Binary Feature Selection Classifier Ensemble for Fault Diagnosis of Submersible Motor Pump

Francisco de Assis Boldt*, Thomas Walter Rauber*,
Thiago Oliveira-Santos*, Alexandre Rodrigues†, Flávio M. Varejão*
Universidade Federal do Espírito Santo, 29075-910, Vitória, Brazil
* Departamento de Informática - {fboldt,thomas,todsantos,fvarejao}@inf.ufes.br
† Departamento de Estatística - alexandre.rodrigues@ufes.br

Marcos Pellegrini Ribeiro
Petrobras, CENPES/PDP/TE
Av. Horácio Macedo 950 - Ilha do Fundão
21941-598, Rio de Janeiro, Brazil
mpellegrini@petrobras.com.br

***Abstract*—The main motivation to develop this work is to create a diagnosis system able to facilitate the work of human experts responsible for detecting faults before acquisition of submersible petroleum motor pump systems. A new approach for multiclass learning by reduction to multiple, binary classifiers, in a one-versus-one scheme, is presented as an alternative artificial intelligence solution to diagnose faults. Such an idea is based on the hypothesis that each pair of process conditions has different optimal feature sets to improve the classification performance. Thus, features are selected from datasets containing only two classes. Then, classifiers are trained with the selected features. The combination uses the average confidence of each classifier pair prediction to calculate the ensemble answer. Experimental results show that the proposed approach improves classification performance in a statistically significant way, when compared with correlated work. A secondary contribution is the analysis of the most difficult fault to be identified, namely rubbing.**

I. Introduction

In some cases, oil wells elevate its product naturally to the surface, aka petroleum seep. When an increase in the fluid pressure or in the production rate is needed, an artificial support method has to be applied. Electric submersible pump (ESP) systems are often used as an artificial lifting method in offshore oil exploration. ESP systems utilizes a submerged multistage centrifugal pump driven by an electrical motor, whose power is supplied from the surface by an electric cable [1]. These systems work inside the oil well and its installation and eventual removal due to maintenance are expensive operations. Before an ESP system is put into operation in sub-sea installations, it needs to be carefully tested for avoiding initial failures, and high intervention costs. Thus, to avoid posterior problems during the operational phase, rigorous reliability evaluation is performed before ESP deployment [2], [3]. This evaluation is made in laboratory where large amounts of data are used by experts who analyze the ESP system. In vibratory analysis, accelerometers are attached to several ESP body positions, during an operational test in which water is pumped. In the test, 36 accelerometers are fixed pairwise with a 90 degrees phase offset in the axial direction. The accelerometers are equally distributed along the motors, protectors and pumps. Hence, one pair of accelerometers is connected at the bottom, middle and top of each component, as shown in figure 1. Finally, the collected data is analysed and labeled by an expert, using as the principal technique the visual inspection of the frequency spectra obtained from the Fourier transform of the raw vibration signal.

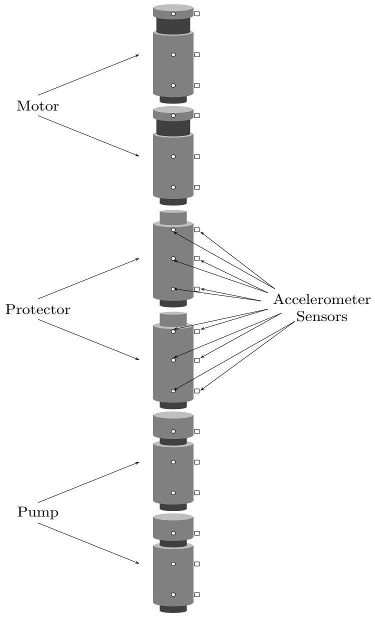

Fig. 1. Accelerometers position to test an ESP system. Six accelerometers are placed in each ESP component.

Human experts usually expend several days to provide the final diagnostic. Not rarely, the decision whether the pump system is able to be put into operation relies on professional experience. An automatic diagnosis system can help human experts, experienced or novice, to identify failures. Supervised learning is often used to develop model-free diagnosis systems [3]. The model-free approach has the advantage of avoiding explicit expert knowledge. Its main drawback is the necessity of a significant amount of labeled examples. In a previous work presented in [4], a comparative study of classification methods applied to diagnose faults in ESPs is presented. That work uses eight features extracted from the frequency domain and presents the K-Nearest-Neighbour as an equivalent classifier to Random Forest, Support Vector Machine and Decision Trees, when the training data is standardized.

A model-free diagnosis system does not need to be restricted to a single feature extraction model. It can use heterogeneous feature models since redundant or irrelevant features are discarded by feature selection techniques [5]. The current work investigates the automatic selection of features for the pump diagnose problem, it adds twelve more features, also extracted from the frequency domain, to enhance the classification performance. Based on the hypothesis that different feature sets increases the separability of each pair of classes, this work proposes a new approach to combine classifier algorithms trained with different feature sets. A large amount of data has been generated over years of cumulative knowledge acquired by ESPs diagnostic tests. Comparable experiments with this data is computably expensive. Techniques like Error-Correcting Output Codes (ECOC) [6] and feature selection are also expensive. Using these approaches simultaneously would make comparison of the algorithms nearly unfeasible. Therefore, a lighter binary classifier ensemble algorithm is proposed. The results are compared against different approaches, using the ESP data and reliable statistical methods. Details about the proposed method are presented in section III. Section II explains how the ESP data is used to create a model-free diagnosis system and which features and conditions are considered here. The experimental method is explained in section IV, results and discussion in section V. Final remarks and future work are outlined in section VI.

II. Data Acquisition and Diagnosis

The vibration data is collected with a sample rate of 4096 Hz by accelerometers placed in the ESP system, as shown in figure 1. When a human expert performs a diagnosis, the data in the time domain is transformed to the frequency domain and plotted to allow a better visual inspection, as shown in figure 2. The red and green horizontal lines are similar to alarm thresholds where the amplitude usually cannot be higher. However, they are actually guidelines, because there are situations where the amplitude surpasses them and the fault is not considered, and situations where the amplitude does not surpass them but the fault is considered. The diagnostic also depends on the signature of the other components, or the component behavior in lower and higher frequency rotations. Thus, the decision if there is a fault or not frequently relies on experience of the human expert. This study tries to create an expert system that does not depend on a threshold or considers different signals to diagnose a component. Consequently, additional features have to be added to enhance the classification performance. The dataset used in this work has 4570 examples distributed a priori as shown in table I.

TABLE I
A-PRIORI PERCENTAGES OF NORMAL AND FAULT CLASSES.

Condition class	A priori class distribution [%]
Normal condition	81.10
Unbalance	10.61
Misalignment	1.09
Rubbing	0.77
Accelerometer fault	6.43

In a typical normal condition all amplitudes are below the red and green lines. The unbalance signature is characterized by a high peak in the $1x$ frequency, where x is the rotation frequency of the shaft. A misalignment fault signature is characterized by an abnormal high peak in the $2x$ frequency. Rubbing is characterized by high energy in the lower frequencies and the presence of peaks in $0.5x$, $1x$ and $2x$. It is not uncommon that accelerometers fail. Despite it is not an ESP defect, it caches the eventual existence of an ESP fault. Thus, the accelerometer fault is also considered. It has high energy in the lower frequencies, but does not present any peak in the $0.5x$, $1x$ and $2x$ frequencies.

Since the size of each time domain signal is much higher than the number of examples, this signal cannot be used directly in supervised learning [5]. Thus, descriptive features must be extracted before training a classifier via supervised learning. The features extracted in [4] are described below:

- x_{rf}: Rotation frequency (first harmonic) of the submersible motor pump during the test.
- x_{rfm}: Magnitude in the rotation frequency (first harmonic);
- x_{rfm2}: Magnitude in the double of the rotation frequency (second harmonic);
- x_{rfrms}: Root mean square of the magnitudes around the rotation frequency, [x_{rf} - 1, x_{rf} + 1];
- x_{rfmm}: Median of the magnitudes around the rotation frequency, [x_{rf} - 1, x_{rf} + 1];
- x_{m3to5}: Median of the magnitudes of the low frequencies, interval [3Hz, 5Hz];
- x_{ilr}: Intercept (a) of the linear regression of logarithm of the frequency magnitudes (Mag) over the interval of frequencies (Fq) [5Hz, 19Hz], equation 1;
- x_{slr}: Slope (b) of the linear regression of logarithm of the frequency magnitudes (Mag) over the interval of frequencies (Fq) [5Hz, 19Hz], equation 1;

$$\log(Mag) = a - b \times Fq \qquad (1)$$

Following the principle of heterogeneous feature extraction presented in [5], twelve more features have been added to the dataset used in [4]. The additional features are the average, standard deviation and the maximum of four frequency windows. A zoom-in of the spectra in figure 2 shows the considered intervals by red curly brackets. The intervals are $0x$ to $0.25x$, $0.375x$ to $0.625x$, $0.875x$ to $1.125x$ and $1.875x$ to $2.125x$. The example in figure 2 was collected from an ESP system working in a frequency rotation of 60 Hz ($1x = 60$). It is worth to highlight that 60 Hz is the nominal frequency rotation, that is the frequency desired for a given power supply. However, the natural friction of the components always reduce the real rotation. That is why the highest peak ($1x$) in figure 2 is presented in a smaller frequency than 60 Hz. The difference of nominal and real rotation frequency is the main motivation to frequency windows.

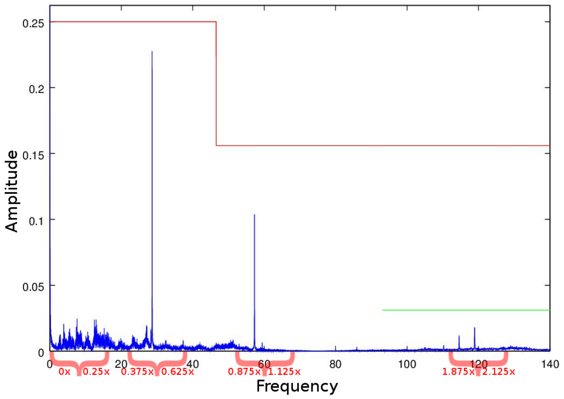

Fig. 2. Frequency bands of the additional feature set. The red and green horizontal lines are similar to thresholds which the amplitude usually cannot surpass them. The red curly brackets show the frequency windows from where the additional features were extracted.

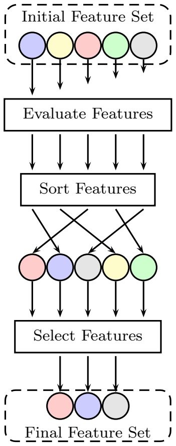

Fig. 3. Ranking Feature Selection

III. Binary Feature Selection Classifier Ensemble

Feature selection [7] and classifier ensembles [8] are established research areas. Classifier ensembles are a combination of accurate classifiers that commit errors in different regions of the multivariate input space [9]. The main goal of a classifier ensemble is to improve classification performance. Thus, the run-time overhead that such an approach might have is usually tolerated. On the other hand, feature selection methods aim to speed up processing and augment the performance of the final classifiers simultaneously [7]. Feature selection can be seen as a combinatorial optimization problem, composed of a selection criterion and a search strategy, improving prediction performance, and reducing problem dimensionality [7], [5]. Feature selection algorithms, both optimal and suboptimal, commonly are used as filter, wrapper, embedded or ranking methods. Filter methods calculate the goodness of the combination of some features based on the data only, so they are faster than wrapper methods, which need a classifier to evaluate a feature combination according to some validation method. Commonly, wrapper methods achieve better results [10] than filter methods with the drawback of a higher computational burden.

Ranking methods provide a univariate evaluation of the features without considering possible mutual dependencies among them. The feature evaluation can be done using only the data, like filter methods, or using some performance estimation, like wrapper methods. Ranking methods return a list of features sorted in descendant order of each feature evaluation. The final feature subset has the k highest ranked features in the returned ordered list, where k is the number of features to be selected. Figure 3 illustrates how a ranking feature selection algorithm selects three features ($k = 3$) from a dataset with five features. Usually ranking methods are faster and have inferior performance than multivariate methods, that consider mutual dependencies among the features. On the other hand, many feature selection algorithms include feature ranking as a principal or auxiliary selection mechanism because of its simplicity, scalability, and good empirical results.

The combination of heterogeneous feature extraction models and feature selection methods has shown promissory results in automatic fault diagnosis [5], [11]. However, the optimal set of features to identify multiple conditions might be different, depending of which conditions are considered. Here a classification scheme is presented where the features are selected by pairs of conditions, aiming to maximize the identification of each class with distinct feature sets. In order to implement this idea, a one-versus-one performance estimation arrangement has to be defined. Nevertheless, even ranking feature selection might be expensive when large datasets as the ESP dataset are studied. Therefore, techniques like Error-Correcting Output Codes (ECOC) [6] used with feature selection for large datasets may be excessively expensive. For instance, table II shows a one-versus-one code design for five classes, where $\mathcal{C}$ represents classes and $\mathcal{L}$ represents learners, or classifiers. Let M be the coding design matrix of table II, with elements m_{kl}, and s_l be the predicted classification score, or confidence, for the positive class of learner l. Five classes ($k = 5$) demand $k \times (k-1)/2 = 10$ learners in a one-versus-one arrangement. ECOC solves an optimization problem, equation 2, for each sample to give the final prediction.

$$\hat{k} = \underset{k}{\operatorname{argmin}} \frac{\sum_{l=1}^{L} |m_{kl}| g(m_{kl}, s_l)}{\sum_{l=1}^{L} |m_{kl}|} \tag{2}$$

For a dataset with 4570 samples, it takes a long time to perform multiple cross-validations which have comparable results considering statistical significance, since each sample would have to perform an optimization over matrix M. Whence, a simpler one-versus-one scheme is proposed here.

TABLE II
One-versus-one code design for five classes.

	$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$	$\mathcal{L}_4$	$\mathcal{L}_5$	$\mathcal{L}_6$	$\mathcal{L}_7$	$\mathcal{L}_8$	$\mathcal{L}_9$	$\mathcal{L}_{10}$
$\mathcal{C}_1$	1	1	1	1	0	0	0	0	0	0
$\mathcal{C}_2$	-1	0	0	0	1	1	1	0	0	0
$\mathcal{C}_3$	0	-1	0	0	-1	0	0	1	1	0
$\mathcal{C}_4$	0	0	-1	0	0	-1	0	-1	0	1
$\mathcal{C}_5$	0	0	0	-1	0	0	-1	0	-1	-1

In the proposed method the learners $\mathcal{L}$ answer the actual label $\mathcal{C}$ instead of a code in M. Thus, the learners design is arranged as shown in table III. Let P be the prediction matrix with elements p_l, and s_l be the predicted classification confidence of the learner prediction. The label with the maximal average confidence is returned as the predicted label by the ensemble, equation 3.

$$\hat{C} = \max_{\mathcal{C}} \frac{\sum_{l=1}^{L} \begin{cases} s_l, & p_l = \mathcal{C} \\ 0, & \text{otherwise} \end{cases}}{\sum_{l=1}^{L} \begin{cases} 1, & p_l = \mathcal{C} \\ 0, & \text{otherwise} \end{cases}} \tag{3}$$

No optimization needs to be performed and the calculus of the predicted label can be done by efficient matrix calculations, present in many software libraries, like e.g. Matlab. Preliminary experiments have shown better classification results for average confidence than for the sum of confidence, because the average confidence rewards the highest confidence levels instead of the highest number of votes. It is expected that learners do not give high scores for samples with actual labels that were not present in the classifiers training dataset. Hence, a low score reduces the average confidence for that label.

TABLE III
PROPOSED ONE-VERSUS-ONE APPROACH FOR FIVE CLASSES.

$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$	$\mathcal{L}_4$	$\mathcal{L}_5$	$\mathcal{L}_6$	$\mathcal{L}_7$	$\mathcal{L}_8$	$\mathcal{L}_9$	$\mathcal{L}_{10}$
$\mathcal{C}_1$	$\mathcal{C}_1$	$\mathcal{C}_1$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_3$	$\mathcal{C}_4$
$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_5$	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_5$	$\mathcal{C}_4$	$\mathcal{C}_5$	$\mathcal{C}_5$

The training and test procedure of the proposed approach is presented in figure 4. Right angle rectangles represent datasets, round corner rectangles represent processes and ellipses represent classifiers or predictions. The training process is represented by the largest dashed rectangle. This process splits the training dataset into smaller datasets with samples of only two classes each. Considering that the ESP dataset has $n = 5$ conditions, $n * (n - 1)/2 = 10$ datasets have to be assembled. Feature selection is performed over each binary dataset generating potentially reduced datasets, probably with different features. Classifiers that are trained with these datasets compose the ensemble classifier. The test process, represented by the smaller dashed rectangle, receives the test dataset and the ensemble classifier. The prediction label process performed by the ensemble classifier is represented by the dotted rectangle inside the test process. Each classifier in the ensemble uses the whole test dataset to predict the label of each sample. Obviously, many of those predictions will be wrong, since the classifiers in the ensemble were not trained with all classes. However, it is expected that classifiers that try to identify samples of classes with which they were not trained, give a low score for those samples. Then, the low score reduces the average confidence for the wrong predicted label. The label with the highest average confidence is chosen as the predicted label.

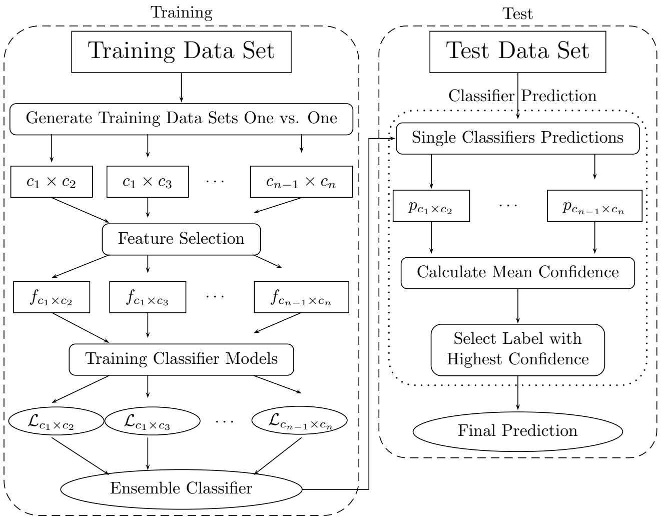

Fig. 4. One vs. One Feature Selection.

IV. Experimental Evaluation

Publications that performed experiments using datasets that have no explicit separation of training and test datasets may use cross-validation to evaluate and compare different techniques. However, a single cross-validation has a high variability due to its randomness. Different fold separations may result in significantly different values, favoring one method over other. Consequently, the replicability of this validation method is low. Replicability of an experiment is a measure of how well the outcome of an experiment can be reproduced [12]. A stronger approach, also very common, is the repeated cross-validation, e.g. ten rounds of 10-fold-cross-validations. Nevertheless, this validation method might be also unfair, because the evaluated methods probably do not use the same data division. In the present work, a multiple paired nested cross-validation was performed. It means that all evaluated methods used the same data division for training and test. Despite this validation method provides no guarantee that one method is really better than other, it ensures that both methods use equal dataset divisions. Therefore, this kind of paired validation tends to reduce the probability of Type I error [13]. A Type I error is a conclusion of an experiment that there is a difference between algorithms, while in reality there is none.

The experiments performed for this work use multiple paired nested cross-validation, i.e. ten rounds of 10-fold-cross-validation. Nested means that the training process does not have access to the test dataset. All feature selection processes use only the nine folds reserved to train the classifier, ignoring the fold reserved for test. This procedure aims to guarantee that the experiments are bias aware. Ranking feature selection applied as wrapper was used to select the features. This combination is easy to understand and implement, because ranking is a simple feature selection algorithm and the wrapper approach aims to optimize the actual final performance instead of intermediate criteria, as in the filter approach. Using the wrapper approach, the same classifier that performs the classification performs the feature selection. Once the nested validation is performed, the validation used to tune the classifier or select features by a wrapper does not need to be the same as for the outer validation. Therefore, the ranking feature selection method used the average performance of a five rounds holdout 80/20 to be executed as a wrapper.

As the ESP dataset has a very unbalanced number of samples for each class, accuracy is not a good choice as the classification performance metric. The Fscore is more appropriate. Since the ESP data is multiclass the Fscore macro-averaged ($Fscore_M$) [14] was used. Its calculus (equation 6) is based on the Precision macro-averaged (equation 4) and the Recall macro-averaged (equation 5), where l is the number of labels, and β is a weighting parameter. According to [14], macro-averaging treats all classes equally while micro-averaging favors bigger classes.

$$Precision_M = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fp_i}}{l} \tag{4}$$

$$Recall_M = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fn_i}}{l} \tag{5}$$

$$Fscore_M = \frac{(\beta^2+1)Precision_M Recall_M}{\beta^2 Precision_M + Recall_M} \tag{6}$$

All experiments were performed using dataset standardization, because the standardization process has shown significant improvement of performance classification for the K-NN classifier in [4]. It is worth to highlight that the standardization process performed here uses only the average and standard deviation obtained exclusively from the training dataset to avoid overoptimistic results. In experiments presented in [4] the K-NN classifier has a tuning phase to chose the value of K of the number of neighbors as part of the training process. This tuning phase with wrapper feature selection and one-versus-one arrangement would expend an excessive amount of computational resource and, consequently, a prohibitive long time to perform multiple cross-validations. Thus, preliminaries experiments were performed to chose the value of K in advance. The 1-NN has shown the best classification performance when compared to 3-NN and 5-NN, using the original dataset. Therefore, the experiments were performed with the 1-NN classifier. Finally, robust statistical tests were also performed following the same procedure presented in [4].

V. Experimental Results

Figure 5 shows the boxplot chart of four methods compared, where KNN is the 1-NN classifier using the original ESP dataset from [4], KNN+ is the 1-NN classifier using the extended dataset, that has the same features of [4] plus the twelve new features presented in section II, KNN+FS is the 1-NN with the extended dataset and a multiclass feature selection. KNN+EFS is the proposed approach using the extended dataset and the 1-NN classifier. The search strategy to select the features for both, KNN+FS and KNN+EFS, was the ranking feature selection applied as wrapper (figure 3).

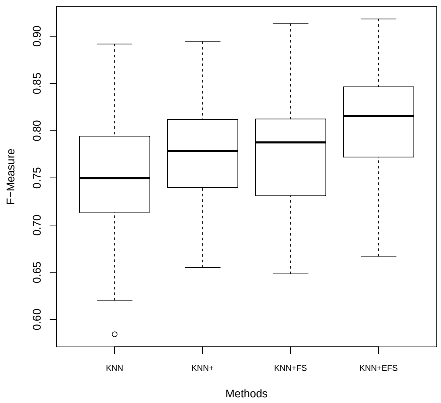

Fig. 5. F-Measure boxplot.

By visual inspection it is easy to see the improvement provided by the conjunction of additional features and the proposed approach. Table IV numerically repeats the parameters used to plot the chart of figure 5, where 'Min.' is the minimum value, 'Q_1' is the first quartile, 'Q_3' is the third quartile and 'Max' is the maximum value obtained for each classification method. The highest values are in bold face.

TABLE IV
Parameters of the boxplot in figure 5.

	KNN	KNN+	KNN+FS	KNN+EFS
Min.	0.5843	0.6550	0.6486	**0.6671**
Q_1	0.7148	0.7402	0.7312	**0.7722**
Median	0.7496	0.7786	0.7876	**0.8157**
Mean	0.7522	0.7749	0.7752	**0.8068**
Q_3	0.7941	0.8115	0.8124	**0.8461**
Max.	0.8918	0.8942	0.9133	**0.9183**

Figure 6 shows a matrix to compare the four methods by pairs. In the lower triangle, the histograms of the differences of the Fscores for each pair of method are shown. The zero of the horizontal axis is in the center of each histogram box. The upper triangle presents the corresponding p-values of the correlated t-test for each pair of methods. The value in bold face, 0.00576 in the top right square, indicates a statistically significant difference. In other words, the KNN+EFS method is statistically significant better than the KNN method. It can be seen that, despite the small improvement of the multiclass feature selection (KNN+FS), such an improvement was not enough to provide statistical significant difference to the 1-NN with the original ESP dataset. Further explanations about how the matrix of figure 6 is generated can be found in [4].

It is also important to know how the behavior of each method relates to each condition. Figure 7 shows such an information. It can be seen that the additional features improve the recall for all conditions, when compared with the original dataset. When the KNN+FS is compared with KNN+, the former improves the recall for all conditions, except for the rubbing fault. This behavior can theoretically be expected. According to [8], a classifier ensemble only can improve the

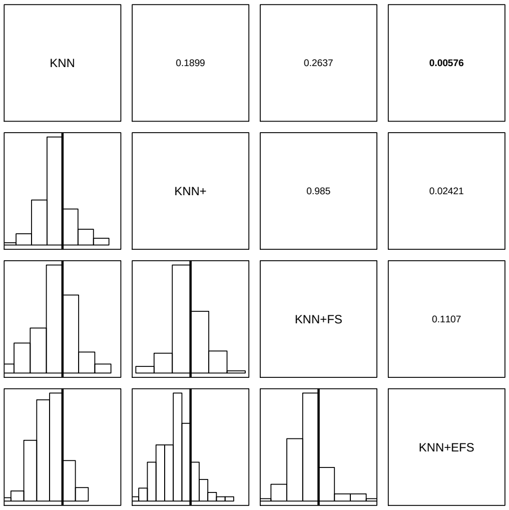

Fig. 6. Pairwise comparison between methods. The lower triangle shows the histograms of the differences of the Fscores for each pair of method. The upper triangle presents the corresponding p-values of the correlated t-test.

classification if the individual performance of each learner is higher than 0.5, otherwise, it tends to decrease performance. This happens for the rubbing condition, where multiclass classifiers have poor classification performances for this class.

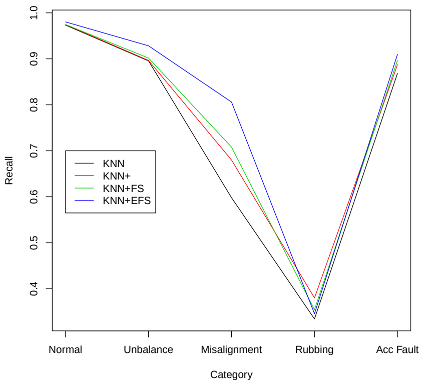

Fig. 7. Recall per condition.

Knowing that rubbing is the most difficult condition to be identified, two research approaches can be envisioned. Firstly, rubbing is the condition with less examples in the dataset. Only 35 of the 4570 samples are labeled as rubbing. Oversampling techniques might solve this problem. Secondly, the used features are not discriminative enough to discern rubbing properly, and more features have to be extracted and tested. The new features might be designed exclusively for this condition or a range of generic features can be extracted and filter through feature selection methods.

VI. Conclusion

The union of a one-versus-one classification approach bundled with feature selection of an extended dataset has shown statistically significant improvement in classification performance. The additional features and feature selection with a multiclass classifier improved the overall classification, but without statistical significance. Only the feature selection method with one-versus-one arrangement was able to improve the classification scores significantly. This result suggests that different feature sets have significantly different performances to identify each group of process conditions, corroborating the fundamental hypothesis of this work. As the proposed approach did not improved the classification of the rubbing fault, future works will try oversampling and extract new features to improve the rubbing condition classification performance. It is also envisioned to use the Cascade Feature Selection [15] technique to boost the selection speed and improve the overall classification performance.

References

[1] G. Takács, *Electrical Submersible Pumps Manual: Design, Operations, and Maintenance*. Elsevier, 2009.

[2] P. J. Tavner, L. Ran, J. Penman, and H. Sedding, *Conditiong Monitoring of Rotating Electrical Machines*. London: The Institution of Engineering and Technology, 2008.

[3] F. de Assis Boldt, T. W. Rauber, F. M. Varejao, and M. Pellegrini Ribeiro, "Performance analysis of extreme learning machine for automatic diagnosis of electrical submersible pump conditions," in *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*. IEEE, 2014, pp. 67–72.

[4] T. Oliveira-Santos, T. W. Rauber, F. M. Varejão, L. Martinuzzo, and W. Oliveira, "Submersible motor pump fault diagnosis system: A comparative study of classification methods," in *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*. IEEE, 2016.

[5] T. W. Rauber, F. de Assis Boldt, and F. M. Varejao, "Heterogeneous feature models and feature selection applied to bearing fault diagnosis," *Industrial Electronics, IEEE Transactions on*, vol. 62, no. 1, pp. 637–646, 2015.

[6] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of artificial intelligence research*, vol. 2, pp. 263–286, 1995.

[7] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[8] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine learning*, vol. 51, no. 2, pp. 181–207, 2003.

[9] D. W. Opitz, "Feature selection for ensembles," in *AAAI/IAAI*, 1999, pp. 379–384.

[10] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern recognition*, vol. 33, no. 1, pp. 25–41, 2000.

[11] R. S. Broetto and F. M. Varejão, "Heterogeneous feature models and feature selection applied to detection of street lighting lamps types and wattages," in *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*. IEEE, 2016, pp. 933–938.

[12] R. R. Bouckaert, "Estimating replicability of classifier learning experiments," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 15.

[13] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[14] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[15] F. de Assis Boldt, T. W. Rauber, and F. M. Varejão, "Cascade feature selection and elm for automatic fault diagnosis of the tennessee eastman process," *Neurocomputing*, vol. 239, pp. 238 – 248, 2017.