frequency regulation verification can be reduced to a supervised learning problem in the following way.

Suppose that from the SCADA we can collect measurements of system frequency at a reasonable rate (say every second). Then we can identify from these measurements situations where there has been a significant change in frequency (a drop or, less frequently, an increase) and collect the data corresponding to the active power generation of the power plants that were supposed to participate in primary frequency control at that moment. The problem of verifying the service provision then amounts to the comparison of the frequency variation with the variation of active power of a particular generator, and deciding whether or not its response is appropriate.

Suppose that, for a number of frequency events, we have recorded this information and obtained an expert classification of the quality of the response. This data can then be used in order to build a training set corresponding to frequency responses of generators, where for each case the input is provided by the time-series of frequency and active power and the output by the decision associated to that case by the expert.

If the sample of responses is representative of all possible situations, and if the expert decisions are consistent and based only on the information contained in the time series, one can hope to derive a classifier by automatic learning which decisions will be coherent with those of the expert. Otherwise, supervised learning will attempt to build a decision rule which is based solely on the time series information and which on the average agrees with the expert decision. Thus, if the expert decisions are not perfectly coherent or based on supplementary information not contained in the time-series data, the decision rules produced by supervised learning will be consistent and based only on this information, by construction.

IV. Results on the Belgian system

A. *Data sets description*

The results reported below were gathered on a data set containing more than 700 objects corresponding to 35 frequency incidents (see Figure 2 for an illustration). Each learning object has two synchronised time series as input information, one is the measurement of the network frequency over a time period of 30 minutes with one sample every two seconds, the other is the measurement of the power produced by a generator at the same sampling times. The inputs of each object are hence described by a vector of $2 \times 30 \times 30 = 1800$ real numbers.

Each object is assigned an output value by a human expert, namely a class related to the apraisal by the expert of the generator behaviour. The objects whose behaviour is judged correct are labelled OK. Those which are deemed to behave in an incorrect way are labelled NOK_i, $i = 1, 2, ...6$, depending on the root cause identified by the expert for incorrect behavior. Notice that in our analyses we have considered the determination of two types of classifiers, targetting respectively a binary OK/NOK classification (where all NOK_i classes are merged), and a more refined classification where the aim is also to reproduce the root cause judgement of the expert (in terms of NOK_i classes).

B. *Brute force approach*

In our first attempt to build a classifier, we applied the methods exposed in Section II to the raw data, with little preprocessing. The signals were normalised to zero mean and unit variance, to remove from the input information hints about the size of the generating units. In the original data, the interval between two power (or frequency) time samples was of two seconds. In order to reduce the amount of data, without loss of information since the underlying dynamic process has much slower time constants, we grouped time samples five by five and took their average value, increasing the interval between two items of the time series to ten seconds and reducing the number of input variables to 360.

Each object of the learning set is described by two time series concatenated, one for frequency, one for power, and its output value is a label corresponding to a classification criterion OK or NOK_i. Notice that two of the NOK_i classes could be easily identified in an automatic way with very simple rules and we decided to focus on the more difficult classes. Also, two classes which were leading to very similar behaviours and which distinction could only be done using side information that we decided not to use in this study were merged. Consequently, the study was carried out using a total number 652 power/frequency curves classified into on of 4 classes, OK and $NOK'_i, i = 1, 2, 3$.

In the sequel, two kinds of tests were carried out. In the first (OK/NOK classification) we only tried two distinguish good from bad behaviours. The NOK_i' objects were grouped in a single class NOK. In the second (Full classification) we tried to identify the reasons why generators did not react properly to primary control. The results are reported in Table I. In these tests, the SVM method uses a Gaussian kernel of variance σ^2 and the parameter C is related to the SMO (Sequential Minimal Optimisation) approach used to solve the optimisation problem (cf. section II-C).

(a) OK/NOK classification

Method	Parameters	Error rate
ET	$T = 100, K = 19$	17.2 %
Boosting	$T = 100$	19.2 %
SVM (SMO)	$C = 10,\ 2\sigma^2 = 0.25$	**15.5 %**

(b) Full classification

Method	Parameters	Error rate
ET	$T = 100, K = 19$	29.6 %
Boosting	$T = 100$	30.6 %
SVM (SMO)	$C = 10,\ 2\sigma^2 = 0.25$	**28.1 %**

TABLE I

RESULTS OF THE BRUTE FORCE APPROACH

The brute force approach provides us an upper bound of the error rate with the supervised learning methods that we used. The SVMs give the best results in the two tests, but are the most difficult to tune, and a small variation in parameters causes a major change in the results. The ensemble tree methods are more robust from this point of view. Analysing the results more in depth, we see that correct behaviours are a little bit better detected in the full classification test, even if the error rate is greater than in the OK/NOK classification. We found out that this is due to the fact that NOK_3' objects look like OK objects. Attribute importance analysis shows that nearly only the power time-series is taken into account by the classifier, and that we cannot really isolate very important attributes, which is obvious with this low level data. The methods are unable to model the link between power and frequency.

C. Adding features

The lack of accuracy of the brute force approach suggested to elaborate the pre-processing of data before the application of automatic learning methods.

First, the region of the frequency variation or "reference interval" were (automatically) isolated, inducing three regions on each frequency signal (Figure 2). These regions are transposed on all the corresponding power signals. The aim of this step is to decrease the model sensitivity to the the incident's duration. Then we proceeded to a piecewise linearisation of the curves on the three regions, to gain some synthetic information on the behaviour of the curves. We created features from the lines coefficients and introduced them in the input data. We also computed lag correlation coefficients on these three intervals to make the link between power and frequency variation. It is the maximum correlation over a predefined lag period which was introduced in the input data, as well as the lag of the maximum. We also investigated other new attributes such as Fourier and wavelet transforms coefficients to model oscillations and high frequencies in the signals. Several attributes sets were defined to measure the influence of new features on the results reported in Table II. The results of the SVMs are not reported here because they were not better than in section IV-B, because we were not able to find parameter values yielding better results. On the other hand, one can see that Extra-Trees and Boosting succeeded in selecting the good features and in decreasing the error rate by about 5 %. The attributes importance evaluation for this experiment are depicted on Figure 3. They show that the correlation coefficients bring the major part of information to the classifiers. In order of importance, we observe attributes related to the reference interval, to the post-incident interval and to the pre-incident interval.

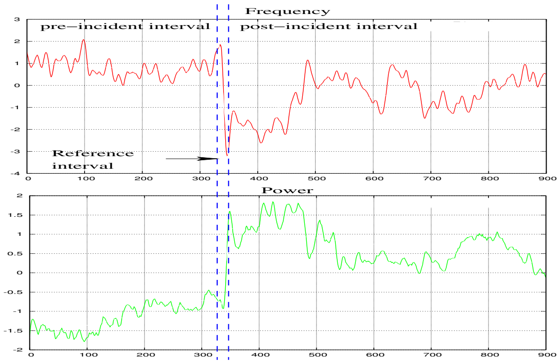


Fig. 2. Reference interval

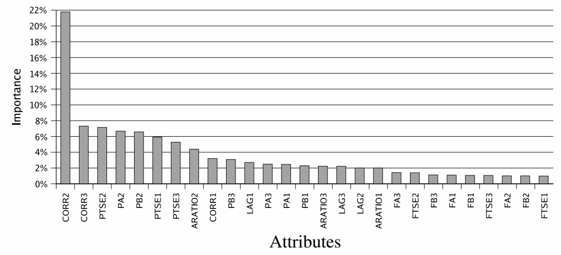


Fig. 3. Attributes importances

(a) OK/NOK classification

Method	Parameters	Attribute set	Error rate
ET	$T = 100, K = 989$	all	12.5 %
Boosting	$T = 100$	all	10.3 %

(b) Full classification

Method	Parameters	Attribute set	Error rate
ET	$T = 100, K = 5$	features	23.8 %
Boosting	$T = 100$	all	23.8 %

TABLE II

RESULTS AFTER FEATURES ADDITION

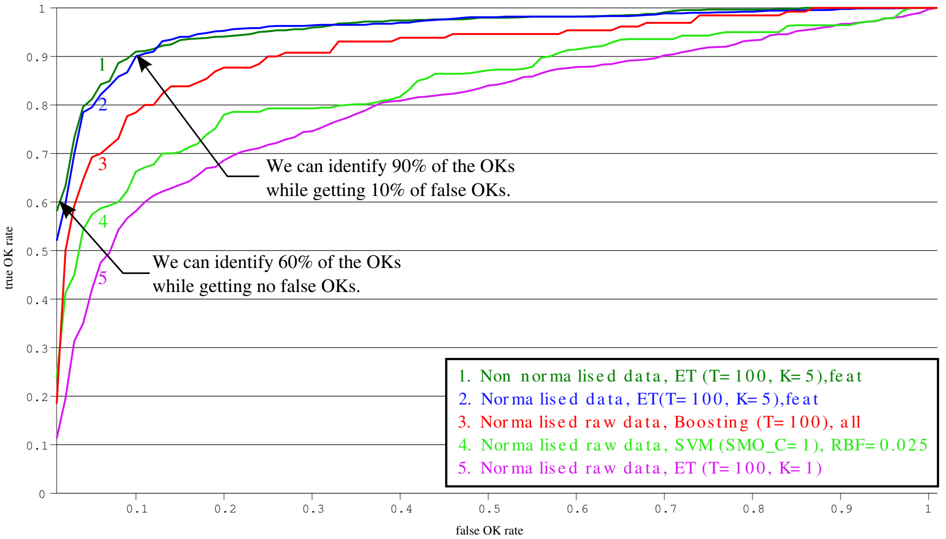


Fig. 4. ROC curves results

D. ROC curves analysis

ROC curves analysis yields some encouraging results, as depicted on Figure 4. This graph represents the ROC curves of classifiers resulting from the full classification, NOK_i being aggregated after the classification process. When using one method we can choose to identify a more or less big part of the OK objects, capturing a more or less big fraction of NOK_i objects, by fixing the decision threshold. The curves 4 and 5 result from the brute force approach. The curve number 4 is obtained with the best method, and the curve number 5 is obtained with the extreme version of the Extra-Trees where K is set to one. The best two methods, corresponding to curves 1 and 2 on the graph, are obtained when only the features added in section IV-C are used, in opposition to curve number 3. These curves show that it is possible to identify 60 % of the OK objects while declaring nearly no NOK_i objects as OK or, on the other hand, to identify correctly 90 % of the OK objects while getting a False OK rate of 10 %.

V. Conclusion

This paper has presented a systematic approach based on automatic learning for deriving verification rules for the primary frequency ancillary service provision. The results obtained on a moderate sample of real-life data from the Belgian system are quite promising. They have shown the interest of adequate pre-processing and tree-based ensemble methods and the possibility to reach error rates in the order of 10%. Complementary investigations not reported in the paper seem to show that this error rate with respect to expert classification is difficult to reduce. Further investigations will be required to assess whether this is related to the limitations of automatic learning methods, the intrinsic difficulty of the problem, or whether it is necessary to use information not contained in the power/frequency curves to reach better agreement with the expert's judgement.

While this paper has dealt with primary frequency ancillary service verification, the presented approach is generic and could be applied to other ex post analyses of ancillary services and market operation performances.

In this context, we believe that automatic learning not only allows to automate tasks currently carried out by human experts, but also to filter out subjectivity and inconsistencies in their judgement.

Acknowledgements

Bertrand Cornélusse thanks Elia (the Belgian System Operator) and FRIA (Belgian Fund for Research in Industry and Agriculture) for allowing him to carry out this research as well as PEPITe, a spin-off of the University of Liège, who gave him the opportunity to use the data mining software PEPITo®.

References

[1] Tom Fawcett. Roc graphs: Notes and practical considerations for researchers. *HP Laboratories, MS 1143, 1501 Page Mill Road, Palo Alto, CA 94304*, 2004.

[2] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, 1996.

[3] Pierre Geurts. *Contributions to decision tree induction: bias/variance tradeoff and time series classification*. PhD thesis, University of Liège, Belgium, May 2002.

[4] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 36(1):3–42, 2006.

[5] Bernard Scholkopf and Alexander J. Smola. *Learning with Kernels*. MIT press, Cambridge, Massachussets, 2002.

[6] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Kluwer Academic Publishers*, 1999.

[7] Louis Wehenkel. *Automatic learning techniques in power systems*. Kluwer Academic, Boston, 1998.