IT 10 035
Examensarbete 15 hp
Juni 2010

Mining unexpected behaviour from equipment measurements

Paolo Pareti

Institutionen för informationsteknologi
Department of Information Technology

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Mining unexpected behaviour from equipment measurements

Paolo Pareti

Modern physical systems tend to have a high level of complexity that hinders the efficiency of human-based condition monitoring. Automatic and intelligent strategies, on the contrary, easily outperform the human expertise in terms of speed, accuracy and scalability. Focusing on faults, probably the most critical issue in condition monitoring, this paper presents a selected survey on the data-driven Fault Detection and Diagnosis (FDD) field analysed from a data mining perspective.
Data pre-processing is identified as a fundamental step to reach satisfactory results in the FDD process. In this respect, Empirical Mode Decomposition, Wavelet and Walsh transforms are effective signal transformation tools. Principal Component Analysis and Fisher Discriminant Analysis are often used for feature reduction.
Machine Learning techniques, such as Support Vector Machines and Neuro-Fuzzy are used to solve the core tasks of FDD, namely classification and novelty detection. Genetic Algorithms and Swarm Intelligence methods are usually applied for parameter optimization for the above mentioned techniques.
It has also been observed that a particular approach, namely fault classification, is the most common FDD strategy. However, since it requires supervised learning, it is limited to applications where supervised data is available.

Handledare: Tore Risch
Ämnesgranskare: Tore Risch
Examinator: Anders Jansson
IT 10 035
Tryckt av: Reprocentralen ITC

INDEX OF CONTENTS:

1) INTRODUCTION

While physical engineering systems (such as industrial machineries) grows in complexity, human expertise is always less capable of managing them without the use of automatic intelligent systems. To substitute the human expertise, another source of knowledge has to be used. The massive amount of operational data produced in the above mentioned systems (for example equipment measurements), is seen as the most promising source of such knowledge. First of all, this kind of data is easy to obtain. It is usually formed by raw data that can be collected in large amounts from sensor measurements. Furthermore, while human knowledge is limited, the amount of information contained in operational databases usually scales with the complexity of the systems where it comes from.

This problem will be analysed from an information technology perspective. To reduce the scope of this research to manageable dimensions, it will be limited to the Fault Detection and Diagnosis (FDD) problem for physical systems. Software FDD or other related problems (such as quality optimization) are beyond the scope of this research.

This thesis presents a selected survey on data driven FDD from a data mining point of view. After a first chapter that describes the ideas behind FDD, the application of data mining techniques in this context will be discussed. To provide an overview of the current state of art in this field, the following chapters will then describe among the most recent data driven FDD techniques developed. Among those, this research selects the most popular techniques, those that better exploit the knowledge hidden in the operational databases.

2) FAULT DETECTION AND DIAGNOSIS

2.1) OVERVIEW ON FAULT DETECTION AND DIAGNOSIS

Fault detection and diagnosis is a research field dedicated to automate the process of discovering faults and diagnose their causes in physical systems by extracting the relevant information from raw data. It is a fundamental process of every engineering system nowadays.

In some cases, faults are common events across the lifetime of physical machinery. Due to deteriorating forces, this kind of faults may happen regularly after a machine has reached a certain age. Being common, it is often easy to categorize these faults into well-known categories. Classification tools have proven to be very reliable FDD techniques and their strength lies in the ability of exploiting supervised data. Supervised learning consists in approximating a function that, given a data point as input, generates a particular output. When a dataset containing input output pairs is available for a particular task, supervised learning tools can be used to learn the input/output mapping.

Other times, faults are infrequent, unexpected, and very little knowledge about them is available before. These faults are not-trivial to detect and, once detected, it is generally very hard to diagnose their exact type and magnitude. Novelty detection is the standard solution to deal with this kind of faults. However the lack of prior knowledge makes it difficult to design an accurate detection system without increasing too much the false-alarm rate.

In any case, faults are a major issue that has to be dealt with very carefully. If not detected/diagnosed in time, faults can generate serious consequences such as damage to other machines, waste of products due to unacceptable quality or even injury to humans.

2.2) A GLOSSARY OF FUNDAMENTAL TERMS IN FDD

Here it follows a definition of the most important concepts in data driven FDD:

- *Fault*. A fault can be defined as an anomalous deviation from the normal behaviour in a system or in one of its sub-parts.

- *Fault detection*. This is the indication that a fault is present in a system. Solving the fault detection problem means developing a technique to reliably and quickly identify the occurrence of a fault.

- *Fault diagnosis*. This process involves the extraction of information that is necessary to deal with an already detected fault to restore normal working conditions.

While this is the term that will be generally used in this survey, sometime, in the literature, different terms are used. For instance fault isolation, fault identification or fault classification. The definition of those terms does not seem to be very clear and their usage is sometime inconsistent among different papers. However the most common definition of those terms will be here presented.

- *Fault isolation*. It refers to the location of the fault in complex system (for instance, which machine is faulty).

- *Fault identification*. It refers to the quantification of the magnitude of the fault.

- *Fault classification*. It refers to the categorization of a fault into well-known categories. If a fault is successfully categorized, prior-knowledge about the fault category can be used to deal with the fault.
From these definitions, fault isolation, identification and classification can be considered sub-problems of fault diagnosis. It has to be noted that in most applications the diagnosis process consists only in isolation or classification. In fact, fault isolation or fault classification are processes that, on their own, are capable of generating enough information to deal with the fault. For this reason many authors uses the terms of fault isolation or classification as synonyms of fault diagnosis.

- *Raw data*. The data directly obtained from the sensors.

- *Quality data*. This term refers to raw data that has been pre-processed. The most important pre-process steps are: feature reduction techniques (such as Principal Component Analysis) and feature extraction techniques (such as Fourier or Wavelet transforms). Noise reduction techniques are used as well, but their importance is usually secondary in FDD.

- *Healthy data*. This term refers to raw or quality data obtained monitoring the process while no faults are present. This type of data is practically always indispensable since it is used to identify the normal behaviour of the system. Since the system is healthy most of the time, it is very easy to accumulate large amounts of this type of data.

- *Faulty data*. This term refers to raw or quality data obtained monitoring the process while one or more faults are present. Since faulty conditions are not common, it is generally hard to obtain large amount of faulty data. Some time, no faulty data is available at all. If used correctly, however, it is a very valuable resource to use in the FDD process. If the exact type of the fault that was present in the system is known, this type of data can be used for fault classification.

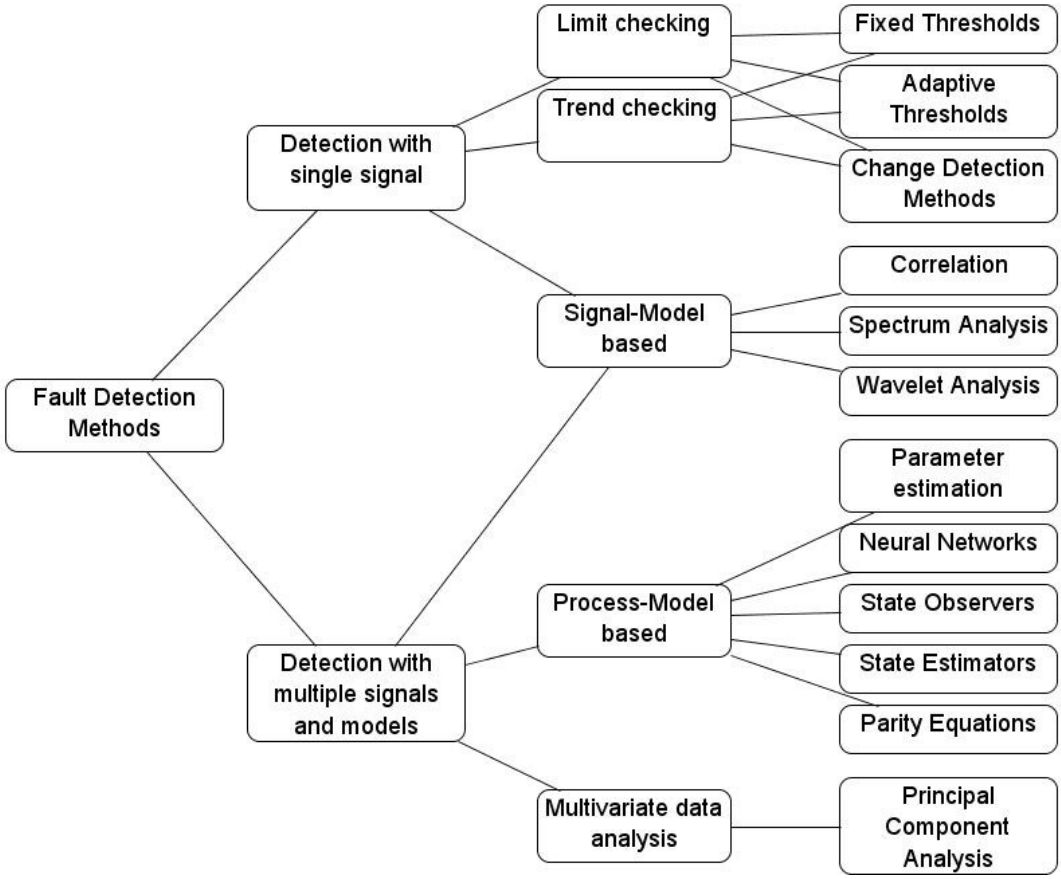
2.3) MODEL BASED AND DATA DRIVEN FDD

One possible categorization of FDD techniques involves a division into two groups: model based and data-driven techniques. The model based approach to FDD was the first to be extensively studied and it is based on a mathematical model of the system. Model based approaches suffer from a number of serious shortcomings:
- they are usually not scalable for high dimensional systems [17],
- a considerable amount of prior-knowledge is necessary to develop and validate the mathematical model,
- in case of very complex systems, the process of constructing the model becomes excessively difficult,
- it is very hard to develop a complete model comprehensive of all possible faults,

Nowadays, the increasing complexity of industrial systems hinders the applicability of model based techniques that cannot scale to many contemporary real-world situations. Data driven methods, on the contrary, can better adapt to high dimensionality and system complexity. A common feature of every data-driven FDD method is, according to [1], that they all use raw data to process the required knowledge. They are mostly based on the analysis of large historical databases.
A basic prior knowledge of the system is still necessary to achieve good performances (for instance to choose the best methods to use). However, an important note is that the amount of prior knowledge required to apply a data-driven FDD strategy does not increase significantly if the target system is very complex. These techniques, therefore, better scale with the system complexity.
Additional case specific prior-knowledge can, of course, be effectively used, but it is not a major requirement of this technique. A better understanding of the system can also be used to better tune the learning parameters, choose the optimal amount of data to be analysed or, eventually, to perform minor modifications on existing methods in order to improve their effectiveness for the specific problem.

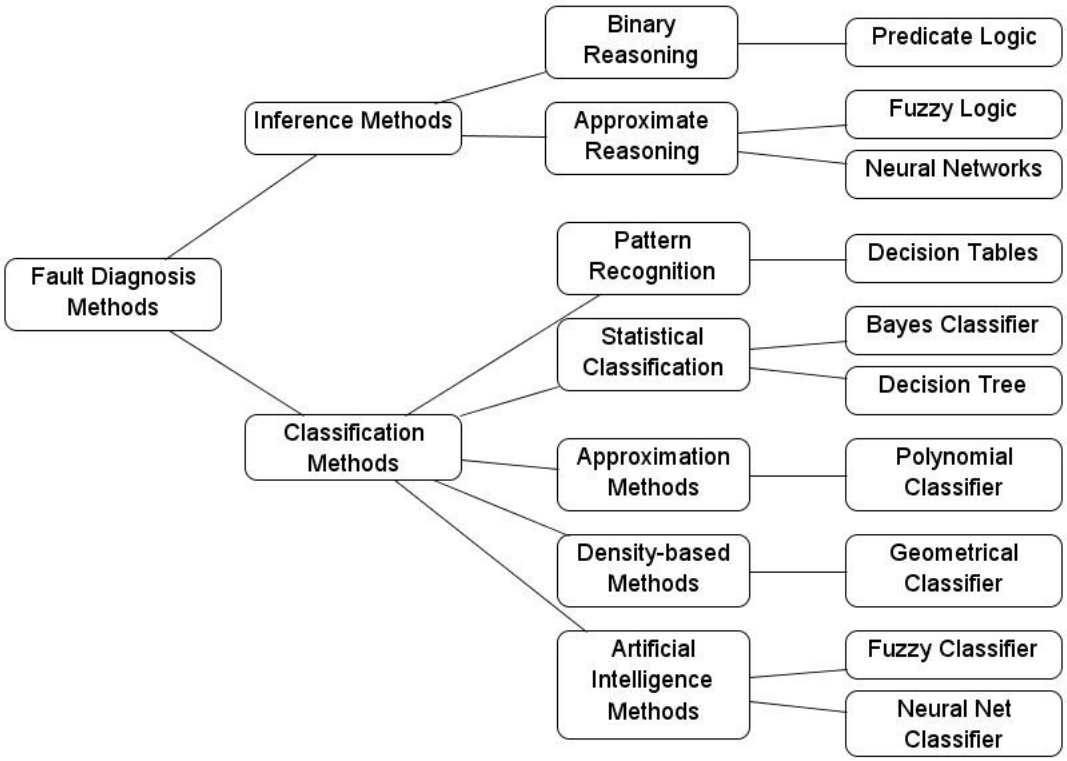
2.4) Hierarchical structure of data-driven FDD methods

A number of different hierarchical structures of the FDD process have been proposed. They are strongly dependent on the criterions that are used to categorize the methods.
In [20] fault detection methods are categorized according to the following scheme, that highlights the division between single and multiple signal detection.



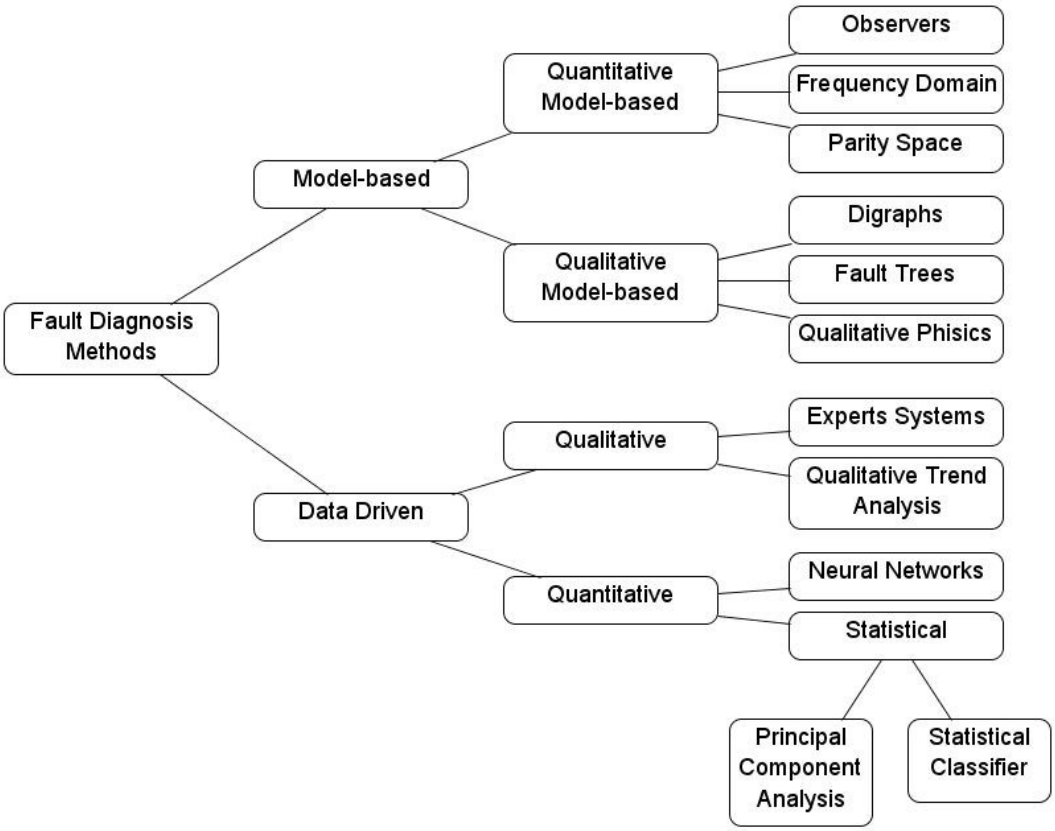
The categories of Signal-Model Based, Process-Model Based and Multivariate Data Analysis contains some of the methods analysed in the present survey. For example Wavelets, Principal Component Analysis and derivates of Artificial Neural Networks.

In [20] a schematic representation of fault diagnosis methods is presented as well:



As we can see from this scheme, classification methods are the most developed branch of fault diagnosis. In particular, fuzzy classifiers, decision trees and derivates of Artificial Neural Networks will be discussed in the present work. Inference methods, on the contrary, will not be analysed here because they were not commonly found in the recent literature.

Another possible categorization of fault diagnosis is presented in [44], as shown below.



In this case, the division of model based and data driven fault diagnosis is highlighted. Furthermore, the methods are divided in quantitative and quantitative techniques. Concerning the relationship between input and output, the quantitative approach models this relationship with mathematical functions. The qualitative approach, instead, uses qualitative functions.

2.5) THREE PHASES TO DESCRIBE DATA-DRIVEN FDD

An effective description of data-driven techniques is presented in [17]. In this paper, the evolution of data-driven FDD methods is described in three main phases. Each phase is represented by a category of techniques that address the FDD problem with a different level of sophistication. These categories will here be presented.

- *Signal based FDD*. The methods in this category are based on signal processing such as Fourier or Wavelet transforms. The basic idea is that an unexpected change in a signal can be interpreted as a sign of a faulty condition. Thresholds can be computed to detect abnormal signals features such as amplitude or frequency in spectral signals.

- *Multi-variable statistics based FDD.* Those methods are a natural extension of signal based FDD. Signal based FDD has the drawback of ignoring the correlation between different signals. In real cases, in fact, the system is highly interconnected and the values of a single signal are usually insufficient to effectively detect a fault. A global view on multiple sensors is then necessary. In Multi-variable statistics based FDD, principal component analysis or partial least squares can be used to reduce the multi-variable data dimensionality. The quality data obtained in such way can be used to calculate the statistical distribution of the healthy data and consequently to identify abnormal data points (possible faults).

- *Knowledge based FDD*. While Multi-variable statistics based FDD can be used effectively for fault detection it does not generate convenient information to effectively diagnose the fault. This is because it has the limit of analysing the data only from a statistical point of view and it does not carry any information related to the system structure. To solve this problem process knowledge is required. In model based FDD, this type of knowledge is usually included in the model. In data driven FDD, it can be obtained from operational data. An example is fault classification that can be carried on by classification algorithms or machine learning approaches such as Artificial Neural Networks or Support Vector Machines. In these cases, the prior-knowledge on the different class faults is the operational data included. Another example is cause-effect analysis using symbolic graphs or fault trees.

3) DATA MINING PERSPECTIVE ON FDD

3.1) A DATA MINING DEFINITION FOR FDD

"Data mining is defined to be the exploration and analysis, by automatic or semiautomatic means, of large quantities of data stored either in databases, data warehouses, or other information repositories to discover interesting knowledge including meaningful patterns and rules."
[8]

"Data mining (DM) is the process of selecting, exploring and modelling large amounts of data in order to discover unknown patterns or relationships which provide a clear and useful result to the data analyst."
[6]

"Data mining is a blend of concepts and algorithms from machine learning, statistics, artificial intelligence, and data management."
[16]

"Data mining is not only database analysis method, but also an important machine learning tool."
[47]

(these definitions are taken from FDD related papers)

In the literature, web sites or other sources, it is possible to find an impressive number of definitions of the data mining concept. For the most part of them, the central idea is the extraction of not-trivial information from data. With a similar definition, two problems arise. Firstly, a neat boundary between "what is" and "what is not" data mining is not established. Secondly, a too wide range of techniques will eventually fall into this definition. Data-driven FDD, here, is a fitting example. This whole research field can be considered data mining. However with such level of generalization, the concept of data mining becomes less meaningful and its use less opportune.

A possible restriction to the general definition of data mining methods is the requirement of having a small computational complexity needed to analyse large amounts of data. Several definitions of data mining, in fact, mentions that the methods should be applicable to large databases. One obvious problem is the sub-definition of the concept "large" in this context. From a human point of view, for instance, even data sets that are considered "small" from a computer perspective are still too large to be effectively analysed by human experts.

This complexity-based restriction to the definition has another problem: a data mining method should at the same time be "intelligent". This is a fundamental requirement. The information discovered should be useful and not-obvious. However, as every algorithm strive for intelligence and computational simplicity at the same time, the no-free-lunch theorem tells that should exist a balance between those two positive qualities. Very scalable techniques, are usually less informative. An example are statistical analysis of equipment measurements that can deal with huge amount of data. Their output, however, can be used for fault detection but is generally not sufficient for a proper diagnose of the fault. On the other hand, strategies to automatically generate the system model or to induce classification rules are generally computationally more expensive.

To achieve good performances both in computational complexity and usefulness the most common strategy is the following. At first, feature extraction techniques are used to produce highly informative data. Then, the dimensionality of the problem is reduced using feature reduction or feature selection techniques. If possible, the length of the data sample is also reduced. The data obtained in this way, being reduced in length and dimensionality, can still be processed even by a computational intensive algorithm in reasonable time.

Not being able to find a main definition of data mining, in the following sections more arguments will be discussed. In particular, to have a broader view on this issue, real examples of data mining approaches to FDD will be presented in the following sections. These examples should give an idea on the usage of the data mining concept in the FDD literature.

3.2) LOW ENTROPY IN EQUIPMENT MEASUREMENTS DATABASES

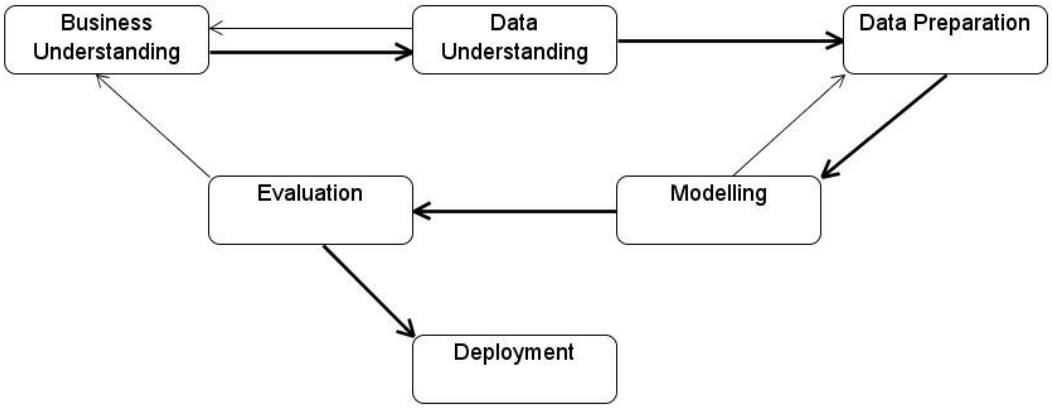
If all the data collected by industrial machinery had to be stored in databases, an impressive amount of data would be available. However, this amount would exceed the real necessity by far. A very straightforward example is the vibration data obtained by a rotating machinery. In vibration data, frequency and amplitude are the most important features to be extracted. They are generally the only features needed to determine if new data samples are healthy or faulty.
Assuming a slow rotating speed, such as 100 Hz, the measurements obtained in a few minutes are enough to extract features such as normal vibration frequency and amplitude in healthy conditions. Measuring this kind of data for extended periods of time, for instance for several hours, will not give significantly better results.
This example is, on purpose, very simple. For instance vibrations could vary during the start-up or shout down phases of the machine or depending on the current temperature or the humidity. In these cases, the data collected in a few minutes could not be enough to perform an accurate classification of the healthy features. Nevertheless this example should give an idea on why, the larger a databases generated from machinery measurements is, the higher probability it has of containing redundant information. Depending on the type of measurement and the type of system where the FDD approach

is used, a smaller or larger data set is needed. An arbitrary increase of the size of the data set will not improve significantly the accuracy of a FDD method used and it will hinder the computational complexity. The strength behind data-driven FDD techniques does not lie in the in the amount of data collected but in the correct choice of the types of data to use.

3.3) A DATA MINING STANDARD: CROSS INDUSTRY STANDARD PROCESS FOR DATA MINING (CRISP-DM)

"Most published research on data mining in manufacturing reports dedicated applications or systems, tackling specific problem areas, such as fault detection (see Fig. 1). Only limited research has been done to address the integration of data mining with existing manufacturing-based enterprise reference architectures, frameworks, middleware, and standards such as Common Object Request Broker Architecture (CORBA), Model-Driven Architecture (MDA), or Common Warehouse Metamodel (CWM).
[16]"

CRISP-DM methodology provides a detailed description of six major phases to develop a data mining technique. The complete life cycle of a data mining project is described but, in real world situations, one or more of the described steps can be skipped if not required.



1) *Business Understanding*. The data mining project should start by defining its goals in terms of business requirements. This specification should then be converted into a data mining problem definition.
2) *Data Understanding*. To effectively operate on the data in the later phases, some knowledge has to be obtained on the characteristics of the data itself. In FDD is very important to detect which kind of data has been collected and what it represents.
3) *Data Preparation*. This is the process of producing quality data from raw data. Typical pre-processing tasks are noise-cleaning, feature extraction and feature selection.
4) *Modelling*. In this phase, a number of data mining techniques are proposed and their parameters are adjusted to the specific problem.

5) *Evaluation*. This stage involves further evaluation of the techniques of sufficient quality. Particular attention has to be directed to possible problems that has not been previously considered. It is also necessary to be confident that the methods will actually deal with the original goals of the project.
6) *Deployment*. This last phase involves the necessary steps to make the user able to exploit the data mining method developed in the previous steps.

3.4) AN EXAMPLE OF CRISP-DM APPLICATION

A recent example of CRISP-DM application in the FDD field is described in [6]. The Business understanding and and the Data understanding phases were not accurately discussed in the paper. This does not mean that those steps are not necessary or less important. Regarding the business understanding, a deep knowledge of the FDD field is required before developing the data mining model. However the purpose and the reasons of FDD are assumed to be well known. The data understanding phase is also essential. This latter step is usually under the form of a general description of data sets and of the system where the FDD will operate.
The first phase to be discussed in [6] is Data preparation. Inaccurate data is detected and removed assuming a normal distribution range. The data is then divided into two subsets: the 80% is destined for training purposes and the remaining 20% for testing. A genetic algorithm has been used for feature selection. As a result, among the monitored parameters, only two are identified as the most effective ones.
In [6], during the modelling phase, several models are created using different techniques. A multilayer perceptron is trained with the backpropagation algorithm. The M5 model tree algorithm is used to induce a tree model. Other approaches used are decision tables, Kstar algorithm, Sequential Minimal Optimization, linear regression and pace regression.
For the evaluation phase, the accuracy of the models was tested computing the RMSE (root mean square error) and R^2 (correlation coefficient). The experimental results show that M5P has the highest R^2 value and the lowest RMSE.
The deployment phase described in [6] involves the development of a java program to predict the level of supply unbalance of the induction motor, the parameter used to detect a fault. Given a new data set, the program can dynamically evaluate the most efficient model and use it to compute the prediction.

3.5) A FOUR STEP DATA MINING FRAMEWORK

In the research described in [8], a data mining framework is proposed to detect and diagnose *yield defects* in semiconductor manufacturing. It involves 4 major steps:
-*Problem definition*. This step shares many similarities with the business understanding step of the CRISP-DM framework. The business problem can be defined as a FDD issue. In the experimental case, the specific problem was fault diagnosis relative to a monitored period of low yield rates.

-*Data preparation*. This step represent the data pre-processing part of the framework. The data is transformed in different formats according to the requirements in the following step. In the experimental case this phase starts with cleaning the data from noise and inconsistent data. The problem of missing values has also to be dealt with. New data is generated, such as labels that identify the time and the machine that generated the data.
-*Data mining*. In the central phase of the framework, data mining techniques are used to identify problems and extract patterns from the data. The approach used from the authors of [8] is the following. K-means clustering algorithm is used to distinguish between low and normal yield lots. Basically, this is the fault detection step. In this case, the yield is mono-dimensional and the result of the K-means algorithm is the threshold of 57% yield. Lots with a lower yield rate are considered proofs of the existence of a fault. Among 71 lots, 12 were classified as low yield lots.
The Kruskal–Wallis test is then applied to each process stage to determine if the output of the related machines are subject to significant variation between low and high yield data. This process gives insight on the process steps (and the related mechanical devices) that are more concerned with the low yield problem. This analysis can also be used to screen not-relevant process phases and improve the efficiency. The system was composed of 455 process stages. This number indicates a very high complexity. After the Kruskal–Wallis was applied, only 168 stages were considered relevant in this context and the remaining 63% was discarded from the analysis.
A decision tree is then developed with the ANOVA F-test splitting criterion using the available data with the task of deciding which machine is probably the root cause of the yield problem. The decision tree created not only identifies the process stage where the fault is supposed to be located, but also the specific machine. In fact its leaf nodes are machine identifiers.
-*Evaluation and interpretation*. This last phase involves the real usage of the information obtained from the previous step. No matter how accurate the results of the data mining process are, in most cases this information has to be evaluated by human experts.
In the experimental example, the results obtained had proven to be very accurate. The engineers managed to detect the machines that had problems very quickly despite the very high complexity of the system.

3.6) AN EXAMPLE OF DATA MINING METHODS FOR SENSORS FDD

A FDD method based on data mining is described in [49]. Has to be noted that the authors of this paper justify the decision of using a data mining approach. The mentioned reasons are two. Firstly, when the volume of the data is too big, conventional database techniques are less practical and harder to use. Secondly, when the future behaviour has a strong similarity to the past behaviour, it is computationally more efficient learning from the historical data compared to "learning from scratch". It is also mentioned that a major advantage of the data mining approach is being able to explicitly

detect changes in a system and consequently in its environment. The authors of this paper mention rough sets, decision trees, clustering methods and artificial neural networks as some of the most important data mining tools.

The specific data mining process presented in [49] addresses the problem of FDD in the sensors used for system control. The complex structure of the monitoring instruments increases the risk of faults in the sensors that can mislead the main control system and induce invalid strategies for fault recovery.
When a sensor is reading results with values too different from the expected ones, it might be labelled as faulty. However, this decision has to be taken considering its environment and this last one is measured using other sensors. Sensor fault detection, therefore, should not be done in isolation but the measurements of other sensors should be taken into account to verify if they justify or not the unexpected valued of a specific sensor.
The techniques used are rough sets and artificial neural networks. Rough sets are used as a pre-processing tool for feature reduction. Multiple artificial neural networks trained with the Levenberg–Marquardt algorithm are used to learn a set of rules to predict the system output. If the predicted value is different from the measured one, a possible fault is detected.

3.7) USAGE OF DATA MINING IN THE FDD LITERATURE

Beside the previous examples, that used the term data mining to describe the framework used, in most of the literature on FDD, the words "data mining" are never mentioned. However is doubtless that several techniques used in FDD should be considered authentic examples of data mining. In the data driven FDD field most of the techniques used can be generally considered, at least, border-line examples of data mining. FDD methods shares many similarities with data mining approaches, but they are rarely good examples of them. This is the reason why the term data mining, although an important concept in this context, will not be used to describe the FDD techniques in the present survey.

4) SIGNAL TRANSFORMATIONS

4.1) THE NEED TO TRANSFORM RAW DATA

The data based FDD process has its roots in the analysis of a certain amount of data, generally acquired from various types of sensors. The data obtained in such way is called "raw" data and it is usually a function of time (it is in time-domain). A few times, raw data can be directly used by the detection or diagnosis algorithms. More often, however, it has to be pre-processed in order to be effectively analysed. One of the most common examples is vibration data. This is an extremely common type of data obtained by sensors in mechanical processes. In its raw form, it is an inconvenient type of data to be analysed by common FDD methods. The reason is that the relevant features (e.g. frequency and peaks of the vibration) are hidden in the raw data and require additional computation to be extracted. The usual solution is to perform a data transformation, that effectively transforms the raw data into quality data where the relevant features are made explicit. The most important transforms will be now described. The information presented in the following paragraphs comes from various essay on transforms such as [27] and from the study of FDD publications that make use of transforms such as [24].

4.2) FOURIER TRANSFORM (FT)

This rather old technique is probably a very common transform. It has been applied extensively in the FDD field. FT identifies which frequencies are present in the signal data. The definition of a FT as an integrable function is given below:

$$Y(f) = \int_{-\infty}^{+\infty} x(t)\, e^{-2j\pi ft}\, dt$$

$Y(f)$ is the amplitude of the FT at a given frequency f. The value $x(t)$ corresponds to the amplitude of the signal at a given time t. The result of a FT can be plotted in a two dimensional graph with frequency and amplitude as dimensions.
The main drawback is that the information produced by the Fourier Transform have no time-dimensionality. Therefore the FT does not carry information about "when" certain frequency has appeared in the signal. This makes this technique not feasible for non-stationary signals when it is relevant the time when specific spectral components occur. In FDD, this is often the case. In fact, not considering the time-dimensionality reduces significantly the amount of information extracted from a signal.

4.3) SHORT TERM FOURIER TRANSFORM (STFT)

Short Term Fourier Transform can be seen as an intermediate step between Fourier and Wavelet Transform. STFT works in the same way as the standard FT, but instead of evaluating a frequency over all the signal data, it is restricted to a specific time-window. The window is mathematically implemented by a window function ω that translates

across all the data as the value of τ changes. The window function is multiplied to the amplitude function x. The effect is that values of the signal far in time from the translation point τ will be multiplied to a value close to 0 and therefore will not contribute significantly to the STFT value. For every frequency and translation point the value of the STFT can be defined by the following function:

$$Y^{\omega}(\tau, f) = \int_{-\infty}^{+\infty} x(t)\, \omega(t-\tau)\, e^{-2j\pi f t}\, dt$$

The result of a STFT can be plotted in three dimensions: frequency, amplitude and time. STFT can be used to analyse non-stationary signals but it has a major drawback that can hinder its performance in a real application: the window function and consequently its width, is fixed. It does not change dynamically during the evaluation of the function. The narrower the window is, the worse frequency resolution it is obtained (the identifiable frequencies are restricted to a limited band). In case of a wide window, on the contrary, we have a poor time resolution. Using this method, therefore, it is not possible to obtain a good time and frequency resolution at the same time.

4.4) BASIC THEORY OF WAVELET TRANSFORM (WT)

In the standard wavelet transform, the window functions are modified (scaled and translated) from an original prototype called mother wavelet. The amplitude value of the WT, in the formula below called $W(\tau, s)$, is dependent of two variables. One of them is the translation value τ of the window across the time-domain. It is dependent on the time and therefore it can be interpreted as a time measure. The second is the scale s of the width of the window. The width of the window determines the limited band of frequencies that the window is capable of detecting. It can be interpreted as a frequency measure because high scales will correspond to low frequencies and low scales to high frequencies. The scale concept in WT, that substitute the frequency concept in SFTS is one of the major differences between those two methods. WTs can be defined by the following formula, where $\overline{this\ notation}$ indicates the complex conjugate:

$$W(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} x(t)\, \overline{\omega * \left(\frac{t-\tau}{s}\right)}\, dt$$

The scale and translation variables, in real computation, are discrete variables. The amplitude function is calculated for every translation point and for every scale that needs to be computed. The main difference between WT and STFT is that the latter one has constant time and frequency resolution. WT, instead, has good time resolution and low frequency resolution for smaller scales (higher frequencies). In real world data, high frequencies are often of brief duration and therefore it is convenient to have a good time resolution to analyse them. Symmetrically, WT has low time resolution and good frequency resolution for larger scales (lower frequencies). Again, this is a convenient choice for real world data, since low frequencies are usually prolonged in time and therefore they don't need a good time resolution to be analysed.

4.5) WAVELET TRANSFORM, DECOMPOSITION AND OPTIMIZATION

It is possible to observe that with larger scales (lower frequencies), we need to compute less translation points compared to the points needed by smaller scales (higher frequencies). This is a consequence of the fact that, to analyse a signal entirely, several window functions with different translation values are needed to cover the whole signal length. The wider the windows are (the greater is the scale), the less of them is required to cover the whole signal. Therefore the number of translation points to be considered can be reduced dynamically as the scale increases.
Discrete Wavelet Transform is the most common type of WTs applied in real computations. Besides having discrete translation and scale values, it uses a filtering and subsampling process. Using a half band lowpass filter, the highest half of the frequencies are removed. Then the signal is subsampled by two (one sample point every two is discarded). Subsampling by two has the same effect as doubling the window size. This procedure, and the complementary half band highpass filter, are used to decompose a signal into two signals, each one of half the length of the original one, and each one composed only by a half of the frequencies included in the original one. The signals generated in such way can be further decomposed. Discrete Wavelet Transform can also be used for data reduction if the decomposed parts of a signal that carry too little information are deleted.

4.6) APPLICATIONS OF WAVELET TRANSFORM IN FDD

WT can be applied to a considerably broad range of tasks in FDD. The authors of [24] presents an overview on the usage of WTs in FDD. In this paper, four major successful application areas of WTs can be identified.

- *Fault feature extraction*. Using the decomposition method, a signal is decomposed in several coefficients. The coefficients with low values can be discarded, the ones that carry significant information can be used as major features to analyse in the FDD process. For example, they can be used as highly-informative features for fault classification. This feature extraction technique can considerably enhance fault detection, as well.

- *Singularity detection*. It can be observed that the singular and irregular components in a signal (such as peaks or discontinuities) are the ones that carries the most information about possible faults. Wavelets can be used to highlight and detect singularities. In [24], the wavelet modulus maxima method is described as a standard for this category of problems.

- *Denoising and extraction of the weak signals*. A common problem in signal analysis for FDD is the unavoidable presence of noise. It has been also noted that, some time, relevant information for FDD is located in weak signals components. This type of components have a high risk of being ignored, especially when standard denoising

techniques are used. The wavelet approach for denoising is different from the standard filtering strategy that consider noise the frequencies outside a certain range. From a wavelet perspective, noise is identified on the base of the amplitude of the coefficients, that is usually low for noise, and high even for weak signal components. The application wavelets for denoising is considered very successful in the FDD field from the authors of [24].

- *Vibration signal compression*. The need for data compression might not seem a central issue in FDD. For practical purposes, nevertheless, it is a problem that has to be considered. High sampling rate is very common in machinery vibration sensors, and a massive amount of data has to be stored for extended periods of time. Effective data compression can reduce the cost of maintaining such memory. In addition, the digital transmission (e.g. internet) of such data is sometime required. An excessively massive amount of data will make this process unachievable in reasonable time. Distributed or remote diagnosis systems, for example, require good performances in data transmission.
Wavelets are described in many sources as an outstanding tool for signal compression. The reason is that a large percentage of the coefficients calculated have a low amplitude value and can be discarded.

4.7) RECENT EXAMPLES OF WAVELET TRANSFORM IN FDD

The number of papers in the FDD literature that use WT-based techniques is really considerable. Among the most recent examples, wavelets has been used to solve the FDD problem in an induction motor in [34]. Discrete Wavelet Transform with 5 Daubechies decomposition levels is used to extract information on the differences between faulty and normal signals. In this paper, the Wavelet theory has also been used to develop a Support Vector Machine with a wavelet based kernel function. This last approach, however, proved to be not very accurate in clustering the classes. The final results, nevertheless, showed a high accuracy level in classification but the time required to compute them was intensive.
In [10] a similar study was conducted. Discrete Wavelet Transform was used on 50,000 training samples with seven decomposition levels. The authors mention that an extensive knowledge of signals is required to use this technique. The choice of the mother wavelet and the main frequency supply is, for instance, critical to obtain good performances.

4.8) WALSH TRANSFORM

Walsh transforms are a non-sinusoidal type of transforms. An example of its application in FDD is discussed in [42]. Two main advantages of Walsh transforms over most typical transforms such as FT or WT, are presented in this paper. Firstly, this type of transform has a small computational complexity. Its complexity is $O(m \log m)$ where m is the number of samples. Secondly, sinusoidal transforms such as FT have difficulties in analysing impulse signals. Walsh transform, on the contrary, is a suitable tool to analyse this kind of signals.

In [42], an experiment is conducted using different methods, namely Walsh Transform, Fourier Transform, Daubechies Wavelet and Symlets Wavelet. The problem addressed is FDD for rolling element bearings. Fault diagnosis is based on decision rules extracted using Rough Set Theory. The results of this experiment shows that the diagnosis accuracy obtained using Walsh transform is some time inferior and some time superior to the accuracy obtained by other methods. While not superior in accuracy, Walsh Transforms are proven to have a significant better algorithmic efficiency compared to the other methods. The authors claims that the proposed method that combines Walsh Transforms and Rough Set Theory has to be preferred to the other methods in case the computational efficiency is a major issue of the FDD problem.

4.9) HILBERT–HUANG TRANSFORM AND THE EMPIRICAL MODE DECOMPOSITION

Hilbert–Huang transform is a common tool in signal processing and its applications to FDD are numerous. It is composed of two steps: the empirical mode decomposition (EMD) and the Hilbert spectral analysis.

Empirical Mode Decomposition is a technique to decompose a signal into its components. The description of this method comes from the one provided by [19]. The principle behind EMD is that a complex oscillatory signal can be decomposed in a relatively small number of simple oscillatory components. These components are called intrinsic mode functions (IMF) and are defined by the following two statements: 1) in the whole dataset, the number of extrema and the number of zero-crossings must either be equal or differ at most by one, 2) at any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero.

An IMF differs from a simple harmonic functions because its peak amplitude and frequency can vary in time. Given a signal $x(t)$, a way to calculate an IMF is the following.

- Identify the points of local maxima and minima of the signal x.
- Connect all the local maxima by a cubic spline line $e^{max}(t)$ so that $\forall\, i, e^{max}(i) \geq x(i)$. More precisely, $e^{max}(i)$ is equal to $x(i)$, if and only if i corresponds to a local maxima in x. In a similar way, connect all the local minima by a cubic spline line $e^{min}(t)$ so that $\forall\, i, e^{min}(i) \leq x(i)$. More precisely, $e^{min}(i)$ is equal to $x(i)$ if and only if i corresponds to a local minima in x. The line computed among the maxima (or minima) points does not have to be necessary a cubic spline line. This line, however, has proved to be the most efficient in experimental results.
- Define the mean function $m(t) = \frac{e^{min}(t)+e^{max}(t)}{2}$.
- Calculate the first component $h(t)$ as the difference between the signal and the mean function: $h(t) = x(t) - m(t)$. This sifting process can be repeated several times, considering iteratively $h(t)$ the new signal. This process highlights low amplitude waves, eliminate riding waves and makes the profiles of the waves more symmetric. This iterative process terminates according to a certain criterion and the result is a IMF $c_1(t)$.

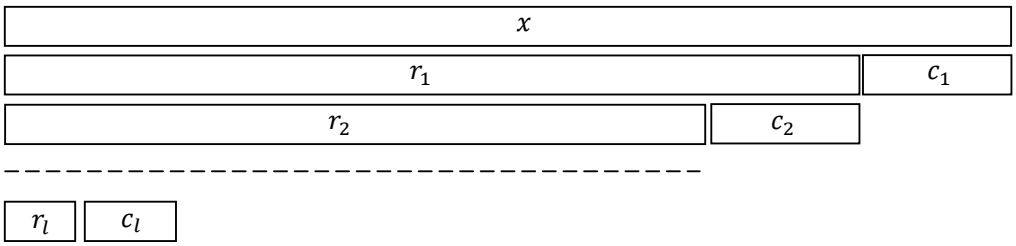
- To calculate more IMF, this process has to be repeated considering the residual $r(t)$ as the new signal. The residual can be defined as: $r(t) = x(t) - c(t)$.
The number of local extrema in the residual is reduced after each iteration. This guarantees the termination of this process in a finite number of steps and the extraction of a finite number l of IMFs. It stops when the residual becomes a monotonic function or when it becomes too small to further extract meaningful IMFs.
The final residue should represent the global trend of the signal. Summing up the final residue with all the IMF extracted should reconstruct the original signal:

$$x(t) = \sum_{i=1}^{l} c_i(t) + r_l(t)$$

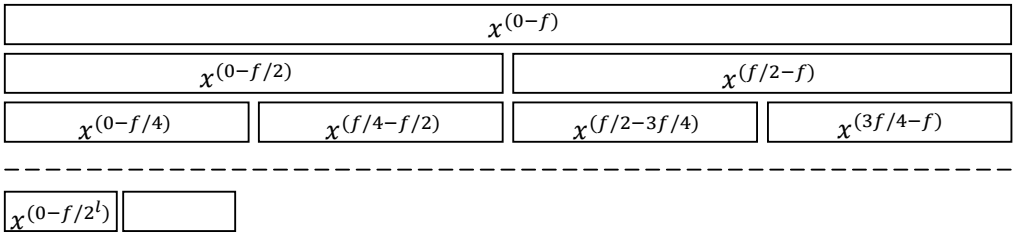
After the EMD process is completed, the Hilbert spectral analysis can be computed. The result is an amplitude function dependent on frequency and time.

4.10) COMPARISON BETWEEN WT AND EMD IN FDD

In EMD, the decomposition involves the extraction of components from a signal generating a residual. A graphical representation of the EMD decomposition is shown below:



Compared to WT, the decomposition method is very different. WT decomposes a signal in frequency bands, and not in IMFs. In the image below, a graphical representation of a wavelet decomposition with l levels is shown:



The WT's decomposition is dependent on the mother wavelet function chosen and on the limit to the decomposition level. In EMD, the decomposition is more *self-adapting* meaning that its characteristics, for instance the number of levels, are mostly dependent on the specific signal. While still dependent on the characteristics of the signal, a stopping criterion has to be chosen to limit the number of decompositions.
EMD is considered a better signal processing tool than WT if the signal shows a non-stationary and non-linear behaviour [36]. The type of decomposition used in EMD, in addition, is very convenient to accurately evaluate the energy of a particular component.

In WT, the components are not as well separated and this task result more difficult and less accurate. For example, the energy entropies calculated on the same signal in [43], gave different results between the EMD and the WT approach. Furthermore, in the same example, EMD recognized more difference in energy entropy between faulty and healthy data. Using WT, instead these two types of data appeared less differentiable.

4.11) EXAMPLES OF APPLICATIONS OF EMPIRICAL MODE DECOMPOSITION IN FDD

In FDD, while WT is also used for feature and data reduction, EMD is mostly used in feature extraction. Similarly to WT, EMD is a powerful tool to analyse signals decomposing them into basic components whose changes from faulty and healthy data can be easily identified. In [43], both EMD and WT are used for gear FDD. The authors justify this choice asserting that these two transforms analyse the signal from two different perspectives and each of them brings different advantages to the analysis. The 14 features considered came from the first three decomposition levels of the WT (8 features) plus the first 6 IMFs. The transformed data was used to train a hybrid intelligent system based on multiple classifiers such as Radial Basis ANN, Multi Layer Perceptron ANN and KNN algorithm. On the fault classification problem the hybrid system achieved a very high classification accuracy, 98.33%, significantly superior to the one obtained by the individual classifiers. In [46], a EMD is used for roller bearing FDD in combination with a SVM. The method described can be summarized as follows. At first, $4n$ data points are gathered collecting n data samples for each one of the following 4 conditions, namely: normal working conditions, out-race fault, inner-race fault, roller fault. The data extracted in such way is decomposed by the EMD in a number of IMFs. Only the most informative IMFs are selected. On those, the Hilbert transform is performed and their envelop spectrum is calculated. The envelop spectrum is then used to evaluate their characteristic amplitude ratios. Multiple Support Vectors Machines (a better description of this concept will be given in section 6.2) are generated to solve the fault classification task. The Support Vector Machines are trained using the amplitude ratios as features and the results shown a high classification accuracy.

5) FEATURE REDUCTION

5.1) FEATURE REDUCTION IN FDD

In data-driven FDD, feature reduction is often a very important pre-processing technique. In complex systems, it is possible to measure a large number of different features from different sensors. This leads to the problem of an excessively high dimensionality of the data. High dimensionality, not only decrease the computational speed of the FDD algorithms, but can also hinder their accuracy. In fact, it has often been noticed that, without a proper feature reduction phase, relevant knowledge becomes hard to extract from the data.
Despite the large number of raw features, in practice, only a relative small subset of the data is relevant for FDD purposes. Selecting the most informative features is, in general only a partial solution. The features discarded might carry a little but relevant amount of knowledge. Superior performances can be obtained projecting a subset of the initial features into new features vectors. These new vectors are chosen according to some criterion to maximize the amount of relevant knowledge they contain. In this way, all the original features can contribute to the final, smaller set of new features. Feature reduction methods like Principal Component Analysis and Fisher Discriminant Analysis are commonly used for these purpose.

5.2) PRINCIPAL COMPONENT ANALYSIS (PCA)

PCA is a fundamental statistical tool in data driven FDD. Among its advantages, PCA is able to process massive amount of data thanks to its efficiency.
Assuming a n-dimensional data, a few important concepts of Principal Component Analysis will here be summarized.
- *Covariance matrix*. The covariance matrix is a $n \times n$ matrix calculated in such way that, given two indices i and j, the value $x_{i,j}$ (or $x_{j,i}$) is the covariance between the ith dimension and the jth dimension for the given data.
- *Eigenvectors*. Using the covariance matrix n eigenvectors are calculated. These vectors are n-dimensional and orthogonal to each other.
- *Eigenvalues*. For each eigenvector an eigenvalue can be calculated. If the data points had to be cast in the dimensionality indicated by the corresponding eigenvector, its eigenvalue indicates, intuitively, the amount of variance that the data points would have. More intuitively, the eigenvalue indicates how "good" a vector would be if it had to be used as a dimensionality.
The eigenvectors are ranked from the highest eigenvalue, the principal component, to the lowest. According to the same principle, the eigenvectors with the lowest eigenvalues can be discarded. Casting the data only into the remaining p eigenvectors, the dimensionality of the data is reduced. If the eigenvalues of the eigenvectors discarded is low, the information lost in the dimensionality reduction is low.

- *Feature vector*. A feature vector is a $n \times p$ matrix with one of the remaining eigenvectors for every column. The dimensionality reduction step is done multiplying two matrixes. The first one is the feature vector transposed to have the eigenvectors in the rows and not in the columns (the matrix becomes $p \times n$). The second matrix is made having a data point for each column. If the data points are l, the matrix is $n \times l$. The result of this matrix multiplication is the n-dimensional dataset reduced to p dimensions.

5.3) STANDARD APPLICATIONS OF PCA IN FDD

A standard application of PCA in typical FDD problems is described in [11]. This process can be divided in three steps. The data samples are assumed to be n and their dimensionality to be m. If no data pre-processing is performed before the use of PCA, the dimensionality should correspond to the number of sensors in the system.
- *Data collection and normalization*. The n data samples are stored in a matrix with dimensionality $n \times m$. This matrix is scaled to 0 mean (i.e. the mean of every dimensionality is scaled to 0) generating a new matrix X.
- *Threshold identification*. The covariance matrix is formed and then the principal components are identified. Singular Value Decomposition is a typical way of doing it. Using the principal components, thresholds for the Squared Prediction Error and the Hotelling T^2 distribution are computed. This is just an example. These methods can, of course, be substituted by others.
- *Fault detection*. New measurements are scaled and the Squared Prediction Error and the Hotelling T^2 distribution values are computed. If both those values are outside the thresholds, a fault is detected.

5.4) FISHER DISCRIMINANT ANALYSIS

Fisher's Discriminant is a dimensionality reduction technique particularly suitable for classification tasks. This technique is data-driven and its effectiveness strictly depends on the quality and the quantity of the training data. A Fisher Discriminant aims to make the classes as separable as possible. This is a difference compared to PCA. PCA, in fact, strives for the ability to represent the data with a smaller dimensionality. However this last technique is not concerned about the division of the classes.
In the following formulas, two data classes C_1 and C_2, and a training set $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 \dots \boldsymbol{x}_{n_1+n_2}$ of d-dimensional data points will be considered for the classification problem. More specifically, we will assume that n_1 data samples $\in C_1$ and that n_2 data samples $\in C_2$. A linear combination of the dimensions can be defined as follows.

$$y_i = \boldsymbol{w}\, \boldsymbol{x}_i$$

Using this combination, the data is projected in $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3 \dots \boldsymbol{y}_{n_1+n_2}$. The projected data is still divided into two classes represented by the two sets Y_1 and Y_2. The mean of the data of class j (in this example j is either equal to 1 or to 2) is defined as follows.

$$\boldsymbol{m}_j = \frac{1}{n_j} \sum_{\forall \boldsymbol{x} \in C_j} \boldsymbol{x}$$

It is possible to calculate the mean of the projected points as follows.

$$m_j = \frac{1}{n_j} \sum_{\forall y \in Y_j} y = \frac{1}{n_j} \sum_{\forall \boldsymbol{x} \in C_j} \boldsymbol{w}\boldsymbol{x} = \boldsymbol{w}\boldsymbol{m}_j$$

The distance δ between the mean of the classes is defined as follows.

$$\delta = |m_1 - m_2|$$

Intuitively, the greater the value of δ is, the better the classes are separated by the linear discriminant. However, the more the data is scattered in the projection, the worse the separation is. The value of scatter s_j can be defined as follows.

$$s_j = \sum_{\forall y \in Y_j} \left(y - m_j\right)^2$$

Using the previous definitions, we define the objective function G as follows.

$$G(\boldsymbol{w}) = \frac{\delta^2}{(s_1 + s_2)}$$

The identification of the $\boldsymbol{w}$ that maximize the value of $G(\boldsymbol{w})$ is the principle behind Fisher Discriminants. In practice, however, further mathematical formulations are used in its algorithmic implementation. A *scatter matrix*, for example, is used to define $G(\boldsymbol{w})$ as an explicit function of $\boldsymbol{w}$.

5.5) Applications of Fisher Discriminant analysis in FDD

An application of Fisher Discriminant analysis in FDD is presented in [7]. At first, Genetic Algorithms are used to identify the optimal variables to use in the computation of the Fisher Discriminant. This technique has been tested in the Tennessee Eastman benchmark problem. After a feature selection phase, Fisher Discriminants and SVMs are used for classification. The misclassification rate of SVMs is found to be around 6%. Fisher Discriminant Analysis, however, reached only a 18% misclassification rate. This result, in [7] pointed to the observation that Fisher Discriminant Analysis is not able to adapt very well to non-linear problems.

To overcome this drawback, in [48], the standard Fisher Discriminant Analysis is extended with the use of kernels to adapt to a non-linear distribution of the classes. This technique is, again, applied to the Tennessee Eastman benchmark problem. This time, the misclassification rate of most of the faults was evaluated inferior to 5%.

5.6) The use of decision trees in FDD, the C4.5 algorithm.

Decision trees are a common type of classifiers particularly appreciated in the FDD field for the interpretability of their decisions. In fact, a classification task using a decision tree is performed using a set of if-then rules. This allows the decision tree to motivate its

output. A characteristic not present in all the classification techniques. The Artificial Neural Networks are a standard example of not interpretable reasoning process.
The C4.5 algorithm is a statistical classifier. It is fed by a set of already classified training samples and its output is a decision tree. A main idea exploited by this algorithm is the entropy gain. Given an attribute x, its entropy gain can intuitively be described as the difference between the amount of information needed to classify an element of the training set and the amount of information needed to perform the same classification task, but in the case the attribute x is known. In every node of the decision tree, the attribute with the highest entropy gain is the one selected to form the decision rule.
Many applications of this algorithm can be found in the literature. For example, the C4.5 algorithm has been successfully used in [31] for fault classification in a chemical plant case study. In this work, this algorithm has demonstrated a high classification accuracy while keeping the classification rules few and simple.
In [4] another example of a decision tree classifier is presented. This research used the J48 algorithm for fault classification in FDD of bearing defects. J48 is a Java open-source implementation of the C4.5 algorithm. J48 achieved a satisfactory classification accuracy (larger than 93%).

5.7) A NOTE ON THE C4.5 ALGORITHM AS FEATURE SELECTOR.

Based on the entropy gain principle, the C4.5 algorithm has been proposed for feature selection in a few papers on FDD. In each node of the decision tree generated by the C4.5 algorithm, a feature is used to evaluate the decision rule. If a node is close to the root in the decision tree, the feature it uses has a high discrimination ability over the given classification problem. On the other hand, if a feature is used in a node far from the root, it is probably a bad classification feature. Furthermore, features that are not used at all by the decision tree should be discarded as they are not necessary to solve the classification problem.
This use of the C4.5 algorithm is documented in [12]. At first the training set was composed by 11 features. Then the classification accuracy of a Bayesian Net was evaluated considering only the top-x features according to the C4.5 generated decision tree. Using only the first feature (the one used in the root) a classification accuracy of 73.33% was reached. A significant difference was noted adding a second feature, the accuracy ratio reached the 87.22%. Adding more features, from the second to sixth feature added, in general increased the accuracy, but not significantly. Then, from the sixth feature on, the accuracy slightly decreased. After this experiment, the 6 top-most features in the decision tree were identified as the optimal set of features to use in the classification problem.
In [12] and in related works, C4.5 has been proven to have feature selection abilities. However this is a side effect of the entropy gain principle and this algorithm is not meant to be used as a feature selector. In fact in papers like [25] the efficacy of this strategy is discussed and several drawbacks are identified. For example, only in the root node the best feature is selected. In any other node, the choice of the best feature is

dependent on the subset of attributes that the node can use. It has also been noted that the choice of the parameters is strongly dependent on the size of the data set. In general, C4.5 algorithm will not lead to an optimal feature selection and dedicated techniques are considered much more effective for this task.

6) MACHINE LEARNING APPROACHES

6.1) MACHINE LEARNING IN FDD

In the field of Fault Detection and Diagnosis (FDD), machine learning approaches are fundamental techniques. The analysed papers show that a fundamental step of fault diagnosis, that is fault classification, is usually based on machine learning techniques. These techniques, in fact, are very suitable for this kind of tasks and in practice they can easily outperform statistical classifiers. Artificial Neural Networks (ANN), though the most typical example of this kind of methods, will not be discussed in this survey. ANN is a traditional approach and nowadays, in the field of FDD, it is rarely used in one of its typical implementations. Having a "long" history, a large number of different techniques have been developed either altering the standard ANNs algorithms or being inspired by some of their principles. It is believed that the description of similar but more promising techniques would be more informative in this context. In particular, Support Vector Machines and Fuzzy-Networks shares many similarities with ANN but have proven to be significantly more effective.
Machine learning techniques can be used as novelty detectors for fault detection as well. However, for this task, other very effective strategies exist.
Many machine learning methods have the drawback of requiring parameter tuning. It is often the case that the final outcome of a machine learning algorithm is strongly dependent on the initial setting of the parameters. Evolutionary computing algorithms, such as Genetic Algorithms or Swarm Intelligence are common techniques to solve this kind of problem, which is usually formulated as an optimization problem. Exceptionally, they can also be used for main tasks in FDD. Ant Colony Classifier is an example of it.

6.2) SUPPORT VECTOR MACHINE (SVM) IN THE LAST DECADE

Support Vector Machine is a supervised learning method that has been recently used and studied intensively in the field of FDD. In the ScienceDirect database, dozens of articles were found concerning SVM in FDD. The number of those papers is given in the table below. Though these numbers are not comprehensive of all the literature about SVMs in FDD, they gives an idea on their distribution over the last decade.

Number of papers on SVM in FDD (in the ScienceDirect database).

Year	Papers
2000	1
2001	0
2002	1
2003	1
2004	3
2005	4
2006	3
2007	11
2008	11
2009	19
2010	7

Approximately the 78% of them were published between 2007 and 2010. These numbers indicate a significant interest in this relatively-new technique concentrated in the recent few years.

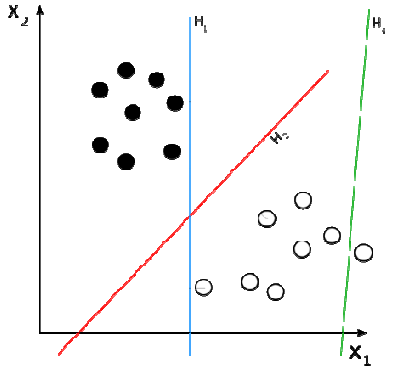
6.3) SUPPORT VECTOR MACHINE, A BRIEF DESCRIPTION

The existing literature about the theory of Support Vector Machine includes a large number of papers. Several of them, like [3] and [5], deal with the general aspects of this method. Here it will be presented a brief description of the most significant characteristics of this learning method.

- *Finding the optimal hyperplane: maximization of the margin*. In the simplest case, a classification problem is linearly separable by a hyperplane. Being $\boldsymbol{x}$ the n-dimensional input vector, it can be classified in two categories (identified by the numbers 1 and -1) using the following function:

$$f(x) = sign(\boldsymbol{w} \cdot \boldsymbol{x} + b)$$

The hyperplane is determined by the constant b and the n-dimensional vector $\boldsymbol{w}$. The image below shows three hyperplanes (H_1, H_2, H_3) in a 2-dimensional problem where black and white data points belong to two different data sets.



We can easily observe that H_3 is not correctly separating the two classes. It is, intuitively, the worse solution among those three. H_1 is correctly separating the classes, but not in a intuitive way. H_2 is correctly separating the classes and it appears like a "correct and good solution". In fact if we had to solve this problem, we would probably draw a line similar to H_2.
We can define the value of the "margin" as the sum: $d_+ + d_-$, where d_+ and d_- are the distances respectively, between the hyperplane and the closest (to the hyperplane) data point of the positive and negative classes. A very important characteristic of this problem is that the previous three observations can be collapsed into one principle to obtain the best solution: maximize the "margin" of the hyperplane.
The problem of maximization of the hyperplane can be transformed in the following Lagrangian:

$$L(\alpha) = -\sum_{i=1}^{l} \alpha_i + \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j)$$

Given l traning samples, the value to maximize is $L(\alpha)$. This maximization problem is a "Quadratic Programming Problem" and several methods to solve it have already been developed.

- *Identification and use of Support Vectors*. In the mathematical formula used, The α_i terms are called Lagrange multipliers and there is one for every training point. If the value α_i is > 0, the sample point i is called a support vector. Otherwise the value of α_i is equal to 0 and the sample point does not contribute to the final solution. Intuitively, only the borderline examples (i.e. the closest data points to the hyperplane) contribute to the computation of the classification problem. All the others can be ignored.

- *Mapping the Inputs to other dimensions*. When a problem is not linearly separable, it is customary to use kernel functions to map the data points into a higher-dimensional space where the classes are linearly separable. It is possible to define a kernel function for every particular case but, in practice the most used ones (that tend to work well in most of the cases) are the "polynomial kernel" and the "Gaussian radial basis function" (with centres in the support vectors). This strategy is not new in the machine learning field. Radial Basis Neural Networks apply exactly the same principle: the radial basis hidden layer is used to make the problem linearly separable for the linear activation functions of the output layer.

- *Not dealing with the higher dimensionality*. It is a common rule of thumb that the higher the dimensionality of the problem is, the more complex is the problem. The increase of dimensionality in Support Vector Machines, however, is not much of a trouble. In the mathematical formulas used, all the n-dimensional vector (that should be mapped into a higher z-dimensional space), are always in a dot product with another n-dimensional vector (e.g. $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$). This leads to an important observation: if we know the formula (kernel) for the dot product in the higher dimensional space, we will not have to

compute the mapping of all the n-dimensional vectors in the z-dimensional space and therefore we avoid directly dealing with the higher dimensional space.

- *Soft and hard margin*. Some time the distribution of the data points is such that, to achieve a perfect separation of the classes, the problem has to be cast in a very high dimensionality, diminishing significantly the performances. Furthermore, this increase of complexity can easily generate overfitting and reduce the ability to generalize. This happens for instance when, due to noise, the classes are not easily separable. To overcome this problem we can mathematically define the margin to be "soft": to allow misclassification of a number of data points. The softness of the margin can be controlled by a variable that determines how flexible the margin is to adapt to irregular distribution of the classes.

- *Dealing with more than two classes*. To deal with more than two classes to classify, several methods have been developed. A common strategy is "one-against-all" where we simply construct a model for every class. Every model will be trained to classify this specific class and all the other classes will be merged together. This decompose a N-class classification problem in N binary classification problems. As a consequence, N SVMs has to be trained. "One-against-one" technique is more complex but more precise, the models to be created are N(N-1)/2. In this technique a model is created for every possible class pair.

- *The result*. The unique and optimal (according to the training samples) hyperplane will be found in some unknown feature space. In the original input space the hyperplane may result in a complex, curved, non-continuous line.

6.4) THE COMPLEXITY OF SVM

Training a SVM involves solving a quadratic programming problem whose complexity is dependent on the number of training samples. The most traditional solutions require the construction of a matrix, this results in a memory complexity in the order of $O(n^2)$ where n is the number of training samples. To overcome this limit it is possible to divide the whole data into several "chunks" and train on each them individually, accumulating the support vectors incrementally. The computational complexity of those methods is $O(v^3)$ where v is the number of support vectors.
In [40] Chaos Particle Swarm Optimization has been successfully applied as an alternative method to solve the quadratic programming problem necessary to train a Support Vector Machine. Using Chaos Particle Swarm Optimization (CPSO), each particle represent a vector of Lagrange multipliers: the variables that are needed to be optimized to solve the quadratic programming problem. The fitness function computational complexity is dependent on the number of training samples.
After testing several methods on some benchmark problems, the authors conclude that CPSO-VSM has better performances than Least Squares Support Vector Machine, under the condition of training on a small training set.

6.5) OPTIMIZATION TECHNIQUES TO IMPROVE SVM

In [3] several optimization techniques are mentioned.
- *Stochastic gradient ascent*. It reduces the memory complexity to O(1) since it deals with only one example at a time but its convergence is not guaranteed.
- *Newton method* and *conjugate gradient descent* divides the problem in steps of length q. These techniques have a $O(q^2)$ memory consumption and require $O(q^3)$ computational effort.
- *Sequential Minimal Optimization* decompose the problem to its extremes (only 2 samples at a time). A big advantage is that (since only 2 samples are present at a time), there is no need to solve a quadratic problem. However, due to the reduced number of samples, it has to iterate several times.
- *Geometric approach*. It is possible to determine which data points are not support vectors and therefore remove them from the computation. Following a geometric approach. This involves solving a series of Linear Programming problems.

6.6) WHY SVM AND NOT ANN?

SVM are often used as an alternative technique to solve problems traditionally solved by ANN. The reason to substitute ANN is rooted in a number of its shortcomings. Here is a list of the most mentioned:
- ANN can suffer from multiple local minima. The solution reached by a SVM, instead, is global and unique. This is the opinion of the authors of [13], [39], [40] and [38].
- ANN have a poor generalization ability if trained with few training samples. SVM, on the contrary, achieve a good generalization ability even with a few training samples. This is the opinion of the authors of [39], [40], [23] and [38].
- ANN have a higher risk of overfitting compared to SVM. This is the opinion of the authors of [13] and [39].
- ANN have a lower convergence rate than SVM. This is the opinion of the authors of [13] and [39].
- ANN is based on the principle of Empirical Risk Minimization that usually give worse generalization ability compared to the Structural Risk Minimization principle (generally considered a better principle) that is the base of SVM. This is the opinion of the authors of [40].
- SVMs are more robust to corrupted data and can adapt more easily to complex systems compared to ANN. This is the opinion of the authors of [40]

6.7) EXPERIMENTAL RESULTS TO COMPARE SVM AND OTHER TECHNIQUES

The research done in [29] shows experimental results that can be used to compare back-propagation ANN and SVM. After several tests to "tune" the parameters of the learning techniques (like number of hidden nodes) the authors used the best parameters setting of each to evaluate the two different methods.

Method	Efficiency	Time to train
Artificial Neural Network	97.5%	More than PSVM
Proximal Support Vector Machine	97%	Less than ANN

The authors believe that the two learning algorithm have shown a very similar accuracy. It is therefore their opinion that the slightly superior efficiency of ANN over PSVM is not significant and PSVM should be considered a better approach due to its faster training speed. The validity of these results, however, has to be considered restricted to the problem fault diagnosis of spur bevel gear box.

A similar study is described in [39]. The accuracy of SVMs have been compared to Back Propagation Neural Networks for fault diagnosis in rotating machinery. In this experiment, Back Propagation NN had an average accuracy of 89.58% and Hybrid SVM had an average accuracy of 92.50%.

In [23], SVM has been compared to fuzzy logic classification. Fuzzy logic classification resulted in an accuracy of 96.701%, inferior to 98.703% calculated using SVM.

6.8) SVM RECENT APPLICATIONS IN FDD

The most common task of SVMs is classification and therefore its most common use in FDD is fault diagnosis. There are several examples of this in the recent literature. For instance SVM has been used for fault diagnosis for power transformer ([13]), car assembly line ([38] and [37]), spur bevel gear box ([29]), rotating machinery ([40] and [39]), series compensated transmission line ([23]), low speed bearing ([33]), induction motor([35] and [34]) and compressor valve ([9]).

There are however examples of SVM used for other tasks. In [18], SVM has been used for time-series prediction model for vibration signals and Lorenz signal. In [21], SVM has been used for fault detection for the Tennessee Eastman benchmark problem.

6.9) DIFFERENT TYPES OF SVM USED IN FDD

Several modifications to the standard SVM have been proposed in numerous papers. Among those, it is worth mentioning the following ones.

- *Proximal Support Vector Machine* (PSVM). In the literature, [29] and [30] deal with this type of SVM. The standard SVM is based on an optimization problem and therefore is computationally expensive. PSVM, on the contrary, is based on a system of linear equations that achieve similar results as SVM but with less computation. In [7], PSVM is roughly estimated three times faster than a standard SVM.
- *Gaussian Support Vector Machine*. Presented in [38], this SVM deals with the problem of Gaussian noise in the data.
- *One-class SVM*. In the literature, [21] discuss this type of SVM and its use for novelty detection. While a standard SVM requires to be trained both on faulty and healthy data to be able to distinguish between classes, a one-class SVM can be trained only with healthy data points. It will compute the boundary that accommodates most of the training samples and it will recognise a new sample as an “outlier” if it lies outside the

boundaries. The major advantage of this technique is that it can capture and model the non-linearity of a system. The most traditional approaches like Principal Component Analysis, on the contrary, work with linear assumptions. On the Tennessee Eastman benchmark problem this technique gave better result than Principal Component Analysis.

6.10) THE PROBLEM OF TUNING THE SVM'S PARAMETERS

The problem of tuning the parameters in a SVM is a very delicate one. Several studies proved that a bad choice of the parameters could lead to overfitting or underfitting. The main parameters to tune are: 1) the softness of the margin (often called C), 2) the parameters of the kernel function such as the radius σ of the radial basis functions. The most traditional strategy to solve this problem is cross-validation.
In the literature it is possible to find several attempts to solve this optimization problem. The most common of them are Evolutionary Computing and Swarm Intelligence methods. For instance, in [38], the optimal parameters are found using Particle Swarm Optimization, in [18] the method used is Evolution Strategy with Covariance Matrix Adaptation, in [13] a Genetic Algorithm is used.

6.11) NEURO-FUZZY

Neuro-Fuzzy learning systems attracted significant interest in the FDD literature. The advantage of this hybrid method is the combination of two complementary techniques. ANN can be easily trained to learn the behaviour of non-linear systems and Fuzzy Logic is a powerful tool for high-level qualitative reasoning.
The major drawback of ANN in FDD is that this technique is unable to provide an intuitive interpretation of its reasoning. This problem is solved by the Fuzzy Logic interpretability. Classification, in Fuzzy logic, can be expressed by a set of IF<*condition*>THEN<*class*> rules. Furthermore, these conditions can be expressed in terms of not-numerical imprecise qualities such as "low", "medium" or "high".
The problem with Fuzzy Logic, instead, is how to partition the input space into Membership Functions and how to select the fuzzy rules. Typically this is done with the help of human knowledge. Using ANN, instead, this process can be automated.
Moreover, of a Neuro-Fuzzy classification tool has the peculiar characteristic of outputting the degree of membership for every class. This additional information can be useful in the diagnosis step, as it carries more knowledge on the fault detected.

6.12) AN EXAMPLE OF NEURO-FUZZY APPLICATION IN FDD

In [50], Neuro-Fuzzy is used for FDD for rotating machinery. A Gaussian Membership Function is selected to partition the input data into linguistic terms. This function is a standard in this respect and can be easily updated modifying only its position and its radius. In the cited work, considering n input variables and m output variables, the rules are in the conjunctive form:

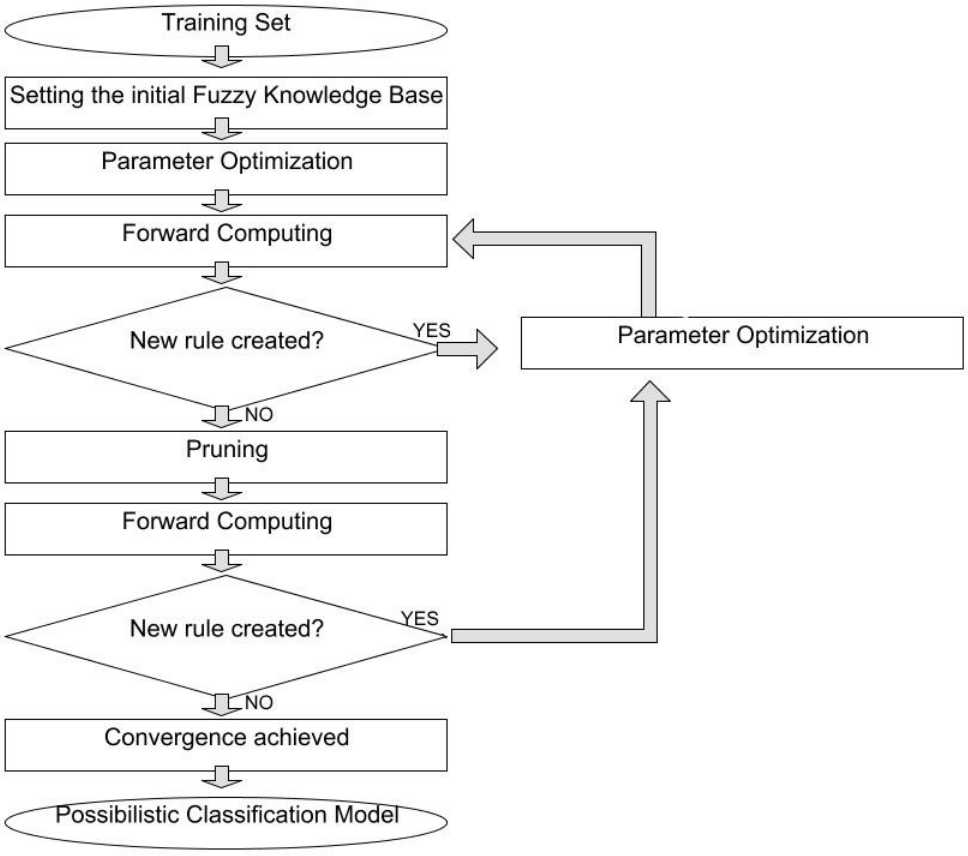
$$IF : x_1 is\ y_1\ AND\ \ x_2 is\ y_2\ AND \ldots\ AND\ x_n is\ y_n$$
$$THEN : t_1 is\ z_1\ AND\ \ t_2 is\ z_2\ AND \ldots\ AND\ t_m is\ z_m$$

The position and the radius of the Membership Functions, as well as the output membership grades, are adjusted using the backpropagation algorithm.

The optimal input partitioning interface, according to [50], follows the following principles.

- *Moderate number of Membership Functions*: if the number is high, the learning system risks to get *overfitted*. In this case, the Neuro-Fuzzy model will have higher accuracy on the training set but lower extrapolation ability on unseen data.
- *Normality*: at least one number in the fuzzy partition should have full membership grade.
- *Coverage*: the Membership Functions should cover the whole multidimensional space.
- *Distinguishability*: each membership function should be clearly distinguishable with each other. They should not overlap too much.

The method described in [50] can be schematized in the following diagram.



To compute the initial Fuzzy Knowledge Base, every feature is divided in three qualitative values [*small*, *medium*, *big*]. To accomplish this, three Gaussian Membership Functions are created for every feature. The three centres of these functions are positioned, considering the training set, at the lowest value, at the highest and in the centre. The radius of the Gaussian function is initialized equally for each of the three

Membership Functions. It is calculated as the minimum radius that maintains the coverage property.
These linguistic terms are used to calculate all the 3^n possible antecedents (and related consequent) in a problem with n inputs. It has to be remembered that the antecedents are a conjunction of n terms as described above. This number, in a system with many features, can be excessively high. They can be reduced selecting, for each class, only the antecedent that fires most often. The number of antecedents is then reduced to the number of classes in the problem. During the training, new rules can be automatically added if necessary. The parameters are then tuned using the backpropagation learning algorithm. It is also possible to reduce the number of rules in the Fuzzy Knowledge Base using pruning strategies. For example it is possible to delete rules that have not fired enough or merge together rules that have a very similar antecedent. The final results demonstrated a high classification accuracy on a the FDD problem of motor bearing faults combined with a high interpretability of the model.

6.13) GENETIC ALGORITHMS, A BRIEF DESCRIPTION

A typical problem where Genetic Algorithms (GA) are very effective is finding the optimal combination of parameters for a process. GAs are, in general, very effective to solve difficult problems (such as NP-hard problems) compared to traditional search strategies (that are, instead, usually faster on simpler problems).
The principle behind these methods is inspired by genetics and the evolutionary process driven by natural selection.
Genetic algorithms perform parallel search. A major benefit from this is the *global view* on the problems that makes the search faster and significantly less likely to get stuck into a sub-optimal minima. A common problem that hinders the *global view* is premature convergence. It happens when the population converge into a small subspace too soon causing the search to become "less" parallel. The advantages brought by the individual diversity, then, are mostly lost.
The most important concepts of genetic algorithms will be here summarized.
-*Genotype*. The Genotype is a vector of values that encodes all the information carried by an individual. In its most simple form, it is a binary string.
-*Phenotype*. The Phenotype is the interpretation of the genotype according to a specific problem. It represents a possible solution to the problem.
-*Individual*. An individual is defined by a genotype and its related phenotype. Therefore it can be considered a solution to the problem.
-*Fitness*. A fitness function to evaluate an individual is required. The fitness value is the value that the algorithm will try to maximize. It is usually computed considering the phenotype.
- *Population*. Evolutionary algorithms involve the "evolution" of a population of individuals. The evolution process will aim to generate a population of well-fit individuals.
-*Evolutionary operators*. These are the operators used to generate a new generation from a previous one. Usually, well-fit individuals have a higher probability of being used by

evolutionary operators compared with individuals with a lower fitness value. Their genotype, then, has a higher probability to spread through the population and survive across multiple generations (elitism).
-*Reproduction*. This evolutionary operator consist in copying an individual into a new population without modifications in its genotype. This is fundamental to preserve a "good" solution.
-*Recombination*. This evolutionary operator generates a new individual performing a cross-over operation on the genotype of two individuals.
-*Mutation*. This evolutionary operator generates a new individual changing randomly the genotype of another individual. This is a fundamental operator that allows "new" genotypes to be used in a population. It also prevents the population to converge into a sub-optimal fitness minima. A problem that reduces the efficiency of mutation is that this operation can generate individuals with an illegal genotype that does not correspond to any meaningful phenotype (it does not represent a solution to the problem).

6.14) USAGES OF GENETIC ALGORITHMS IN FDD

In [45] GAs have been used to train the hidden layer of a Neural Network adjusting the centers and the width of the radial-basis activation function. Pseudo-inverse matrix algorithm has been used to train the output layer. This technique has efficiently solved the fault classification problem in Air Handling Unit.
In [41] a GA has been used to determine the optimal parameters of the Gaussian kernel of a Support Vector Data Description, a method that can "describe" a dataset and detect outliers. This method has been successfully used in one-class bearing fault detection.
In [1], fault detection in electronic circuits is done using a negative selection algorithm. This type of algorithm is used to detect explicitly faulty data points using detectors. Detectors results in hyperspheres whose task is to cover faulty regions in the data space. GAs has been used to find the optimal centers of the detectors. The same strategy has been used in [2] as well. In the latter case, however, the problem addressed is to detect broken rotor bar and broken connectors in induction motors.
GAs has also been used to train the weights of an Elman Neural Network which is a locally recurrent but globally feedforward neural network. In [15] this last method is used in motor fault detection.
In the experiment conducted in [13], GAs has been used to train a SVM. The genotype of the population encoded the σ (the radius of the radial basis function) and the C (softness of the margin) parameters in binary notation. The probability of the evolutionary operators was the following: recombination 80%, reproduction 15%, mutation 5%. The fitness function was given by the classification accuracy (on a few data samples) of the SVM adjusted with the given σ and C values.

6.15) SWARM INTELLIGENCE, ANT COLONY OPTIMIZATION (ACO)

Ant colony optimization algorithms are a group of optimization algorithms inspired to the social structure of an ant colony. The main application of this technique is directed to solve the problem of finding an "optimal" path in a graph.
A population of ants is positioned in a graph to explore it. The initial positions of the ants is problem-dependent (for instance the ants can be distributed randomly over the nodes, but more often they have a fixed starting node). The specific problem also determines the rules that controls the movement of the ants (in the travelling salesman problem, for instance, the rule of not visiting the same node more than once). In general, the choice of an ant on which node to visit next is determined by static variables such as the distance between nodes (for instance the ants could prefer to travel to closer nodes) or by dynamic variables such as pheromone trails that are modified during the optimization process. Pheromone trial is a variable related to the environment and it is updated (usually increased) by each ant during its path. It is a measure of how many ants have passed through a specific portion of a path.
Each ant is unable to communicate directly to any other in the colony. The communication is done indirectly trough modification of the environment. This type of communication is called stigmergy and its main advantage is being very scalable. The reason is that, without direct communication, the computational complexity of an ant is not dependent on the number of the other ants in the colony.
Each ant has also the possibility (with a certain probability), to choose a not-optimal path to allow exploration.

6.16) ANT COLONY FOR FEATURE SELECTION

In [32] Modified Discrete Binary Ant Colony Optimization has been used for feature selection and SVM has been used for fault classification for the problem of fault diagnosis in the Tennessee Eastman benchmark problem. This particular type of ant colony optimization uses a binary representation. Features marked by 1 are selected and those marked by 0 are not.
According to the results presented in [32], Modified Discrete Binary Ant Colony Optimization obtained better results than the classical discrete ant colony optimization technique. Furthermore, these two ant colony optimization methods resulted better in performances compared to genetic algorithms.

6.17) ANT COLONY AS CLASSIFIER

ACO can be used as a classification tool as well. In the latter case, the paths of the ants are interpreted as IF<condition>THEN<class>. The nodes of the graph are conditions/part of conditions or classes. After the path of each ant in the colony has been computed, the pheromone trails are used to select the rule discovered in the current iteration. Further iterations are necessary to discover new rules and particular attention is required to avoid the colony to discover the same rules again. In [31] this ant colony

classifier method has been used for fault classification for chemical plants. In this paper ACO has been compared to C4.5 algorithm. ACO resulted better in accuracy but C4.5 algorithm resulted better in simplicity (number of rules and rule's terms). In [31], thanks to the high scalability of this system, the number of ants in the colony has been set to 500. This is a very high number of individuals compared to other population-based techniques. Particle Swarm Optimization and Genetic Algorithms, for instance, usually have a population of less than 100 individuals.
Despite ACO has been successfully used for classification, the authors of [31] mentions a number of drawbacks of this technique:
- the data has to be discretized in categories before the ACO algorithm can be used (this is the approach followed by [31]),
- the computation required to achieve optimal models is high,
- problem-dependent information is required for an optimal usage of ACO.

6.18) SWARM INTELLIGENCE, PARTICLE SWARM OPTIMIZATION (PSO)

Particle swarm optimization is a global optimization algorithm where solutions are encoded by a feature vector that determines the position of a number of particles. Unlike other evolutionary computing techniques, the search for the best solution is not computed trough evolutionary operators like recombination and mutation. The search is driven by the velocity of the particles that moves them trough the search-space. The speed and the direction of the particles are typically determined by a combination of factors such as the best solution found so far by the particle and the best solution found so far by the swarm.

6.19) USAGES OF PARTICLE SWARM OPTIMIZATION IN FDD

PSO can be applied as a training algorithm for neural networks. In [14] an ANN is trained using PSO to solve a fault detection problem. It is applied in FDD of a crack seeded in an intermediate gearbox of a helicopter's main transmission using vibration data.
In [28] particle swarm optimization is used to detect faults and identify their magnitude in an induction Motor. After creating a mathematical model of the motor, PSO with a population of 100 particles solves the problem of FDD trough parameter identification. Using the set of known parameters, the unknown parameters, such as the magnitude of the fault, are identified as the ones that maximize the fitness function.

6.20) NOTE ON THE SCALABILITY OF PARTICLE SWARM ALGORITHMS

It has been noted in [26], that particle swarm algorithms have difficulties in dealing with high-dimensional problems. However "high-dimensional", in this context, means a number of dimensions above 100. In real Fault Detection and Diagnosis applications, however, this does not seem to be a problem since the number of features is frequently

inferior to 100 by far. Using feature selection and reduction techniques, the average number of features analyzed is between 10 and 20.

7) DEALING WITH NO FAULTY-DATA AVAILABLE

7.1) THE PROBLEM OF THE DIAGNOSIS OF UNKNOWN FAULTS

Fault detection can be done by means of clustering, following the principles of novelty detection. In this case, the solution is found using an unsupervised learning method. Being unsupervised, no prior knowledge of the faults is necessary and this method can be applied in most of the real world problems. The most common approaches to fault diagnosis, instead, are supervised learning methods. As such, those methods require prior knowledge on the fault types and characteristics. Monitoring the process in faulty conditions is also usually necessary. When no information about the faults is available, different approaches have to be taken.

One possible solution is to generate a faulty data set not by direct measurements but using a simulation. In general, if the real physical faults cannot be monitored, this solution is feasible only if the system is simple enough to develop an accurate mathematical model. A fault is simulated in the model and its outputs are collected in a faulty data set. This strategy allows the use of supervised learning methods, however it has a few obvious drawbacks. For one thing, a model-based FDD approach is required. From model-based FDD, this strategy inherits low scalability and the complexity of developing an accurate and comprehensive model.

7.2) ADDING MASSES TO A SYSTEM: A POSSIBLE PARTIAL SOLUTION

In the Single Degree of Freedom theory, in a system with stiffness k and mass m, the natural frequency ω of its undamped free vibration is defined as follows.

$$\omega = \sqrt{\frac{k}{m}}$$

The research presented in [22] is based on the idea that a fault introduces local flexibility in the system. A fault often results in a change in the natural frequency. It is usually a reduction of the normal value due to a stiffness loss. However, the same effect would be caused by an increase in the mass.
In the experiment conducted in [22] an aircraft wingbox was used. A few particular characteristics of this problem have to be mentioned to justify the use of this unusual diagnosis strategy. First of all, aircraft equipment is extremely expensive and deliberately inducing a fault is not a feasible option. Secondly these kind of faults require an immediate troubleshooting reaction and accurate detection and diagnosis strategies have to be implemented. Unlike a normal industrial system, the working condition of an aircraft cannot be “temporary shutdown” to perform maintenance operations as soon as a fault is detected. In addition the particular physical properties of these equipments are suitable to be modeled in terms of mass, stiffness and vibration frequency.

The experiments conducted in [22] demonstrated that the natural frequencies decreased as the fault introduced in the system became more severe. A similar trend in the natural frequencies was calculated increasing the mass instead. Increasing the mass could be a strategy to generate *pseudo faults*. Pseudo faults are particular system conditions that, though fault-free, generates fault-like effects. The obvious advantage is that faulty conditions can be measured on the system without the need of having a real fault. The authors of [22] claims that this method can be effectively used, under some circumstances, to generate pseudo faults. If the faults are not related to a frequency dependent on the mass, however, this technique is not applicable. In general, it is not a global solution to the problem of lack of fault information.

7.3) THE DIAGNOSIS OF UNKNOWN FAULTS: AN UNSOLVED PROBLEM

The existing literature concerning the automatic diagnosis of unknown faults is not massive. The lack of research in this respect has its root in a general low interest for this kind of problem combined with a high complexity of the problem itself. Most of the real world faults can be described in one of the following way.

- Common faults. These kind of faults occur often during the normal working operations of a system. Being common, they are not usually the source of catastrophic damages and it is often possible to deliberately cause them in order to measure their effects. These kind of faults can be analysed a-priori and the fault diagnosis techniques used have the advantage of exploiting supervised data. For this kind of problems supervised learning methods are probably the best solution developed so far. These techniques can be implemented efficiently and can easily reach high levels of accuracy.

- Uncommon faults. These faults are unpredictable both in location and in magnitude. Being infrequent, it is not possible to obtain accurate information about them. Their effect, however, could be severe, and FDD approaches have to be used. It is possible to develop an accurate fault detection method using the available healthy data. In case of a fault is detected, the system can be shut down and maintenance operations are performed. The diagnosis is done by human experts. Relevant information about the fault is generally generated automatically but the actual diagnosis operations are not automated. Since this kind of faults are uncommon, the interruption of the working process and the use of human expertise is a cost that can be sustained.

The above mentioned examples shows that in real world problems, the automation of the diagnosis step of unknown faults is not strictly necessary, as long as the detection task can be accomplished. Moreover, the difficulties behind finding a solution to this problem are several. The monitored healthy data might not contain enough knowledge of the system to perform the diagnosis task. A possible solution to this problem usually requires the exploitation of particular properties of the data to be analysed.

8) CONCLUSION

The Fault Detection and Diagnosis (FDD) field has been analysed, in this survey, as the main research field that deals with abnormal conditions in engineering physical systems. In particular, data-driven FDD is identified as the branch of FDD that is closest to the principles of Data Mining. The strength of data-driven FDD methods lies in the databases of historical process data. The amount of data points used in the data-driven FDD process is not necessary very large, but it is very common to have a very high dimensionality of the inputs. Probably more than the dimension of the data set, the dimensionality of the problem is the characteristic that makes human expertise not feasible in FDD and automatic techniques are nowadays necessary.
Most of the FDD strategies described in the analysed literature focused on the following three points.

- *Accurate data pre-processing*. This phase mostly comprehend denoising, feature reduction, feature extraction and feature selection. Without this step, it is very common to achieve unacceptable accuracy in the detection and diagnosis models.
Feature reduction techniques such as Principal Component Analysis and Fisher Discriminant Analysis are very common in FDD applications. These methods deal with the high dimensionality of the problem and, at the same time, strive for a better representation of the data.
Feature extraction is a very common process to elaborate vibration signals. They are the main data type measured from mechanical machines. However this data type, in its raw form, is a very hard type of data to be analysed automatically. This problems is solved by a pre-processing step to extract and highlight the relevant features from the data. Fourier Transforms are probably the oldest approach to deal with this issue. Nowadays, Wavelet Transforms, Walsh Transforms and Empirical Mode Decomposition are more effective and therefore more popular techniques.

- *Fault detection*. The main problem in FDD is the detection of the occurrence of a fault. This problem is generally treated as a novelty detection problem. Clustering algorithms are very common solutions. Another very common strategy is the analysis of the statistical properties of the data in order to define thresholds to discriminate if new data points are faulty or not.

- *Fault classification*. The most intuitive way to diagnose a recurrent fault is to monitor the system during its occurrence. If data sets representing the different faults are available, it is possible to train a model to distinguish between the different faults classes (and the healthy class) given a new unseen data point. Fault diagnosis can, in practice, be considered a classification problem.
For this reason, almost every FDD strategy makes use of Machine Learning techniques. In fact, this field seems to provide the most promising classification tools. In this respect, Support Vector Machines and Neuro-Fuzzy classification have proven to

outperform the already efficient ANN approaches. Swarm Intelligence and Evolutionary Computing methods, while not very widespread in the FDD field yet, are considered interesting techniques to solve optimization problems such as parameter tuning of other algorithms. Many Machine Learning methods, in fact, require a smart choice of the initial parameters in order to achieve good results.

While a lot of research focused on data pre-processing and fault classification, very little has been written about the problem of diagnosing faults which are unknown a-priori. If data sets representing faults are not available, a general strategy to perform accurate fault diagnosis has not been published yet. This is an open problem that requires further research.

The use of data from system measurements for FDD has proven to be very effective. However there is a high number of different techniques available and each of them can be used or tuned in several different ways. Human expertise still appears to be fundamental in the choice of those variables. In addition, clear and global standards in data driven FDD have not yet been established.

REFERENCES

1] J.L.M. Amaral, J.F.M. Amaral, D.Morin, R.Tanscheit, An Immune Fault Detection System with Automatic Detector Generation by Genetic Algorithms. *Seventh International Conference on Intelligent Systems Design and Applications IEEE (2007) 283-288*

2] I. Aydin, M. Karakose, E. Akin, Artificial Immune Inspired Fault Detection Algorithm Based on Fuzzy Clustering and Genetic Algorithm Methods. *CIMSA 2008 – IEEE International Conference on Computational Intelligence for Measurement Systems And Applications Istanbul – Turkey , 14-16 July (2008)*

3] D. Boswell, Introduction to Support Vector Machines. *(August 6, 2002) 1-15*

4] M. Boumahdi, J.P. Dron, S. Rechak, O. Cousinard, On the extraction of rules in the identification of bearing defects in rotating machinery using decision tree. *Expert Systems with Applications 37 (2010) 5887–5894*

5] C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery 2 (1998) 121–167*

6] A. Çakır, H. Çalıs, E.U. Küçüksille, Data mining approach for supply unbalance detection in induction motor. *Expert Systems with Applications 36 (2009) 11808–11813*

7] L.H. Chiang, M.E. Kotanchek, A.K. Kordon, Fault diagnosis based on Fisher discriminant analysis and support vector machines. *Computers and Chemical Engineering 28 (2004) 1389–1401*

8] C.F. Chien, W.C. Wang, J.C. Cheng, Data mining for yield enhancement in semiconductor manufacturing and an empirical study. *Expert Systems with Applications 33 (2007) 192–198*

9] H. Cui, L. Zhang, R. Kang, X. Lan, Research on fault diagnosis for reciprocating compressor valve using information entropy and SVM method. *Journal of Loss Prevention in the Process Industries 22 (2009) 864–867*

10] J. Cusidó, L. Romeral, J.A. Ortega, A. Garcia, J.R. Riba, Wavelet and PDD as fault detection techniques. *Electric Power Systems Research 80 (2010) 915–924*

11] S. Ding, P. Zhang, E. Ding, S. Yin, A. Naik, P. Deng, W. Gui, On the Application of PCA Technique to Fault Diagnosis. *Tsinghua Science And Technology Volume 15, Number 2 (2010) 138-144*

12] M. Elangovan, K.I. Ramachandran, V. Sugumaran, Studies on Bayes classifier for condition monitoring of single point carbide tipped tool based on statistical and histogram features. *Expert Systems with Applications 37 (2010) 2059–2065*

13] S. Fei, X. Zhang, Fault diagnosis of power transformer based on support vector machine with genetic algorithm. *Expert Systems with Applications 36 (2009) 11352–11357*

14] H. Firpi, Swarmed Neuro-Artificial Features from Vibration Data for Fault Detection and Isolation. *IEEE Congress on Evolutionary Computation (2006) 871-877*

15] X.Z. Gao, S.J. Ovaska, Genetic Algorithm Training of Elman Neural Network in Motor Fault Detection. *Neural Computing & Applications 11 (2002) 37–44*

16] J.A. Harding, M. Shahbaz, Srinivas, A. Kusiak, Data Mining in Manufacturing: A Review. *Journal of Manufacturing Science and Engineering Vol. 128 (2006) 969-976*

17] W. Hong, C. Tian-You, D. Jin-Liang, M. Brown, Data Driven Fault Diagnosis and Fault Tolerant Control: Some Advances and Possible New Directions. *Acta Automatica Sinica Vol. 35, No. 6 (2009)*

18] S. Hou, Y. Li, Short-term fault prediction based on support vector machines with parameter optimization by evolution strategy. *Expert Systems with Applications 36 (2009) 12383–12391*

19] N.E. Huang, Hilbert-Huang Transform And its Applications. *World Scientific, Interdisciplinary Mathematical Sciences, Vol. 5 (2005) 1-25*

20] R. Isermann, Fault-diagnosis systems: An introduction from fault detection to fault tolerance. *Berlin, Springer–Verlag 1st ed. (2006)*

21] S. Mahadevan, S.L. Shah, Fault detection and diagnosis in process data using one-class support vector machines. *Journal of Process Control 19 (2009) 1627–1639*

22] E. Papatheou, G. Manson, R.J. Barthorpe, K. Worden, The use of pseudo-faults for novelty detection in SHM. *Journal of Sound and Vibration 329 (2010) 2349–2366*

23] U.B. Parikh, B. Das, R. Maheshwari, Fault classification technique for series compensated transmission line using support vector machine. *Electrical Power and Energy Systems 32 (2010) 629–636*

24] Z.K. Peng, F.L. Chu, Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography. *Mechanical Systems and Signal Processing 18 (2004) 199–221*

25] P. Perner, C. Apte, Empirical evaluation of feature subset selection based on a real-world data set. *Engineering Applications of Artificial Intelligence 17 (2004) 285–288*

26] S. Piccand, M. O'Neill, J. Walker, Scalability of particle swarm algorithms. *London, England, United Kingdom ACM 978-1-59593-697-4/07/0007 (2007) 179*

27] R. Polikar, The wavelet tutorial. *Rowan University, College of Engineering (1999). http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html*

28] V. Rashtchi, Detection and Magnitude Determination of Turn Faults in Induction Motor By Using of Particle Swarm Optimization Algorithm. *The 2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology IEEE (2009)*

29] N. Saravanan, S.V.N.S. Kumar, K.I. Ramachandran, Fault diagnosis of spur bevel gear box using artificial neural network (ANN), and proximal support vector machine (PSVM). *Applied Soft Computing 10 (2010) 344–360*

30] N. Saravanan, S.V.N.S. Kumar, K.I. Ramachandran, A comparative study on classification of features by SVM and PSVM extracted using Morlet wavelet for fault diagnosis of spur bevel gear box. *Expert Systems with Applications 35 (2008) 1351–1366*

31] P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni, An ant colony classifier system: application to some process engineering problems. *Computers and Chemical Engineering 28 (2004) 1577–1584*

32] L. Wang, J. Yu, A Modified Discrete Binary Ant Colony Optimization and Its Application in Chemical Process Fault Diagnosis. *T.-D. Wang et al. (Eds.): SEAL 2006, LNCS 4247 (2006) 890-896*

33] A. Widodo, E.Y. Kim, J.D. Son, B.S. Yang, A.C.C. Tan, D.S. Gu, B.K. Choi, J. Mathew, Fault diagnosis of low speed bearing based on relevance vector machine and support vector machine. *Expert Systems with Applications 36 (2009) 7252–7261*

34] A. Widodo, B.S. Yang, Wavelet support vector machine for induction machine fault diagnosis based on transient current signal. *Expert Systems with Applications 35 (2008) 307–316*

35] A. Widodo, B.S. Yang, D.S. Gu, B.K. Choi, Intelligent fault diagnosis system of induction motor based on transient current signal. *Mechatronics 19 (2009) 680–689*

36] F. Wu, L. Qu, Diagnosis of sub harmonic faults of large rotating machinery based on EMD. *Mechanical Systems and Signal Processing 23 (2009) 467– 475*

37] Q. Wu, Car assembly line fault diagnosis based on modified support vector classifier machine. *Expert Systems with Applications 37 (2010) 6352–6358*

38] Q. Wu, Fault diagnosis model based on Gaussian support vector classifier machine. *Expert Systems with Applications 37 (2010) 6251–6256*

39] G.M Xian, B.Q. Zeng, An intelligent fault diagnosis method based on wavelet packer analysis and hybrid support vector machines. *Expert Systems with Applications 36 (2009) 12131–12136*

40] T. Xianlun, Z. Ling, C. Jun, L. Changbing, Multi-fault classification based on support vector machine trained by chaos particle swarm optimization. *Knowledge-Based Systems 23 (2010) 486–490*

41] T. Xin-min, C. Wan-Hai, D. Bao-Xiang, X. Yong, D. Han-Guang, A Novel Model of one-class bearing fault detection using SVDD and Genetic Algorithm. *Second IEEE Conference on Industrial Electronics and Applications (2007) 802-807*

42] X. Xiuqiao, Z. Jianzhong, L. Chaoshun, L. Qingqing, L. Zhimeng, Fault diagnosis based on Walsh transform and rough sets. *Mechanical Systems and Signal Processing 23 (2009) 1313–1326*

43] L. Yaguo, J.Z. Ming, H. Zhengjia, Z. Yanyang, A multidimensional hybrid intelligent method for gear fault diagnosis. *Expert Systems with Applications 37 (2010) 1419–1430*

44] Q. Yang, Model-based and data driven fault diagnosis methods with applications to process monitoring. *Ph.D. Dissertation in Electrical Engineering and Computer Sciences, Case Western Reserve University (2004)*

45] H. Yonghong, L. Nianping, H. Yonghong, S. Yangchun, Automated Fault Detection and Diagnosis for an Air Handling Unit Based on a GA-Trained RBF Network. *Proceedings of ICCCAS (2006) 2038-2041*

46] Y. Yu, Y. Dejie, C. Junsheng, A fault diagnosis approach for roller bearing based on IMF envelope spectrum and SVM. *Measurement 40 (2007) 943–950*

47] G. Zhao, D. Jiang, K. Li, J. Diao, Data Mining for Fault Diagnosis and Machine Learning for Rotating Machinery. *Key Engineering Materials Vols. 293-294 (2005) pp 175-182*

48] Z. Zhi-Bo, S. Zhi-Huan, Fault diagnosis based on imbalance modified kernel Fisher discriminant analysis. *Chemical Engineering Research and Design (2010), article in press*

49] H. Zhijian, L. Zhiwei, Y. Ye, Y. Xinjian, Data mining based sensor fault diagnosis and validation for building air conditioning system. *Energy Conversion and Management 47 (2006) 2479–2490*

50] E. Zio, G. Gola, A neuro-fuzzy technique for fault diagnosis and its application to rotating machinery. *Reliability Engineering and System Safety 94 (2009) 78–88*