Estefhan Dazzi Wandekokem

Support Vector Machine Ensemble Based on Feature and Hyperparameter Variation

Vitória - ES, Brasil

23 de fevereiro de 2011

Estefhan Dazzi Wandekokem

Support Vector Machine Ensemble Based on Feature and Hyperparameter Variation

Dissertação apresentada para obtenção do Grau de Mestre em Informática pela Universidade Federal do Espírito Santo.

Orientador:
Dr. Thomas W. Rauber

Co-orientador:
Dr. Flávio M. Varejão

DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória - ES, Brasil

23 de fevereiro de 2011

Dissertação de Mestrado sob o título *"Support Vector Machine Ensemble Based on Feature and Hyperparameter Variation"*, defendida por Estefhan Dazzi Wandekokem e aprovada em 23 de fevereiro de 2011, em Vitória, Estado do Espírito Santo, pela banca examinadora constituída pelos professores:

Dr. Thomas W. Rauber
Orientador

Dr. Flávio M. Varejão
Co-orientador

Dr. Renato Krohling
Universidade Federal do Espírito Santo

Dr. Roberto M. Cesar, Jr.
Universidade de São Paulo

Resumo

Classificadores do tipo máquina de vetores de suporte (SVM) são atualmente considerados uma das técnicas mais poderosas para se resolver problemas de classificação com duas classes. Para aumentar o desempenho alcançado por classificadores SVM individuais, uma abordagem bem estabelecida é usar uma combinação de SVMs, a qual corresponde a um conjunto de classificadores SVMs que são, simultaneamente, individualmente precisos e coletivamente divergentes em suas decisões. Este trabalho propõe uma abordagem para se criar combinações de SVMs, baseada em um processo de três estágios. Inicialmente, são usadas execuções complementares de uma busca baseada em algoritmos genéticos (GEFS), com o objetivo de investigar globalmente o espaço de características para definir um conjunto de subconjuntos de características. Em seguida, para cada um desses subconjuntos de características definidos, uma SVM que usa parâmetros otimizados é construída. Por fim, é empregada uma busca local com o objetivo de selecionar um subconjunto otimizado dessas SVMs, e assim formar a combinação de SVMs que é finalmente produzida. Os experimentos foram realizados num contexto de detecção de defeitos em máquinas industriais. Foram usados 2000 exemplos de sinais de vibração de moto bombas instaladas em plataformas de petróleo. Os experimentos realizados mostram que o método proposto para se criar combinação de SVMs apresentou um desempenho superior em comparação a outras abordagens de classificação bem estabelecidas.

Abstract

The support vector machine (SVM) classifier is currently considered one of the most powerful pattern recognition based techniques for solving binary classification problems. To further increase the accuracy of an individual SVM, a well-established approach relies on using a SVM ensemble, which is a set of accurate, divergent SVMs. In this work we investigate composing an ensemble with SVMs that differ among themselves on the feature subset and also the hyperparameter value they use. We propose a three-stage method for building an SVM ensemble. First we use complementary Genetic Ensemble Feature Selection (GEFS) searches to globally investigate the feature space, aiming to produce a set of diverse feature subsets. Further, for each produced feature subset we build a SVM with tuned hyperparameters. Finally, we employ a local search to retain an optimized, reduced set of these SVMs to ultimately comprise the ensemble. Our experiments were performed in a context of real-world industrial machine fault diagnosis. We use 2000 examples of vibration signals obtained from motor pumps installed on oil platforms. The performed experiments show that the proposed SVM ensemble method achieved superior results in comparison to other well-established classification approaches.

Agradecimentos

Meus sinceros agradecimentos a todos os que colaboraram, direta ou indiretamente, à realização deste trabalho.

À Petrobrás S.A. pelo apoio financeiro e pela disponibilização da base de dados utilizada. À CAPES, pelo apoio financeiro.

Aos meus orientadores Thomas Rauber e Flávio Varejão, cujo conhecimento e entusiasmo foram fundamentais para o sucesso deste trabalho.

Aos professores Roberto Cesar e Renato Krohling, membros da Banca Examinadora, que gentilmente aceitaram dispor de seu tempo e conhecimento.

Aos colegas do laboratório NINFA, em especial Mendel, Fábio e Marcelo, pelas contribuições, ajudas, conselhos e amizade. Também, aos meus amigos em geral, da engenharia e de fora dela.

À minha família, pelo amor e apoio.

À Adriana, pelo amor e melhor companhia.

Contents

List of Figures

List of Tables

1 Introduction

"The machine does not isolate man from
the great problems of nature but plunges
him more deeply into them."
- Antoine de Saint-Exupéry, Wind, Sand, and Stars, 1939.

This chapter presents the objective and structure of this work.

Section 1.1 introduces support vector machine classifiers, classifier ensembles, and the machine fault diagnosis problem. Section 1.2 presents the further structure of this work.

1.1 Introduction

Dichotomizers (i.e. two-class classifiers) are used in many important applications, such as automated diagnosis, fraud detection, currency verification and document retrieval. In order to achieve a high discriminative power, a well established approach relies on using a *classifier ensemble* [Kuncheva 2004] [Wandekokem et al. 2011] to take classification decisions. An ensemble is a set of accurate classifiers that disagree among themselves as much as possible. Several works have showed that employing an adequate ensemble provides a higher classification accuracy than employing a single accurate classifier.

The *support vector machine* (SVM) [Vapnik 1998] classifier is currently considered one of the most powerful machine learning techniques for solving two-class classification problems. The classification hypothesis limit of a SVM corresponds to the hyperplane providing the maximum separation margin between the two classes, constructed in a high-dimensional transformed feature space defined implicitly by a *kernel* mapping [Miller et al. 2001].

The kernel function used by a SVM estimates the similarity between two patterns $\mathbf{x}$ and $\mathbf{y}$. We employ the widely adopted radial basis function (RBF) kernel $k(\mathbf{x},\mathbf{y}) = \exp(-\gamma||\mathbf{x}-\mathbf{y}||^2)$. It is critical to consider that the performance of a SVM strictly depends on its hyperparameters, and choosing an adequate hyperparameter value depends on its turn on the used feature subset. For instance, for a SVM using the RBF kernel, even a slight variation of the used feature subset (i.e. the set of features composing $\mathbf{x}$ and $\mathbf{y}$) or a slight variation of the kernel parameter γ alter the values computed by $k(\mathbf{x},\mathbf{y})$, therefore changing the transformed feature space in which the SVM discriminant hyperplane is defined.

Even thought the SVM is currently a very popular classification technique, by now few works have studied SVM ensembles, and most of them have focused on the traditional approach based on employing, for each classifier, a resampled training data set [Li, Wang e Sung 2008, Hu et al. 2007, Bertoni, Folgieri e Valentini 2005, Kim et al. 2003]. But considering that the SVM is a stable classifier, in the sense that a small variation of the training data causes only a small variation of the SVM decision function, we argue that a more natural and powerful approach to generate diversity in a SVM ensemble should take advantage of the high sensitivity of the SVM discriminant function to a variation of the employed feature subset and hyperparameter value.

The proposed SVM ensemble method is based on a three-stage process. First, we use different Genetic Algorithm (GA) searches to globally investigate the space of feature subsets, with each GA search using a fixed, different hyperparameter value to build SVMs to estimate

the quality of the feature subsets. Using these complementary GA searches allows many accurate feature subsets to be found, which are also divergent since they were investigated in feature spaces defined by different kernel mappings. In the second stage, for each produced feature subset we build a SVM, which uses tuned hyperparameters aiming to achieve a better classification performance. The use of different hyperparameter values increases the collective diversity of SVMs. Finally, in the third stage, we employ a local search aiming to retain an optimized, reduced set of these produced SVMs to ultimately compose the ensemble.

Our experiments were performed in the context of fault detection and diagnosis of industrial machines [Widodo e Yang 2007]. We used data from real-world operating industrial machines instead of using data from a controlled laboratory environment which is almost always found in the literature (see for instance [Zio, Baraldi e Gola 2008], [Hu et al. 2007]). From the engineering point of view, that is an important novelty of our research, since laboratory hardware in general cannot realistically represent real-world fault occurrences. We work with 2000 examples of vibration signals obtained from operating partially faulty motor pumps, installed on 25 oil platforms off the Brazilian coast; the signals were obtained during a period of five years. To generate the labeled training data, experts in maintenance engineering provided a label for every fault present in each acquired example.

In the diagnosis of an input pattern $\mathbf{x}$ (which represents the acquired signals of a motor pump), each considered fault is detected by an independent SVM ensemble, with the SVMs in an ensemble considering $\mathbf{x}$ as belonging to the positive class ω_{pos} if $\mathbf{x}$ presents the fault considered by this ensemble, or as belonging to the negative class ω_{pos} if $\mathbf{x}$ does not present this fault (although $\mathbf{x}$ may present other faults).

1.2 Structure of this Work

The chapters of this work are structured as follows. Chapter 1 introduces classifier ensembles and support vector machine classifiers. Chapter 2 is concerned with classifier ensembles in general. Chapter 3 considers the specificities of SVM classifiers and the use of SVMs as component classifiers in ensembles. Chapter 4 outlines the proposed method for building SVM ensembles, based on feature and hyperparameter variation. Chapter 5 presents the motor pump equipment, the considered faults, and the extracted features. Chapter 6 shows the experimental results achieved by the studied classification models using the acquired database of motor pump vibration signals. Finally, chapter 7 draws conclusions and points out to future research.

2 *Classifier Ensembles*

"Vox populi, vox Dei."

This chapter is concerned with classifier ensembles in general.

Section 2.1 discusses why an ensemble should be composed of accurate, divergent classifiers, in order to achieve a high prediction accuracy. Section 2.2 presents a method for combining decisions of different classifiers into a single classification decision. Section 2.3 is concerned with approaches for generating a set of divergent classifiers in order to compose an ensemble.

2.1 Ensemble of Classifiers

To achieve a high classification accuracy, a well-established approach relies on combining decisions from complementary, divergent classifiers, instead of employing just a single, fixed classifier. In this context, divergence means that each classifier gives wrong prediction in a different region of the *global* feature space (obtained by considering every available feature). Thus divergent classifiers make errors for different testing patterns.

Figure 2.1 shows the global feature space region in which classifiers C_1, C_2 and C_3 give wrong predictions, respectively R_1, R_2 and R_3. The shaded area is the region in which the ensemble composed of the three classifiers, using majority vote, gives a wrong prediction. As one can see, the error region of the ensemble is smaller than the error region of any individual classifier. In the example presented in this figure, both classifiers C_1 and C_3 give a correct decision for the testing pattern $\mathbf{x}_2$, thus $\mathbf{x}_2$ is correctly classified by the ensemble, even with the classifier C_2 giving a wrong prediction for $\mathbf{x}_2$. However testing pattern $\mathbf{x}_1$ is incorrectly classified by the ensemble, since both C_1 and C_3 give a wrong decision for this pattern.

The general motivation behind the use of classifier ensembles is reflected in figure 2.1. If the component classifiers *diverge* on their predictions (i.e. if each classifier corresponds to a different error region in the global feature space), then, for some testing patterns, the wrong decision given by some classifiers can be corrected by the right decision given by others. Besides, if the component classifiers are *accurate* (i.e. if each classifier corresponds to a small region of wrong decision in the global feature space), then the ensemble composed of them might correspond to an even smaller region of wrong decisions.

Creating a classifier ensemble entails addressing two issues: how to generate a set of divergent classifiers to compose the ensemble; and how to aggregate decisions from these distinctly trained classifiers into a single, combined decision.

2.2 Combining Decisions from Distinct Classifiers

A widely employed method for combining decisions from distinct classifiers is the majority vote. In this approach, each classifier in the ensemble assigns a testing pattern $\mathbf{x}$ to one class; then the ensemble ultimately assigns $\mathbf{x}$ to the class indicated by most classifiers. Majority vote can be naturally employed with classifiers that only provide the predicted class, for instance the K-Nearest Neighbors classifier.

Considering SVM classifiers, the discriminant function computed by a SVM is a real-valued

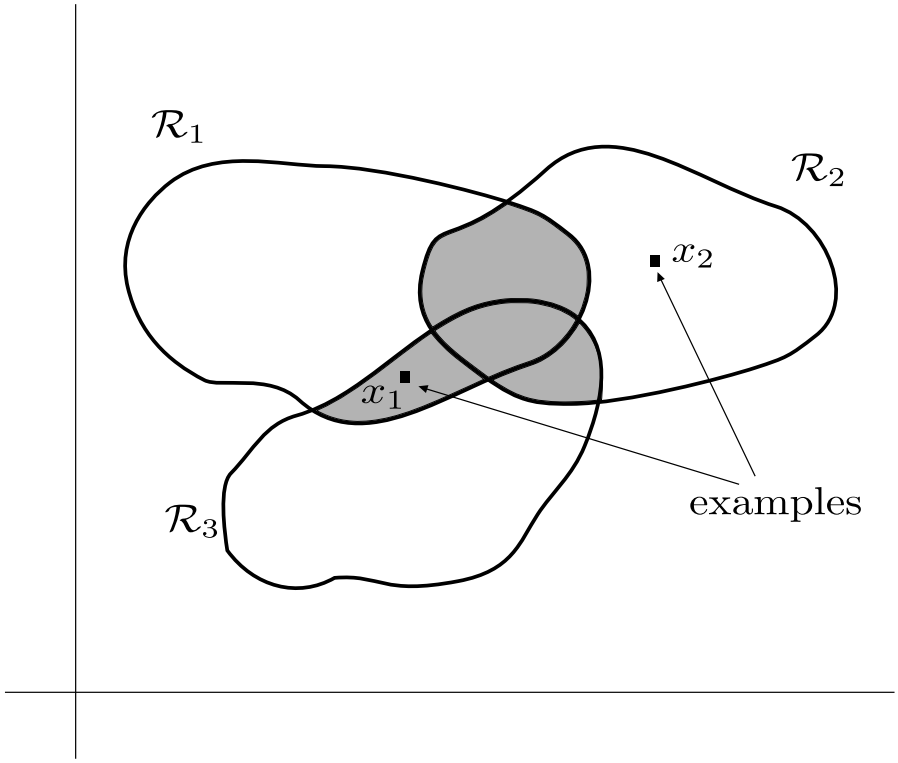

Figure 2.1: The region of wrong decision of an ensemble (the shaded area) is smaller than the region of wrong decision of any individual component classifiers (the regions R_1, R_2 and R_3).

function that corresponds to the degree of support of belonging to a class, which is more informative than just providing the predicted class. As it is more convenient to use degrees of support in the interval $[0,1]$ (with 0 meaning "no support" and 1 meaning "full support"), we use a logistic discrimination [Theodoridis e Koutroumbas 2006] to estimate the a posteriori probability $\hat{P}_{\text{pos}}(\mathbf{x})$ that a pattern $\mathbf{x}$ belongs to the positive class ω_{pos}.

An advantage of using classifiers that produce a degree of support is that it allows taking into account their certainty of decision. For that aim, we use the *averaging* method to combine the decisions of the individual classifiers. In this approach, an ensemble $\mathscr{E}$ estimates the probability $\hat{P}^{\mathscr{E}}_{\text{pos}}(\mathbf{x})$ of an input pattern $\mathbf{x}$ belonging to the positive class ω_{pos} as the average of the $\hat{P}^{c_m}_{\text{pos}}(\mathbf{x})$ score value that the $|\mathscr{E}|$ classifiers c_m in $\mathscr{E}$ produce for $\mathbf{x}$,

$$\hat{P}^{\mathscr{E}}_{\text{pos}}(\mathbf{x}) = \frac{1}{|\mathscr{E}|}\sum_{m=1}^{|\mathscr{E}|}\hat{P}^{c_m}_{\text{pos}}(\mathbf{x}). \tag{2.1}$$

Thus $\mathbf{x}$ is predicted as belonging to ω_{pos} if $\hat{P}^{\mathscr{E}}_{\text{pos}}(\mathbf{x}) > 0.5$ or as belonging to ω_{neg} otherwise.

2.3 Generating Divergent Classifiers

A classifier takes decisions according to its hypothesis, defined by the training of this classifier. Before being trained, a classifier has a set of hypotheses that are accessible to it, according to the available training data and the used classifier architecture. The classifier training algorithm then starts an a point in the hypothesis space, traverses through this space and stops in one of the accessible hypothesis.

Ensemble methods can be grouped according to how they guarantee that component classifiers use different hypotheses [Brown et al. 2005]. We make a distinction between two general groups: methods based on training each classifier with the use of the same set of accessible hypotheses; and methods based on training each classifier with the use of a different set of accessible hypotheses.

The approach based on employing the same set of accessible hypotheses relies on starting the training of each classifier in a different point in the hypothesis space, or employing, for each classifier, a different approach for traversing the space of possible hypotheses. For instance, [Opitz e Maclin 1999] built a neural network ensemble by training each network using different random initial weights, and [Brown et al. 2005] used a penalty term in the error function of a neural network ensemble to encourage some overfitting in the individual networks to occur.

The second general ensemble approach relies on training each classifier with the use of a different set of accessible hypotheses. One can vary three things among classifiers: architecture (classifier model and value of intrinsic parameters); training patterns; and the feature subset.

A natural approach to create diverse classifiers is based on setting the intrinsic parameters of each classifier to a different value. For instance, [Islam, Yao e Murase 2003] investigated ensembles of neural networks with each classifier using a different, fixed number of neurons in its hidden layer.

Probably the most studied ensemble method is based on employing a different training data set for each classifier. For instance, in Bagging [Breiman 1996], each classifier samples N training patterns, with equal probability and with replacement, from an available set of N different examples; thus a training set might not contain some of the available patterns while it contains other repeated patterns. The AdaBoost method [Freund e Schapire 1996] is a variation of Bagging, in which an iterative process is employed to progressively increase the probability of sampling difficult patterns. The ensemble methods based on resampling the training data work well with the use of unstable classifiers, for instance neural networks, in which a small variation of the training data set might cause a large variation of the classifier discriminant

function [Kuncheva 2004].

Another useful approach for building ensembles is based on using a different feature subset for each classifier [Zio, Baraldi e Gola 2008] [Wandekokem et al. 2011]. Indeed, [Ho 1998] showed that even randomly sampling the features used by each component classifier is effective for producing an ensemble. Other works have investigated approaches for searching the space of feature subsets, aiming to define more accurate ensembles. A well-established method is the *Genetic Ensemble Feature Selection* (GEFS) proposed by Opitz [Opitz 1999], that relies on a Genetic Algorithm (GA) based global search. Using neural networks as component classifiers, Opitz showed that ensembles built by GEFS achieved better performance than ensembles built by Bagging or AdaBoost. Several works have employed GEFS for comparing results; in fact, previous work shows that the GEFS method usually achieves a higher prediction accuracy in comparison to other ensembles methods [Tsymbal, Pechenizkiy e Cunningham 2005].

3 Support Vector Machine Ensembles

"If in other sciences we should arrive at
certainty without doubt and truth without error,
it behooves us to place the foundations of knowledge
in mathematics."
- Roger Bacon.

This chapter is concerned with the specificities of support vector machine (SVM) classifiers and the use of SVMs as component classifiers in ensembles.

Section 3.1 presents the SVM classification architecture. Section 3.2 details previous work on SVM ensemble construction.

3.1 The Support Vector Machine Classifier

The SVM discriminant function corresponds to the hyperplane that provides the maximum-margin separation between patterns belonging to the two considered classes. To deal with non-linearly separable problems, a kernel function is used, which implicitly performs a non-linear mapping of the input feature space into a high-dimensional transformed feature space in which the separating hyperplane can be defined.

We use the widely employed radial basis function (RBF) kernel $k(\mathbf{x},\mathbf{y}) = \exp(-\gamma||\mathbf{x}-\mathbf{y}||^2)$. As the computed value $k(\mathbf{x},\mathbf{y})$ estimates the similarity between patterns $\mathbf{x}$ and $\mathbf{y}$ in the transformed feature space, the kernel parameter γ controls decisively the non-linear mapping from the input feature space. Using a high γ causes distance between patterns to be increased, thus employing a very high γ may cause overfitting. On the other hand, using a low γ causes distance between patterns to be decreased, thus employing a very low γ may cause underfitting.

During the training of a SVM classifier it is possible to allow some training patterns to be misclassified. That is controlled by a regularization parameter C which determines the cost of allowing a training pattern to remain in the wrong side of the separating hyperplane. A very high value for C determines a very high cost for misclassification, producing a complex discriminant function that may overfit the training data. On the other hand, if C is set to a very low value, the SVM may not be able to learn an effective discriminant rule, since too many training patterns are allowed to be misclassified.

After training, the SVM discriminant function which defines the side of the hyperplane of an unknown pattern $\mathbf{x}$ is $\text{sgn}(g(\mathbf{x}))$, with

$$g(\mathbf{x}) = \sum_{k=1}^{N_s} \lambda_k t_k k(\mathbf{x},\mathbf{x}_k) + w_0 \tag{3.1}$$

being the unnormalized distance of the pattern $\mathbf{x}$ from the maximum-margin separating hyperplane defined by the SVM training. The vectors $\mathbf{x}_k$ are the support vectors (the training patterns that are ideally closest to the decision boundary); t_k are the class labels of each $\mathbf{x}_k$ (1 for the positive class, -1 for the negative class); and λ_k are the Lagrange multipliers obtained from the convex quadratic optimization problem [Tu et al. 2007] formulated by the SVM approach (thus λ_k is a linear weight corresponding to the relevance of $\mathbf{x}_k$), and $k(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ is a kernel function that calculates the inner product of two patterns $\mathbf{x},\mathbf{y}$ implicitly mapped from the original feature space to the usually nonlinear mapped space by the implicit feature extraction

function ϕ. We employ the radial basis function (RBF) kernel

$$k(\mathbf{x},\mathbf{y}) = \exp(-\gamma||\mathbf{x}-\mathbf{y}||^2). \tag{3.2}$$

The distance of a pattern to the separating hyperplane $g(\mathbf{x})$, followed by a logistic discrimination [Theodoridis e Koutroumbas 2006], is used to estimate the a posteriori probability $\hat{P}_{\text{pos}}(\mathbf{x})$ that a pattern $\mathbf{x}$ belongs to the positive class ω_{pos}:

$$\hat{P}_{\text{pos}}(\mathbf{x}) = \frac{1}{1+\exp(A+Bg(\mathbf{x}))}. \tag{3.3}$$

The parameters A and B in (3.3) are determined after training the SVM, by minimizing a cross-entropy error on the training set [Bishop 2007].

We use the `libsvm` library [Chen, Lin e Schölkopf 2005] to implement SVM classification. This provides C++ code to implement tasks such as scaling input features to a range of $[-1,1]$ (which is more adequate for SVMs), training and evaluating SVMs, and hyperparameter tuning.

3.2 Previous Work on Support Vector Machine Ensemble Methods

Since the SVM training algorithm investigates the space of accessible hypotheses and then finds the global best solution, a natural approach to create a SVM ensemble relies on training each classifier with the use of a different set of accessible hypotheses. In this case, one can vary three things among SVMs: training patterns, its architecture (i.e. employed hyperparameter values and the kernel function), and the feature subset.

Although some works reported success in building SVM ensembles by using traditional data resampling methods such as Bagging or AdaBoost [Hu et al. 2007] [Kim et al. 2003], other works did not, for instance [Evgeniou, Pontil e Elisseeff 2002] which stated that single SVMs with tuned hyperparameters had performed as well as SVM ensembles defined by Bagging. As a matter of fact, building SVM ensembles by employing training data resampling may seem like going against the SVM principle, since the SVM is a stable classifier i.e. a small variation of the training data might cause only a small variation of the SVM discriminant function.

To better adapt the AdaBoost method to SVMs, [Li, Wang e Sung 2008] proposed varying the value of the kernel parameter γ as the AdaBoost iteration proceeds, starting with low γ values (implying weak learning) and then increasing γ progressively. This process generates SVMs that differ on training data and also on hyperparameter values. The authors reported success in

problems with unbalanced classes, as AdaBoost focuses on selecting difficult patterns that tend to belong to the less frequent class. Other works, taking advantage of the high influence of γ in the definition of the SVM discriminant function, have employed SVM ensembles with component SVMs differing among themselves solely on the value of γ [Sun, Zhang e Wang 2007], [Valentini e Dietterich 2000].

Another approach for building SVM ensembles is based on using different feature subsets for generating diversity among SVMs. For instance, [Bertoni, Folgieri e Valentini 2005] stated that, in a classification task with many available features and with few training patterns, SVM ensembles using randomly defined feature subsets performed better than single SVMs with an optimized feature subset defined by feature selection [Kudo e Sklansky 2000].

Reference [Verikas et al. 2010] considered SVM ensembles with component SVMs differing on feature subset and also hyperparameter values. They used a GA method which performs feature selection and hyperparameter tuning to produce an accurate single SVM. This GA method was used to initially build a SVM having access to all the available features during training (thus being able to select an optimized feature subset). Further, this GA search was used to independently build each other component SVM, one by one, but using as features available to be selected just a randomly defined subset of all the initially available features.

4 SVM Ensemble Based on Feature and Hyperparameter Variation

"A designer knows he has achieved perfection not
when there is nothing left to add, but
when there is nothing left to take away."
- Antoine de Saint-Exupéry.

This chapter outlines the proposed method for building support vector machine (SVM) ensembles, based on feature and hyperparameter variation.

Section 4.1 introduces the proposed SVM ensemble construction method, based on a three-stage approach. The first stage, described in section 4.2, builds a set of feature subsets. The second stage, presented in section 4.3, builds, for each previously defined feature subset, a SVM with tuned hyperparameters; these SVMs are candidates to compose the ensemble. The third stage, described in section 4.4, selects a subset of SVMs from all these produced SVMs, to ultimately comprise the SVM ensemble.

4.1 Three-stage Approach to Build SVM Ensemble

The traditional approach to build a SVM based predictor relies on determining a single accurate SVM. Under that perspective, first, the space of feature subsets is investigated to find *one* accurate feature subset; this process is denoted as feature selection [Kudo e Sklansky 2000]. Further, the space of hyperparameters is investigated to find a hyperparameter value providing an accurate SVM using that feature subset; this process is denoted as hyperparameter tuning [Widodo e Yang 2007].

In this work, we propose an adaptation of that traditional single-SVM based approach to the modern perspective of classifier ensembles. Our SVM ensemble method first employs a global search to investigate the space of feature subsets, aiming to find a *set* of feature subsets corresponding to accurate, divergent classifiers. Further, for each produced feature subset, the methods builds a SVM with tuned hyperparameters. Finally, to increase the ensemble accuracy besides reducing the number of component SVMs, the method uses a local search to determine an optimized, reduced SVM subset to compose the final ensemble.

The proposed ensemble method is based on a three-stage process. First it produces a set $\mathcal{F}$ of feature subsets. Then for each feature subset in the set $\mathcal{F}$ the method builds a SVM with tuned hyperparameters, which generates a set $\mathcal{H}$ composed of $|\mathcal{H}| = |\mathcal{F}|$ SVMs, each of which associated to a feature subset and to a hyperparameter value. Finally, the method selects a subset $\mathscr{E}$ of SVMs from $\mathcal{H}$ to form the final ensemble, composed of $|\mathscr{E}|$ SVMs. Figure 4.1 presents a diagram of the proposed SVM ensemble method.

4.2 First stage: Feature Variation

The objective of the first stage is generating diversity by using feature subsets that allow complementary classification decisions to emerge. We achieve this by producing a set $\mathcal{F}$ of diverse feature subsets. Since searching the space of feature subsets is a NP-hard problem, we rely on a suboptimal search strategy, namely the well-established GEFS [Opitz 1999] method.

GEFS originally employed neural networks as component classifiers. To better adapt GEFS using component classifiers that are very sensitive to the definition of their parameters (such as SVMs), in this work we propose a *multiple-GEFS* approach to search the space of feature subsets more profoundly, by evolving independent ensembles. Each of the feature subsets represents one SVM classifier and uses a different, fixed hyperparameter value.

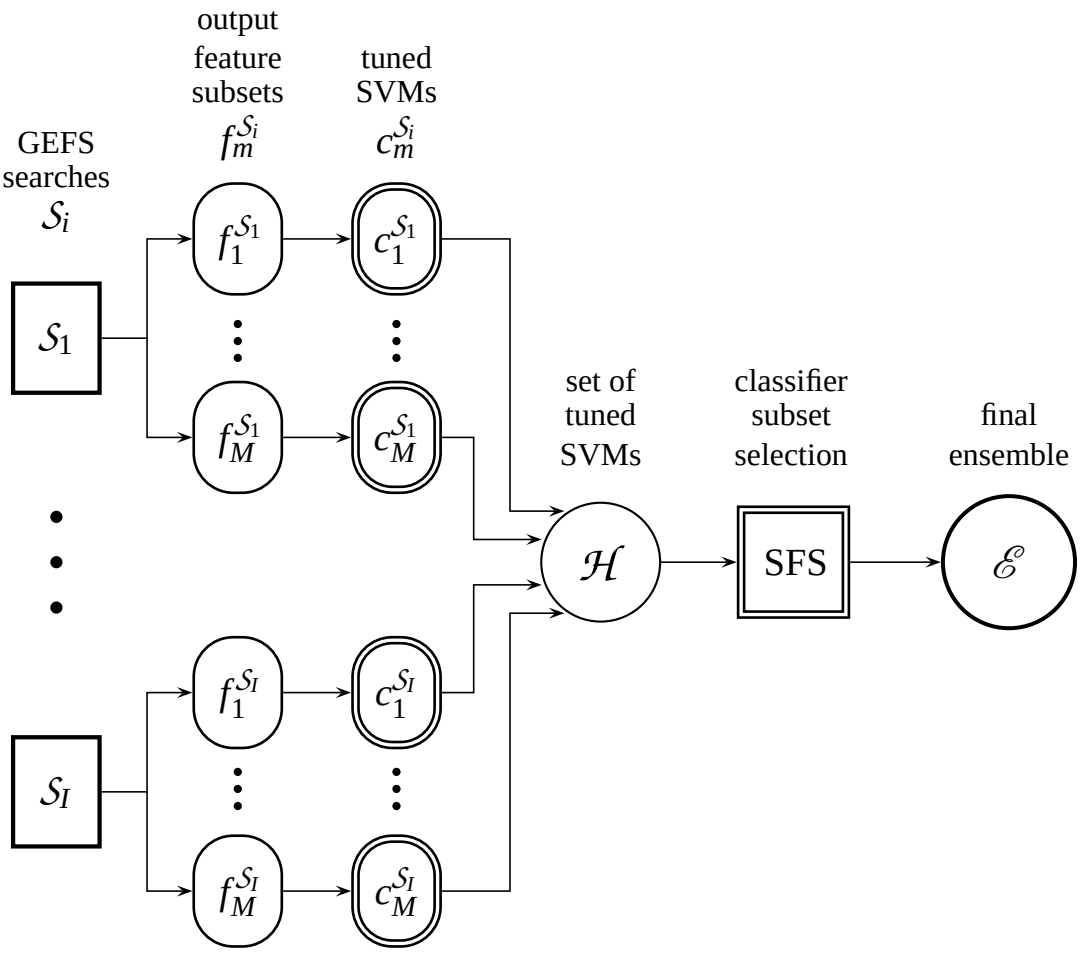

Figure 4.1: Construction of an ensemble $\mathscr{E}$ by the proposed Multiple-GEFS ensemble method.

4.2.1 The GEFS Method

Opitz [Opitz 1999] proposed the Genetic Ensemble Feature Selection (GEFS) that relies on a Genetic Algorithm (GA) global search to investigate the space of feature subsets. Since the efficacy of a feature subset to learning depends on the learning algorithm itself, GEFS relies on the wrapper approach which directly estimates the quality of a feature subset by using k-fold cross-validation to evaluate a classifier employing this feature subset. GEFS considers that a member of the population represents one feature subset, implemented as a vector storing the index of each component feature.

The parameter M determines the number of feature subsets composing the ensemble at the end of every generation. Considering that we have D globally available features, the size of each initial feature subset is randomly defined from 1 to $2 \times D$, with features being sampled with replacement; repeating a feature in a chromosome might increase its chance of surviving to future generations besides increasing its importance to classification.

In each generation, starting from the M current feature subsets, the *cross-over* operator produces m_{cro} new feature subsets, and the *mutation* operator produces m_{mut} new feature subsets. Then, from all these $M + m_{\text{cro}} + m_{\text{mut}}$ available feature subsets, only a total of M feature subsets presenting the highest fitness values are selected to compose the ensemble at the end of the current generation (the other feature subsets, with smaller fitness, are discarded). These M feature subsets correspond to the output of the current generation.

The fitness Fit_m of a feature subset f_m is estimated as a linear combination of the accuracy Acc_m achieved by a classifier c_m which uses f_m and the diversity Div_m of this classifier,

$$\text{Fit}_m = \text{Acc}_m + \lambda \text{Div}_m, \tag{4.1}$$

where λ is a regularization parameter that controls the trade-off between accuracy and diversity. The diversity Div_m of a classifier c_m is defined as the average difference between its prediction and the prediction of the ensemble $\mathscr{E}'$ of M classifiers corresponding to the population of the current generation for a training set composed of R training examples $\mathbf{x}_r$ as

$$\text{Div}_m = \frac{1}{R} \sum_{r=1}^{R} |\hat{P}^{c_m}_{\text{pos}}(\mathbf{x}_r) - \hat{P}^{\mathscr{E}'}_{\text{pos}}(\mathbf{x}_r)|. \tag{4.2}$$

Since there is no obvious way to set the value of the parameter λ, GEFS dynamically adjusts λ after each generation, based on the discrete derivatives of the ensemble error, the average population error and the average diversity within the ensemble. If the ensemble error is not decreasing, then λ is modified by 10% of its current value: λ is increased if the average population

error is not increasing and the average diversity is decreasing; or λ is decreased if the average population error is increasing and the average diversity is not decreasing.

The cross-over operator works as follows. It randomly selects, proportionally to fitness, two feature subsets from the current M feature subsets. These two parents generate one child, which is a new feature subset that uses a randomly defined number of features (from 1 to $2 \times D$); the percentage of these features that comes from each parent is also randomly defined. Then each parent contributes with a number of features, each feature being sampled, with replacement, from the feature subset of this parent.

The mutation operator works as follows: It uses one feature subset, that is randomly selected from the current M feature subsets. Then a new feature subset is produced, using the same number of features, but having a total of Perc_{mut} percent of its features being randomly selected and then changed to a different, randomly defined feature.

4.2.2 The Multiple-GEFS Approach

Running the GEFS algorithm produces a set of feature subsets that have a good potential to compose an ensemble. These feature subsets were selected because they presented a higher fitness, estimated by constructing SVMs and performing cross-validation. But this optimized performance was achieved with SVMs using a fixed hyperparameter value, which provides a fixed perspective of the transformed feature space defined by the kernel mapping. As defining a fixed, global best SVM hyperparameter value is against the principle of classifier ensembles, in this work we use *multiple-GEFS* searches, each of which employing a different hyperparameter value. The use of different hyperparameter values might allow more diverse feature subsets to be found, which ultimately might compose a more divergent, accurate ensemble.

For creating the set $\mathcal{F}$ of feature subsets, we run I independent GEFS searches, $\{\mathcal{S}_1, \ldots, \mathcal{S}_i, \ldots, \mathcal{S}_I\}$, each of which using a different, fixed hyperparameter value $(C, \gamma)_i$ to build RBF-kernel C-SVMs to estimate the quality of each feature subset. The output of a GEFS search $\mathcal{S}_i$ corresponds to M feature subsets, $\{f_1^{\mathcal{S}_i}, \ldots, f_m^{\mathcal{S}_i}, \ldots, f_M^{\mathcal{S}_i}\}$. The set $\mathcal{F}$ of feature subsets is then composed of every feature subset $f_m^{\mathcal{S}_i}$, which is the m-th feature subset produced by the search $\mathcal{S}_i$. So $|\mathcal{F}| = I \times M$.

4.3 Second Stage: Hyperparameter Variation

The objective of the second stage is to improve divergence in an ensemble besides improving the SVMs accuracy, by better adapting the SVMs to their specific feature subset. That is done by tuning the hyperparameters of each SVM. Although this approach does not explicitly increase a metric of divergence among the SVMs, tuning each SVM does improve their diversity and disagreement, since the assigned hyperparameter value is likely to be quite distinct among different SVMs due to the diverse feature subsets employed.

We use a simple, widely employed method to tune the SVM hyperparameters. We use the grid-search on the log-scale of the parameters in combination with cross-validation on each candidate parameter vector. Basically, pairs (C,γ) from a set of predefined values are tried by evaluating RBF-kernel C-SVMs which use them, and the pair that provided the highest cross-validation accuracy is finally selected to be used with this SVM. The `libsvm` [Chen, Lin e Schölkopf 2005] library provides an implementation of grid-search, in which the investigated values of C are $\{2.0, 8.0, 32.0, 128.0, 512.0, 2048.0, 8192.0, 32768.0\}$, and the investigated values of γ are $\{0.0078125, 0.03125, 0.125, 0.5, 2.0, 8.0\}$.

To define the SVM set $\mathcal{H}$, for each feature subset $f_m^{S_i}$ in the set $\mathcal{F}$ we employ the grid-search method to build a SVM $c_m^{S_i}$ using this feature subset and employing tuned hyperparameters. Thus $\mathcal{H}$ is composed of every produced SVM, i.e. $|\mathcal{H}| = |\mathcal{F}|$.

4.4 Third Stage: Selection of the Final Ensemble

The objective of the third stage is discarding most of the overproduced SVMs in the set $\mathcal{H}$, in such a way that just an optimized SVM subset $\mathscr{E} \subset \mathcal{H}$ is finally retained to comprise the ensemble. This classifier selection process is useful for increasing the ensemble accuracy and to reduce the number of component SVMs in the ensemble. Building a large classifier set and further searching for an optimized classifier subset is an ensemble construction strategy known as *overproduce-and-choose* [Kuncheva 2004].

Considering that the SVM set $\mathcal{H}$ was defined by a global search, in the sense that multiple, complementary GEFS searches were used to investigate the space of feature subsets, it seems adequate to employ a local search to define the classifier subset $\mathscr{E} \subset \mathcal{H}$. Since $\mathcal{H}$ is composed of many promising SVMs, this local search should be able to precisely investigate the candidate ensembles, aiming to find a SVM subset with an optimized trade-off between accuracy and diversity, i.e. with a higher estimated ensemble accuracy.

We use the *sequential forward selection* (SFS) search method [Kudo e Sklansky 2000] to select the classifiers, due to the good performance of hill-climbing approaches in performing local search. The SFS search starts with an empty set $\mathcal{V}_k$ of selected SVMs composing the ensemble, and at each step one SVM is included in $\mathcal{V}_k$. Consider that k SVMs have already been selected and included in $\mathcal{V}_k$. If $\mathcal{H}$ is the set of all $|\mathcal{H}|$ available SVMs, then $\mathcal{H} \setminus \mathcal{V}_k$ is the set of $|\mathcal{H}| - k$ candidates SVMs c_t. To include one more SVM in $\mathcal{V}_k$, each non-selected SVM c_t must be tested individually together with the already selected SVMs and ranked according to the criterion L, so that

$$L(\mathcal{V}_k \cup \{c_1\}) \geq L(\mathcal{V}_k \cup \{c_2\}) \geq \ldots \geq L(\mathcal{V}_k \cup \{c_{|\mathcal{H}|-k}\}). \tag{4.3}$$

As a result of the current inclusion step, the SVM c_1 that provided the highest criterion $L(\mathcal{V}_k \cup \{c_1\}) = L(\mathcal{V}_{k+1})$ is included in the set of selected SVMs; this corresponds to the $(k+1)$-th inclusion step.

We define the criterion $L(\mathcal{V}_k)$ of a candidate SVM set $\mathcal{V}_k$ to be the Area Under the Receiver Operating Characteristics (ROC) Curve (AUC) [Fawcett 2006] achieved by this candidate ensemble $\mathcal{V}_k$. Since the score that every SVM in $\mathcal{H}$ gives to a training pattern $\mathbf{x}$ was previously estimated by cross-validation (during the hyperparameter variation stage), then the criterion $L(\mathcal{V}_k)$ can be readily estimated, by obtaining, for every training pattern $\mathbf{x}$, the score $\hat{P}_{\text{pos}}^{\mathcal{V}_k}(\mathbf{x})$ assigned to $\mathbf{x}$ by the candidate ensemble $\mathcal{V}_k$. $\hat{P}_{\text{pos}}^{\mathcal{V}_k}(\mathbf{x})$ is obtained by averaging the k scores $\hat{P}_{\text{pos}}^{c_k}(\mathbf{x})$ given to $\mathbf{x}$ by the k SVMs c_k in $\mathcal{V}_k$.

The AUC value, used to directly estimate the quality of a candidate ensemble, is similar to the traditional accuracy value, but using AUC is more useful for comparing classifiers in problems with unbalanced classes in which negative class examples are usually much more common than positive ones. The AUC value achieved by a classifier corresponds to the probability that, given a positive class example p and a negative class example n, both randomly sampled, this classifier predicts $\hat{P}_{\text{pos}}(p) > \hat{P}_{\text{pos}}(n)$.

5 Oil Rig Motor Pump Fault Diagnosis

"The engineer's first problem in any design situation
is to discover what the problem really is."
- George C. Beakley.

This chapter details the mechanical engineering problem focused on this work, namely the diagnosis of faults in industrial machines.

Section 5.1 is concerned with the fault diagnosis problem and the model-free approach based on pattern recognition techniques. Section 5.2 describes the motor pump equipment. Section 5.3 presents the considered fault categories. Section 5.4 describes the extracted features.

5.1 Model-free Fault Diagnosis

The early detection of faults in complex industrial machinery is advantageous for economical and security reasons [Bellini, Filippetti e Capolino 2008]. An effective diagnostic system can aid relatively unskilled operators in making reliable decisions about machinery condition as well as aiding experts in making decisions about intricate fault occurrences. This might decisively contribute to the main objective of maintenance engineering, which is repairing damaged components during planned maintenance aiming to minimize machinery downtime and to improve security.

There are two main approaches to the machine fault diagnosis problem: model-based techniques and model-free techniques. The model-based line of research relies on an analytical model of the studied process, involving time dependent differential equations. Usually the experimental process setup is installed in a controlled laboratory environment and is embedded in a control loop in which inputs, controlled variables and sensor outputs are modeled. However in real-world processes the availability of an analytical model is often unrealistic or inaccurate due to the complexity of the process. In this case model-free techniques are an alternative approach [Bellini, Filippetti e Capolino 2008], which relies on pattern recognition based techniques for automatically learning fault describing rules from training data.

5.2 Motor Pump Equipment

Rotating machinery covers a wide range of mechanical equipment and plays an important role in industrial applications. In this work we focus on a specific rotating machine model, namely horizontal motor pumps with extended coupling between the electric motor and the pump. Accelerometers are placed at strategic positions along the main directions to capture specific vibrations of the main shaft which provides a multichannel time domain raw signal. Figure 5.1 shows a typical positioning configuration of the accelerometers in the equipment.

5.3 Considered Fault Categories

Several faults can simultaneously occur in a motor pump. Such a high diversity of defects has a direct impact on the subsequent classifier. Many faults cause vibrations in similar frequency bands, for instance 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

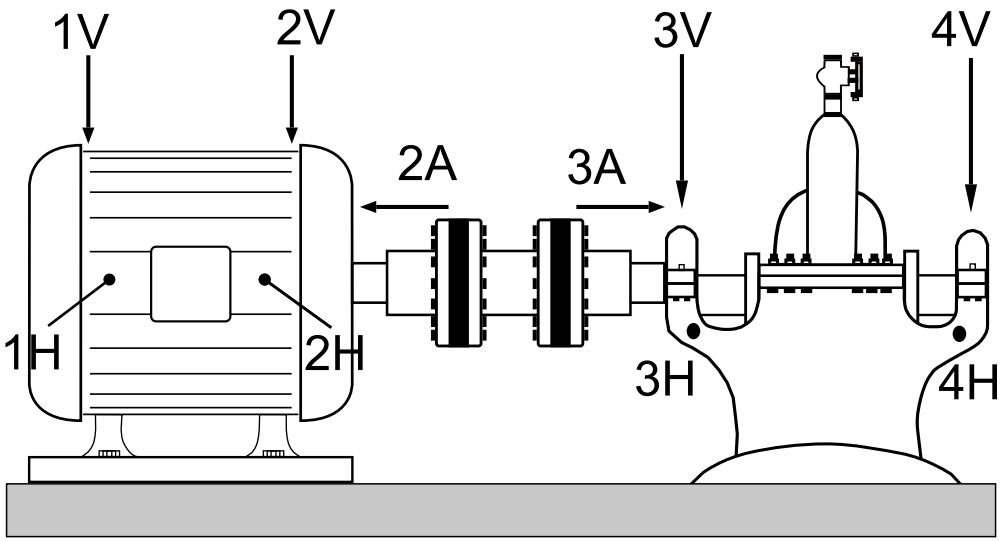

Figure 5.1: Motor pump with accelerometers placed along the horizontal (H), axial (A) and vertical (V) directions. The motor corresponds to positions the 1 and 2, and the pump to the positions 3 and 4.

characteristic signature.

We build an independent predictor for detecting the individual occurrence of each of the following fault categories: pump blade unbalance; hydrodynamic fault (due to blade pass and vane pass, cavitation or flow turbulence); shaft misalignment; rolling element bearing failures; mechanical looseness; and structural looseness. For the latter three fault categories, since they individually occur in a motor or in a pump, we make a distinction between the predictor of a fault occurrence in the motor and the predictor of a fault occurrence in the pump.

Table 5.1 shows, for each considered fault category, the percentage of the 2000 examples that presented this fault. Examples presenting multiple faults are more common than examples in which just one fault is occurring.

Figure 5.2 illustrates how the frequency spectrum is associated with two of the faults. The figure presents the vibration signal Fourier spectrum of a motor pump with misalignment; this fault manifests itself in the frequency spectrum at the first three harmonics of the shaft rotation frequency. Besides, the high energy in the fifth harmonic, as well as the noise in low frequencies, indicate that additionally a hydrodynamic fault is emerging. This signal was measured from the position 3, in the horizontal direction.

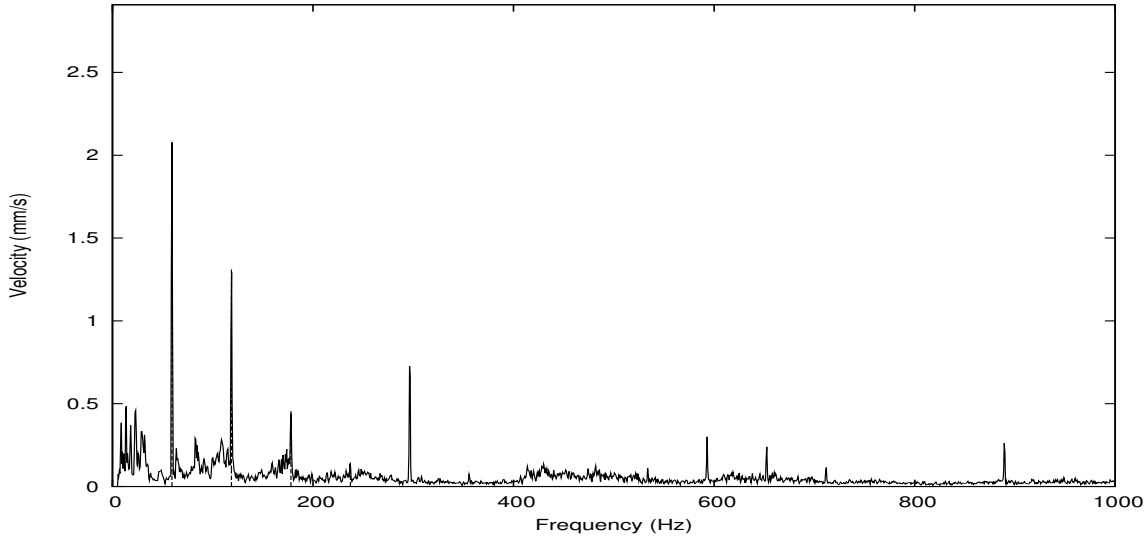

Figure 5.2: Vibration signal Fourier spectrum of a motor pump presenting misalignment and also an emerging hydrodynamic fault.

Table 5.1: Fault occurrence.

Fault class	Percentage of faulty data
Misalignment	42.6%
Hydrodynamic	42.4%
Unbalance	24.9%
Bearing - motor	24.9%
Bearing - pump	16.6%
Structural looseness - motor	26.6%
Structural looseness - pump	13.9%
Mechanical looseness - motor	12.1%
Mechanical looseness - pump	8.6%

5.4 Extracted Features

Our general classification strategy is based on providing as much information as possible in the initial *feature extraction* stage, and further using ensemble construction to automatically prioritize more relevant features.

It would be desirable to extract features from distinct, complementary information sources, for instance electrical current, chemical, thermal, and mechanical vibration sensors. Also, it would be desirable to employ different signal preprocessing techniques, thus obtaining features from different domains aiming to reflect complementary perspectives. For instance, for mechanical vibration sensors, the features can include [Lei et al. 2010]: time domain statistical features such as mean, root mean square (RMS), variance, skewness and kurtosis; frequency domain features such as the amplitude of the spectrum and the energy in specific frequency bands;

and time-frequency domain features obtained by using advanced time-frequency analysis techniques, such as the empirical mode decomposition (EMD) and the wavelet packet transform (WPT) [Lei et al. 2010].

But the information format of the acquired examples was previously fixed as the frequency and the envelope spectrum of machine vibration signals. We work with well-established signal processing techniques, namely the Fourier transform, envelope analysis based on the Hilbert transform [Mendel, Rauber e Varejao 2008] and median filtering. Thus the extracted features correspond to the vibrational energy of predetermined frequency bands of the spectrum.

In the initial feature extraction stage, we extract the same feature categories for building the predictor of every considered fault. The initially extracted feature set is composed of a total of $D = 81$ features, with 68 of them from the Fourier spectrum and 13 of them from the Envelope spectrum.

Before extracting the $D = 81$ features that globally describe the condition of a motor pump, it is necessary to specify which one of the four machine positions will be employed as the source of the vibration signal. This depends on which fault category is currently under consideration. Specifically, the shaft misalignment predictor chooses

between position 2 or 3, actually selecting the one which presents the higher total RMS vibration energy of the velocity signal, for the testing pattern $\mathbf{x}$ under consideration. Similarly, the predictor of unbalance or hydrodynamic fault chooses between positions 3 or 4. For mechanical looseness, structural looseness or a bearing defect, since they can independently occur in a motor or in a pump, we build an independent predictor to detect a fault occurrence in the motor (thus choosing between the position 1 or 2) and another independent predictor to detect a fault occurring in the pump (thus choosing between the position 3 or 4).

In summary, the complete diagnostic system is composed of nine independent SVM ensembles, each of which individually detects one considered fault category.

5.4.1 Fourier Spectrum Features

We extract a total of 68 features from the Fourier spectrum. We use the RMS value of a 10% large narrow band around each of the following harmonics of the shaft rotation frequency: 0.5x, 1x, 1.5x, 2x, 2.5x, 3x, 3.5x, 4x, 4.5x, 5x and 5.5x, obtained for each of the three directions of measurement, namely horizontal, vertical and axial (this generates a total of $11 \times 3 = 33$ features).

Besides, we use as features the sum of the RMS value of a 10% large narrow band around the harmonics 1x, 2x, ..., 5x, in each direction (3 features), and also similarly the sum of the RMS value of bands around the inter-harmonics 0.5x, 1.5x, ..., 5.5x (3 features).

We also use the RMS value of the noise calculated with the median filtering, in the bands 0x-1x, 0x-2x, 0x-3x, 0x-4x and 0x-5x, in each direction ($5 \times 3 = 15$ features).

Additionally, we use the 10% large narrow band around harmonics of the pump blade pass frequency (BPF), namely $0.5 \times$ BPF, $1.0 \times$ BPF and $2.0 \times$ BPF, in each direction ($3 \times 3 = 9$ features).

We also use the vibration signal total RMS value, in each direction for the velocity signal (3 features) and in horizontal and vertical direction for the acceleration signal (2 features).

5.4.2 Envelope Spectrum Features

We extract a total of 13 features using the Envelope analysis [McInerny e Dai 2003]. Specifically, these features correspond to the RMS of 10% large narrow bands around the first, the second and the third harmonics of the bearing characteristic frequencies, namely BPFI, BPFO, FTF and BSF [Mendel, Rauber e Varejao 2008], in the horizontal direction (this generates a total of $4 \times 3 = 12$ features). The bearing characteristic frequencies depend of the bearing model of the machine under consideration. We also use as a feature the total RMS value of the Envelope spectrum (1 feature).

6 *Experimental Results*

"Computers are useless.
They can only give you answers."
- Pablo Picasso.

This chapter shows the experimental results achieved by the studied classification models, using the acquired database of real-world industrial machine vibration signals.

Section 6.1 details the 5×2 cross-validation method, employed to estimate the quality of the studied classification models which are presented in section 6.2. Section 6.3 provides the classification accuracy estimated by 5×2 cross-validation. Section 6.4 details experiments performed to provide a better insight into important aspects of the proposed ensemble method.

6.1 Cross-validation 5×2

To assess the effectiveness of the studied classification approaches we performed a stratified 5×2 cross-validation [Kuncheva 2004]. This corresponds to five replications of a 2-fold cross-validation. In each replication, the complete database of 2000 examples was randomly partitioned, in a stratified manner, into two sets each one with approximately 1000 examples (the stratification process preserves the distribution of the nine fault categories between both sets). So in each replication each considered classification model for creating the predictor of a fault was trained on a set and tested on the remaining one; after the five replications, the final test data AUC achieved by this classification model in predicting this fault is then obtained as the average of the ten estimated testing data AUC values.

6.2 Studied Classification Models

For each considered fault category, we studied four different classification models for building the predictor of this fault: a single SVM; a SVM ensemble built by the traditional GEFS method, with every SVM using the same hyperparameter value; a SVM ensemble built by a straightforward upgrade of GEFS, namely tuning the hyperparameters of every SVM ultimately produced by GEFS; and a SVM ensemble built by the proposed multiple-GEFS method described in chapter 4.

6.2.1 The `SVM` Classification Model

This classification model is a single SVM classifier, using all the $D = 81$ available features. We used the grid-search method to tune the hyperparameters as explained in section 4.3.

6.2.2 The `GEFS` Classification Model

This classification model is a SVM ensemble built by the traditional GEFS method, described in section 4.2.1. We employed the hyperparameter value $(C = 8.0, \gamma = 0.5)$ to build RBF-kernel C-SVMs to estimate fitness; this value was chosen since it was frequently selected by grid-search in preliminary experiments, thus suggesting that this hyperparameter value tends to produce more accurate SVMs.

We set the initial value of the λ regularization parameter used for fitness evaluation as $\lambda = 1.0$. We use $M = 20$ classifiers (feature subsets) in the ensemble. In each generation,

starting from these $M = 20$ feature subsets, we produced $m_{\text{mut}} = 10$ new feature subsets by using mutation (randomly changing $\text{Perc}_{\text{mut}} = 30\%$ of features) and more $m_{\text{cro}} = 10$ new feature subsets by using cross-over; from these $M + m_{\text{cro}} + m_{\text{mut}} = 40$ feature subsets, the $M = 20$ top ones with higher fitness were selected to compose the ensemble at the end of this generation. The population evolved for a total of $N = 100$ generations.

A single run of the GEFS algorithm demanded a total of $20 + N \times 20 = 2020$ feature subsets to be evaluated. Since each feature subset evaluation corresponded to a 5-fold cross validation, a run of the GEFS algorithm demanded a total of $2020 \times 5 = 10100$ SVMs to be constructed.

In summary, this **GEFS** classification model corresponds to an ensemble of 20 RBF-kernel C-SVMs, each of which used the hyperparameter values $(C = 8.0, \gamma = 0.5)$.

6.2.3 The GEFS-Tuned Classification Model

This classification model corresponds to a straightforward upgrade of GEFS, namely tuning the hyperparameters of each SVM in an ensemble built by GEFS. First we used the **GEFS** classification model (presented in section 6.2.2) to generate an ensemble of $M = 20$ SVMs which differ among themselves solely on their feature subset. Further, for each of these 20 produced SVMs we employed the grid-search method to tune its hyperparameters. Thus the final ensemble was composed of $M = 20$ SVMs which differ among themselves on their feature subset and also on their employed hyperparameter value.

In summary, this **GEFS-Tuned** classification model corresponds to an ensemble of 20 RBF-kernel C-SVMs, each of which used tuned hyperparameters.

6.2.4 The Multiple-GEFS Classification Model

This classification model corresponds to a SVM ensemble based on feature and hyperparameter variation, built by the multiple-GEFS ensemble method proposed in this work (presented in section 4.1).

For creating the set $\mathcal{F}$ of feature subsets, we ran $I = 5$ independent GEFS searches, each of which used a different, fixed hyperparameter value $(C, \gamma)_i$ to build SVMs to estimate the quality of the feature subsets. After preliminary experiments to evaluate the hyperparameter values investigated by the grid-search tuning method (which are presented in section 4.3), we defined the following hyperparameter values to be used: $\{(C = 8.0, \gamma = 0.5), (C = 128.0, \gamma = 0.03125), (C = 128.0, \gamma = 0.125), (C = 128.0, \gamma = 2.0), (C = 128.0, \gamma = 8.0)\}$. The former value was

chosen since it provided more accurate SVMs, while the latter values were chosen because they correspond to a large range of γ values besides using a relatively high C value. Every GEFS search evolved for a total of $N = 100$ generations.

To ultimately compose the set $\mathcal{F}$ of feature subsets we used the outputs of those five GEFS searches, thus $|\mathcal{F}| = 5 \times 20 = 100$. Further, for composing the SVM set $\mathcal{H}$, for each feature subset in $\mathcal{F}$ we used the grid-search method to build a SVM with tuned hyperparameters. Thus $|\mathcal{H}| = |\mathcal{F}| = 100$. Finally, from all these $|\mathcal{H}| = 100$ produced SVMs, we used the SFS search to select a reduced SVM subset $\mathscr{E}$ as explained in section 4.4. We set the ensemble size as $|\mathscr{E}| = 40$, so the ultimately produced ensemble $\mathscr{E}$ was composed of 40 SVMs. The other parameters of the GEFS algorithm were set as for the **GEFS** classification model presented in section 6.2.2.

In summary, this **Multiple-GEFS** classification model corresponds to an ensemble of 40 RBF-kernel C-SVMs, each of which used tuned hyperparameters.

6.3 Cross-validation 5×2 Estimated Results

Table 6.1 presents the testing data AUC values and the standard deviations estimated by 5×2 cross-validation. For each considered fault, the result of the classification model which provided the most accurate predictor is showed in bold.

The consistently higher accuracy achieved by the proposed **Multiple-GEFS** classification model, in comparison to the accuracy achieved by the **GEFS** or the **GEFS-Tuned** classification models, suggests the importance of employing a powerful search to deeply investigate the space of feature subsets and the space of hyperparameter values, aiming to produce an optimized SVM ensemble based on feature and hyperparameter variation. Results show that the **Multiple-GEFS** classification model achieved the highest accuracy for every fault category, and the lowest standard deviation for six of the nine considered faults.

To corroborate the superiority of the multiple-GEFS method, we used the statistical testing procedure proposed by Dietterich (described in [Kuncheva 2004]) to be employed with the 5×2 cross-validation process, which determines whether the estimated difference of AUC values is statistically significantly different.

The level of significance is the 0.05 percentile. For misalignment, hydrodynamic and bearing-motor faults, the statistical test confirmed that the **Multiple-GEFS** classification model performed significantly better than the **GEFS** or the **GEFS-Tuned** classification models. Also, comparing the **Multiple-GEFS** model to the **SVM** model (which corresponds to

Table 6.1: Test data AUC estimated by 5×2 cross-validation

Fault category	SVM	GEFS	GEFS -Tuned	Multiple -GEFS
Misalignment	.834 $\pm$.011	.862 $\pm$.007	.865 $\pm$.009	**.882** $\pm$.006
Unbalance	.909 $\pm$.014	.933 $\pm$.008	.929 $\pm$.006	**.942** $\pm$.005
Hydrodynamic	.923 $\pm$.010	.931 $\pm$.012	.935 $\pm$.010	**.942** $\pm$.008
Bearing - motor	.935 $\pm$.006	.955 $\pm$.004	.957 $\pm$.004	**.969** $\pm$.005
Bearing - pump	.877 $\pm$.020	.927 $\pm$.014	.926 $\pm$.019	**.944** $\pm$.010
Structural L. - motor	.914 $\pm$.009	.931 $\pm$.006	.934 $\pm$.007	**.943** $\pm$.008
Structural L. - pump	.857 $\pm$.028	.893 $\pm$.011	.896 $\pm$.012	**.911** $\pm$.013
Mechanical L. - motor	.862 $\pm$.013	.888 $\pm$.012	.888 $\pm$.012	**.895** $\pm$.012
Mechanical L. - pump	.886 $\pm$.022	.908 $\pm$.016	.908 $\pm$.018	**.920** $\pm$.014

a single SVM), the statistical test confirmed a significantly superior performance for all those mentioned faults and additionally for the bearing-pump fault. On the other hand, for the **GEFS** or the **GEFS-Tuned** classification models, the statistical test confirmed a superior performance of the ensemble in comparison to a single SVM just for the bearing-motor fault.

6.4 Influence of the Number of Evolved Generations and the Number of Component SVMs

In this section we show experimental results aiming to provide an insight into two important aspects of the proposed ensemble method: the influence of the number of evolved generations and the influence of the number of component SVMs in the ensembles.

6.4.1 Influence of the Number of Evolved Generations

An important parameter of the GEFS algorithm is the maximum number of evolved generations. Using a very high number of generations causes two main problems, namely a high computational cost and overfitting. In this work, to build the SVM ensembles, we evolved the GEFS searches for a total of $N = 100$ generations.

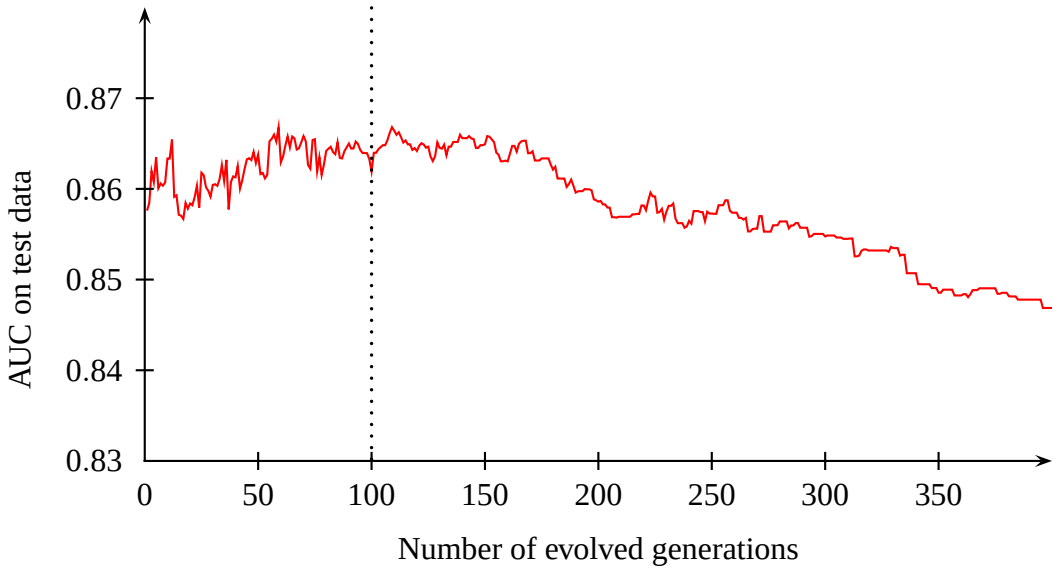

Figure 6.1: AUC on test data achieved by each evolved generation of the GEFS method, for the misalignment predictor.

Figure 6.1 presents the behaviour of a GEFS search concerning the number of evolved generations. The figure shows results obtained for the first pair of training and testing data generated for the 5×2 cross-validation process, considering the misalignment predictor. One can observe the AUC on test data achieved by the SVM ensemble produced by each generation of the **GEFS** classification model, starting from the generation number 1 and finishing in the generation number 400; every SVM used the hyperparameter value $(C = 8.0, \gamma = 0.5)$. It can be seen that the generation number 100 corresponded to an ensemble with a relatively high accuracy. The figure also shows that after the generation number 150 the test AUC presented a tendency of decreasing, as a consequence of overfitting; surprisingly, the ensemble produced by the generation number 400 presented a lower estimated test data AUC than the ensemble defined by the first generation.

To present the behaviour found for some of the other faults, figures 6.2 and 6.3 present results for the bearing - pump and the bearing - motor fault predictor, respectively; these experiments also used the first pair of training and testing data generated by the 5×2 cross-validation.

6.4.2 Influence of the Number of Component SVMs

To provide an insight into the influence of the number of component SVMs in an ensemble, we show the AUC on test data estimated during the classifier selection stage of the **Multiple-GEFS** classification model, performed using the sequential forward selection (SFS) search strategy.

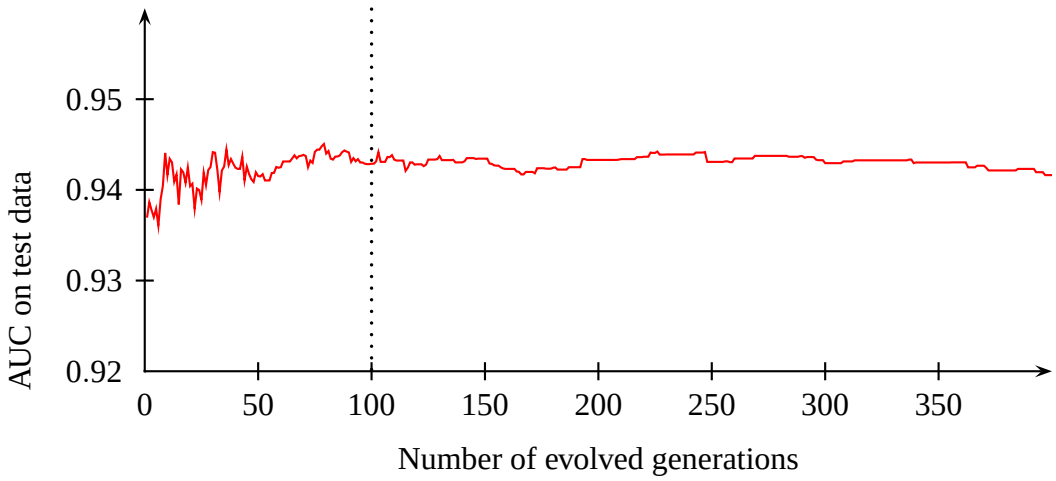

Figure 6.2: AUC on test data achieved by each evolved generation of the GEFS method, for the bearing - pump fault predictor.

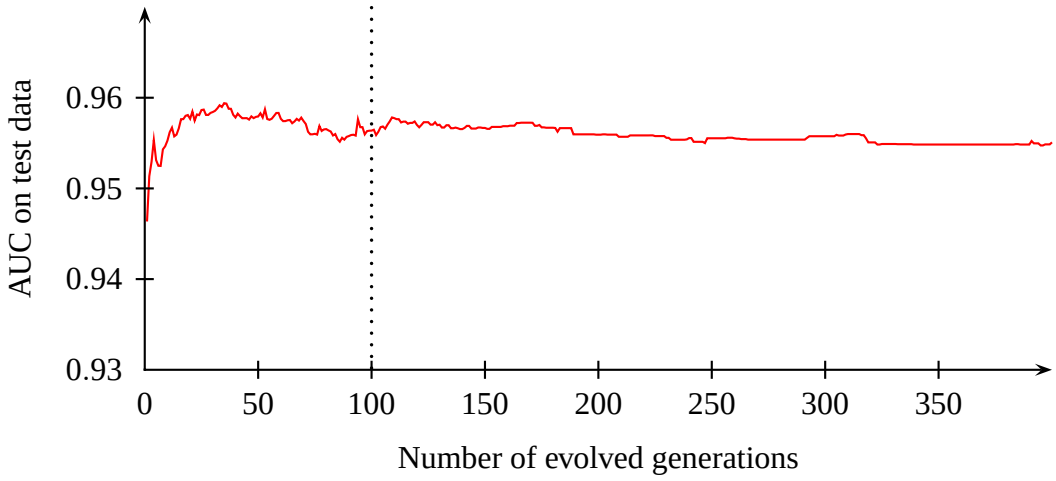

Figure 6.3: AUC on test data achieved by each evolved generation of the GEFS method, for the bearing - motor fault predictor.

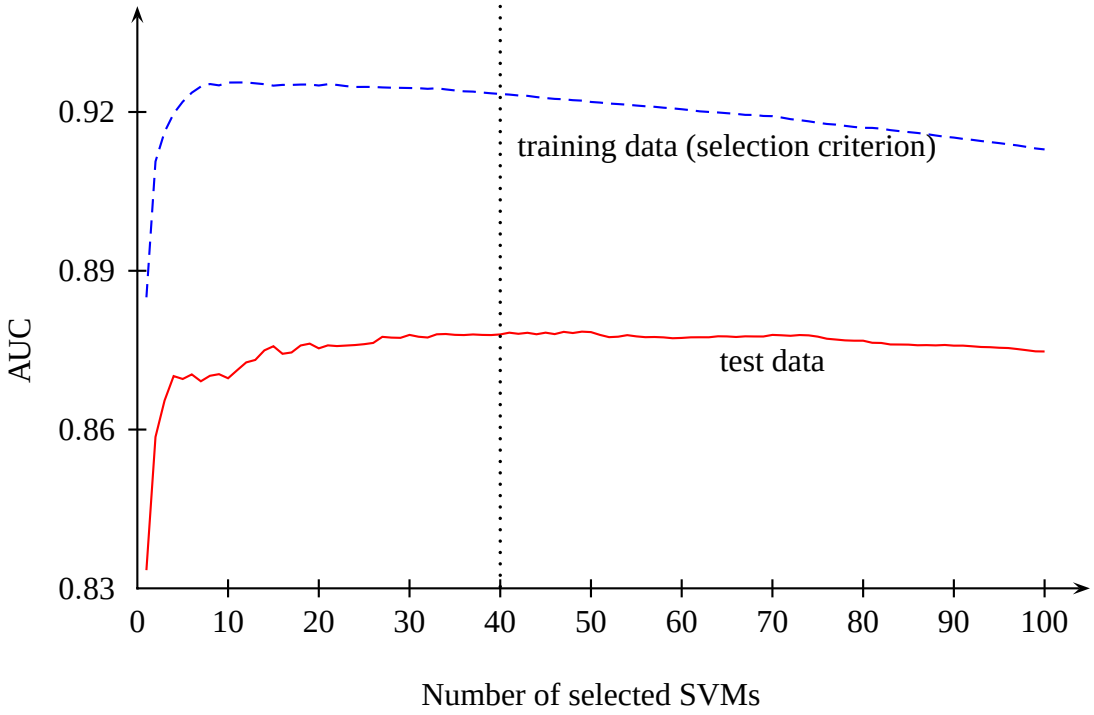

Figure 6.4: AUC on training and testing data achieved by each number of component SVMs, during the classifier selection stage of the multiple-GEFS method, for the misalignment predictor.

Figure 6.4 shows results for the first generated pair of train-test data of the 5×2 cross-validation, for the misalignment predictor. The figure presents the AUC of training data (which is the selection criterion) and the AUC of testing data achieved by the SVM ensemble defined by using each number of selected SVMs, from 1 SVM to 100 SVMs. The final ensemble was composed of the $|\mathscr{E}| = 40$ firstly selected SVMs, since we observed a general tendency of an AUC decrease with the use of a larger set.

To present the behaviour found for some of the other faults, figures 6.5 and 6.6 present results for the structural looseness - pump and the structural looseness - motor predictor, respectively; these experiments also used the first pair of training and testing data generated by the 5×2 cross-validation.

6.5 Usefulness of Hyperparameter Tuning to Improve SVM Diversity

This experiment provides an insight into the effectiveness of hyperparameter tuning aiming to improve SVM diversity. First, we used feature selection to generate a set of different feature

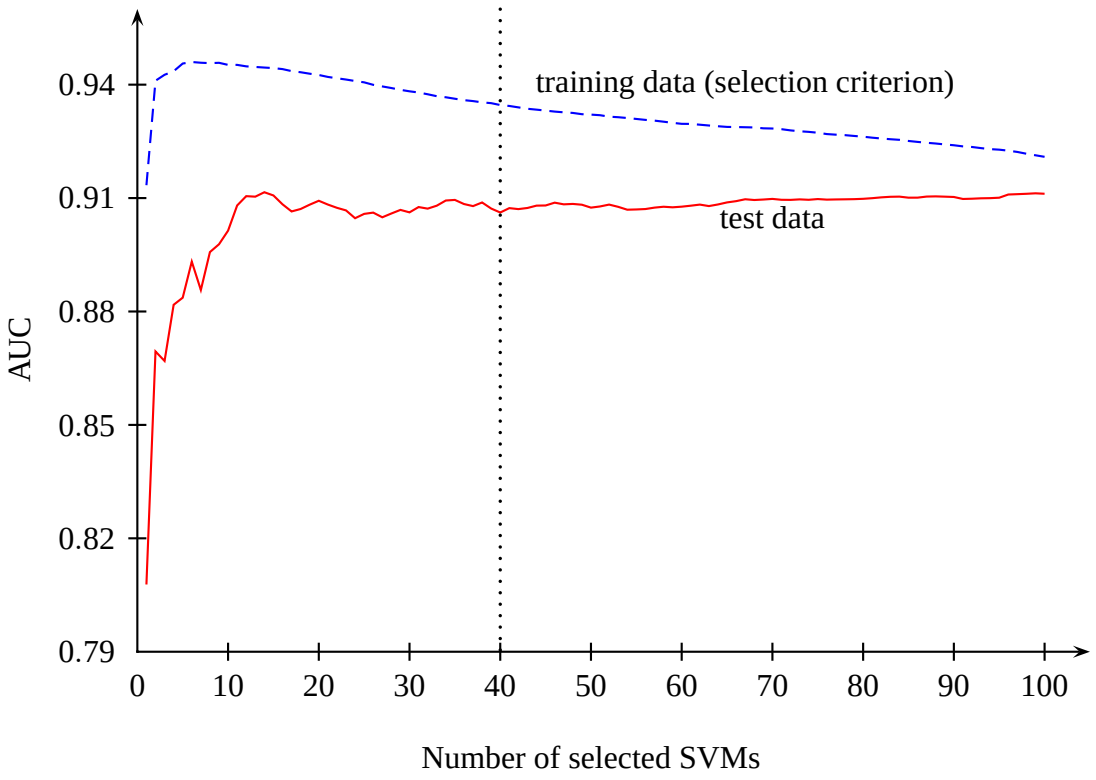

Figure 6.5: AUC on training and testing data achieved by each number of component SVMs, during the classifier selection stage of the multiple-GEFS method, for the structural looseness - pump predictor.

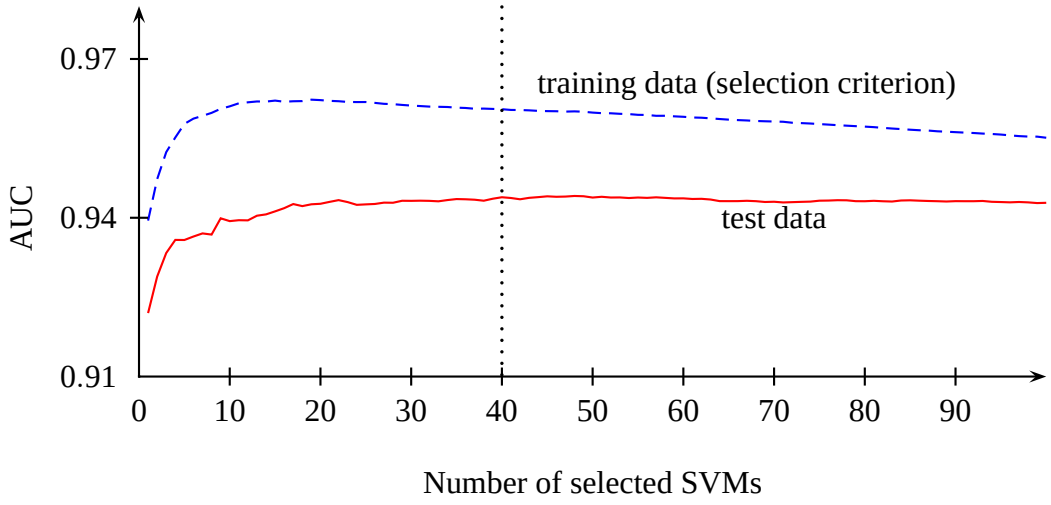

Figure 6.6: AUC on training and testing data achieved by each number of component SVMs, during the classifier selection stage of the multiple-GEFS method, for the structural looseness - motor predictor.

subsets. Further, for each produced feature subset, we estimated the classification accuracy achieved by a SVM using this feature subset. We evaluated two approaches: building each SVM with the hyperparameter value $(C = 8.0, \gamma = 0.5)$ (which tends to provide accurate SVMs) and building each SVM with a tuned hyperparameter.

To perform feature selection, we used the traditionally employed sequential forward selection (SFS) search strategy [Kudo e Sklansky 2000]. The SFS search was presented in section 4.4 in the context of selecting classifiers to compose an ensemble. But in the present section, we use SFS to select features aiming to generate an accurate single SVM.

The SFS search starts with an empty set of selected features, and at each step one feature is included in this set, namely the feature that provided the higher selection criterion with its individual inclusion in the current set of selected features. The selection criterion of a feature was estimated as the 5-fold cross-validation AUC achieved by a SVM using the currently selected features and also this new feature under evaluation. The SVMs used the hyperparameter value $(C = 8.0, \gamma = 0.5)$.

From the total of $D = 81$ available features, we required SFS to select $D - 1 = 80$ features. This produces 80 different feature subsets; for each one we evaluated two classifiers, namely a SVM with fixed hyperparameters and a SVM with tuned hyperparameters.

Figure 6.7 presents the results obtained for the first pair of training and testing data defined by 5×2 cross-validation, for the misalignment predictor. For each number of selected features, from $k = 1$ to $k = 80$, the figure shows the AUC on the test data achieved by the SVM with hyperparameter fixed as $(C = 8.0, \gamma = 0.5)$, and the AUC on test data achieved by the SVM with tuned hyperparameters. Figure 6.7 also shows, as a horizontal line, the test data AUC achieved by the SVM ensemble built by the **`Multiple-GEFS`** classification model.

By comparing the performance of the non-tuned SVMs versus the tuned SVMs, it is interesting to see that tuning the hyperparameters of each *individual* SVM tends to increase the *collective* diversity of SVMs, since the figure shows that the AUC achieved by the tuned SVMs can strongly vary even among SVMs that employ similar feature subsets (i.e. SVMs that use a similar number of selected features). Such an improvement in SVM diversity is useful since it corresponds to an increasing in the disagreement among the SVMs, which tends to improve the ensemble accuracy. Indeed, the figure shows that the performance of the ensemble with tuned SVMs (indicated as a dashed horizontal line) was significantly better than the performance of any single SVM.

Figures 6.8 and 6.9 present results for the structural looseness - pump and the structural

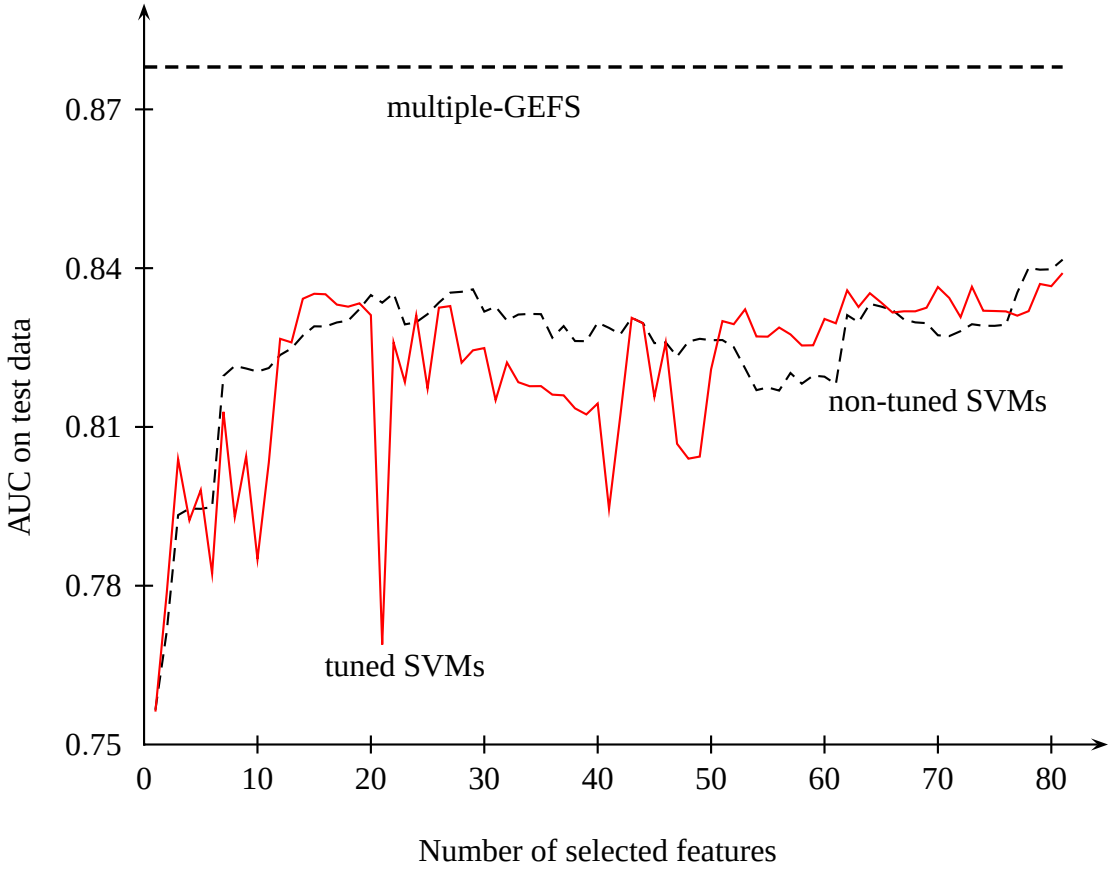

Figure 6.7: AUC on test data achieved by the Multiple-GEFS method, and by individual SVMs which use selected feature subsets with tuned or non-tuned hyperparamenters, for the misalignment predictor.

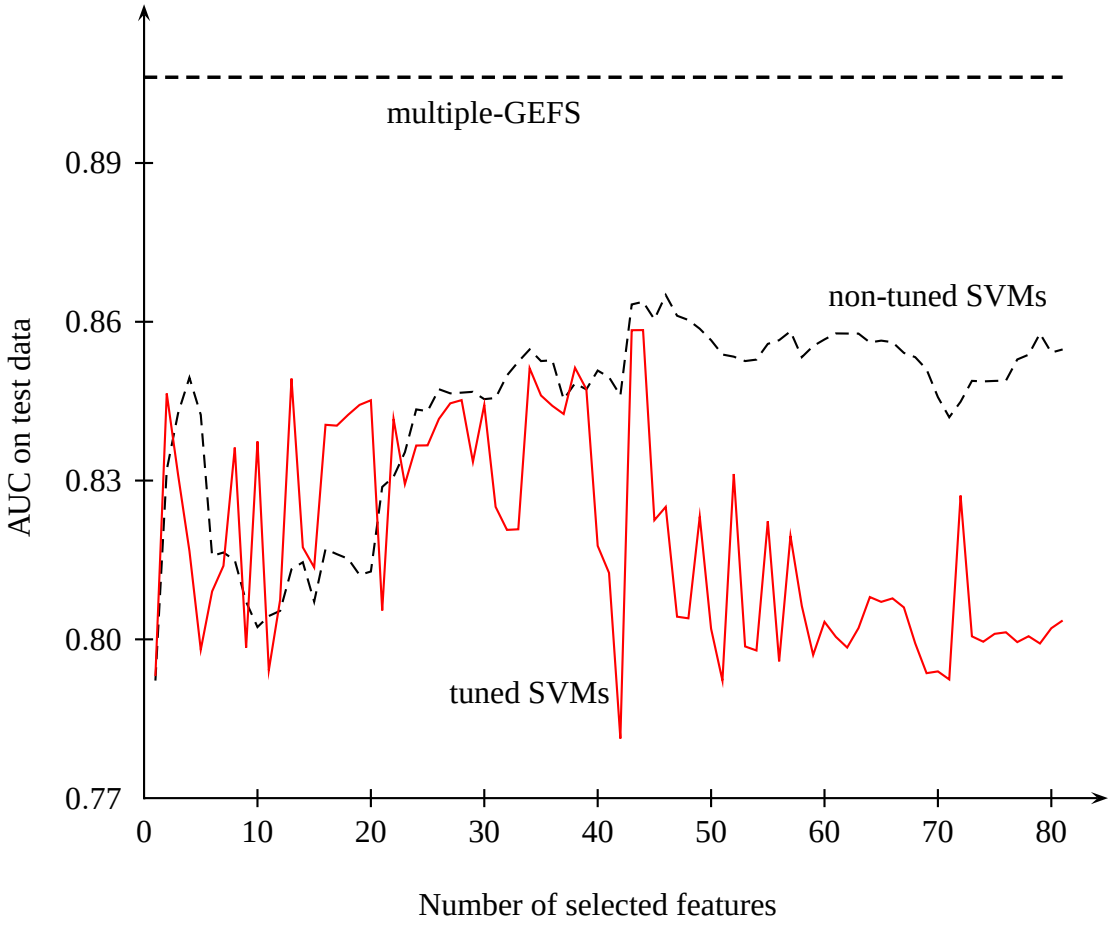

Figure 6.8: AUC on test data achieved by the Multiple-GEFS method, and by individual SVMs which use selected feature subsets with tuned or non-tuned hyperparamenters, for the structural looseness - pump predictor.

looseness - motor predictor, respectively. Figures 6.10 and 6.11 present results for the mechanical looseness - pump and the mechanical looseness - motor predictor, respectively. These experiments also used the first pair of training and testing data generated by the 5×2 cross-validation.

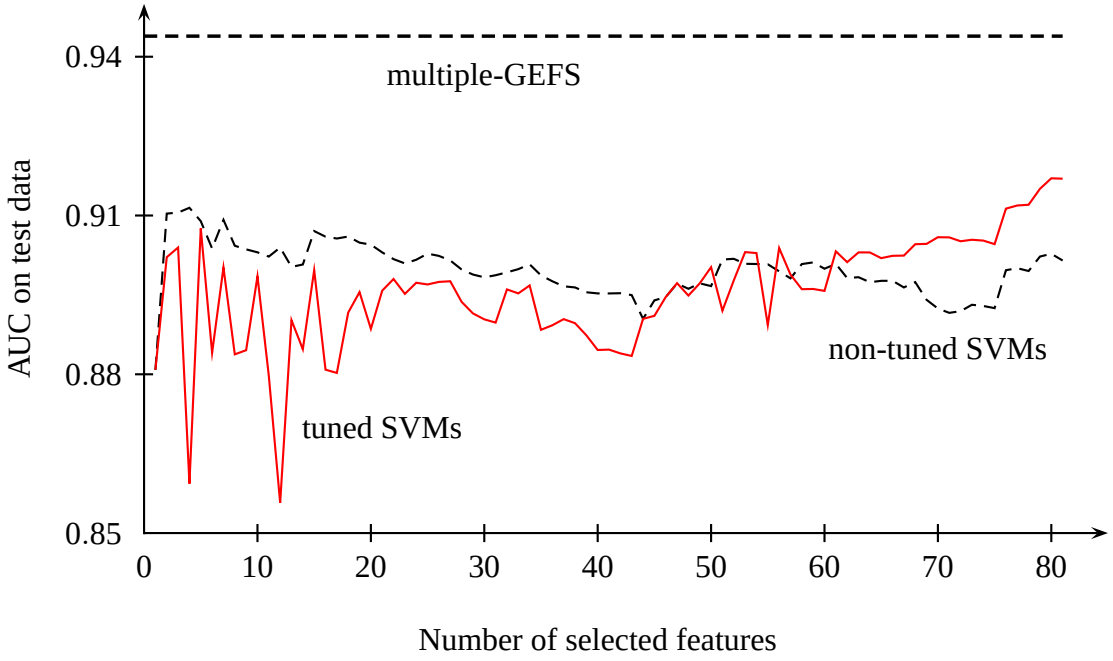

Figure 6.9: AUC on test data achieved by the Multiple-GEFS method, and by individual SVMs which use selected feature subsets with tuned or non-tuned hyperparamenters, for the structural looseness - motor predictor.

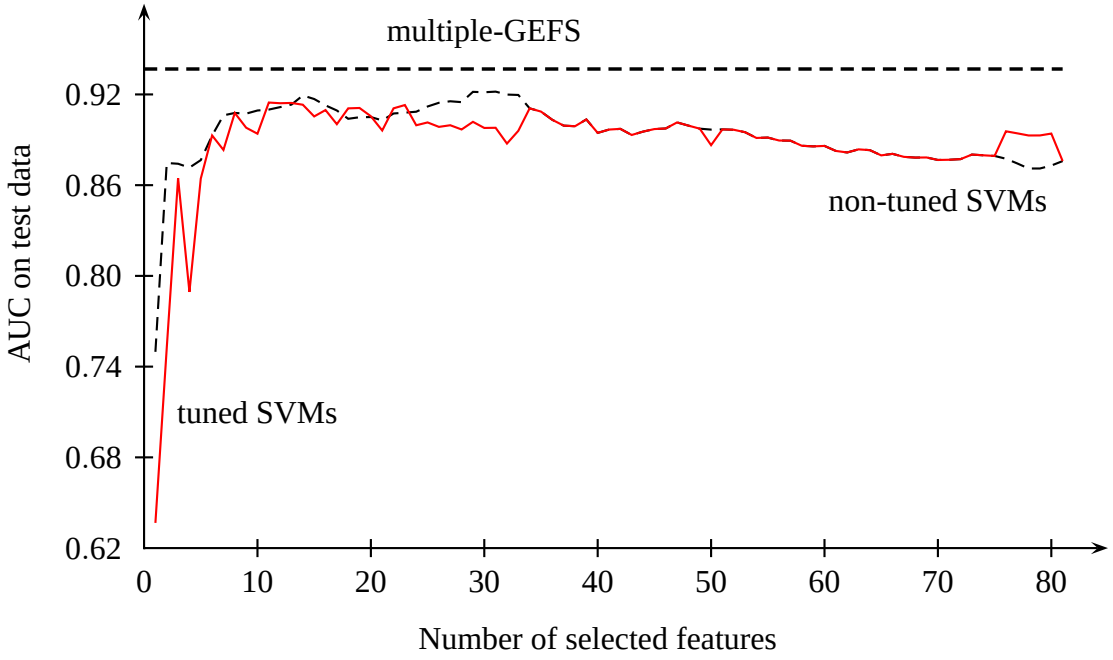

Figure 6.10: AUC on test data achieved by the Multiple-GEFS method, and by individual SVMs which use selected feature subsets with tuned or non-tuned hyperparamenters, for the mechanical looseness - pump predictor.

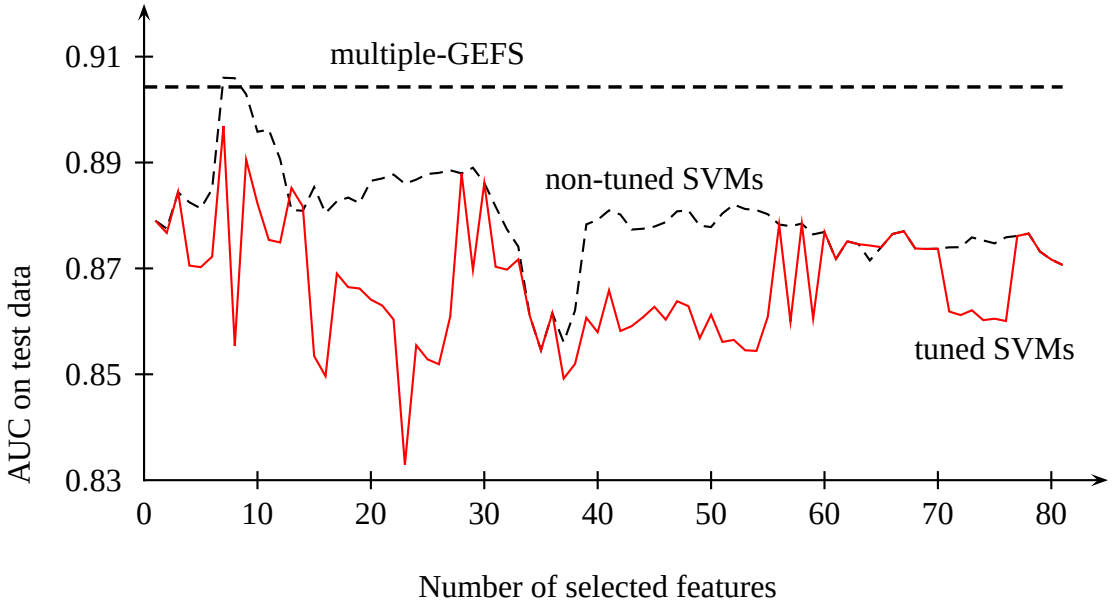

Figure 6.11: AUC on test data achieved by the Multiple-GEFS method, and by individual SVMs which use selected feature subsets with tuned or non-tuned hyperparamenters, for the mechanical looseness - motor predictor.

7 Conclusions and Future Work

"Around computers it is difficult to find the
correct unit of time to measure progress.
Some cathedrals took a century to complete.
Can you imagine the grandeur and scope of a
program that would take as long?"
- 'Epigrams in Programming', SIGPLAN, Association for Computing Machinery, 1982.

This chapter summarizes the main results of this work, and offer concluding remarks.

Section 7.1 draws conclusions. Section 7.2 points out to future research.

7.1 Conclusions

In this work we presented a novel approach for creating a SVM ensemble with component classifiers differing among themselves on the feature subset and the hyperparameter value they use. The performed experiments show a consistent improvement in the prediction accuracy in comparison to using a single accurate SVM or using the well-established GEFS method for creating ensemble classifiers.

This work presented an optimization method to improve the prediction accuracy, which is generally the main objective of classification. But it is important to note that this improvement in accuracy demands an even higher growth of the computational cost, specially during training, that demands building and evaluating several SVMs to investigate the space of feature subsets. Fortunately, SVMs can naturally be employed using parallel computing in order to drastically reduce the processing time.

7.2 Future Work

We plan to work on two main directions of future research. First, acquiring data from more sources than just vibration signals. Second, improving the proposed SVM ensemble method, specifically by increasing the accuracy gain provided by the hyperparameter variation stage.

7.2.1 Using Data from Different Sources

We plan to acquire more real-world data, from different machines and from more sources than just vibration signals. Thus we plan to develop a multiparametric diagnostic system, which uses vibration signals complemented with electrical signals such as current, power and torque; besides, we plan to employ new features which carry specific information about the occurrence of some faults, for instance structural resonance [Wandekokem et al. 2011].

The use of features from different sources might increase the prediction accuracy, since more diversified information may become available, with each information source considering a different perspective of the current motor pump condition. Classifier ensembles can naturally take advantage of multiple information sources, since using features extracted from different sources has a good potential for generating classifiers that give wrong predictions in different regions of the global feature space.

As the SVM ensemble method proposed in this work relies on initially extracting as much

features as possible and further using ensemble construction to automatically prioritize more relevant features, we expect this method to be able to naturally deal with features from multiple sources. For instance, considering that the available features comprise a total of D_{vib} vibrational features and D_{elec} electrical features, than the GEFS searches can be employed to directly investigate all these $D_{\text{global}} = D_{\text{vib}} + D_{\text{elec}}$ features. In this case, the generated feature subsets would be composed of features from both electrical and vibrational sources, as a result of the performed GEFS searches that automatically determine optimized feature subsets.

7.2.2 Using Particle Swarm Optimization to Tune Hyperparameters

The performed experiments showed the consistent improvement in SVM diversity and ensemble accuracy that was provided by the hyperparameter tuning stage. Considering that in this work we employed the simple, exhaustive grid-search method to tune hyperparameters, there is still much space for further improvement. Specifically, we plan to investigate more powerful approaches to tune SVMs.

A straightforward improvement of the proposed SVM ensemble method may be employing a more powerful method for tuning the hyperparameters of each produced SVM aiming to directly increase its individual accuracy; this also implicitly increases the collective diversity of SVMs.

A more complex improvement of the proposed SVM ensemble method may be using hyperparameter variation not only to increase the individual accuracy of each SVM, but also to *directly* improve the collective divergence of a set of SVMs. That is already done by the GEFS method, but using feature variation instead of hyperparameter variation.

Particle Swarm Optimization (PSO) based techniques have been succesfuly employed for SVM hyperparameter tuning [Li e Tan 2010]. By now, these approaches have showed promising results for building accurate single SVMs. Thus a straightforward improvement of the proposed SVM ensemble method may be using a PSO search to tune hyperparameters, instead of using the simple grid-search. Also, since PSO is a population-based search algorithm (like GEFS), PSO seems approppriate to be used for the more complex task of tuning the hyperparameters of a set of SVMs aiming to directly increase their diversity and accuracy.

Bibliography

[Bellini, Filippetti e Capolino 2008]BELLINI, A.; FILIPPETTI, F.; CAPOLINO, G.-A. Advances in diagnostic techniques for induction machines. *IEEE Transactions on Industrial Electronics*, v. 55, 2008.

[Bertoni, Folgieri e Valentini 2005]BERTONI, A.; FOLGIERI, R.; VALENTINI, G. Biomolecular cancer prediction with random subspace ensembles of support vector machines. *Neurocomputing*, v. 63, p. 535–539, 2005.

[Bishop 2007]BISHOP, C. M. *Pattern Recognition and Machine Learning*. Berlin: Springer, 2007. ISBN 978-0387310732.

[Breiman 1996]BREIMAN, L. Bagging predictors. *Machine Learning*, v. 24(2), p. 123–140, 1996.

[Brown et al. 2005]BROWN, G. et al. Diversity creation methods: a survey and categorization. *Journal of Information Fusion*, v. 6(1), 2005.

[Chen, Lin e Schölkopf 2005]CHEN, P. H.; LIN, C. J.; SCHöLKOPF, B. A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry*, v. 21, p. 111–136, 2005.

[Evgeniou, Pontil e Elisseeff 2002]EVGENIOU, T.; PONTIL, M.; ELISSEEFF, A. Leave-one-out error, stability, and generalization of voting combinations of classifiers. *Machine Learning*, 2002.

[Fawcett 2006]FAWCETT, T. An introduction to ROC analysis. *Pattern Recog. Letters*, v. 27, 2006.

[Freund e Schapire 1996]FREUND, Y.; SCHAPIRE, R. Experiments with a new boosting algorithm. In: *Proc. 13th International Conference on Machine Learning (ICML'96)*. Bari, Italy: [s.n.], 1996.

[Ho 1998]HO, T. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20(8), p. 832–844, 1998.

[Hu et al. 2007]HU, Q. et al. Fault diagnosis of rotating machinery based on improved wavelet package transform and SVMs ensemble. *Mechanical Systems and Signal Processing*, v. 21, p. 688–705, 2007.

[Islam, Yao e Murase 2003]ISLAM, M. M.; YAO, X.; MURASE, K. A constructive algorithm for training cooperative neural network ensembles. *IEEE Transactions on Neural Networks*, v. 14(4), p. 820–834, 2003.

[Kim et al. 2003]KIM, H. C. et al. Constructing support vector machine ensemble. *Pattern Recognition*, v. 36, p. 2757–2767, 2003.

[Kudo e Sklansky 2000]KUDO, M.; SKLANSKY, J. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, v. 33, p. 25–41, 2000.

[Kuncheva 2004]KUNCHEVA, L. I. *Combining Pattern Classifiers*. [S.l.]: Springer, 2004.

[Lei et al. 2010]LEI, Y. et al. A multidimensional hybrid intelligent method for gear fault diagnosis. *Expert Systems with Applications*, v. 37, p. 1419–1430, 2010.

[Li e Tan 2010]LI, S.; TAN, M. Tuning SVM parameter by using a hybrid CLPSO-BFGF algorithm. *Neurocomputing*, v. 73, p. 2089–2096, 2010.

[Li, Wang e Sung 2008]LI, X.; WANG, L.; SUNG, E. Adaboost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence*, v. 21, 2008.

[McInerny e Dai 2003]MCINERNY, S. A.; DAI, Y. Basic vibration signal processing for bearing fault detection. *IEEE Transactions on Education*, v. 46, n. 1, p. 149–156, 2003.

[Mendel, Rauber e Varejao 2008]MENDEL, E.; RAUBER, T. W.; VAREJAO, F. M. Automatic bearing fault pattern recognition using vibration signal analysis. In: *Proc. of the IEEE Int. Symp. on Ind. Electronics ISIE 2008*. [S.l.: s.n.], 2008.

[Miller et al. 2001]MILLER, K. R. et al. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, v. 12, p. 181–201, 2001.

[Opitz e Maclin 1999]OPITZ, D.; MACLIN, R. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, v. 11, p. 169–198, 1999.

[Opitz 1999]OPITZ, D. W. Feature selection for ensembles. In: *Proc. of the Sixteenth National Conf. on Artificial Intelligence AAAI'99*. [S.l.: s.n.], 1999.

[Sun, Zhang e Wang 2007]SUN, B.-Y.; ZHANG, X.-M.; WANG, R.-J. On constructing and pruning SVM ensembles. In: *Proc. of the 2007 IEEE Conf. on Signal-Image Technologies*. [S.l.: s.n.], 2007.

[Theodoridis e Koutroumbas 2006]THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition, Third Edition*. Orlando, FL, USA: Academic Press, Inc., 2006. ISBN 0123695317.

[Tsymbal, Pechenizkiy e Cunningham 2005]TSYMBAL, A.; PECHENIZKIY, M.; CUNNINGHAM, P. Diversity in search strategies for ensemble feature selection. *Information Fusion*, v. 6, p. 83–98, 2005.

[Tu et al. 2007]TU, C.-J. et al. Selection using PSO-SVM. *IAENG International Journal of Computer Science*, v. 33, p. 111–116, 2007.

[Valentini e Dietterich 2000]VALENTINI, G.; DIETTERICH, T. G. Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods. *The Journal of Machine Learning Research*, 2000.

[Vapnik 1998]VAPNIK, V. *The Nature of Statistical Learning Theory*. New York: Wiley, 1998.

[Verikas et al. 2010]VERIKAS, A. et al. Selecting features from multiple feature sets for SVM committee-based screening of human larynx. *Expert Systems with Applications*, v. 37, p. 6957–6962, 2010.

[Wandekokem et al. 2011]WANDEKOKEM, E. D. et al. Diagnosing multiple faults in oil rig motor pumps using support vector machine classifier ensembles. *Integrated Computer-aided Engineering*, v. 18, 2011.

[Widodo e Yang 2007]WIDODO, A.; YANG, B. S. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, v. 21, 2007.

[Zio, Baraldi e Gola 2008]ZIO, E.; BARALDI, P.; GOLA, G. Feature-based classifier ensembles for diagnosing multiple faults in rotating machinery. *Applied Soft Computing*, v. 8, p. 1365–1380, 2008.

APPENDIX A – Attached reference

Attached is the paper "Diagnosing multiple fault in oil rig motor pumps using support vector machine classifier ensembles", Integrated Computer-Aided Engineering, vol. 18, 2011.

Diagnosing multiple faults in oil rig motor pumps using support vector machine classifier ensembles

Estefhan Dazzi Wandekokem[1]*, Eduardo Mendel[1], Fabio Fabris[1],
Marcelo Valentim[1], Rodrigo J. Batista[2], Flávio M. Varejão[1], and Thomas W. Rauber[1].

[1]Department of Computer Science, Federal University of Espírito Santo
Av. Fernando Ferrari s/n, 29075-910 Vitória, ES, Brazil
phone: + (55) 27 3335-2654; fax: + (55) 27 3335-2850;
email: {estefhan, emendel, ffabris, marcelo, fvarejao, thomas}@inf.ufes.br
[2]Espírito Santo Exploration and Production Business Unit
Petróleo Brasileiro S.A. PETROBRAS, Av. Fernando Ferrari 1000, 29075-973 Vitória, ES, Brazil,
email: rodrigojb@petrobras.com.br

Abstract.
We present a generic procedure for diagnosing faults using features extracted from noninvasive machine signals, based on supervised learning techniques to build the fault classifiers. An important novelty of our research is the use of 2000 examples of vibration signals obtained from operating faulty motor pumps, acquired from 25 oil platforms off the Brazilian coast during five years. Several faults can simultaneously occur in a motor pump. Each fault is individually detected in an input pattern by using a distinct ensemble of support vector machine (SVM) classifiers. We propose a novel method for building a SVM ensemble, based on using hill-climbing feature selection to create a set of accurate, diverse feature subsets, and further using a grid-search parameter tuning technique to vary the parameters of SVMs aiming to increase their individual accuracy. Thus our ensemble composing method is based on the hybridization of two distinct, simple techniques originally designed for producing accurate single SVMs. The experiments show that this proposed method achieved a higher estimated prediction accuracy in comparison to using a single SVM classifier or using the well-established genetic ensemble feature selection (GEFS) method for building SVM ensembles.

1 Introduction

The detection and diagnosis of faults in complex industrial machines is advantageous for economical and security reasons [7]. The early detection of a fault allows damaged components to be repaired during planned maintenance, which minimizes machinery downtime besides providing more secure operations. Recent progress in sensor technology and computational intelligence permit the construction of powerful diagnostic systems, which can aid relatively unskilled operators in making reliable decisions about the machine condition as well as providing valuable information to experts in making decisions about intricate fault occurrences.

A single reliable diagnosis procedure for any type of fault based on noninvasive signals is still not established [1]. Noninvasive monitoring relies on easily measured signals, for instance electrical and mechanical quantities like current, voltage, flux, torque and speed. In this work we present a generic procedure for diagnosing faults using features extracted from machine sig-

nals. Our approach is based on the *supervised learning* [2] classification paradigm as the primal mechanism to automatically generate fault classifiers in a model-free context. This has as an advantage the requirement of a minimum of a priori knowledge about the plant, as the fault predictor is automatically defined based on the training data, which allows the diagnosis procedure to be easily extended to many types of equipments, faults and sensors.

Supervised learning based diagnosis requires the use of a large number of labeled examples of each fault category in order to build a classifier with a good generalization capacity. An important novelty of our research is the use of data from real-world operating industrial machines instead of using data from a controlled laboratory environment which is almost always found in the literature (see for instance [34]). This is highly desirable, as laboratory hardware cannot realistically represent intricate real-world fault occurrences. We work with 2000 examples of vibration signals obtained from operating partially faulty motor pumps, acquired from 25 oil platforms off the Brazilian coast during five years. After extensive analysis, human experts provided a label for every fault present in each acquired example, relying on their practical experience in maintenance engineering.

Several faults can simultaneously occur in a motor pump. We formulate the fault diagnosis problem as a *multi-label* [26] classification task in which several labels (fault categories) may be simultaneously assigned to a pattern; in this context, a pattern represents the signals of a motor pump and a label represents a specific fault category. Each fault is individually detected in an input pattern by a distinct binary predictor. Specifically, each fault category is detected by a distinct *ensemble* [14] of *support vector machine* (SVM) [4] classifiers. The SVM classifier is currently considered one of the most powerful binary classification techniques; to further increase the accuracy of a single SVM we use an ensemble of SVMs, composed of accurate SVMs that disagree on their predictions as much as possible. An SVM ensemble assigns a pattern $\mathbf{x}$ to the positive class ω_{pos} or to the negative class ω_{neg}, with the positive class meaning that the fault considered by the ensemble is present in the pattern $\mathbf{x}$ and the negative class meaning that this considered fault is not present in $\mathbf{x}$ (but other faults may be present). We build a fault predictor able to diagnose six fault categories, so it is composed of six independent ensembles of SVM classifiers, each ensemble considering the occurrence of a different fault.

We propose a novel method for building an accurate SVM ensemble. By now very few papers have investigated SVM ensembles based on varying the feature set of the classifiers besides also varying their SVM parameter value. It can be expected that using different feature subsets and SVM parameter values might increase the divergence among the SVMs in an ensemble, therefore increasing the ensemble accuracy. We propose a novel method for constructing an SVM ensemble, based on using hill-climbing *feature selection* [11] to create a set of accurate, diverse feature subsets, and further using a grid-search *parameter tuning* technique to vary the parameters of SVMs aiming to increase their individual accuracy. Thus our ensemble composing method is based on the hybridization of two distinct, simple techniques originally designed for producing accurate single SVMs. The experiments show that this proposed method achieved a higher estimated prediction accuracy in comparison to other well-established approaches for building ensembles.

The remainder of this paper is organized as follows. Section 2 is concerned with the motor pump equipment, the considered fault categories and the extracted features. Section 3 details feature selection. Section 4 describes the SVM classifier. Section 5 presents the ensemble approach for classification. Section 6 details the proposed method for building SVM ensembles based on varying both the features and the parameters of the classifiers. In section 7 we show the experimental results achieved by the studied classification models. Finally, section 8 draws conclusions and points out future research.

2 Model-free approach to motor pump fault diagnosis

There are two main approaches to the machine fault diagnosis problem: model-based techniques and model-free techniques. The model-based approach relies on an analytical model of the studied process, involving time dependent differential equations. In this

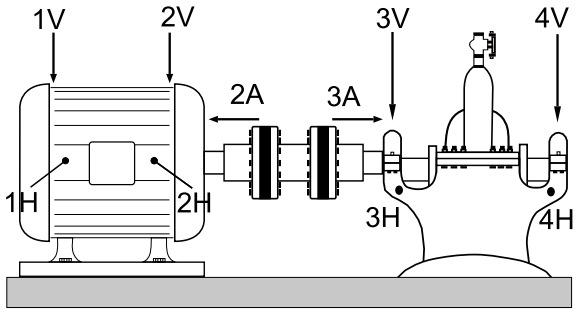

Figure 1: Horizontal motor pump with extended coupling between the motor and the pump. Accelerometers are placed along the main directions to capture specific vibrations of the main axes (H=horizontal, A=axial, V=vertical.)

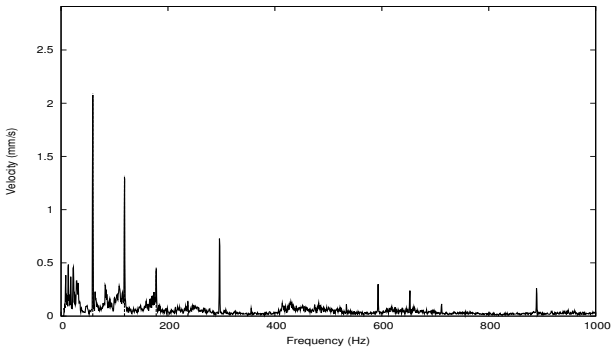

Figure 2: Misalignment fault and its manifestation in the frequency spectrum at the first three harmonics of the shaft rotation frequency. The high energy in the fifth harmonic, as well as the noise in low frequencies indicate that additionally a hydrodynamic fault is emerging.

case, usually the experimental process setup is installed on a controlled laboratory environment and is embedded in a control loop in which inputs, controlled variables and sensor outputs are modeled. However in real-world processes the availability of an analytical model is often unrealistic or inaccurate due to the complexity of the process. In this case model-free techniques are an alternative approach [6]. This paper is concerned with model-free diagnosis of multiple faults in motor pumps, relying on supervised learning based techniques.

2.1 Motor pump equipment

Rotating machinery covers a wide range of mechanical equipment and plays an important role in industrial applications. In this work we focus on a specific rotating machine model, namely the horizontal motor pump with extended coupling between the electric motor and the pump. Accelerometers are placed at strategic positions along the main directions to capture specific vibrations of the main shaft which provides a multichannel time domain raw signal. Figure 1 shows a typical positioning configuration of the accelerometers on the equipment.

2.2 Considered fault categories

Several faults can simultaneously occur in a motor pump. Such a high diversity of defects has a direct impact on the subsequent classifier. For instance, 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.

We build a predictor for individually detecting each of the following six fault categories in an input pattern: 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.

Figure 2 illustrates how the frequency spectrum is associated with two of the faults, presenting the Fourier spectrum of the vibration signal (measured from the position three, horizontal direction) of a faulty motor pump with a misalignment fault and also an emerging hydrodynamic fault.

Every example in the database of 2000 machine signal acquisitions presented the occurrence of at least one fault. Examples presenting the occurrence of multiple faults are more common than examples in which just one fault is occurring. Table 1 shows, for each of the six considered fault categories, the percentage of the 2000 examples that presented this fault.

Table 1: Fault occurrence.

Fault class	A priori class distribution
Misalignment	42.6%
Hydrodynamic	42.4%
Bearing	35.7%
Unbalance	24.9%
Structural looseness	21.2%
Mechanical looseness	12.0%

2.3 Extracted features

The first step to diagnose the faults in a motor pump is to *extract* a global feature vector G to describe many relevant aspects of the current motor pump condition. The general strategy is to provide as much information as possible in the feature extraction stage and further use feature selection and ensemble construction to prioritize more relevant features. So the feature vector G should be composed of features extracted from distinct, complementary information sources, for instance electrical current, chemical, thermal, and mechanical vibration sensors. Also, for each information source, different signal preprocessing techniques should be used, thus producing features extracted from different domains that can reflect complementary perspectives. For instance, for mechanical vibration sensors, the extracted features can include [16]: time domain statistical features such as mean, root mean square (RMS), variance, skewness and kurtosis; frequency domain features such as the amplitude of the spectrum and the energy in the frequency band considered by a feature; and time-frequency domain features obtained by using advanced time-frequency analysis techniques, such as the empirical mode decomposition (EMD) [16] and the wavelet [22] packet transform (WPT) [16].

In this work we build a diagnostic system that uses vibration signals. We work with well-established signal processing techniques, namely the Fourier transform, envelope analysis based on the Hilbert transform [21] and median filtering. The reason that we use solely vibration signals besides not using other signal sources as for instance electrical current is that the information format of the acquired examples was previously defined only as the frequency and the envelope spectrum of the machine vibration signals. In this context, features correspond to the vibrational energy in a predetermined frequency band of the spectrum.

To build the predictor of a specific fault, an important aspect to analyze is the occurrence of the fault in each of the four machine positions of signal acquisition. For instance, the hydrodynamic fault only occurs in a pump (positions 3 and 4), the occurrence of a misalignment fault is better detected by measuring vibrations close to the shaft (positions 2 and 3), and the mechanical looseness fault usually occurs independently in a motor (positions 1 and 2) or in a pump (positions 3 and 4).

We performed preliminary experiments aiming to define a set of relevant features to be extracted. The cardinality of a feature vector G is 95 regardless of the fault under consideration, with 69 of them from the Fourier spectrum and 26 of them from the envelope spectrum.

2.3.1 Fourier spectrum features

For extracting features from the Fourier spectrum, for the hydrodynamical fault predictor, one feature vector is extracted from the machine position which has the highest total root mean square (RMS) value of the velocity signal selected from position 3 or 4 (the pump); for misalignment, one feature vector is similarly extracted from position 2 or 3 (which are close to the shaft); for unbalance, 3 or 4; and for the bearing fault predictor, from position 1, 2, 3 or 4. On the other hand, for structural looseness and mechanical looseness, one pattern is extracted from position 1 or 2 (motor) and another distinct pattern is extracted from position 3 or 4 (pump). A motor pump is diagnosed as faulty if any of the extracted feature vectors is diagnosed as faulty, so the SVM classifiers of the first group of faults are trained with one pattern per machine signal acquisition, while for the second group of faults a SVM is trained with two patterns per acquisition (each of which independently labeled as belonging to the positive ω_{pos} or to the negative ω_{neg} class).

Most of the Fourier features correspond to the RMS value of frequency bands defined as harmonics of the shaft rotation frequency, which is 60Hz for most of the studied motor pumps (thus in this example 1x means 60

Hz and 1.5x means 90 Hz). We use the RMS value of a 10% large narrow band around each of the following frequencies: 0.5x, 1.0x, 1.5x, ..., 5.5x, obtained for each of the three directions of measurement (horizontal, vertical and axial) (which generates $3 \times 11 = 33$ features, for instance the feature `rms_V_2.0x` corresponds to the vertical direction and to the second harmonic). Besides, we take as features the sum of the RMS value of a 10% large narrow band around the harmonics 1x, 2x, ..., 5x, in each direction (3 features, for instance `sum_harm_X` which corresponds to the axial direction), and also similarly the sum of the RMS value of bands around the inter-harmonics 0.5x, 1.5x, ..., 5.5x (3 features, for instance `sum_interharm_H` which corresponds to the horizontal direction). We also use the RMS value of the noise calculated with the median filtering, in the bands 0x-1x, 0x-2x, 0x-3x, 0x-4x and 0x-5x, in each direction ($3 \times 5 = 15$ features, for instance `noise_0-4x_H` which corresponds to the band 0x-4x and to the horizontal direction). Additionally, we use the 10% large narrow band around harmonics of the pump blade pass frequency (BPF), namely $0.5 \times BPF$, $1.0 \times BPF$ and $2.0 \times BPF$, in each direction ($3 \times 3 = 9$ features, for instance `bpf_1x_X` which corresponds to BPF frequency and to the axial direction). We also use the vibration signal total RMS value, in each direction for the velocity signal (3 features, for instance `total_rms_HV` which corresponds to the horizontal direction) and in horizontal direction for the acceleration signal (1 feature, `total_rms_HA`). Finally we take the total RMS value of the acceleration signal in the horizontal direction, specifically for position 1 or 2 (1 feature, `rms_motor_A`) and also 3 or 4 (1 feature, `rms_pump_A`).

2.3.2 Envelope spectrum features

For extracting features from the Envelope spectrum to compose a pattern, regardless of the fault under consideration, a group of features is taken from position 1 or 2 (selecting the one with the highest total RMS value) and another group of features is similarly taken from position 3 or 4. Each group is composed of features defined as the RMS of 10% large narrow bands around the first, the second and the third harmonics of the bearing characteristic frequencies [21] (BPFI, BPFO, FTF and BSF, each one being a constant value determined by the machine bearing model), in the horizontal direction (the only one available for the Envelope signals), for instance the feature `bpfo_pump_2x` corresponds to the second harmonic of the BPFO frequency and is extracted from the pump (position 3 or 4), and the feature `bsf_motor_1x` corresponds to the BSF frequency and is extracted from the motor (position 1 or 2). We also use as a feature the total RMS value of the vibration signal Envelope spectrum, for instance the feature `rms_pump_E` which is extracted from the pump. Once a group of Envelope spectrum features is extracted from position 1 or 2 and another from position 3 or 4, they result in a total of $2 \times 13 = 26$ features.

3 Feature selection

A central issue in fault diagnosis is the definition of which aspects of the input signals, i. e. features, a fault predictor should analyze. The traditional approach is the manual definition of the used features (see for instance [17]). But the handpicking of the fault descriptive features demands specialized knowledge and can result in predictors with low accuracy, for instance due to multiple coexistent faults [30]. An approach for avoiding the manual definition of the important features relies on the initial extraction of a large, comprehensive feature set, and on the further use of *feature selection* techniques [11] to retain a reduced set of relevant features that are used to form the feature space of a classifier (see for instance [32]). An alternative approach is assigning a different weight to each of the extracted features [15]. In this work we use feature selection techniques to create a SVM classifier ensemble instead of searching for a single accurate SVM.

Feature selection is the process of choosing an optimized subset of features for classification from a larger set that may contain irrelevant and redundant information. The two common approaches to feature selection are the filter and the wrapper methods. The former assess the saliency of feature subsets from data properties, without training a classifier. The latter uses the learning algorithm itself to estimate the usefulness of features by evaluating classifiers which use the candidate feature subsets. Wrappers methods are computationally more expensive but usually allow more accurate feature

subsets to be found [24] and thus are used in this work.

Feature selection is composed of two ingredients: the selection criterion and the search strategy. The selection criterion is used to estimate the performance of a feature subset. A suboptimal search strategy is needed since an exhaustive search is not feasible to investigate the space of feature sets.

3.1 Selection criterion

We estimate the criterion $J(X_k)$ of a candidate feature set X_k as the Area Under the ROC Curve (AUC) [10] achieved by a SVM classifier which uses X_k, estimated by cross-validation on the training data.

The Receiving Operating Characteristics (ROC) analysis is very useful for comparing classifiers in problems with unbalanced classes in which negative class examples are usually much more common than positive ones. Table 1 shows that the predictor of some considered fault categories (for instance mechanical looseness) must deal with the unbalance problem. A ROC graph represents a classifier as a point in a two-dimensional space where the true positive rate is plotted on the Y axis and the false positive rate is plotted on the X axis. The threshold of the estimated a posteriori probability of belonging to the positive class ω_{pos} is set to 0.5 by default, producing the X-Y point. By varying the threshold between zero and one, the X-Y point traces the ROC curve and a high area under it indicates an accurate classifier; an AUC value of 0.5 corresponds to a random classifier. So the AUC value can be seen as the probability that, given a positive class example p and a negative class example n, both randomly sampled, the classifier outputs $\hat{P}_{\text{pos}}(p) > \hat{P}_{\text{pos}}(n)$.

3.2 Search strategy

An exhaustive search takes $\binom{|G|}{d}$ attempts to select d features from an available global pool G of $|G|$ features, which is computationally unfeasible in general. Thus we must rely on a suboptimal search strategy.

We use the *Sequential Backward Selection* (SBS) [13] search strategy, which operates based on a hill-climbing greedy search. The SBS strategy allows the more important features to be prioritized, which is useful for building accurate SVMs and thus accurate ensembles. The SBS method starts with every feature (from the global pool G) included in the set of selected features, and at each step one feature is removed from this set. Consider that k features are included in the set of selected features X_k. To remove the worst feature from X_k, each currently selected feature ξ_j must be evaluated by being individually removed from X_k and ranked following the criterion J, so that

$$J(X_k \setminus \{\xi_1\}) \geq J(X_k \setminus \{\xi_2\}) \geq \ldots \geq J(X_k \setminus \{\xi_k\}). \quad (1)$$

As a result of the current exclusion step, the updated selected feature set is given as $X_{k-1} = X_k \setminus \{\xi_1\}$, having $|X_{k-1}| = (k-1)$ features in it. The exclusion process stops when the desired number of features is selected.

4 The support vector machine classifier

The support vector machine (SVM) [4] classification architecture has been extensively used during the last decade in many distinct domains, for instance bioinformatics [33] and machine fault diagnosis [31], and is currently considered one of the most powerful methods in machine learning for solving binary classification problems; SVMs also have been successfully used for regression tasks [20]. We experimentally compared SVM classifiers with Multi-layer Perceptron (MLP) [2] artificial neural network classifiers and found that SVMs achieved a consistent higher accuracy, besides the MLP being computationally much more expensive during training and thus less appropriate to be used for feature selection.

The objective of the SVM training is to create a maximum-margin separating hyperplane that lies in a transformed feature space defined implicitly by a kernel mapping. The hyperplane splits the mapped space into two regions, one associated to the positive class ω_{pos} and the other to the negative class ω_{neg}; a SVM considers a pattern $\mathbf{x}$ as belonging to the positive class ω_{pos} if $\mathbf{x}$ presents the fault category considered by the SVM or as belonging to the negative class ω_{neg} if $\mathbf{x}$ does not present this fault. The distance of a pattern $\mathbf{x}$ to the separating hyperplane, followed by a logistic discrimination, is used to estimate the a posteriori probability $\hat{P}_{\text{pos}}(\mathbf{x})$ that $\mathbf{x}$ belongs to the positive class ω_{pos}.

We use a widely adopted SVM model, namely a Radial Basis Function (RBF) kernel and the C-SVM architecture [2]. So we work with two SVM parameters, namely the regularization parameter C which controls the model complexity and the kernel parameter γ which controls the nonlinear mapping of the features.

The performance of a SVM classifier strictly depends on its parameters. We use an effective, simple method to tune the SVM parameters, namely the *grid-search* on the log-scale of the parameters in combination with cross-validation on each candidate parameter vector. Basically, pairs (C, γ) from a set of predefined values are tried by evaluating SVMs which use them, and the pair that provided the highest cross-validation accuracy defines the best parameters.

We use the `libsvm` library [3] to implement the SVM classification.

5 Classifier ensembles

Combining decisions of multiple accurate, divergent predictors into an ensemble decision is becoming one of the most important techniques to improve classification accuracy [34]. In this context divergence means that each classifier gives erroneous answers in a different region of the global feature space. Creating a so-called *classifier ensemble* entails addressing two issues: the construction of the base classifiers which constitute the ensemble and the combination of their individual predictions. To combine classifier predictions we use an effective, simple method, namely averaging the scores assigned to an input pattern by the classifiers being combined.

The focus of this work is on methods for building a set of classifiers to compose an ensemble. By now, three approaches have become popular for achieving diversity in an ensemble: using a different set of training data for each classifier; using a distinct feature set for each classifier; and setting different values of the classifier intrinsic parameters.

5.1 Data-based ensembles

The classical approach to create a set of classifiers to compose an ensemble relies on using a different training data set for each classifier. For instance, in bagging [2], each classifier in the ensemble samples N training patterns (with equal probability and with replacement) from an available set of N different examples; so the training set of a classifier might not contain some of the available patterns besides containing some patterns that are repeated.

Bagging works well with unstable classifiers, for instance MLP neural networks, in which a small variation of the training data set may cause a large variation of the classifier decision function. But the SVM is a stable classifier, in the sense that a small variation of the training data set tends to cause a small variation of the SVM decision function, as just a reduced subset of the training patterns are retained as support vectors (namely the ones close to the decision boundary). Although some works have reported that SVM ensembles built by using bagging achieved a high prediction accuracy [12], other works have reported negative experiments about bagging based SVM ensembles [8]; see for instance [9] which stated that single SVMs with tuned parameters performed as well as SVM ensembles built by using the bagging method. For the fault diagnosis problem studied in this work we found that the traditional bagging method was not effective to increase the accuracy achieved by a single SVM.

5.2 Feature-based ensembles

A useful approach for building an ensemble is to employ a different feature set for each classifier; see [34] for a reference on fault diagnosis.

Opitz [23] proposed the genetic ensemble feature selection (GEFS) method which relies on a Genetic Algorithm (GA) based search to investigate the space of feature sets. In GEFS, a member of the population (a chromosome) represents the feature set of a single classifier. Starting with randomly defined feature sets, the genetic operators (selection, cross-over and mutation) are used to evolve the population aiming to increase the classifiers fitness. The fitness of a classifier is estimated as a linear combination of its accuracy and its diversity, the latter being defined as the average difference between the prediction of this classifier and the prediction of the current ensemble. At the end of every generation the algorithm outputs the feature subsets of the classifiers in the current ensemble, thus the last generation

defines the final produced feature subsets.

Previous work shows that the GEFS method usually achieves a higher prediction accuracy in comparison to other approaches for building ensembles [28]. Inspired by GEFS, other GA based methods for building feature-based ensembles have been investigated; see [25] for a reference on SVM ensembles.

5.3 Parameter-based ensembles

A natural approach for generating divergence among the decisions of a set of classifiers is the use of different classifier intrinsic parameter values. For instance, one of the first studied ensemble models was an ensemble of MLP neural network classifiers with each MLP having a different number of neurons in its hidden layer. Considering SVM classifier ensembles, varying the kernel parameter (in our case the RBF γ) of a SVM classifier decisively changes its decision function [29], so using different kernel parameter values might allow the construction of divergent SVMs which is useful for building ensembles [19]. For instance, in [27] an SVM ensemble was built with each SVM using a different predefined value for γ (every SVM used the same value for the parameter C).

In this work, we use the grid-search parameter tuning technique to vary the parameters of SVMs aiming to increase their individual accuracy. Although the grid-search method does not directly work aiming to increase a metric of the diversity among the classifiers, we observed that the tuned SVM parameter value assigned to each SVM was likely to be distinct among many of the produced SVMs, as each SVM used a distinct feature subset. Thus this parameter tuning process is useful for building accurate SVMs which are also divergent among themselves.

6 The Best Selected Feature Subsets method

We propose a novel method for building an SVM ensemble, based on varying the features and also the value of the parameters of the classifiers. We call it Best Selected Feature Subsets (BSFS). By now very few papers have investigated SVM ensembles based on varying both features and parameters. It can be expected that using different feature subsets and also parameter values might increase the collective divergence among the SVM classifiers in an ensemble, therefore increasing the ensemble accuracy.

The BSFS method operates as follows. We use complementary SBS feature selection searches combined with the grid-search parameter optimization technique to build a large set $\mathcal{L}$ of classifiers that are candidates to constitute the ensemble. Further we use a sequential forward search to select just a reduced, optimized subset $\mathscr{E}$ of them to compose the final ensemble. This approach of building a large classifier set $\mathcal{L}$ and further searching for a subset $\mathscr{E}$ is known as *overproduce-and-choose* [14].

As we combine the decisions of the classifiers by averaging the assigned scores, an ensemble $\mathscr{E}$ estimates the probability $\hat{P}_{pos}(\mathbf{x})$ that an input pattern $\mathbf{x}$ belongs to the positive class ω_{pos} as the average of the $\hat{P}_{pos}(\mathbf{x})^{c_h}$ classification score values that the classifier c_h in $\mathscr{E}$ outputs for $\mathbf{x}$ (considering every classifier in the ensemble $\mathscr{E}$ i. e. for $h = 1$ to $h = |\mathscr{E}|$). Thus $\mathbf{x}$ is predicted as belonging to ω_{pos} if $\hat{P}_{pos}(\mathbf{x}) > 0.5$ or as belonging to ω_{neg} otherwise.

6.1 The classifier overproduction stage

To create the set $\mathcal{L}$ of candidate classifiers to compose the ensemble, we first build a set Ξ of feature sets composed of several promising feature sets. The feature sets use features from the global pool G and differ on their cardinality. To define Ξ we perform m distinct SBS feature selection searches, $\{\mathcal{S}_1, \ldots, \mathcal{S}_i, \ldots, \mathcal{S}_m\}$, which differ among themselves on the SVM parameter value they use to create SVMs to estimate the selection criterion; this allows feature subsets to be found using complementary kernel mapped feature spaces. We require each search $\mathcal{S}_i$ to select a total of 1 feature (which is equivalent to require $\mathcal{S}_i$ to exclude $|G| - 1$ features); the exclusion of each feature defines a new feature subset and hence a new candidate classifier to compose the ensemble. So the feature sets in Ξ are determined by taking each produced feature set $X_k^{\mathcal{S}_i}$ which uses k features selected by the search $\mathcal{S}_i$, considering every k and i, that is $k = 1, 2, 3, \ldots, |G|$ and $i = 1, 2, \ldots, m$ (thus $|\Xi| = m \times |G|$).

Then the classifier set $\mathcal{L}$ is defined by building, for each feature set $X_k^{\mathcal{S}_i}$ in the set Ξ, a classifier $c_k^{\mathcal{S}_i}$ that uses

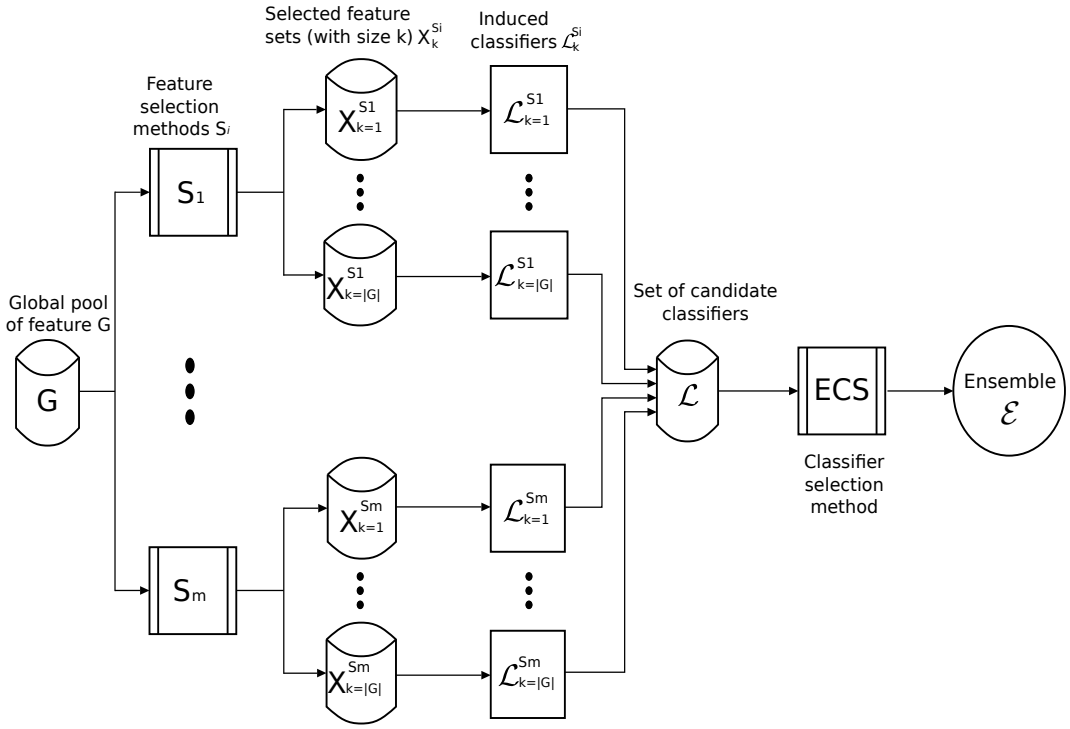

Figure 3: A diagram of the training process of an ensemble built by the BSFS method (used as the predictor of a fault). After the initial overproduction stage every candidate SVM classifier in the set $\mathcal{L}$ uses a distinct feature subset (defined by feature selection) and also tuned parameters. In the Ensemble Classifier Selection (ECS) stage the ensemble is finally built by selecting an optimized subset of classifiers from $\mathcal{L}$.

this feature set, and we use the grid-search method to tune the SVM parameters of every classifier $c_k^{S_i}$ aiming to increase its accuracy. Thus $\mathcal{L}$ is composed of every produced $c_k^{S_i}$, each of which associated to a feature set $X_k^{S_i}$ and to a tuned SVM parameter value (C', γ').

6.2 The Ensemble Classifier Selection (ECS) stage

After building the set $\mathcal{L}$ of candidate classifiers, we use the *Sequential Forward Selection* (SFS) search to select an optimized set of $|\mathscr{E}|$ classifiers to compose the final ensemble $\mathscr{E}$, selecting from $\mathcal{L}$, cf. figure 3. The SFS search operates in a similar way as SBS, but SBS removes objects, while SFS includes objects.

SFS starts with an empty set of selected classifiers, and at each step one classifier is included in this set, namely the one that provided the highest criterion with its individual inclusion in the current set of selected classifiers. We define the criterion J of a candidate ensemble (a subset of classifiers from $\mathcal{L}$) to be the AUC on training data achieved by this candidate ensemble. The score that each classifier in $\mathcal{L}$ gives to a training pattern $\mathbf{x}$ is previously estimated by cross-validation. Thus to obtain the criterion J of a candidate ensemble the score $\hat{P}_{pos}(\mathbf{x})$ assigned by this ensemble to each training pattern $\mathbf{x}$ must be obtained by averaging the scores $\hat{P}_{pos}(\mathbf{x})^{c_j}$ assigned to $\mathbf{x}$ by the classifiers c_j in the candidate ensemble.

The first selected classifier $c_k^{S_i}$ (from $\mathcal{L}$) is the one with the highest individual cross-validation AUC. In the following, each next selected classifier is the currently non-selected one which enabled the highest criterion J

achieved by an ensemble composed of the currently selected classifiers and also this new selected one; thus the second selected classifier is the one which provided the highest criterion for an ensemble of two classifiers, namely the first and the second selected ones. When the desired number $|\mathscr{E}|$ of classifiers are selected, the inclusion process stops, so the ensemble $\mathscr{E}$ is finally built.

7 Experimental results

To assess the effectiveness of the studied classification approaches we performed a stratified 5×2 cross-validation [14]. So in the experiments we performed five replications of a 2-fold cross-validation. In each replication, the complete database of 2000 examples was randomly partitioned, in a stratified manner, into two sets each one with approximately 1000 examples (the stratification process preserves the distribution of the six fault categories between both sets). Then in each replication each considered classification model for creating the predictor of a fault was trained on a set and tested on the remaining one; after the five replications we averaged the ten distinct test accuracies.

7.1 Studied classification approaches

For each of the six considered fault categories, we studied three different classification models for creating the predictor of that fault: a single SVM classifier; a SVM ensemble built using the GEFS method; and an SVM ensemble built using the proposed BSFS method.

7.1.1 The SVM classification model

This classification model is a single SVM classifier. We used the global pool of features G as the feature set, and used the grid-search method to tune the SVM parameters explained in section 4.

7.1.2 The GEFS classification model

This classification model is a SVM ensemble built using the GEFS method. So the feature sets of the classifiers in the ensemble are defined by the last generation of classifiers of the GEFS algorithm. During the ensemble construction (i. e. the GA evolutionary process) we set the SVM parameter values as $(C = 8.0, \gamma = 0.5)$ in order to build SVMs to estimate the fitness; this value was chosen for providing more accurate SVMs in preliminary experiments with the grid-search parameter tuning method. We set the GEFS ensemble size parameter to 20, so the final ensemble produced by GEFS was composed of 20 SVMs, each of which uses a different feature subset (with features from the set G) and using the SVM parameter value $(C = 8.0, \gamma = 0.5)$.

The GEFS algorithm has several parameters [23], and we performed preliminary experiments aiming to find parameter values that provided more accurate ensembles. We set the initial value of the GEFS λ parameter as 1.0, to estimate the initial value of the fitness of a classifier. We set 20 classifiers in the ensemble; in each generation, starting from the ensemble of 20 classifiers, we produce 10 new classifiers by mutation (randomly changing 10% of the features of each classifier) and more 10 new classifiers by cross-over (the two parents of each classifier are randomly selected from the current ensemble, proportionally to the fitness), and from these 40 classifiers the 20 fittest are selected to compose the ensemble at the end of this generation. The population evolved for 200 generations. We used 5-folds cross-validation to estimate the accuracy and fitness of each classifier.

7.1.3 The BSFS classification model

This classification model is a SVM ensemble built using the proposed BSFS method, as showed in section 6. To build the set Ξ of feature sets we ran four SBS feature selection experiments $\{\mathcal{S}_1, \ldots, \mathcal{S}_4\}$, each of which uses a different SVM parameter value to create SVMs to estimate the selection criterion (which was the 5-fold cross-validation AUC). We performed preliminary experiments to define the SVM parameters values to be used. We used the SVM parameter values $(C = 8.0, \gamma = 0.5)$ which provided accurate SVMs. For the other three values, we focused on varying the γ parameter in order to introduce diversity among the SBS searches, allowing the selection of accurate feature subsets from different mapped feature spaces. So we used a low, a medium and a high value for the γ parameter; for the C parameter we used a high value, as it may cause some overfitting which is useful for

increasing the diversity [5] (we used $C = 30000.0$ according to experiments with the grid-search method). So the four SVM parameter values $(C = 8.0, \gamma = 0.5)$, $(C = 30000.0, \gamma = 0.002)$, $(C = 30000.0, \gamma = 2.0)$ and $(C = 30000.0, \gamma = 36.0)$ were used to run the four SBS feature selection searches.

After performing the four SBS searches, to obtain the feature sets $X_k^{S_i}$ to form Ξ, we employed, for each SBS search S_i, the feature subsets defined by using each number of selected features from $k = 1$ to $k = |G|$. Thus $|\Xi| = 4 \times 95 = 380$ feature sets.

Then for each feature set in Ξ we built a SVM classifier using the grid-search method to tune its SVM parameters, including this classifier in the set $\mathcal{L}$ of candidates to compose the ensemble. In order to finally select a subset $\mathscr{E}$ of classifiers from $\mathcal{L}$ we applied the SFS search as explained in section 6.2. We set the desired ensemble size $|\mathscr{E}| = 10$ as we observed a general tendency of an AUC decrease with the use of a larger set.

7.2 5×2 *cross-validation estimation results*

Table 2 presents, for each considered fault, the AUC estimated on test data by the 5×2 cross-validation estimation process, achieved by each considered classification model predictor.

A comparison among the studied classification models suggests two main conclusions. First, building an SVM ensemble was an effective method for improving the accuracy achieved by a single SVM. Second, SVM ensembles built by the BSFS method achieved a higher prediction accuracy than SVM ensembles built by the well-established GEFS method

We performed the statistical testing procedure suggested by Dietterich [14] to be used with the 5×2 cross-validation process, aiming to determine whether there is a significant difference of the accuracy achieved by using a single SVM, a SVM ensemble built by BSFS and a SVM ensemble built by GEFS. The level of significance of the statistical test is 0.05. For the GEFS method, for none of the considered fault categories it was possible to accept that the accuracy achieved by the ensemble had a significant difference to the accuracy achieved by a single SVM. On the other hand, for the BSFS method, for the misalignment, structural looseness and hydrodynamic fault the statistical test confirmed that there is a significant difference of the accuracy achieved by an ensemble in comparison to the accuracy achieved by a single SVM.

Table 2: Estimated test data AUC by 5×2 cross-validation

Fault classifier	Single SVM	GEFS	BSFS
Misalignment	0.829	0.852	**0.876**
Bearing	0.909	0.934	**0.942**
Unbalance	0.836	0.866	**0.883**
Hydrodynamic	0.912	0.929	**0.936**
Structural L.	0.873	0.874	**0.918**
Mechanical L.	0.878	0.892	**0.901**

7.3 *Using the BSFS method to build the misalignment predictor*

Figure 4 presents results for the misalignment predictor, for the first pair of train-test data of the 5×2 cross-validation process, considering the use of the BSFS method. Figure 4 presents the AUC estimated during feature selection by two SBS searches. We present for each search, for each number of selected features, the 5-fold cross-validation AUC estimated on the training data (the feature selection criterion), and also the estimated AUC on test data achieved by a SVM using this feature subset and with parameters tuned by the grid-search method. For comparison we also show as a horizontal line the estimated test data AUC achieved by the ensemble produced by the BSFS method.

Figure 4 illustrates that using different SBS searches which differ on their SVM parameter value is able to generate diverse SVMs. The SBS search A used the SVM parameter value $(C = 8.0, \gamma = 0.5)$ and the SBS search B used the value $(C = 30000, \gamma = 0.002)$. As the parameter γ is related to the kernel mapping of the feature space and each search used a different value for γ, the SVMs investigated during the searches tend to be

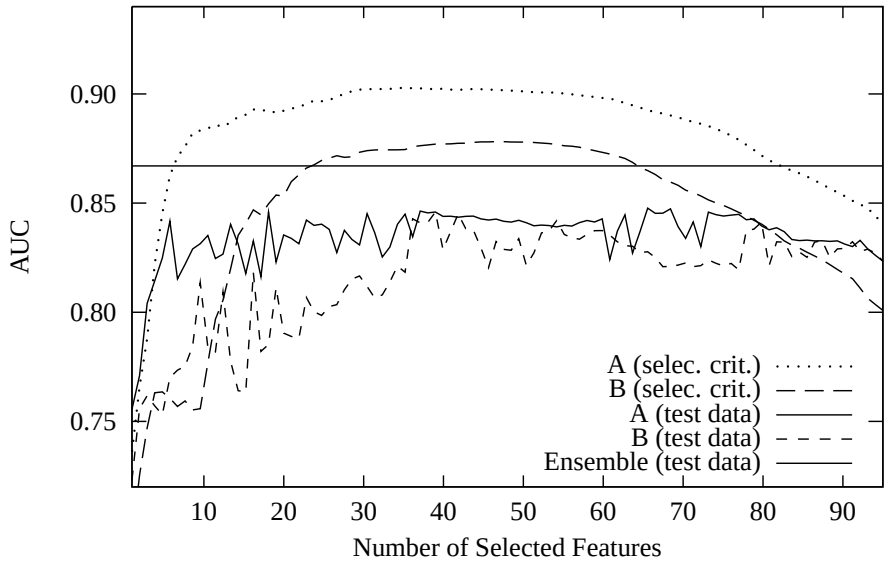

Figure 4: AUC achieved by the SVMs defined during the feature selection searches, estimated on the test data (using tuned SVM parameters) and on the training data (using fixed SVM parameters, to estimate the feature selection criterion). As a comparison we also show as a horizontal line the AUC on test data achieved by the ensemble built using the proposed BSFS method.

distinct, which is suggested by the fact that the AUC achieved by the search *A* was consistently higher than the AUC achieved by the search *B*. So we expected that investigating diverse SVMs might allow diverse feature subsets to be found during the feature selection searches. Figure 4 also shows that the process of tuning the SVM parameters of each produced SVM, besides increasing their accuracy, also tends to increase the diversity among the SVMs, as it can be seen that the test data AUC achieved by the SVMs would strongly vary even among SVMs built using a similar number of features.

Figure 5 is concerned with the Ensemble Classifier Selection (ECS) stage of the BSFS method for selecting SVMs to compose the ensemble, i.e. selecting a subset $\mathscr{E}$ from the classifier set $\mathscr{L}$. Figure 5 presents the selection of the SVMs obtained for the feature selection processes shown in figure 4 and also the other two SBS searches, not shown in figure 4. Figure 5 shows the test data AUC and also the training data cross-validation AUC (which is the selection criterion) achieved by each number of selected SVMs composing the ensemble, from 1 to 70; above 70 classifiers the curve presented a tendency of decreasing slightly. The final ensemble $\mathscr{E}$ was composed of the ten first selected SVMs.

7.4 Useful fault indicators

Aiming to provide an engineering insight into the fault indicators used by the classifiers, we show in table 3 an ordered list of the most common features in the ensemble built for each considered fault by the BSFS method. Thus the firstly listed features are used by a higher number of classifiers in the ensemble than the posteriorly listed ones.

One can observe that the most common features are usually related to characteristic aspects of the considered fault. For instance, for the bearing fault predictor the total RMS energy of the Envelope spectrum of acceleration signal (`total_rms_HA`), the BPF-based

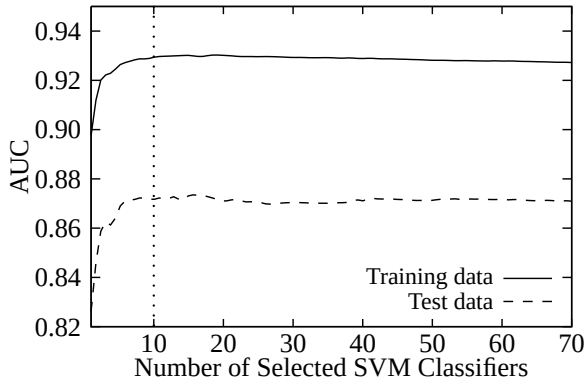

Figure 5: AUC on test and training data for each number of classifiers in the ensemble, from 1 to 70. The selection criterion was the 5-fold cross validation AUC estimated on the training data. The final ensemble was composed of 10 classifiers. An ensemble composed of all 380 produced SVMs achieved a test data AUC of 0.863.

features for the hydrodynamical fault classifier (like `bpf_1x_X` which is around the first harmonic of BPF, in axial direction), and for the mechanical looseness predictor the RMS around 0.5x in vertical direction (`rms_0.5x_V`).

The features listed next are less crucial and can be grouped into two categories. The first one encompasses features describing less characteristic aspects of the fault, which manifest themselves just in a fraction of the training examples, for instance the RMS energy of harmonics above 2x for the hydrodynamic fault predictor (for instance `rms_5x_H`). The second category encompasses features related to the occurrence of other defects and thus used to detect the considered fault in the occurrence of mutually influential defects, for instance the Envelope-based features used by the mechanical looseness predictor which are useful for distinguishing between that fault and a severe bearing defect (for instance `ftf_pump_3x` which is the RMS energy around the FTF third harmonic, extracted from a pump position). Additional useful features are the median filtering based noise (for instance `noise_0-4x_H` which is the noise in band 0x-4x) and the sum of the RMS from harmonics or inter-harmonics (for instance, respectively `sum_harm_X` and `sum_inter_H`).

We observed that classifiers with reduced feature sets usually presented a good performance, since using the characteristic features of a fault might be sufficient to detect it, for instance in an isolated occurrence of the fault. On the other hand, when a complex combination of faults is happening, some of the classifiers with a higher number of features, namely the ones with a feature set better adapted to the current machine condition, presented a good performance. This suggests that building an ensemble is an interesting approach for dealing with the multiple faults problem, as each classifier can reflect a different perspective (according to its feature set) and thus contribute with a complementary decision.

7.5 Using a knowledge-based system to detect structural resonance

Our approach of extracting a general global pool of features *G* for detecting every considered fault category, with the features representing the vibrational energy in predetermined frequency bands, requires little a priori knowledge about the plant. The experiments show that this approach provided accurate predictors for the six studied fault categories. However, there are other fault categories that may not be effectively detected by using the presented extracted features in the set *G*. To detect these faults, new features should be defined to describe their relevant aspects.

Probably the definition of such new features would demand the system developer to consider intricate, specific aspects of the considered fault, thus increasing the demand of a priori knowledge about the plant. To illustrate the definition of such new features, in this section we show the results achieved by the predictor of another fault category, not considered in the previous sections. For diagnosing the fault of structural resonance, we created a so-called knowledge-based system, which predicts this fault by directly searching its characteristic aspect. So the structural resonance predictor worked based on an if-then-else rule.

Structural resonance is characterized by a very high vibrational energy occurring at a frequency, usually not at a harmonic of the shaft rotation frequency. The

Table 3: Ordered list of the features most used by the classifiers in the ensembles.

Misalignment	Structural looseness	Unbalance	Hydrodynamic	Mechanical looseness	Rolling bearing
rms_H_2x	rms_V_1x	rms_H_2x	bpf_X_1x	rms_V_0.5x	total_rms_HA
rms_V_2x	sum_harm_H	rms_H_1x	bpf_V_1x	bpfo_motor_1x	rms_motor_A
noise_0-5x_V	bpfi_motor_1x	rms_V_0.5x	bpf_H_1x	rms_H_1	bpfo_pump_1x
bsf_motor_2x	rms_X_3.5x	noise_0-2x_H	total_rms_V	bpfi_pump_2x	ftf_motor_2x
bpfo_motor_3x	rms_V_0.5x	bpfi_pump_1x	rms_V_3x	bpfi_motor_1x	rms_pump_E
rms_X_2x	noise_0-5x_H	rms_H_2.5x	bpfi_pump_3x	bpfo_motor_3x	bpfi_motor_1x
rms_H_0.5x	rms_V_5x	rms_X_1x	rms_V_2x	noise_0-5x_H	bsf_motor_1x
noise_0-4x_V	rms_H_1x	bpfi_motor_3x	noise_0-1x_X	ftf_motor_3x	bpfo_motor_2x
total_rms_VV	noise_0-4x_H	rms_V_2.5x	rms_X_2.5x	bpf_X_0.5x	bpfo_motor_1x
rms_H_1x	bpf_V_2x	noise_0-1x_V	rms_V_4.5x	bpf_V_2x	rms_V_2x
rms_H_6.5x	bpf_H_0.5x	sum_interharm_X	noise_0-2x_H	bpf_H_3	rms_H_0.5x
noise_0-4x_H	total_rms_V	sum_harm_V	noise_0-1x_H	rms_H_3x	rms_pump_A
sum_interharm_X	total_rms_H	bpfi_motor_2x	ftf_pump_1x	rms_H_2.5x	bpfi_pump_2x
sum_interharm_V	bpfo_pump_2x	bpfi_motor_2x	sum_harm_X	ftf_motor_2x	rms_X_0.5x
sum_harm_X	rms_V_2x	bpfo_pump_4x	sum_harm_H	bpf_H_1x	rms_V_3.5x
sum_harm_V	rms_H_4.5x	bpfo_pump_3x	rms_pump_A	sum_harm_H	rms_V_2.5x
total_rms_HA	rms_H_5.5x	rms_X_3.5x	bpfi_pump_3x	rms_pump_E	rms_H_2x
bpfi_motor_2x	noise_0-3x_H	rms_X_2.5x	rms_X_2.5x	noise_0-5x_V	rms_H_5.5x
bpfo_motor_1x	ftf_motor_2x	rms_X_5x	rms_X_3x	noise_0-3x_V	noise_0-5x_H
rms_X_1x	bsf_pump_3x	rms_H_3x	rms_X_2x	noise_0-3x_H	ftf_motor_3x

source of such a high vibrational energy is external to the motor pump equipment, for instance a damaged equipment that is located close to the motor pump. Figure 6 shows the Fourier spectrum of the vibration signal obtained from a motor pump in which structural resonance is occurring. It can be seen that this fault category is naturally predicted by a rule-based classifier. In fact, no feature in the set G might carry specific information about an energy peak likely to occur at any frequency of the spectrum (and not usually at a harmonic or a subharmonic frequency).

The fault of structural resonance was present in 9.1% of the data. The rule-based predictor achieved a test data accuracy of 94.3% in predicting this fault (estimating by the 5×2 cross-validation); the test data AUC value could not be estimated since the structural resonance predictor did not provide scores for the input patterns as it directly predicts a pattern as belonging to the positive class ω_{pos} or to the negative class ω_{neg}.

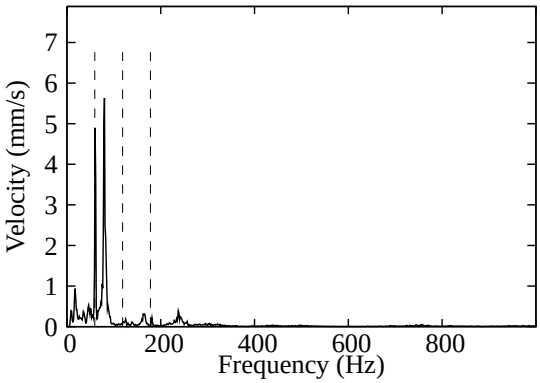

Figure 6: Structural resonance fault and its manifestation in the frequency spectrum. The defect is characterized by a high vibrational energy at the frequency 80 Hz. The energy around 1x (the first harmonic at the leftmost dashed line) indicates a coexistent unbalance fault.

8 Conclusions and Future Work

We presented a novel method for building an accurate ensemble of SVM classifiers, aiming to construct a fault predictor to diagnose six different fault categories in the vibration signals of oil rig motor pumps. This method is based on the hybridization of two distinct, simple techniques originally designed for building accurate SVMs, namely hill-climbing sequential feature selection and grid-search SVM parameter tuning. We use an overproduce-and-choose strategy, first building SVMs which differ on their feature set and also on their SVM parameter, and further searching for an optimized subset of the produced SVMs. The experiments show that such SVM ensembles achieved a higher prediction accuracy in comparison to using single SVM classifiers or using SVM ensembles built by the well-established GEFS method.

To further increase the prediction accuracy, we plan to study more powerful approaches for tuning the parameters of the SVM classifiers. This might produce SVMs that are more accurate and also more divergent among themselves in comparison to the SVMs produced by the grid-search parameter tuning method. Specially, Particle Swarm Optimization (PSO) based techniques have been successfully used to tune SVM parameters [18].

We plan to acquire more real-world data, from different machines and from more sources than just vibration signals. Thus we plan to develop a multiparametric diagnostic system, which uses vibration signals complemented with electrical signals such as current, power and torque. This should increase the prediction accuracy as more information sources will be available to compose the global pool G of extracted features.

Acknowledgments

We would like to thank Petrobras for the financial support of this research and for providing the examples used for training our classifiers.

References

[1] A. Bellini, F. Filippetti, C. Tassoni, and G.-A. Capolino. Advances in diagnostic techniques for induction machines. *IEEE Transactions on Industrial Electronics*, 55, Dec. 2008.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, Berlin, 2006.

[3] P. H. Chen, C. J. Lin, and B. Schölkopf. A tutorial on ν-support vector machines. *Applied Stochastic Models in Business and Industry*, 21:111–136, 2005.

[4] C. Cortes and V. Vapnik. Support-vector network. *Machine Learning*, 20:273–297, 1995.

[5] P. Cunningham. Overfitting and diversity in classification ensembles based on feature selection. *Trinity College Dublin, Dublin (Ireland), Computer Science Technical Report: TCD-CS-2000-07*, 2000.

[6] V. Devedzic. Knowledge modeling - state of the art. *Integrated Computer-Aided Engineering*, 8, 2001.

[7] G. Dounias, G. Tselentis, and V. S. Moustakis. Machine learning based feature extraction for quality control in a production line. *Integrated Computer-Aided Engineering*, 8, 2001.

[8] T. Evgeniou, L. Perez-Breva, M. Pontil, and T. Poggio. Bounds on the generalization performance of kernel machine ensembles. In *Proc. of the Seventeenth International Conference on Machine Learning (ICML 2000)*, 2000.

[9] T. Evgeniou, M. Pontil, , and A. Elisseeff. Leave-one-out error, stability, and generalization of voting combinations of classifiers. *Machine Learning*, 2002.

[10] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 2006.

[11] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.

[12] Q. Hu, Z. He, Z. Zhang, and Y. Zi. Fault diagnosis of rotating machinery based on improved wavelet package transform and svms ensemble. *Mechanical Systems and Signal Processing*, 21:688–705, 2007.

[13] M. Kudo and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33:25–41, 2000.

[14] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Springer, 2004.

[15] H. Lee, E. Kim, and M. Park. A genetic feature weighting scheme for pattern recognition. *Integrated Computer-Aided Engineering*, 14, 2007.

[16] Y. Lei, M. J. Zuo, Z. He, and Y. Zi. A multidimensional hybrid intelligent method for gear fault diagnosis. *Expert Systems with Applications*, 37:1419–1430, 2010.

[17] B. Li, M. Y. Chow, Y. Tipsuwan, and J. C. Hung. Neural-network-based motor rolling bearing fault diagnosis. *IEEE Transactions on Industrial Electronics*, 47(5):1060–1069, 2000.

[18] S. Li and M. Tan. Tuning SVM parameter by using a hybrid CLPSO-BFGF algorithm. *Neurocomputing*, 73:2089–2096, 2010.

[19] X. Li, L. Wang, and E. Sung. Adaboost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence*, 21, 2008.

[20] H. Lian. No-reference video quality measurement with support vector regression. *International Journal of Neural Systems*, 19, 2009.

[21] E. Mendel, L. Z. Mariano, I. Drago, S. Loureiro, T. W. Rauber, and F. M. Varejao. Automatic bearing fault pattern recognition using vibration signal analysis. In *Proc. of the IEEE International Symposium on Industrial Electronics ISIE 2008.*, 2008.

[22] A. Mohammed, R. Minhas, Q. Wu, and M. Sid-Ahmed. An efficiient fingerprint image compression technique based on wave atoms decomposition and multistage vector quantization. *Integrated Computer-Aided Engineering*, 17, 2010.

[23] D. W. Opitz. Feature selection for ensembles. In *AAAI '99/IAAI '99: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence*, pages 379–384, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.

[24] N. Sanchez-Marono, A. Alonso-Betanzos, and R. M. Calvo-Estevez. *A Wrapper Method for Feature Selection in Multiple Classes Datasets*. Springer, Berlin, 2009.

[25] J.-D. Son, G. Niu, B.-S. Yang, D.-H. Hwang, and D.-S. Kang. Selecting features from multiple feature sets for SVM committee-based screening of human larynx. *Expert Systems with Applications*, 37:6957–6962, 2010.

[26] A. F. D. Souza, F. Pedroni, E. Oliveira, P. M. Ciarelli, W. F. Henrique, L. Veronese, and C. Badue. Automated multi-label text categorization with vg-ram weightless neural networks. *Neurocomputing*, 72:2209 – 2217, 2009.

[27] B.-Y. Sun, X.-M. Zhang, and R.-J. Wang. On constructing and pruning SVM ensembles. In *Proc. of the 2007 IEEE Conf. on Signal-Image Technologies*, 2007.

[28] A. Tsymbal, M. Pechenizkiy, and P. Cunningham. Diversity in search strategies for ensemble feature selection. *Information Fusion*, 6:83–98, 2005.

[29] G. Valentini and T. G. Dietterich. Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods. *The Journal of Machine Learning Research*, 5, Jan. 2000.

[30] E. D. Wandekokem, F. T. A. Franzosi, T. W. Rauber, F. M. Varejão, and R. J. Batista. Data-driven fault diagnosis of oil rig motor pumps applying automatic definition and selection of features. In *Proceedings of the 7th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives*, Cargèse, France, August 2009.

[31] A. Widodo and B.-S. Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21, 2007.

[32] B. S. Yang, T. Han, and J. L. An. Art-kohonen neural network for fault diagnosis of rotating machinery. *Mechanical Systems and Signal Processing*, 18:645–657, 2004.

[33] Y. Yang and B. Lu. Protein subcellular multi-localization prediction using a min-max modular support vector machine. *International Journal of Neural Systems*, 20, 2010.

[34] E. Zio, P. Baraldi, and G. Gola. Feature-based classifier ensembles for diagnosing multiple faults in rotating machinery. *Applied Soft Computing*, 8:1365–1380, 2008.