Constructing Feature-based Ensemble Classifiers for Real-World Machines Fault Diagnosis

Marcelo V. de Oliveira, Estefhan D. Wandekokem, Eduardo Mendel, Fábio Fabris, Flávio M. Varejão and Thomas W. Rauber
Department of Computer Science
University of Espírito Santo
29060–970, Vitória, ES, Brazil
Email: mvdo123@gmail.com
{estefhan,mendel,ffabris,fvarejao,thomas}@inf.ufes.br

Rodrigo J. Batista
Espírito Santo Exploration and Production Business Unit.
Inspection and Maintenance Engineering.
Petróleo Brasileiro S.A. PETROBRAS.
Vitória, ES, Brazil.
Email: rodrigojb@petrobras.com.br

Abstract—This paper presents the results achieved by fault classifier ensembles based on supervised learning for diagnosing faults on oil rigs motor pumps. The main goal is to apply two feature-based ensemble construction methods to a real-world problem. Recent studies have shown that the use of ensembles of classifiers that are accurate and at the same time have diversifying results can improve the final classification accuracy, compared to a single accurate classifier. The diversification performed by the methods presented in this work is obtained by varying the feature set each classifier uses. We show results obtained with the established genetic algorithm GEFS and a recently developed approach called BSFS, which has lower computational cost. We rely on a database of real data, with 2000 acquisitions of vibration signals extracted from operational motor pumps. Our results show that the ensemble methods had a higher classification accuracy solving a real-world fault diagnosis problem than single classifiers. Also, we present promising results in our experiments with both algorithms, that successfully solves the problem.

I. INTRODUCTION

Oil rigs motor pump fault diagnosis is an important problem in oil industry. Detecting or even preventing failures still at an early stage in complex machines usually benefits in terms of economy and security [1]. Also, predictive maintenance [2] from analysis of vibration signals produced by the process allows monitoring and making conclusions about the operational state of the machine. In addition to that it permits to take appropriate measures to extend the time of use, which minimizes costs resultant from the machine’s downtime.

Several pattern recognition paradigms have been used in the past years to construct fault diagnosis systems. The two principal approaches are: model-based techniques [3] and model-free techniques. In this work we describe two model-free fault diagnosis methods, both using the supervised learning paradigm [4] as the primal mechanism to automatically generate the classifiers. The main advantages of the model-free approach over the model-based techniques is the non-necessity of analytical models to describe the plant. Moreover, the availability of statistically significant amounts of real data from operational oil extraction rigs can overcome the fundamental drawback of such an model-free approach to fault diagnosis: if only a small number of patterns are available in the training phase, the statistical classifiers might be misleading and very sensitive to noise.

A good feature model is important to describe the fault classes. The traditional way to select features is the manual feature selection, in which the specialist uses his knowledge to determine which features have to be employed. In databases of real data this could be a misguided approach because depending on the type of data, it could be very difficult to determine good features that represent the classes with effectiveness, since the difference from one class to another may be very subtle. So we use automatic feature selection methods, that have shown better results so far [5].

Recently in the area of machine learning the concept of ensemble based systems has been shown to produce favorable results compared to those of single-expert systems for a broad range of applications [6]. An ensemble of classifiers is a set of models whose individual decisions are combined in some way to greatly improve generalization accuracy. The main advantages are that ensembles are often more powerful than the individual classifiers [7] and the results are less dependent on particularities of a single training set. A condition for an ensemble to be more accurate than its individual classifiers is to generate a set of classifiers that are accurate and diverse in their predictions, in the sense that prediction errors occur in different regions of the feature space. The diversity between the classifiers can be achieved applying various methods. The most popular approaches are the ones that use different subsets of training data, set different classifier intrinsic parameters or employ a distinct feature subset for each individual classifier. The methods studied in this paper introduce diversity into the ensemble by varying feature sets.

The main goal of our work is to provide two feature-based ensemble construction methods and apply them to our problem. The first one is the algorithm Genetic Ensemble Feature Selection [8]. It is a genetic algorithm widely used for selecting features for ensembles, and it is known for delivering
good classification results. For scoring the fittest individuals, this algorithm takes into consideration both the accuracy of each classifier individually and how different the classification result of one classifier is from the others in the population.

The other method consists of using two widely known algorithms, Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) [9] to create a large number of classifiers for each number of features. Then, a predefined number of classifiers are selected using the same greedy forward searches we used to select features to finally compose the ensemble. This method is called Best Selected Feature Subsets (BSFS) [5]. The main motivation for comparing the results of these two methods is to analyze their strong and weak points and to determine if they are usable approaches for solving this particular problem.

The rest of the paper is organized as follows. In Section II we give a brief explanation about the problem that is being treated here, the details about the database and the particular choices that were made in this work based on what the classifier was intended to do. In Section III we discuss some important points about the support vector machine classifier in general and in this particular application. Section IV explains the GEFS algorithm, how it works, how it is used in the experiments and what measurements are used to determine the fittest classifiers for the ensemble. Section V deals with the specifics of the proposed BSFS method. Section VI discusses the experiments that were made with both methods previously presented and the obtained results. Finally, Section VII draws conclusions and points out future work.

II. THE PROBLEM DOMAIN

The objective is to detect faults in oil motor pumps at an early stage and thus repair damaged components during planned maintenance, thereby minimizing machinery standstill. The considered motor pumps are composed of one-stage horizontal centrifugal pumps coupled to an AC electric motor. Accelerometers strategically placed at points next to bearings and motors allow the displacement, velocity or acceleration of the machine over time to be measured, thus generating a discrete signal of the vibration level. Fig. 1 shows a typical positioning configuration of accelerometers on the equipment. In general, the orientations of the sensors follow the three main axes of the machine, that is, vertical, horizontal, and axial. Vibration signals were collected by means of a closed, proprietary vibration analyzer equipped with a sensor in the time domain and then were submitted to well known signal processing techniques like Fourier transform, envelope analysis [10] based on the Hilbert transform [11] and median filtering.

Normally all work in literature exhibit their results based on experimental data or even simulated data to corroborate the effectiveness of a given method. When well behaved data from controlled laboratory environments are used, the fault classes are perfectly known permitting a doubtless labeling of the data sample for supervised learning. Machine simulations can assist in several aspects of system operation and control, being useful to do preliminary investigations about the capability of a method, though it cannot completely simulate all real-world situations. There are a number of factors that contribute to the complexity of the failure signature that cannot be simulated. Most industrial machinery contains components which will produce additional noise and vibration whereas a simulated environment is almost free from external vibrations.

A strong motivation of our work is the possibility to investigate the effectiveness of two fault diagnosis methods on statistically significant amounts of real data from operational oil rigs. Measurements were regularly taken during five years from 25 different oil platforms operating along the Brazilian coast. A total amount of 3700 acquisitions was collected. Of this total, 2000 examples had some type of defect attributed by a human operator relying on his practical experience in maintenance engineering. The remainder of the examples represented normal operational conditions. Each acquisition labeled as a fault presents some kind of defect that can be divided into electrical, hydrodynamic, and mechanical failures, and may present several types of defects simultaneously. Normal examples, that is, examples without any defect, were discarded. An example is called "normal" when the level of overall RMS (root mean square) is less than a pre-set threshold. In this way we could distinguish a faulty example from an example in good condition without training a sophisticated classifier, doing only a simple pre-processing.

We consider the fault diagnosis problem as a multi-label classification task [12] in which several labels (fault classes) may be simultaneously assigned to an example. Each fault category is represented by a binary predictor, diagnosing the presence or absence of that fault in an input pattern; the global classification system combines the fault specific decisions in order to completely diagnose a pattern. Also, many faults cause vibrations in similar frequency bands, like the first, the second, and the third harmonics of the shaft rotation frequency, in such a way that the faults cannot be detected by just searching for their well-known characteristic signature.

The fault classes that were considered in this paper are: rolling element bearing failures (problems on ball pass inner raceway, ball pass outer raceway, or on bearing cage); pump blade unbalance; hydrodynamic fault (due to blade pass and
vane pass, cavitation or flow turbulence); shaft misalignment; mechanical looseness; and structural looseness.

III. SUPPORT VECTOR MACHINE

The Support Vector Machine (SVM) is a widely used method for constructing classifiers, and is the one that was chosen to perform the classifications in this work. Early experiments were made using a Multi-Layer Perception (MLP) artificial neural network, however the SVM classification accuracy was just as high with a considerably lower cost, making the repeated training in the process of selecting the classifiers feasible. In our domain we have six different classes to be attributed to the examples. Because of that we first separate the classifiers so each class has its own classifier or ensemble that returns the information positive or negative regarding the presence of that fault on the signal, and in the end all classifications are unified. The SVM classifier works building the feature space for all features used in the training, and then creating a separating hyperplane to split the mapped space into two regions, one for the positive class and one for the negative class. The feature mapping is implicitly done by the use of a kernel [4].

For creating the SVMs we use the libsvm library [15]. Using the structures from this library we create the training nodes with the feature selection done already, and then calibrate the parameters. We used the C-SVM architecture with a radial basis function kernel [4], so we had to determine the values of two specific parameters, $C$ and $\gamma$. For both methods we chose empirically the pair $C = 8.0$ and $\gamma = 0.5$ and used them in all classifiers. Experiments were conducted and we determined that this pair of parameters generally built accurate classifiers. We also used the libsvm to normalize the data before we started selecting features and training classifiers in all the experiments.

The construction of the SVM classifier is done using the data structures available in libsvm to determine the features and calibrate the parameters. To get the prediction of the classifier we can use either cross-validation on the training data set, which can return a reliable classification with only one set of examples, or a simple prediction on a test data set to get results. The second one can be done only if a sufficient number of examples are available for building a test set. The results are a set with one posterior probability [4], calculated based on the distance from the separating hyperplane for each example being tested. This score will mean the probability of a given example to have the given fault. The scores range from 0 to 1, so we consider that values higher than or equal to 0.5 to represent a prediction of the positive class.

IV. GENETIC ENSEMBLE FEATURE SELECTION

The GEFS algorithm is an approach created to search for an appropriate set of feature subsets for ensembles and was first introduced in [8]. At each step, this method produces new candidate classifiers by using the genetic operators of crossover and mutation. Each individual is defined by a vector of integers, where each integer indexes a particular feature in the global feature set and each of these vectors may have a size ranging from one to twice the original number of features. In that way, we can produce feature subsets larger or smaller than the original. Also, one can have a feature subset that is larger than the original, but does not have all of its different features. The rest of the characteristics of the algorithm are very similar to most genetic algorithms, so we will explain it briefly.

A. The GEFS

The algorithm starts by creating a random initial population of features sets, which for our purposes are just variant size vectors of indexes for the original feature set. After that, genetic operations are used to produce new individuals for the population. Then the SVMs are trained using only the selected features of each example from the training set, and the results can be obtained by classifying a test set or performing a cross-validation classification in the training set. For each classifier we measure the accuracy and the diversity. The latter is the average of the difference between the prediction of each classifier and the prediction of the ensemble as a whole. The predictions of the ensemble are obtained with the average of the scores given by all classifiers. These two measurements compose the formula for the fitness of each individual, given below:

$$
\text{fitness} = \text{accuracy} + \lambda \cdot \text{diversity},
$$

where $\lambda$ is an experimental constant with the purpose of regularizing the influence of the diversity. The other three values the algorithm uses as metrics are the ensemble error (EE), the average population error (APE) and the average diversity (AD). EE is the error rate comparing the average of the scores from all classifiers with the real class. APE is calculated getting the error rate from each classifier, and then taking their average. AD is the result of the average of all the diversity values that were used to produce the fitness of each feature set. When EE increases, the algorithm analyses the cause. If APE does not increase and AD decreases, it increases $\lambda$ to force Eq. 1 to give more importance to diversity in the next generation. If AD does not decrease and APE increases, it decreases $\lambda$ to force Eq. 1 to give more importance to accuracy.

After the fitness of each classifier is determined, we select $\max$ classifiers with the greatest values, $\max$ being the predetermined maximal number of classifiers to form the ensemble. That way we always generate new possible good fits for the population, calculate how fit they are, and then remove the ones with the lowest fitness value, always keeping $\max$ individuals at the end. This process is repeated until a stopping criterion is reached.

B. Genetic Operations And Parameters

There are two genetic operations used in this algorithm: cross-over and mutation. The GEFS cross-over works by first choosing two parents from the population, where the probability of being chosen is proportional to the classifier fitness. Then the son is generated randomly picking a number
of features from both parents. In that way, the son can be larger or smaller than any of the the two parents. Mutation is done as usual, given a mutation percentage \( m_p \), randomly replacing \( m_p \cdot |F| \) features in the parents feature set, \( F \) being the original feature set. The parent for the mutation is chosen randomly. The number of new individuals we create in each generation is predetermined, as we define beforehand the mutation and cross-over rates.

In [8] parameters were selected that according to tests worked better, so here, for a truthful reference to the algorithm, we use very similar values. The mutation percentage is set to 0.5 and the total number of classifiers is 20. For the stopping criterion, the reference uses the maximum of 250 classifiers produced in the whole process. Here we chose to increase this number to 1000 to give a better convergence to the algorithm, since we are working with a larger number of features and real databases with patterns difficult to generalize. In each generation the population increases by the cross-over rate \( (0.5) \) and the mutation rate \( (0.5) \), so it gets a 100% increase of size. Then we can select the fittest individuals and reduce the population to its original size again, which in our case is 20. Since our stopping criterion is related to the full size of classifiers produced, the number of classifiers created in each generation does not have a major impact on the results, however a large percentage like 100% was chosen so the algorithm could considerably diversify the parents used to create new individuals.

V. BEST SELECTED FEATURE SUBSETS

The second method showed in this paper is called Best Selected Feature Subsets (BSFS) [5], and uses basically two different greedy search algorithms to perform its selection, the Sequential Forward Selection (SFS) and the Sequential Backward Selection (SBS) [9]. The first one starts with an empty set of features, and then proceeds to include one feature at a time, analyzing the selection criterion resulted by the classifier with the given feature united with the existing selected set. The second one does the opposite process, beginning with a full set and removing in each step one feature that worsens the classification result.

In BSFS we use these two algorithms to produce a large number of different classifiers. Considering \( F \) the complete feature set, we can use SFS to select \( |F| - 1 \) feature sets which sizes range from 1 to \( |F| - 1 \). Repeating the process with SBS, the result is a large set \( X \) that is formed by \( |F| - 1 \) feature sets constructed using SFS, \( |F| - 1 \) feature sets constructed using SBS and the full set \( F \). Once we have \( X \), we can use its sets to build a large aggregation of SVM classifiers we call \( S \). So we use a method called overproduce-and-select [14], in which we perform the same greedy forward searches we used in \( F \), now to select classifiers from a classifier set, and the cross-validation accuracy obtained by the formed ensemble with the training data is used as the criterion for selecting a given classifier. The classifiers are selected until a predetermined maximal size \( N_{max} \) is reached, and these \( N_{max} \) classifiers will compose the final ensemble.

The final results are obtained aggregating all the results from the individual classifiers and calculating the average. So when we use the algorithm to select the classifiers, we can use this average as a metric of quality. In that way, we start off with an empty set of classifiers, and always insert into the set the new classifier that provides the greatest final accuracy when used together with the classifier set as an ensemble.

VI. EXPERIMENTS

The database utilized for our tests is composed of 2000 signal examples extracted from the accelerometers in the motor pumps. This set was divided in 5 different pairs of databases, each pair being formed by one training database and one testing database. The tests were conducted separately with each fault class, so each of these had its own accuracy in the end. The process for combining the results of this experiments is called \( 5 \times 2 \) cross-validation, and consists of testing the algorithm five different times with two different sets of examples each time, the training and testing database, and then repeating the process five more times, using the former testing databases as training databases and the former training databases as testing databases. The final accuracy associated with each given fault class is the average of these 10 obtained accuracies.

A. Global Set of Features

Initially a large set of features had to be determined so we could select the optimized sets of features from it. These features were chosen based on specific knowledge on the area of vibration signal analysis and assembled in a total group of 95 features. They were extracted from the Envelope spectrum and Fourier spectrums of the signals, at all 4 positions (1, 2, 3 and 4) and all 3 directions (horizontal, vertical and axial), as it shows in Fig. 1. An example of this type of features would be the RMS value of the 0.9x-1.1x taken from the frequency spectrum of the velocity signal. If there is a peak around the 1x frequency, this feature would capture it.

The positions from which the features will be taken depends on the fault class being analyzed: position 3 or 4 are used in unbalance and hydrodynamic faults, always choosing the position with the highest global RMS value to avoid null feature values due to nonexistent signals; positions 2 or 3, the ones close to the coupling, are used for misalignment; for structural looseness and mechanical looseness all positions are considered, in the way that if the fault is found in either the motor or the pump, the machine will be considered to have it; the rolling fault information is also taken from all positions, but for this one we use the Envelope spectrum to analyze the bearing characteristic frequencies BPFI, BPFO, FTF and BSF [10].

B. Experiments With BSFS

For each fault we have to create a different ensemble because the classification result is binary. We do one separate test with each of the 10 training-test sets of databases and in the end unify the results as the average. In each of these
tests the algorithm creates a large number of feature sets, performing the selection using SFS and SBS. The selection criterion for the selection of features was the area under the curve (AUC) of receiver operating characteristics (ROC) graphs [15] resulted from the 10-fold cross-validation done by the SVM on the training data [5]. The advantage of AUC compared to accuracy is enabling the measurement of the classifier performance without regard to class distribution or error costs.

Having created all these groups of features that diversify by the number of features and the algorithm used to select them, we build the classifiers. Once we have created all the classifiers, we can perform the selection of classifiers that will provide the final ensemble. Using SFS to insert classifiers into the ensemble, we keep including classifiers until it reaches a given size, in our case 15. The selection criterion to select the classifiers is the cross-validation accuracy of the formed ensemble. Thus, every time we will insert a new classifier in the ensemble, we chose the one that brings the best performance together with the classifiers already selected, starting from an empty set.

C. Experiments With GEFS

As with BSFS, each fault is associated to an appropriate ensemble to identify it. Also similar to the previous algorithm, we test it with the 10 databases for training and testing, and the final result is the average of the accuracies. We start with a initial population of 20 feature sets, represented by vectors of indexes that are associated with the position of the features in the original global feature set. The size of the vectors ranges from one to twice the original feature set size, which in our case defines a dimension of 190.

The algorithm performs cross-over and mutation as already discussed to add 10 more classifiers to the set, respecting the declared rate of new classifiers for generation. All the classifiers are scored with the appropriate fitness defined in Eq. 1, calculating the accuracy by performing a 10-fold cross-validation in the training set. After that, the 20 classifiers with the biggest fitness values are selected to form the ensemble. The algorithm adjusts $\lambda$, and then continues to the next generation. This is repeated until 1000 classifiers are produced, and finally, the current set of 20 classifiers forms the final ensemble. Having this final classifier set built, we can use these specified feature sets to train one final ensemble and classify the test data set. In the end we average the 10 classification results that came from the 10 different training-test pairs and return one last result, completing the 5x2 cross-validation process.

D. Results and Discussion

Table I contains the a priori fault class distribution of the examples. The percentages represent the portion of the data base that has the given fault. As each example may present many faults simultaneously, the sum of the values does not equal 100%. The final results are presented in Table II. It shows the the accuracy given a single SVM with a optimized feature set, the results obtained from GEFS and finally the accuracy of the BSFS ensembles. The use of ensembles exhibit a clear improvement over single classifiers [5]. Between the ensemble methods, we noticed two very similar results when BSFS presented its higher results, and a slight superiority of GEFS in the other faults.

Two reasons can be given. The first one is related to the way each method store the features. Because GEFS can repeat features in one set, the use of weights for individual features in feature sets can be considered a good approach for our problem, and could partially explain the high accuracy. The second aspect was the way each method select classifiers to form the ensemble. BSFS just uses a greedy method choosing the next classifier that will maximize the cross-validation accuracy. This does not guarantee that it will get the most diverse set of accurate classifiers possible, which would be the ideal goal. On the other hand, GEFS uses diversity of classification to score its classifiers in each generation. Because of that, it can evolve to a very diverse and individually accurate set in the end, resulting in a very high final accuracy. The other important aspect about these two algorithms is the computational cost. The fact that the higher cost of BSFS is related to the selection of features makes the algorithm have a lower execution time, and still with very satisfying results. GEFS has the cost of training 1000 SVMs and using them to classify the test database 1000 times. This makes GEFS a much more computationally expensive algorithm, justifying also the higher accuracy, as it takes much more time to compute the final results.

<table>
<thead>
<tr>
<th>Fault class</th>
<th>A priori class distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Misalignment</td>
<td>57.4%</td>
</tr>
<tr>
<td>Bearing</td>
<td>64.3%</td>
</tr>
<tr>
<td>Unbalance</td>
<td>75.1%</td>
</tr>
<tr>
<td>Hydrodynamic</td>
<td>57.6%</td>
</tr>
<tr>
<td>Structural looseness</td>
<td>78.8%</td>
</tr>
<tr>
<td>Mechanical looseness</td>
<td>88.0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Considered fault classifier</th>
<th>Single SVM with optimized feature set</th>
<th>Ensemble constructed with GEFS</th>
<th>Ensemble constructed with BSFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Misalignment</td>
<td>76.5%</td>
<td>78.9%</td>
<td>77.8%</td>
</tr>
<tr>
<td>Bearing</td>
<td>86.1%</td>
<td>86.6%</td>
<td>86.7%</td>
</tr>
<tr>
<td>Unbalance</td>
<td>82.2%</td>
<td>83.3%</td>
<td>82.4%</td>
</tr>
<tr>
<td>Hydrodynamic</td>
<td>84.1%</td>
<td>86.0%</td>
<td>85.9%</td>
</tr>
<tr>
<td>Structural looseness</td>
<td>81.5%</td>
<td>87.7%</td>
<td>87.1%</td>
</tr>
<tr>
<td>Mechanical looseness</td>
<td>90.1%</td>
<td>93.7%</td>
<td>93.0%</td>
</tr>
</tbody>
</table>
VI. CONCLUSION AND FUTURE WORK

In this paper we primarily reinforce the notion of effectiveness of the ensemble over the single classifier method, comparing results obtained with a real-world data of 2000 different examples from a single classifier SVM and two feature based ensemble building approaches. We compared results with the proposed feature selection method for ensembles (BSFS) with the results given by our implementation of a widely recognized method based on the genetic algorithm theory (GEFS). We discovered that BSFS gave very satisfactory results, showing a classification accuracy almost as high, with a much lower computational cost, proving it to be a usable approach that can generate many other studies. Our experiments also showed that the ensembles can be applied to our problem of fault diagnosis in vibration signal with good results.

Although the difference in accuracy is small, GEFS did present higher results in almost all classes. We suggested possible reasons for this slight superiority discussing the methods both algorithms use to represent the features and the direct use of the diversity measure by GEFS to build the feature sets.

As future work, a very interesting idea is to use what we think may be the cause of the GEFS qualities to tune BSFS to try to make it a more effective method, keeping its low cost as much as possible. BSFS can be modified to consider diversity of the classification scores from each classifier as we form the classifier set, and not just the final accuracy of the ensemble. Another natural continuity from this work would be to use BSFS to create an effective ensemble as quick as possible, and use this result as a initial population for GEFS, so it can try to evolve to a better ensemble. For these changes would have to be made in the stopping criterion of the genetic algorithm, so it does not build always 1000 classifiers, but stops when it evolves to a good solution. Basically we would be using BSFS create a initial population, thus making GEFS converge faster. We believe the fusion of the two approaches would be a good idea because, as both algorithms succeeded in classification and have different strong points, it could be advantageous to merge these strong points resulting in one more robust process.

Another promising experiment involve changing certain general aspects of both algorithms to improve their results. For example, our algorithms construct the SVM classifiers by just varying the feature sets. We could implant in both algorithms a parameter tuning, so it is possible to diversify the classifiers by SVM parameter also. Another approach is to test other ways to combine the classifiers in the end. In this work, both algorithms use the average of scores but other approaches may represent the problem more accurately.

REFERENCES
