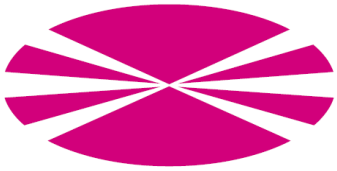


UNIVERSITY OF A CORUÑA

FACULTY OF INFORMATICS

Department of Computer Science

Ph.D. Thesis

On-line learning and anomaly detection methods. Applications to fault assessment

Author: David Martínez Rego
Advisors: Amparo Alonso Betanzos
Óscar Fontenla Romero

A Coruña, July 2013

April 4, 2013
UNIVERSITY OF A CORUÑA

FACULTY OF INFORMATICS
Campus de Elviña s/n
15071 - A Coruña (Spain)

To my late father

Acknowledgements

PhD is a personal journey full of obstacles, dilemmas and crossroads. Sometimes you have the luck of finding the aid and friendship of people that you never would have imagined to meet and connect. During these years I have felt at home in places as far apart as Florida, London, Cambridge and Seville. This places are part of my life now and this is thank to the colleagues and friends I met. In the old town of Seville I would like to thank Carmen Madrigal for receiving me and Rodrigo "Chicho" Tam for showing me the city, all the nice evenings and for revealing me La Eslava (I still owe you climbing the Machu Pichu). I also have to thank Pablo Lorenzino, Carolina Ramiro, Mauricio Ponga, Esteban Tellechea, Federico Buroni and all the people at the School of Engineering of Seville for being like a family for me. In the tropical hot town of Gainesville I have to thank Göktuğ Çinar for being such a good host, Ashwini Amarasinghe and all the colleagues in the CNEL laboratory. I also leave two good friends in London, Bernardino Romera and Jez Kanthan. Last but not least, thanks to Antonio Fabregat and his wife for opening your home to a stranger that you met by chance.

To overcome the obstacles you need guidance and teammates. During these years I have had the help of very good professors that have made this work possible, specially my advisors Amparo Alonso and Óscar Fontenla, and of my colleagues at LIDIA Group at University of A Coruña.

Finally, there is always people that will always have confidence in you and stays on your side in good and hard times. This work is also part of them and would not have any sense without them. I refer to my family, specially my late father who is not here to see this, and you, Rebeca.

David Martínez Rego
March 2013

Abstract

This work lays at the intersection of two disciplines, Machine Learning (ML) research and predictive maintenance of machinery. On the one hand, Machine Learning aims at detecting patterns in data gathered from phenomena which can be very different in nature. On the other hand, predictive maintenance of industrial machinery is the discipline which, based on the measurement of physical conditions of its internal components, assesses its present and near future condition in order to prevent fatal failures. In this work it is highlighted that these two disciplines can benefit from their synergy. Predictive maintenance is a challenge for Machine Learning algorithms due to the nature of data generated by rotating machinery: (a) each machine constitutes an new individual case so fault data is not available for model construction and (b) working conditions of the machine are changeable in many situations and affects captured data. Machine Learning can help predictive maintenance to: (a) cut plant costs though the automation of tedious periodic tasks which are carried out by experts and (b) reduce the probability of fatal damages in machinery due to the possibility of monitoring it more frequently at a modest cost increase.

General purpose ML techniques able to deal with the aforementioned conditions are proposed. Also, its application to the specific field of predictive maintenance of rotating machinery based on vibration signature analysis is thoroughly treated. Since only normal state data is available to model the vibration captures of a machine, we are restricted to the use of anomaly detection algorithms, which will be one of the main blocks of this work. In addition, predictive maintenance also aims at assessing its state in the near future. The second main block of this work, on-line learning algorithms, will help us in this task. A novel on-line learning algorithm for a single layer neural network with a non-linear output function is proposed. In addition to the application to predictive maintenance, the proposed algorithm is able to continuously train a network in a one pattern at a time manner. If some conditions are hold, it analytically ensures to reach a global optimal model. As well as predictive maintenance, the proposed on-line learning algorithm can be applied to scenarios of stream data learning such as big data sets, changing contexts and distributed data.

Some of the principles described in this work were introduced in a commercial software prototype, GIDAS®. This software was developed and installed in real plants as part

of the work of this thesis. The experiences in applying ML to fault detection with this software are also described and prove that the proposed methodology can be very effective. Fault detection experiments with simulated and real vibration data are also carried out and demonstrate the performance of the proposed techniques when applied to the problem of predictive maintenance of rotating machinery.

Keywords: *Machine learning, anomaly detection, on-line learning, stream data, vibration analysis, predictive maintenance, mechanical engineering*

Resumen

La presente tesis doctoral se sitúa en el ámbito de dos disciplinas, la investigación en Aprendizaje Computacional (AC) y el Mantenimiento Predictivo (MP) de maquinaria rotativa. Por una parte, el AC estudia la problemática de detectar y clasificar patrones en conjuntos de datos extraídos de fenómenos de interés de la más variada naturaleza. Por su parte, el MP es la disciplina que, basándose en la monitorización de variables físicas de los componentes internos de maquinaria industrial, se encarga de valorar las condiciones de éstos tanto en el momento presente como en un futuro próximo con el fin último de prevenir roturas que pueden resultar de fatales consecuencias. En este trabajo se pone de relevancia que ambas disciplinas pueden beneficiarse de su sinergia.
El MP supone un reto para el AC debido a la naturaleza de los datos generados por la maquinaria: (a) las propiedades de las medidas físicas recogidas varían para cada máquina y, debido a que la monitorización debe comenzar en condiciones correctas, no contamos con datos de fallos para construir un modelo de comportamiento y (b) las condiciones de funcionamiento de las máquinas pueden ser variables y afectar a los datos generados por éstas.
El AC puede ayudar al MP a: (a) reducir costes a través de la automatización de tareas periódicas tediosas que tienen que ser realizadas por expertos en el área y (b) reducir la probabilidad de grandes daños a la maquinaria gracias a la posibilidad de monitorizarla con una mayor frecuencia sin elevar los costes sustancialmente.
En este trabajo, se proponen algoritmos de AC de propósito general capaces de trabajar en las condiciones anteriores. Además, su aplicación específica al campo del mantenimiento predictivo de maquinaria rotativa basada en el análisis de vibraciones se estudia en detalle, aportando resultados para casos reales. El hecho de disponer sólamente de datos en condiciones de normalidad de la maquinaria nos restringe al uso de técnicas de detección de anomalías. Éste será uno de los bloques principales del presente trabajo.
Por otra parte, el MP también intenta valorar si la maquinaria se encontrará en un estado inaceptable en un futuro próximo. En el segundo bloque se presenta un nuevo algoritmo de aprendizaje en tiempo real (on-line) que será de gran ayuda en esta tarea. Se propone un nuevo algoritmo de aprendizaje on-line para una red neuronas monocapa con función de transferencia no lineal. Además de su aplicación al mantenimiento predictivo, el algoritmo propuesto puede ser empleado en otros escenarios de aprendizaje

on-line como grandes conjuntos de datos, cambios de contexto o datos distribuidos. Algunas de las ideas descritas en este trabajo fueron implantadas en un prototipo de software comercial, GIDAS®. Este software fue desarrollado e implantado en plantas reales por el autor de este trabajo y las experiencias extraídas de su aplicación también se describen en el presente volumen.

Palabras clave: *Aprendizaje Computacional, Detección de Anomalías, Aprendizaje on-line, Datos stream, Análisis de vibraciones, Mantenimiento Predictivo, Ingeniería Mecánica.*

Resumo

O presente traballo sitúase no ámbito de dúas disciplinas, a investigación en Aprendizaxe Computacional (AC) e o Mantemento Predictivo (MP) de maquinaria rotativa. Por unha banda, o AC estuda a problemática de detectar e clasificar patróns en conxuntos de datos extraídos de fenómenos de interese da máis variada natureza. Pola súa banda, o MP é a disciplina que, baseándose na monitorización de variables físicas dos seus compoñentes internos, encárgase de valorar as condicións destes tanto no momento presente como nun futuro próximo co fin último de previr roturas que poden resultar de fatais consecuencias. Neste traballo ponse de relevancia que ambas disciplinas poden beneficiarse da súa sinergia.

O MP supón un reto para o AC debido á natureza dos datos xerados pola maquinaria: (a) as propiedades das medidas físicas recolleitas varían para cada máquina e, debido a que a monitorización debe comezar en condicións correctas, non contamos con datos de fallos para construír un modelo de comportamento e (b) as condicións de funcionamento das máquinas poden ser variables e afectar aos datos xerados por estas. O AC pode axudar ao MP a: (a) reducir custos a través da automatización de tarefas periódicas tediosas que teñen que ser realizadas por expertos no área e (b) reducir a probabilidade de grandes danos na maquinaria grazas á posibilidade de monitorizala cunha maior frecuencia sen elevar os custos sustancialmente.

Neste traballo, propóñense algoritmos de AC de propósito xeral capaces de traballar nas condicións anteriores. Ademais, a súa aplicación específica ao campo do mantemento predictivo de maquinaria rotativa baseada na análise de vibracións estúdase en detalle aportando resultados para casos reais. Debido a contar só con datos en condicións de normalidade da maquinaria, estamos restrinxidos ao uso de técnicas de detección de anomalías. Éste será un dos bloques principais do presente traballo.

Por outra banda, o MP tamén intenta valorar si a maquinaria atoparase nun estado inaceptable nun futuro próximo. No segundo bloque do presente traballo preséntase un novo algoritmo de aprendizaxe en tempo real (on-line) que será de gran axuda nesta tarefa. Proponse un novo algoritmo de aprendizaxe on-line para unha rede neuronas monocapa con función de transferencia non lineal. Ademais da súa aplicación ao mantemento predictivo, o algoritmo proposto pode ser empregado en escenarios de aprendizaxe on-line como grandes conxuntos de datos, cambios de contexto ou datos

distribuídos.
Algunhas das ideas descritas neste traballo foron implantadas nun prototipo de software comercial, GIDAS®. Este software foi desenvolvido e implantado en plantas reais polo autor deste traballo e as experiencias extraídas da súa aplicación tamén se describen no presente volume.

Palabras clave: *Aprendizaxe Computacional, Detección de Anomalías, Aprendizaxe on-line, Datos stream, Análise de vibraciones, Mantemento Predictivo, Enxeñaría Mecánica.*

Contents

1 Introduction **1**

1.1 Anomaly detection 2
1.2 On-line learning 3
1.3 Data-driven predictive maintenance 4
1.4 Outlook of this thesis 6

2 Anomaly detection: background and challenges **7**

2.1 Anomaly: definition 8
2.2 Anomaly detection: problem statement and challenges 8
2.3 Anomaly detection scenarios classification 10
2.4 Anomaly detection techniques taxonomy 13
2.4.1 Classification based anomaly detection techniques 13
2.4.2 Nearest Neighbor based anomaly detection techniques 15
2.4.3 Clustering based anomaly detection techniques 18
2.4.4 Statistical anomaly detection techniques 19
2.4.5 Spectral anomaly detection techniques 21
2.4.6 Information theoretic anomaly detection techniques 22
2.5 Stream anomaly detection 23

3 Anomaly detection: proposals and results **27**

3.1 A minimum volume covering approach with a set of ellipsoids 29
3.1.1 Single minimum volume covering ellipsoid: formulation and improvements 31
3.1.2 Proposed model 32
3.1.2.1 Dealing with the determinant problem 32
3.1.2.2 Considering a set of covering ellipsoids 34
3.1.2.3 Removing outliers 35
3.1.2.4 Avoiding binary variables 36
3.1.2.5 Karush-Kuhn-Tucker conditions 37
3.1.2.6 Proposed bilevel algorithm 42
3.1.3 Sensitivity analysis and classification rule 44
3.1.4 Experimental results 45
3.1.5 Artificial Data sets 45

3.1.5.1 Square example 46
3.1.5.2 Normal data 46
3.1.5.3 Spiral example 48
3.1.5.4 Clustering applications 52
3.1.6 Discussion and future work 53
3.2 Nearest neighbor anomaly detection based on extreme value statistics . 55
3.2.1 Method description 55
3.2.2 Experimental results 57
3.2.2.1 Artificial data sets 58
3.2.2.2 Real data sets 58
3.2.3 Discussion and future work 60
3.3 On line anomaly detection via passive-agressive one class classification . 61
3.3.1 Passive-Aggressive one class classifier 61
3.3.1.1 Background 61
3.3.1.2 Proposed Method 63
3.3.2 Stream anomaly detection algorithm 68
3.3.2.1 The Bernouilli CUSUM chart 68
3.3.2.2 Proposed algorithm: OSDAD 70
3.3.3 Experimental Results 73
3.3.3.1 One class classification in stationary and dynamical environments 73
3.3.3.2 Anomaly detection scenarios 80
3.3.4 Discussion an future work 81

4 On-line learning: incremental, non stationary and distributed scenarios 85
4.1 Background: Non linear single layer neural network learning algorithm . 88
4.1.1 Concept-drift learning algorithm 90
4.2 Diminishing complexity and incrementing efficiency: proposed algorithm 91
4.2.1 Regularization property 93
4.2.2 Main differences and advantages 95
4.3 Experimental Results 96
4.3.1 Regularization behavior 97
4.3.2 Interaction between forgetting factor and regularization 97
4.3.3 Stationary data sets comparison 100
4.3.4 Non stationary scenarios 103
4.3.5 Distributed environments 105
4.4 Dicussion 107

5 Automatic fault detection in rotating machinery: theory and background 109
5.1 Rotating machinery fault detection based on vibration signatures 109
5.2 Mechanical vibration physics: an introduction 112
5.2.1 Basic concepts and terminology 112
5.2.2 Fundamental equations of vibration 116
5.2.3 Physical models of vibrations of a rotating machinery 122
5.3 Mechanical vibration technologies 124
5.3.1 Vibration transducers: characteristics and installation 125
5.3.2 Analysis techniques 129
5.4 Rotating machine fault detection from an anomaly detection perspective 133

6 GIDAS®: An Automatic Fault Detection System based on Vibration Signatures 137
6.1 Aim and scope of the system 138
6.2 Modern wind turbines: concepts and design 139
6.3 Modern horizontal power wind mill main components 143
6.4 Automatic predictive maintenance system requirements 150
6.5 System design 153
6.6 Fault Detection in production environments: Sotavento and Production wind mill farm experiences 160

7 Application of ML to industrial fault detection: Rolling element bearing fault assessment 163
7.1 Rolling element bearings fault detection and diagnosis 164
7.1.1 Bearing Characteristic Frequencies 164
7.2 Rolling element bearing fault assessment: study cases 171
7.2.1 Case 1: UNSW Simulator Data 172
7.2.2 Case 2: Laboratory data I 173
7.2.3 Case 3: Laboratory data II 176
7.2.4 Case 4: Real Scenario, wind mill power turbines 177
7.3 One Class ν-Support Vector Machines 177
7.4 Fault detection: experimental results 180
7.4.1 Vibration data and fault severity coherence 180
7.4.2 Fault detection via frequency domain transformations 184
7.4.3 Fault Detection via on-line anomaly detection 185
7.4.4 Fault detection via alternative transformations 189
7.4.4.1 Recurrence Time Statistics 189

7.4.4.2 Proposed methodology 193
7.4.4.3 Experimental Results 194
7.5 Fault diagnosis strategy: methodology and experimental results 199
7.5.1 Proposed method 199
7.5.2 Experimental results 203
7.6 Fault evolution assessment: methodology and experimental results . . . 208
7.7 Discussion 209

8 Conclusions, main contributions and future work 211

I Resumen del trabajo 215

II Author's key publications and mentions 223

Bibliography 227

List of figures

2.1 A simple example of anomalies in a 2-dimensional data set (red points are anomaly candidates). 9

3.1 Graphic illustration of the classification rule and sensitivity analysis for the Normal data set example. 45
3.2 Evolution of the proposed algorithm for the Square data set: (a) initial step, (b) iteration 2, (c) final solution and (d) solution of the problem with binary variables (equations (3.25)-(3.32)). 47
3.3 Evolution of the proposed algorithm for the Normal data set: (a) initial step, (b) iteration 2, (c) iteration 3, (d) iteration 4, (e) final solution and (f) solution of the problem with binary variables (equations (3.25)-(3.32)). 49
3.4 Evolution of the proposed algorithm for the Normal data set (number of ellipsoids overestimated): (a) initial step, (b) iteration 2, (c) iteration 3, (d) iteration 4, (e) final solution and (f) solution of the problem with binary variables (equations (3.25)-(3.32)). 50
3.5 Evolution of the proposed algorithm for the Spiral data set: (a) initial step, (b) iteration 2, (c) iteration 3, (d) iteration 4, (e) final solution and (f) solution of the problem with binary variables (equations (3.25)-(3.32)). 51
3.6 Illustration of EVOC normal support capture. 58
3.7 Accuracy for Wine data set when changing hyperparameters. 59
3.8 Training data patterns with $\xi_t > 0$. 64
3.9 Outcome of the proposed model for stationary dataset #1. 75
3.10 Outcome of the proposed model for non-stationary dataset #2. 76
3.11 Outcome of the proposed model for non-stationary dataset #3. 76
3.12 Outcome of the proposed model for non-stationary dataset #4. 77
3.13 Scalability of the proposed model for datasets #1, #2, #3 and #4. . . . 77
3.14 AUC obtained varying parameter σ while $C = 0.1$, $C_r = 0.00075$ and $q_i = 1e-5$. 80
3.15 AUC obtained varying parameter C while $\sigma = 8.5$, $C_r = 0.00075$ and $q_i = 1e-5$. 80
3.16 AUC obtained varying parameter C_r while $\sigma = 8.5$, $C = 0.1$ and $q_i = 1e-5$. 82

3.17 AUC obtained varying parameter q_i while $\sigma = 8.5$, $C = 0.1$, and $C_r = 0.00075$. 82
3.18 Change point detection of the proposed model for dataset by Yamanishi and Takeuchi. 83

4.1 Architecture of a single-layer feedforward neural network. 89
4.2 Regularization behavior for Ionosphere data set. 98
4.3 Regularization behavior for linear data set. 98
4.4 Test error for an artificial data set in function of the regularization term and the forgetting factor. 100
4.5 Comparative results for the twelve stationary data sets 102
4.6 Results for the first non stationary data set. 104
4.7 Results for the chaotic time series. 105
4.8 Results for the distributed stairs data set. 106
4.9 Distributed stairs data set with concept drift. 107
4.10 Classification error for the distributed stairs data set with concept drift. 107

5.1 1 Degree of Freedom System. An example of this kind of system are shock-absorbers 116
5.2 Damped Vibration example 118
5.3 Forced vibration example (combination of a transitory and a stationary component) 120
5.4 Vibration transference of a mechanical system 121
5.5 Scheme of a 2 degrees of freedom system. 121
5.6 University of South Wales' fault test rig photo and scheme (courtesy of the authors) 123
5.7 34-DOF modeling of the gearbox (courtesy of the authors) 124
5.8 Velocity pick-ups 126
5.9 Accelerometers 128
5.10 Proximity probe 129
5.11 Example of the spectrum of a healthy and faulty bearing. 131
5.12 Architecture of fault detector based on anomaly detection techniques . . 136

6.1 Examples of horizontal/vertical wind mill designs. 140
6.2 Power and efficiency curve of a wind turbine 142
6.3 Examples of power curves for two types of wind turbines. 142
6.4 Main components of a modern wind turbine. 143
6.5 Section of a blade showing upper and lower shells and two webs, respectively. 144

6.6 View of wind turbine hub. 144
6.7 Wind turbine with main bearing integrated in the gearbox. 145
6.8 Wind turbine with two main bearings. 146
6.9 Spherical roller bearing, from Bonus (1999) 147
6.10 Examples of spur and helical gears. 148
6.11 Examples of planetary gear principle with outer fixed, three revolving planets and a planet carrier in the middle. 148
6.12 Examples of roller bearings: Spherical bearing (left) and cylindrical bearing (right). From SKF (1997). 148
6.13 Block design of the GIDAS system. 153
6.14 Installation of GIDAS system in a production wind mill. 155
6.15 Main control module of GIDAS system. 158
6.16 Analysis modules of GIDAS system. 159
6.17 Sotavento's main control station and wind farm. 161

7.1 Main geometry of a rolling element bearing 167
7.2 Main identities of a rolling element bearing 167
7.3 Impact of a fault for an outer-race fault (equivalent for an inner-race fault with BPFI frequency) and real wear of an inner-race. 168
7.4 Qualitative evolution of the power spectrum of a bearing fault. 171
7.5 UNSW fault test rig photo and scheme (reproduced with permission of the authors) 173
7.6 34-DOF modeling of the gearbox (reproduced with permission of the authors) 174
7.7 Bearing test rig for the test-to-failure experiment: (a) photo; (b) installation schema. 175
7.8 Photography and schematic description of the experimental system. . . 176
7.9 Normal state spectrum of simulated data. 182
7.10 Incipient fault spectrum of simulated data. 182
7.11 Fault detection for the simulated fault growth case. 183
7.12 Correlation between nomal state assessment of one-class ν-SVM and fault width. 183
7.13 The Bernoulli CUSUM charts for Case II dataset. 187
7.14 The Bernoulli CUSUM charts for Case IV data. 188
7.15 Performance of the OSDAD algorithm for the wind mill dataset with different combinations of parameters. 188
7.16 Recurrence Time Statistics calculation illustration. 191
7.17 Proposed method model estimation scheme. 194

7.18 Comparison of proposed methodology vs. human detection 196
7.19 ROC curves of the proposed methodology under different condition (TP: true positives, FP: false negatives) 197
7.20 Fault detection of the proposed method in a real scenario. 198
7.21 Architecture of the automatic bearing diagnosis system (numbers indicate the order of the process). 200
7.22 Signal processing steps for envelope analysis. 203
7.23 (a) Example of normal state power spectrum (b) ν-SVM detection on Case II experiment. 204
7.24 (a) Incipient fault power spectrum and selected band. (b) Incipient fault envelope spectrum. 205
7.25 (a) Advanced fault power spectrum and selected band. (b) Advanced fault envelope spectrum. 205
7.26 Normal state power spectrum. 206
7.27 (a) Power spectrum and selected band of an inner race defect. (b) Envelope spectrum of the inner race defect. 207
7.28 (a) Power spectrum and selected band of a ball defect. (b) Envelope spectrum of the ball defect. 207
7.29 (a) Power spectrum and selected band of an outer race defect. (b) Envelope spectrum of the outer race defect. 207
7.30 Global acceleration energy and prediction. 209

List of tables

3.1 A comparison of the times required by the binary, relaxed and algorithmic approaches respectively. . . . 52
3.2 Results for clustering problems (best results for each data set are boldfaced) 53
3.3 Results of EVOC method for UCI data sets. . . . 60
3.4 Hyperparameters of the proposed model. . . . 64
3.5 Characteristics of experimental datasets. . . . 74
3.6 Parameters of the model for each experiment. . . . 74
3.7 UCI datasets description. . . . 79
3.8 Combinations of parameters used in cross validation. . . . 79
3.9 AUC results and CPU time achieved on UCI benchmarks datasets. . . . 79

4.1 Data sets employed in the comparative study. . . . 101

7.1 Bearing Vibration Frequency Characteristics. . . . 168
7.2 Characteristic fault frequencies of 6205-2RS SKF bearing. . . . 176
7.3 Mean test error for anomaly detection algorithms based on frequency domain feature extraction . . . 185
7.4 Classification accuracy on the experimental data sets for the proposed methodology. . . . 196

List of algorithms

1 Solving minimum volume set of covering ellipsoids (MCSE algorithm). . . 43
2 Proposed EVOC classification method 57
3 OSDAD Algorithm 71
4 Non linear single layer neural network training 94

CHAPTER 1

Introduction

This thesis is devoted both to Machine Learning (ML) research and its application to a relevant industrial field, predictive maintenance of machinery. Predictive maintenance of industrial machinery is the discipline which, based on the measurement of physical conditions of its internal components, assesses its present and near future condition in order to prevent fatal failures. The main objective of the present work is to develop novel ML algorithms which can tackle predictive maintenance problem in real scenarios. Namely, two main restrictions have to be taken into account: (a) values of monitored physical signals vary for each machine and, since each component starts working at good conditions, we will not have data under fault conditions in order to build an individual ML model and (b) working conditions of the machine are changeable in many situations and this is reflected in captured data.

This work flows from general principles to details. First, general purpose ML techniques able to deal with the aforementioned conditions are proposed and, ultimately, we detail their application to the specific field of predictive maintenance of rotating machinery. Availability of only normal state data in order to build a model restricts us to the use of anomaly detection algorithms, which will be one of the main blocks of this work. In addition to current condition determination, predictive maintenance also aims at assessing the state of the machine in the near future. This drives us to the second main block devoted to ML, on-line learning algorithms, which will help us in this task.

Before diving the specific aspects of each topic, in this introduction a summary of the main intentions of the present work in each field is given. Each block of the thesis covers a very specific topic which constitutes a subfield by itself and so they have been made as self-contained as possible.

1.1 Anomaly detection

First block is devoted to the development of ML anomaly detection methods, or in other words, to tackle classification in the absence of counterexamples. Anomaly detection methods are gaining importance in the last decade. In this "data-driven" modern world we are overwhelmed by huge amounts of data impossible to deal with and it turns out that anomalies, or if preferred unexpected events, although being unusual and finding them could seem "mining a needle in a haystack" carry the most important information. These are some examples of this fact: there are millions of credit card payments per day but only a small percentage of them are potential frauds that could cost losses to customers and companies, a commercial web portal receives millions of requests per day but only some requests can carry malicious intentions, stock exchange moves tons of data per second but only a tiny percentage are in the position of being an "inside trader". But not only negative examples can be found: retailers sell different baskets of products to thousands of clients each day, anomalous selling patterns can be a sign of changing tastes that can be analyzed and exploited by the retailer to give a better service to their customers.
Although very different in nature, all anomaly detection algorithms share a common characteristic: they try to determine the region of input space which received data under normal conditions belong to. Some algorithms choose beforehand a specific shape for this "normal region". In this work, three novel anomaly detection algorithms are proposed:

- Minimum Volume Set of Covering Ellipsoids (MSCE). The most common choices in state-of-the art algorithms for the "normal region" shape have been a sphere or an ellipsoid. This choice translates learning process into the classic Minimum Volume Covering Ellipsoid problem, in which a minimum volume ellipsoid which covers "normal" data samples is built. It turns out that this choice can be very restrictive and give poor results in situations where, for example, we have to deal with multi-modal or noisy data. In this first proposed algorithm we extend the classic Minimum Volume Covering Ellipsoid (MVCE) problem to a robust algorithm which obtains a minimum volume set of covering ellipsoids. It will be shown that this algorithm is more flexible and robust than the classic MVCE and can approximate complex regions in an accurate way.

- Extreme Value Statistics One-class Classifier (EVOC). If we could not make any assumption about the shape or nature of the "normal region" and we still need to

comply with an anomaly detection task, we need a criterium to decide whether a new data instance is anomalous or not. The second proposed algorithm aims at obtaining an accurate anomaly detection model based only on a distance measure between instances and a base data set of normal state patterns. This algorithm can be applied to any type of complex data (sequences, graphs, ...) if only a distance measure is available.

- On-line Stream Data Anomaly Detection (OSDAD). Most of state-of-the art anomaly detection algorithms, and also the two previous ones, deal with the case where a dataset of normal data is available beforehand and we train a model in a batch manner. But, in many situations, a stream of data patterns arrives continuously and we want to highlight those parts of the stream where an anomaly (or change) has occurred while continuously adapting to new scenarios. The final proposed model tackles this on-line anomaly detection problem through a Passive-Agressive classification algorithm combined with a well stablished CUSUM chart.

1.2 On-line learning

Second block is devoted to on-line learning on stream data. On-line ML is a model of induction that learns one instance at a time. Its goal is to predict a correct output value for each instance (a label or a real valued property) only based on the current model trained with data previously seen and the current input. In the context of stream data, the algorithm receives periodically (after each input pattern or set of inputs) feedback of the correct output and, based on this information, it has to continuously update its model based only on the last information received. This limitation on the data window available to update the model can be imposed by one of the following requirements: (a) real-time restrictions, (b) database size or (c) change of conditions on data. On-line learning algorithms are gaining applicability also due the proliferation of data in modern times. Web click-through data or real-time stock market analysis tools are examples where stream data ML algorithms find applicability with a huge impact.
In this work we review the problematic of on-line learning in its different flavors (big data sets, changing contexts, distributed data) and propose a novel on-line learning algorithm for a neural node with a non-linear output function. The proposed algorithm is able to continuously train a neuron in a one pattern at a time manner. If some conditions are hold, it analytically ensures to reach a global optimal model. As it will be detailed in this work, proposed algorithm covers previous state-of-the art algorithms

such as classic Recursive Least Squares (RLS) [99] as special cases and is able to tackle the aforementioned three different scenarios of stream data learning: big data sets, changing contexts and distributed data.

1.3 Data-driven predictive maintenance

Third block dives in the details of the application field of this thesis, automatic predictive maintenance software systems. In order to evoke the motivation of the present work in reader's mind we start with a common episode that virtually everybody has suffered at some point in his/her lifetime:

> You are driving your car in the highway and, suddenly, you feel that something is wrong with your car. At first impression you cannot explain it but you feel it somehow. Maybe there's a strange sound, or a strange response of the steering but it is physically noticeable. Eventually, it drives you to the garage for a overhaul where you discover that you have had an unnoticed problem for a long time that will cost you an expensive replacement and inconveniently prevent you from driving.

Fortunately, the end of this episode isn't always like that. Many times, when repairing the fault, you discover that it has a simple solution thanks to having noticed the problem in time. In these occasions, driver is acting as part of the maintenance of his/her own car. Each time he drives his car he has the opportunity of noticing whether there is any kind of internal problem depending on his ability to notice anomalies. Mechanical faults are usually caused by materials fatigue and this is usually noticeable before the breakdown is dramatic. Let's move this episode to an industrial environment:

> A 4$ million wind turbine is working 24/7 offshore in the Irish Sea in the middle of the winter. Due to wear, one of its main components is producing undesired fatigue in the whole mechanism. An early stop would prevent a major breakdown that could cost hundreds of thousand of dollars.

Unfortunately, in this second case there is not any human being around able to notice that something is not working and that the machine should be stopped in order to

avoid a fatal ending. Since human inspection is not possible for this case, we need to build a software able to "hear" or "feel" any fault symptoms in the machinery before they become dangerous, interpret those symptoms and report its presence in order to avoid costly breakdowns. This is the main objective of this work in its application field, fault detection and assessment.

A software of this kind would fall into the Predictive Maintenance paradigm. As previously stated, predictive maintenance aims at reducing the maintenance costs of a plant's budget by determining the status of equipment through performing a periodic monitoring of signals which describe its state. Thanks to an early detection of faults, fatal breakdowns are avoided and scheduling maintenance tasks when they are most cost-effective is possible. Predictive maintenance technologies are key to extending equipment life, reducing maintenance costs and increasing asset exploitation. In order to build such a kind of software, it is necessary to have a physical phenomenon where fault symptoms are apparent. Vibration analysis is one of the most effective techniques to evaluate industrial equipment condition, detecting defects and avoiding fatal failures. As equipment begins to degrade, it may exhibit symptoms that can be revealed, if adequate methodology is used, to detect failure precursors. Integrating sensors with predictive maintenance techniques can avoid unnecessary equipment replacement, save costs, and improve process safety, availability, and efficiency. The impact of advances in this field soars when we take into account the continuous growth of some rotating machinery based markets such as wind mill power generation, which in 2012 had a cumulative power of 238 GW worldwide [57].

But, in order to fully automate the process of detecting a fault, a predictive maintenance software based on vibration captures should be able to distinguish between good and abnormal conditions.

Machine Learning techniques can give an answer to this problematic and will be the topic of the second main part of this thesis. How to deal with fault detection problem from an anomaly detection perspective is detailed and tackled through both state-of-the art and the proposed algorithms in this thesis. Some of this algorithms were made part of GIDAS® software. This system was developed in collaboration with INDRA Systems S.A. by the author of this work. The experiences in applying ML to fault detection in real plants through this software are also described. This thesis extends the work carried out with this software, studying its potential fault detection accuracy through the novel algorithms proposed in this work.

1.4 Outlook of this thesis

In this chapter we have introduced the main topics to be discussed in this work. The first block is covered by chapters 2 and 3. Chapter 2 gives an introduction to the anomaly detection problem, its main challenges, the state-of-the-art algorithms that can be found in the literature and the families in which they can be grouped. Chapter 3 gathers together the anomaly detection algorithms proposed in this work. Subsequently, second block is covered by chapter 4 where an extended introduction to on-line learning and the proposed learning algorithm for neural nodes is presented. Chapter 5 marks the turning point in the reading between Machine Learning contents and the application field. In this chapter, an introduction to vibration analysis and maintenance main concepts is introduced in order to make the book self-contained. Also in this chapter, the anomaly detection strategy for fault detection that is used in the remainder of the work is presented. Chapter 6 presents GIDAS® software, the workbench used in this work for fault detection in real production scenarios and the pilot experience carried out in this work is described. Chapter 7 we detail all the results obtained with the proposed algorithms of this work in the case of fault detection on rolling element bearings, one of the most common components of modern rotating machinery. Finally, in chapter 8 the main conclusions and contributions of this work are summarized.

CHAPTER 2

Anomaly detection: background and challenges

In essence, anomaly detection refers to the problem of detecting data which do not conform to an expected behavior. In mathematical terms, we have a phenomenon which emits descriptive data $\{x_1, x_2, \dots\}$ about itself. Under normal conditions, these data are generated under a probability distribution $P(x)$. Thus, this distribution determines that there are data which are unlikely under normal conditions. Having the actual $P(x)$, we could examine a sample of i.i.d. emitted data samples $\{x_1, x_2, \dots, x_N\}$, for example calculating their likelihood under normal conditions and evaluating whether there is a deviation in the underlying phenomenon. If that is the case, it could be necessary to trigger an assessment and adequate counteractions. For instance, a deep fall in the likelihood of traffic patterns in a news server could indicate that a hacked computer is trying to attack the server or that an unexpected event has occurred and so, an unexpected amount of people are accessing in a novel pattern. This example remarks the fact that anomaly detection is a key model which, in real situations, triggers a further complex decision making process which finally treats the anomaly in a proper way.

It turns out that in real situations we do not have the actual probability distribution $P(x)$ which fully describes the phenomenon, so we have to circumvent this eventuality and come up with a process that emulates the aforementioned detection algorithm. Detecting outliers or anomalies in data has been studied in the statistics community as early as the 19th century [70]. The amount of domains in which this philosophy has encountered applicability (intrusion detection [139], tumor detection [223], fraud detection [13], sensor and machine fault detection [87] and many more) has made impossible to find any generic all-purpose technique. Thus, over time a variety of anomaly detection techniques have been developed in several research communities. Many of these techniques have been specifically developed for certain application domains, while others are more generic.

Anomaly detection will be one of the three main topics of this thesis work so we start up in this chapter with an introduction to the vast literature on anomaly detection techniques through a categorization of the main problems and approaches that can be found. In the next chapter, three novel algorithms for anomaly detection are presented. As we will see, algorithms of chapter 3 expand the range of techniques of the taxonomy presented hereunder.

2.1 Anomaly: definition

Although the notion of what is an anomaly may seems evident at first hand, many definitions in the literature are a bit vague at this point and this does not help to clarify what the techniques are really doing and how to apply them in a specific domain. In this work we will use the following definition for an anomaly which relies on the existence of an unknown probability distribution $P(x)$ of the data:

Anomalies are data patterns which are unlikely to appear under the normal condition's data emission probability distribution $P(x)$.

There are some subtleties emerging from this definition that are worth clarifying. An anomaly is not necessarily an impossible event under normal conditions. For instance, a high vibration level in a machine could be due to a temporal peak load or to a bad sensor capture which are also normal situations that do not mean a need of an overhaul. Whether an anomaly is certainly an abnormal or undesired behavior and not just an unlikely data pattern is a decision which has to be made by higher level complimentary techniques. An anomaly detection technique only assesses whether a data pattern is likely or unlikely to happen.

2.2 Anomaly detection: problem statement and challenges

At an abstract level, anomaly detection consist on building a model able to decide for any data pattern whether it is likely or unlikely under normal conditions. For building that model, a set of data patterns generated by the underlying phenomenon

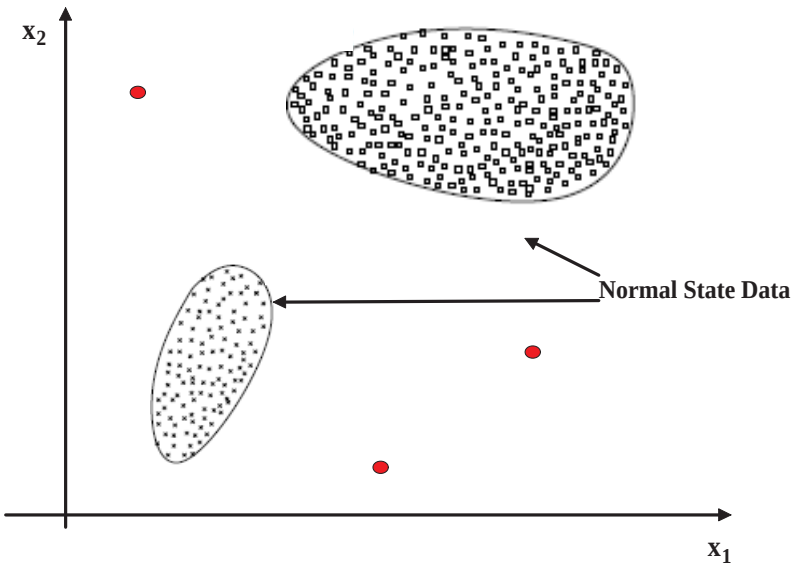


Figure 2.1: A simple example of anomalies in a 2-dimensional data set (red points are anomaly candidates).

$\{x_1, x_2, \ldots, x_N\}$ is available. Thus, the aim is to define a region into the input space which represents a normal behavior and declare any observation which does not belong to this normal region as an anomaly (see figure 2.1). But several factors make this apparently simple problem very challenging:

- The region which encompasses every possible normal behavior could be very complex in nature. Thus, deciding a prior shape of this region (e.g., Gaussian, Mixture of Gaussians, etc.) could be very difficult, if not impossible, for a problem at hand.

- The data set available for building the model is usually unlabeled and includes some unlikely (or even abnormal) data patterns. Thus, it is necessary to decide during the model building phase which data patterns are unlikely (or abnormal) in order not to include them as part of the region of normal behavior. This decision usually leads to an imprecise boundary between normal and anomalous behavior and always involves a trade off between false alarm and true alarm rate. A model that only cares about covering the available data patterns (a simplistic approach could be a model which covers the whole data input space) would never or almost ever detects that something unlikely is happening, leading to a system with a low or null true alarm rate and so making it useless for practical purposes. On the

other hand, a model which rejects many data patterns could lead to a situation in which the learnt normal region is too tight so too many patterns are wrongly considered as unlikely. This could lead to a useless system for practical purposes because it produces higher costs (e.g., if the patterns detected as unlikely are to be assessed by a human) or a higher false alarm rate (e.g., due to the noise that normal patterns incorrectly assessed as unlikely introduce in subsequent phases).

- In many domains normal behavior keeps evolving and a current notion of normal behavior might not be sufficiently representative in the future. These scenarios impose the necessity of detecting when the normal behavior has changed and thus continuously evolving the model.

- Availability of labeled data for training/validation of models used by anomaly detection techniques is usually a major issue. In many situations we can only treat normal data to construct a model, so classical hyperparameter selection techniques like for example cross validation [15] become invalidated or of limited use.

Due to the above challenges, the anomaly detection problem, in its most general form, is not easy to solve. In fact, most of the existing anomaly detection techniques solve a specific formulation of the problem in a very specific domain. The formulation is conditioned by various factors such as: (a) nature of the data, (b) availability of labeled data, (b) type of anomalies to be detected, etc. These factors are usually determined by the application domain in which the anomalies need to be detected. Researchers have adopted concepts from diverse disciplines such as statistics, machine learning, data mining, information theory, spectral theory, and have applied them to specific problem formulations. Anomaly detection has been the topic of a number of surveys and review articles, as well as books. This introduction is mainly based on the recent categorizations of anomaly detection problems and techniques which have been published in the literature. Namely, it is mainly based on the surveys presented in [11][53][108][160][161][186]. The author refers to these publications for further details.

2.3 Anomaly detection scenarios classification

As it was mentioned in the previous section, anomaly detection application domains can impose beforehand restrictions which dramatically determine the design of the

anomaly detection algorithm. Among all the possible restrictions, the following are the most common:

1. Nature of input data: Input is generally a collection of data instances captured under normal conditions of the studied phenomenon. Each data instance can be described using a set of attributes and these can be of different types such as binary, categorical or continuous and in the case of multivariate data instances, all attributes might be of the same type or might be a mixture of different data types. The nature of attributes determines the applicability of anomaly detection techniques. For example, different statistical models have to be used for continuous and categorical data. Similarly, for nearest neighbor based techniques, the nature of attributes would determine the distance measure to be used. Often, instead of the actual data, the pairwise distance between instances might be provided in the form of a distance (or similarity) matrix. In such cases, techniques that require a specific form of input data (e.g., Euclidean space) are not applicable. Input data can also be categorized based on the relationship present among data instances [229]. Most of the existing anomaly detection techniques deal with data in which no relationship is assumed among the data instances. In general, data instances can be related to each other (e.g., temporally or spatially). In this second case, the algorithm may have to take into account this relationship in order to obtain accurate results. In this work we will mainly deal with data in Euclidean spaces, although some of the proposed techniques can deal with data in any metric space.

2. Type of anomaly: Anomalies can be classified into three categories:

   - Point anomalies. If an individual data instance can be considered as anomalous with respect to the rest of data, then the instance is termed as a point anomaly. This is the simplest type of anomaly and is the focus of the majority of research on anomaly detection.
   - Contextual anomalies. If a data instance is anomalous in a specific context (but not otherwise), then it is termed as a contextual anomaly (also referred to as conditional anomaly [222]).
   - Collective anomalies. If a collection of related data instances is anomalous with respect to the entire data set, it is termed as a collective anomaly. The individual data instances in a collective anomaly may not be anomalies by themselves, but their occurrence together as a collection is anomalous.

   In this work we will focus our attention on point anomalies.

3. Data Labels: In this aspect we can found ourselves in any of the following scenarios: (a) supervised anomaly detection, in this scenario we assume that we have labelled instances for both normal as well as abnormal class. Typical approach in such cases is to build a predictive model for normal vs. anomaly classes, (b) semi-supervised anomaly detection, in this scenario we assume that we have labels only for the normal class. Since they do not require labels for the anomaly class, they are more widely applicable than supervised techniques. The typical approach used in such techniques is to build a model for the class corresponding to normal behavior, and use the model to identify anomalies in the test data; (c) unsupervised anomaly detection, in this scenario we assume that we have labels neither for the normal behavior nor for any abnormal case. Thus, the techniques in this category make the implicit assumption that normal instances are far more frequent than anomalies in the test data. Many semi-supervised techniques can be adapted to operate in an unsupervised mode by using a sample of the unlabeled data set as training data. Such adaptation assumes that the training data contains very few anomalies and the model learnt during training is robust to these few anomalies. Since getting a labeled set of data instances is usually difficult and costly in a vast amount of scenarios, the third scenario is the most usual and the most widely studied. Most of the anomaly detection techniques assume that the available dataset can contain a very small fraction of abnormal or rare events (e.g., if we were to model the behavior of a user of an online newspaper, we could use its access log as the dataset representing its normal consumption behavior, but we cannot assume that there are not some abnormal accesses that the user has clicked by error) and try to build a model which covers the majority of the dataset but those data patterns which appear to be rare or abnormal. In this work we will focus on semi-supervised and unsupervised anomaly detection scenarios.

4. Anomaly detection desired output: Typically, the desired outputs produced by anomaly detection techniques are one of the following two types: (a) scores, scoring techniques assign an anomaly score to each instance in the test data depending on the degree to which that instance is considered an anomaly; (b) labels, techniques in this category assign a label (normal or anomalous) to each test instance. Many of the techniques which assign a label to each data pattern possess an underlying score to which they apply a threshold in order to produce a final label. This threshold stage could be avoided if the domain requires a score. Scoring based anomaly detection techniques allow the analyst to use a domain-specific threshold to select the most relevant candidates for anomalies.

The techniques proposed in this work pertain to both the thresholded and labeling groups.

2.4 Anomaly detection techniques taxonomy

As we have mentioned in the Introduction, a vast literature on anomaly detection has emerged due to the large amount of contexts where it can be applied. This variety requires a taxonomy of techniques that brings to order the field and standardizes the terms being used. In this section we introduce such a taxonomy. This taxonomy will also help to classify the proposed techniques of chapter 3.

2.4.1 Classification based anomaly detection techniques

Classification can be defined as the process of learning a model (classifier) from a set of labeled data instances (training) in order to categorize future instances into one of the classes using the learnt model. Classification based anomaly detection techniques operate in a similar two-phase fashion and are named in the literature as One Class Classifiers (OCC) [165]. The training phase learns a classifier using the available labeled (or unlabeled) training data. In case the algorithms have to deal with unlabeled data, the assumption that abnormal events are rare in the available data is used as a regularizer criteria for the normal state region learning. In the detection phase, the classifier is applied to categorize new observations as normal or abnormal. This kind of techniques have attracted much interest in recent years due to its numerous possibilities of application in situations when only data gathered from one class is available and we aim to discriminate between data gathered in an specific state and data gathered in other (possibly still unknown) states [162]. It also has been reported that, depending on the problem at hand, OCC methods can outperform classic two-class classifiers [115][116]. OCC techniques assume a prior basic approximate shape of the normal state data region and try to adapt that shape to the data set in a regularized way (avoiding potential abnormal patterns in the available data set and covering the least input space as possible). This principle avoids the need of trying to, for example, fit a huge amount of parameters in a probabilistic approach, when we only are interested in the final normal state region. In this niche, the kernel methods' literature is specially prolific. We can find mainly two seminal approaches which have remained a constant

in the past years: (a) Support Vector Data Description (SVDD) [234] that adjusts a minimum volume hypersphere to the data set in the feature space, and (b) the approach based on finding the maximum margin hyperplane which separates data from the origin in [209]. When Radial Basis Function (RBF) kernel is used with these two methods (which is the common choice in the majority of the works and applications due to its high performance), both approaches are equivalent and they can be regarded as reduced set Parzen Density Estimators [209]. Recently, new versions of SVDD based on adjusting a ellipsoid to the data in the feature space have been presented with promising results [62]. It has been observed that although these models include a regularizing term, the ability of avoiding potential abnormalities in the data set could be very imprecise in some situations. Improvements to the basic models trying to obtain more robust methods have been proposed such as the work in [199][200], variants of Support Vector OCC models [221] and extensions to detect anomalies in temporal sequences [155][156]. A second type of techniques which can be included in this section are Rule Based Anomaly Detection techniques, which aim at learning rules that capture the normal behavior of a system. If a test instance is not covered by any such rule, then it is considered as an anomaly. The first step is to learn rules from the training data using a rule learning algorithm, such as RIPPER [56], Decision Trees [15], etc. Each rule has an associated confidence value which is proportional to the ratio between the number of training instances correctly classified by the rule and the total number of training instances covered by it. Second step is to find the rule that best captures each test instance. The inverse of the confidence associated with the best rule is the anomaly score of the test instance. Several minor variants of the basic rule based technique have been proposed [78][105][142][202][238]. Association rule mining [9] has also been used for one-class anomaly detection by generating rules from the data in an unsupervised fashion. Association rules are generated from a categorical data set. To ensure that the rules correspond to strong patterns, a support threshold is used to prune out rules with low support [229]. Since frequent item sets are generated in the intermediate step of association rule mining algorithms, [104] proposed an anomaly detection algorithm for categorical data sets in which the anomaly score of a test instance is equal to the number of frequent item sets it occurs in. Applications of these techniques can be found in [21] [35][143] [158][159][179][191][231][249].
In the literature, the following characteristics (see [165]) have been remarked as desirable for OCC models:

1. *Robustness to outliers*: In OCC methods it is assumed that training set characterizes the target distribution. However, in real life scenarios this data set can be

contaminated by outliers. These can lead to great deviations from the optimal OCC model so it is desirable to have an strategy that avoids these outliers.

2. *Incorporation of known outliers*: If some data from the alternative class is available, it is desirable to incorporate them in the analysis in order to tighten the description.

3. *Ease of configuration*: One of the most important feature of one class classifiers is the number of parameters and the availability of a methodology to estimate them for specific situations. A reduced number of parameters with predictable behavior of the model is desirable. The incorporation of *magic parameters* (those such that a bad estimation of them leads to a poor class description) should be avoided.

4. *Resource requirements*: Although this aspect becomes less important with time, the fact that evaluating a single test point takes much time might make the model useless in practice.

2.4.2 Nearest Neighbor based anomaly detection techniques

The concept of nearest neighbor analysis has been used in several anomaly detection techniques. Such techniques are based on the key hypothesis that normal data instances occur in dense neighborhoods, while anomalies occur far from their closest neighbors. Nearest neighbor based anomaly detection techniques require a distance or similarity measure (or a metric) defined between two data instances and can be classified in the following categories:

- Using Distance to k-th Nearest Neighbor: In this case, the anomaly score of a data instance is defined as its distance to its k-th nearest neighbor in a given data set. This basic technique has been applied to detect land mines from satellite ground images [41] and to detect shorted turns (anomalies) in the DC field windings of large synchronous turbine-generators [94]. Usually, a threshold is then applied to the anomaly score to determine if a test instance is anomalous or not. On the other hand, we can use this criterium also to select n instances with the largest anomaly scores as the candidate anomalies [194]. The basic technique has been extended by researchers in three different ways:

– The first set of variants modify the above definition to obtain the anomaly score of a data instance. In [17], [75] and [258], the authors compute the anomaly score of a data instance as the sum of its distances from its k nearest neighbors. A different way to compute the anomaly score of a data instance is to count the number of nearest neighbors n that are not more than d distance apart from the given data instance [132][133][134][135].

– The second set of variants use different distance/similarity measures to handle different data types. A hyper-graph based technique is proposed by [246] called HOT. where the authors model the categorical values using a hyper-graph, and measure distance between two data instances by analyzing the connectivity of the graph. A distance measure for data containing a mix of categorical and continuous attributes has been proposed for anomaly detection [180] by adding distance for categorical and continuous attributes separately. In [182] the authors adapt the technique proposed in [134] to continuous sequences. The work in [136] extends the technique proposed in [194] to spatial data.

– The third set of variants focus on improving the efficiency of the basic technique (the complexity of the basic technique is $O(N^2)$, where N is the data size) in different ways. Some techniques prune the search space by either ignoring instances that cannot be anomalous or by focussing on instances that are most likely to be anomalous. Authors of [27] show that for a sufficiently randomized data, a simple pruning step could result in the average complexity of the nearest neighbor search to be nearly linear. In [194] a partition based technique is proposed, which first clusters the instances and computes lower and upper bounds on distance of a instance from its k-th nearest neighbor for all instances in each partition. This information is then used to identify the partitions that cannot possibly contain the top k anomalies; such partitions are pruned. Anomalies are then computed from the remaining instances (belonging to unpruned partitions) in a final phase. Similar cluster based prunings have been proposed by [75] [166] [233]. To prune the search space for nearest neighbors, several techniques partition the attribute space into a hyper-grid consisting of hypercubes of fixed sizes. The intuition behind such techniques is that if a hypercube contains many instances, such instances are likely to be normal. Moreover, if for a given instance, the hypercube that contains the instance and its adjoining hypercubes contain very few instances, the given instance is likely to be anomalous. Techniques based on this intuition have been proposed by [133]. Angiulli and Pizzuti [17]

extended these models by linearizing the search space through the Hilbert space filling curve.

- Using Relative Density: these algorithms estimate the density of the neighborhood of each data instance. An instance that lies in a neighborhood with low density is declared to be anomalous while an instance that lies in a dense neighborhood is declared to be normal. For a given data instance, the distance to its k-th nearest neighbor is equivalent to the radius of a hyper-sphere, centered at the given data instance, which contains k other instances. Thus the distance to the kth nearest neighbor for a given data instance can be viewed as an estimate of the inverse of the density of the instance in the data set. Density based techniques perform poorly if data has regions of varying densities. To handle the issue of varying densities in the data set, several techniques that compute the density of instances in relation to the density of their neighbors have been proposed. The work in [36][37] assigns an anomaly score to a given data instance, known as Local Outlier Factor (LOF). For any given data instance, the LOF score is equal to the ratio of average local density of the k nearest neighbors of the instance and the local density of the data instance itself. To find the local density for a data instance, the authors first find the radius of the smallest hyper-sphere centered at the data instance, that contains its k nearest neighbors. The local density is then computed by dividing k by the volume of this hyper-sphere. For a normal instance lying in a dense region, its local density will be similar to that of its neighbors, while for an anomalous instance, its local density will be lower than that of its nearest neighbors. Hence the anomalous instance will get a higher LOF score. Tang et al. [232] discuss a variation of the LOF, which they call Connectivity-based Outlier Factor (COF). The difference between LOF and COF is the manner in which the k neighborhood for an instance is computed. In COF, the neighborhood for an instance is calculated in an incremental mode. To start, the closest instance to the given instance is added to the neighborhood set. The next instance added to the neighborhood set is such that its distance to the existing neighborhood set is minimum among all remaining data instances, defining the distance between an instance and a set of instances as the minimum distance between the given instance and any instance belonging to the given set. The neighborhood is grown in this manner until it reaches size k. Once the neighborhood is computed, the anomaly score (COF) is computed in the same manner as LOF. Other variations of these techniques have been proposed in recent years [38][54][97][119][183][188][225][226].

2.4.3 Clustering based anomaly detection techniques

Classical clustering [114][229] aims at grouping similar data instances into homogeneous groups. Clustering is primarily an unsupervised technique though semi-supervised clustering [26] has also been explored lately. Three categories can be detected in this area. The first one embraces those techniques which follow the assumption that normal data instances belong to a cluster in the data, while anomalies do not belong to any cluster. Thus techniques based on the above assumption apply a known clustering based algorithm to the data set and declare any data instance that does not belong to any cluster as anomalous. Several clustering algorithms that do not force every data instance to belong to a cluster, such as DBSCAN [77], ROCK [92], and SNN clustering [74] can be used. The FindOut algorithm [256] is an extension of the WaveCluster algorithm [215] in which the detected clusters are removed from the data and the residual instances are declared as anomalies. A disadvantage of such techniques is that they are not optimized to find anomalies, since the main aim of the underlying clustering algorithm is to find clusters.

The second category includes those techniques which rely on the assumption that normal data instances lie close to their closest cluster centroid, while anomalies are far away from their closest cluster centroid. Algorihtms based on the above assumption consist of two steps. In the first step, the data is divided in groups using a clustering algorithm. In the second step, for each data instance, its distance to its closest cluster centroid is calculated as its anomaly score. A number of anomaly detection techniques that follow this two step approach have been proposed using different clustering algorithms: Self-Organizing Maps (SOM), K-means Clustering, and Expectation Maximization (EM) to cluster training data and then use the clusters to classify test data [219]. Note that if the anomalies in the data form clusters by themselves, the above discussed techniques will not be able to detect such anomalies.

To address this issue a third category of clustering based techniques have been proposed that rely on the hypothesis that normal data instances belong to large and dense clusters, while anomalies either belong to small or sparse clusters. Techniques based on the above assumption declare instances belonging to clusters whose size and/or density is below a threshold as anomalous. Several variations of this third category of techniques have been proposed [75][102][118][159][179][187][189][225].

2.4.4 Statistical anomaly detection techniques

The underlying principle of any statistical anomaly detection technique is mainly the basic definition of anomaly that we stated overleaf: normal data instances occur in high probability regions of an underlying stochastic model, while anomalies occur in the low probability regions of the stochastic model. Statistical techniques fit a probability distribution to normal data and then apply inference to determine if an unseen instance belongs to this model or not. Instances that have a low probability to be generated from the learnt mode are considered as potential anomalies. Both parametric as well as non-parametric techniques have been applied to fit a statistical model. The main difference between both is that the latter do not generally assume any knowledge about the underlying distribution [61].

Parametric Techniques assume that the normal data is generated by a distribution $P(x, w)$, where x is an observation and w is the parameter vector. The parameters w need to be estimated from given data [218]. This is a key disadvantage of this kind of techniques because this parametric assumption does not usually hold true, and parameter estimation can be an issue for high dimensional data sets. Based on the type of distribution assumed, we can distinguish:

- Gaussian Model Based Techniques, which assume that the data is generated from a Gaussian distribution. The parameters are usually estimated using Maximum Likelihood Estimates (MLE). Sophisticated statistical tests have been used to detect potential anomalies, as discussed in [23][24][28].
- Regression Model Based Techniques, which have been extensively investigated for time-series data [2][3][84]. It consists of two steps. In the first step, a regression model is fitted on the data. In the second step, for each test instance, the residual for the test instance is used to determine the anomaly score. The residual is the part of the instance which is not explained by the regression model. The magnitude of the residual can be used as the anomaly score for the test instance, though statistical tests have been proposed to determine anomalies with certain confidence [18][28][98][241]. Certain techniques detect the presence of anomalies in a data set by analyzing the Akaike Information Content (AIC) during model fitting [131]. Presence of anomalies in the training data can influence the regression parameters and hence the regression model might not produce accurate results. A popular technique to handle such anomalies while fitting regression models is called robust regression [111][201]. Variants of the basic regression models based

technique have been proposed to handle multivariate time-series data. In [242] the authors discuss the additional complexity in multivariate time-series over the univariate time-series and come up with statistics that can be applied to detect anomalies in multivariate Autoregressive Integrated Moving Average (ARIMA) models. Another variant to detect anomalies in multivariate time-series data generated by an Autoregressive Moving Average (ARMA) model, was proposed by [88] in which the authors transform the multivariate time-series to univariate time-series and the anomaly detection in each projection is done by using univariate test statistics.

- Mixture Distribution Based Techniques. These techniques model the normal instances as a mixture of parametric distributions. A test instance which does not belong to any of the learnt models is declared to be anomaly. Gaussian mixture models have been mostly used for such techniques [5]. Similar techniques have been applied to detect anomalies in biomedical signal data [4][197][198], where extreme value statistics are used to determine if a test point is an anomaly with respect to the learnt mixture of models or not. In [41] a mixture of Poisson distributions is used to model the normal data and then detect anomalies.

Non-parametric Techniques use non-parametric statistical models - in which the model structure is not defined a prioiri but it is instead directly determined from given data. Such techniques typically make fewer assumptions regarding the data (if any) when compared to parametric techniques. They can be classified in:

- Histogram Based, it is the simplest non-parametric statistical technique and uses histograms to maintain a profile of the normal data. They are particularly popular in intrusion detection community [60][76][218] and fraud detection [81]. For univariate data two steps are considered. The first step involves building a histogram based on the different values taken by that feature in the training data. In the second step, the technique checks if a test instance falls in any one of the bins of the histogram. If it does, the test instance is normal, otherwise it is anomalous. A variant of the basic histogram based technique is to assign an anomaly score to each test instance based on the height (frequency) of the bin in which it falls. The size of the bin used when building the histogram is key for anomaly detection. If the bins are small, many normal test instances will fall in empty or rare bins, resulting in a high false alarm rate. If the bins are large, many anomalous test instances will fall in frequent bins, resulting in a high false negative rate.
  For multivariate data, a basic technique is to construct attribute-wise histograms.

During testing, for each test instance, the anomaly score for each attribute value of the test instance is calculated as the height of the bin that contains the attribute value. The per-attribute anomaly scores are aggregated to obtain an overall anomaly score for the test instance. Note that these techniques present a drawback since they not take into account the relations between the attributes.

- Kernel Function Based. In the non parametric group, the Parzen Density estimation [185] is the most extended and paradigmatic method. It assumes no distribution beforehand and extracts it directly from the data. If an appropriate kernel function and parameters are selected and a reasonable sample size is available, these methods can exhibit a good performance. However, a bad choice of both kernel function and parameters can introduce a large bias in the final model, and sample size requirements can grow exponentially with the dimensionality of the input patterns. Noisy samples can also degrade considerably the performance of these methods. In [61] a semi-supervised statistical technique to detect anomalies is proposed which uses kernel functions to estimate the probability distribution function (pdf) for the normal instances. A new instance which lies in the low probability area of this pdf is declared to be anomalous.

2.4.5 Spectral anomaly detection techniques

Spectral techniques try to find an approximation of the data using a combination of attributes that capture the bulk of variability in the data. The basic assumption is that data can be embedded into a lower dimensional subspace in which normal instances and anomalies appear significantly different. Thus the general approach adopted by spectral anomaly detection techniques is to determine such subspaces (embeddings, projections, etc.) in which the anomalous instances can be easily identified [8]. Using these models, it is possible to calculate a reconstruction of each sample disregarding unimportant features and capturing most of the contained information. These two facts are used for anomaly detection in the following manner: if the difference between a sample and its reconstructed counterpart exceeds a given threshold, this sample is considered a novel or abnormal one. These techniques can work in an unsupervised as well as semi-supervised setting. There are different methods that can be used in this way, such as: (a) k-means [30] in which the distance to the closest center is used as a distance metric, (b) Learning Vector Quantization (LVQ) [220] in which a lattice of centers is adjusted to the normal data following a competitive training rule, and (c) Self Organizing Maps (SOM) [220], in which the difference between a data sample and

its closest node of the lattice is used as the detection feature.
The following methods have a different philosophy and are more commonly applied to real life problems: (a) Principal Component Analysis (PCA) [122][244] in which the most representative principal components of the data set are selected in order to calculate the reconstruction of a new sample in the principal subspace and orthogonal reconstruction error is used to detect novelties, (b) Autoassociative Neural Networks (AARNA) [115] in which a single hidden layer neural network with less units in the hidden layer than the input dimensionality is used to model data, and the error in the output of the network is used as the distance to the true distribution (this method can be demonstrated to be equivalent to the PCA [32] if only a single hidden layer is used), (c) Kernel Principal Component Analysis (KPCA) [31] which is based on the same principle as the Principal Component approach but acomplishes it in feature space thanks to the application of the kernel trick [109] and (d) Diabolo Networks [137][248] which add to the AARNA more hidden layers to obtain non linear reconstruction subspaces (this method can be regarded as equivalent to the KPCA method [137]). Although reconstruction methods are very practical, they can suffer considerably in the presence of noise.
Several variant of this seminal approaches have been proposed. One such technique [184] analyzes the projection of each data instance along the principal components with low variance. A normal instance that satisfies the correlation structure of the data will have a low value for such projections while an anomalous instance that deviates from the correlation structure will have a large value. In [66] the authors adopt this approach to detect anomalies in astronomy catalogs. In [217] an anomaly detection technique is presented where the authors perform robust PCA [110] to estimate the principal components from the covariance matrix of the normal training data.

2.4.6 Information theoretic anomaly detection techniques

Information theoretic techniques analyze the information content of a data set using different information theoretic measures such as Kolmogorov Complexity, entropy, relative entropy, etc. Such techniques are based on the assumption that anomalies induce irregularities in the information content of the data set.
Let $C(D)$ denote the complexity of a given data set, D. A basic information theoretic technique can be described as follows: given a data set D, find the minimal subset of instances, I, such that $C(D) - C(D - I)$ is maximum. All instances in the subset thus obtained are considered as anomalous. The problem addressed by this basic technique

is to find a Pareto-optimal solution, which does not have a single optimum, since there are two different objectives that need to be optimized.
The basic technique described above involves dual optimization to minimize the subset size while maximizing the reduction in the complexity of the data set. Thus an exhaustive approach in which every possible subset of the data set is considered would run in exponential time. Several techniques have been proposed that perform approximate search for the most anomalous subset. He et al. use an approximate algorithm called Local Search Algorithm (LSA) [103] to approximately determine such a subset in a linear fashion, using entropy as the complexity measure. A similar technique that uses an information bottleneck measure was proposed by [16].
Information theoretic techniques can also be used in data sets in which data instances are naturally ordered (e.g. , sequential data and spatial data). In these cases, the data is broken into substructures, and the anomaly detection technique finds the substructure, I, such that $C(D) - C(D - I)$ is maximum. This technique has been applied to sequences [19][52][148], graph data [177], and spatial data [149]. A key challenge of such techniques is to find the optimal size of the substructure which would result in detecting anomalies. The complexity of a data set C can be measured in different ways. Kolmogorov complexity [147] has been used by several techniques [19][128]. The work in [19] uses the size of the regular expression to measure the Kolmogorov Complexity of data (represented as a string) for anomaly detection. On the other hand, authors in [128] use the size of the compressed data file (employing any standard compression algorithm), as a measure of the data set's Kolmogorov Complexity. Other information theoretic measures such as entropy, relative uncertainty, etc., have also been used to measure the complexity of a categorical data set [16][101][103][144].
The performance of such techniques is highly dependent on the choice of the information theoretic measure. Often, such measures can detect the presence of anomalies only when there is a significantly large number of anomalies present in the data. It is also difficult to obtain an anomaly score for a specific test instance using an information theoretic technique.

2.5 Stream anomaly detection

The techniques mentioned in the taxonomy of previous section do not take into account the specific case when a continuous stream of data has to be analyzed in a on-line manner. Stream anomaly detection can be stated as the problem of accurately determining when the process that generates the stream of data has severely changed. This type of

scenario appears in cases such as robotics or web traffic pattern analysis, where batch data analysis is not possible. Usually, the nature of the anomaly is a temporal or permanent change in the source which is generating the data stream and being monitored. In the former case (temporal change), tracking of the subsequent changes is desirable whilst in the latter (permanent change), once the detection has been performed the process ends. Generally, the detection is intended to trigger some counteraction in order to avoid an undesirable effect or to recover a normal functioning of the underlying process. Formally, this problem can be stated as follows [12]:

Definition[On-line Anomaly detection]: An anomaly α_i that starts at instant τ_i and lasts for l_i intervals is represented as $\alpha_i = (\tau_i, l_i)$. Using data obtained so far, on-line stream anomaly detection is to output a signal at interval d_i for every α_i such that $\tau_i \leq d_i \leq \tau_i + l_i$ and $(d_j - \tau_i)$ is minimal.

This kind of task has practical applications in many areas such as fault detection in rotating machinery [79][125][163][170][253], credit fraud detection, intrusion detection, medical anomaly detection, etc. [53], and it is a challenging task due to the following aspects:

- The method has to face the fact that anomalous events may appear rarely and do not have fixed signature. This leads to the necessity of eliciting a detection rule only from normal data.
- The method should be able to adapt to concept drifts when these are not severe in order to avoid false alarms.
- The method should be able to adapt to complex decision boundaries in order to accurately capture the normal support of complex data streams.

Some previous works in stream anomaly detection can be found in the literature. There is a group of algorithms based on principal subspace tracking [63][140] which try to detect deviations from the principal subspace of normal data as a sign of being an anomaly. These techniques can find limitations when dealing with non linear data, thus in [12] this philosophy is applied in feature space through an adaptation of Kernel Recursive Least Squares. In [43] an adaptation for non-stationary data of One-class SVM is introduced. This algorithm obtains good representations but suffers a computa-

tional complexity burden when facing large high dimensional datasets. In [251], a fully probabilistic model for stream anomaly detection is presented. This method obtains also good results but assumes a predefined distribution of data. Other more recent techniques are based on classification trees [230] and clustering [71]. These techniques are adaptations of more classical techniques and need to maintain batches of data in order to update the model and detect abnormalities, which could lead to storage and detection delay issues.

A straightforward manner of tackling this problem is to: (a) continuously capture the support of the probability distribution of the data stream and (b) detecting changes on it. Solving the problem in this way is closely related to concept drift techniques [67][72][172][247]. The first task has been tackled in the literature through one-class classifiers. However, one-class classification has traditionally been addressed from a batch perspective assuming that the whole dataset is available for the training (i.e., it is complete and it can be stored in memory). It turns out that, when facing streaming data, these assumptions are not fulfilled and effective solutions which treat each pattern in a one pass manner are needed. Very few on-line techniques specially designed for on line anomaly detection have been devised. Besides, there is not consensus on the categorization of the types of changes which take place and should be detected. Recently, a categorization of the type of changes has been proposed [173] and will be used hereunder in this work.

CHAPTER 3

Anomaly detection: proposals and results

In this chapter we present the contributions of this thesis to the field of anomaly detection. Three novel anomaly detection algorithms are proposed:

- The first algorithm is an extension of the classic problem that aims at finding the minimum volume ellipsoid that covers a given set of data points to the multiple ellipsoids case. This extension allows for more complex covering of normal data and, in addition, allows to avoid spurious data which can degrade significantly the solution. This algorithm should be included in the One Class Classifiers group described in the previous chapter.

- The second technique is based on nearest neighbor principles. Using concepts from Extreme Value Statistics, the method is able to accurately build a normal state region based only on a distance measure among data patterns. This technique can be useful when dealing with complex input data scenarios where we can count only on distance values. This algorithm should be included in the nearest neighbor group described in the previous chapter.

- The last technique tackles challenging stream anomaly detection problem. In this setting, the data set arrives continuously as a stream and a one pass treatment of the data set is needed. Adopting a passive-agressive perspective [58], an anomaly detection algorithm able to continuously detect changes of context and adapt to them is built. This algorithm should be included in the stream anomaly detection techniques group.

The performance of the three devised algorithms has been tested on benchmark datasets. The application of these techniques to real industrial problems, specifically to machinery fault detection will be detailed in next chapters. As we will see in chapter 5, the most sensible way of treating automatic predictive maintenance of machinery is through

an anomaly detection perspective, due to the lack of fault data to build multiple class classification models beforehand.

3.1 A minimum volume covering approach with a set of ellipsoids

The minimum volume covering ellipsoid (MVCE) problem has been studied since John [120] discussed it for the first time in his work on optimality conditions. Once this problem is solved, an anomaly detection method can be easily derived, using the obtained ellipsoid to decide which data are anomalous (those which are not covered by the ellipsoid). The problem consist on covering a set of points $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_m\} \in \mathbb{R}^n$ with an ellipsoid of minimum volume. The problem can be found in different formulations, each one presenting different properties [34][181]. In its most simple formulation we define an ellipsoid $E \subset \mathbb{R}^n$ as

$$E = \{\mathbf{x} \in \mathbb{R}^n | (\mathbf{x} - \mathbf{a})^T \mathbf{M} (\mathbf{x} - \mathbf{a}) \leq 1\} \tag{3.1}$$

where $\mathbf{a}$ is the center of the ellipsoid and matrix $\mathbf{M}$ determines its shape. In other words, we define the ellipsoid as the set of point which have a restricted Mahalanobis distance with matrix $\mathbf{M}$ from center a. Given this representation, the volume of E is given by the formula

$$\frac{\pi^{n/2}}{\Gamma(n/2+1)} \frac{1}{\sqrt{det(M)}} \tag{3.2}$$

where Γ is the gamma function and the mathematical formulation of the problem is as follows

$$\underset{\mathbf{a}, \mathbf{M}}{\text{Minimize}} \quad det(\mathbf{M})^{-1} \tag{3.3}$$

subject to

$$(\mathbf{x}_i - \mathbf{a})^T \mathbf{M} (\mathbf{x}_i - \mathbf{a}) \leq 1 \tag{3.4}$$

$$\mathbf{M} \succ 0 \tag{3.5}$$

where $\mathbf{x}_i \in X$. It can be noted that the objective function minimizes a quantity proportional to the volume of the ellipsoid and restrictions assure that the data samples are covered by it. Given its applicability in areas such as statistics and data mining, several algorithms for solving it have been developed in the past decades. In [22] the authors provided an algorithm based on matrix eigenvalue decomposition. Posteriorly Khachiyan and Todd [130] first used interior-point methods in developing an algorithm for this purpose. This seminal work is the root of recent developments [138][240].
From a theoretical point of view, several authors obtained bounds for the complexity of the problem. Nesterov and Nemirovskii [175] obtained a complexity bound of

$O(m^{3.5}\log(mR/\epsilon))$ operations for a ϵ-optimal ellipsoid where m is the number of points and R is defined such that the convex hull of the given points contains the unit ball centered at 0 and is contained in the concentric ball of a given radius R. More recently, Khachiyan [129] reduced this to $O(m^{3.5}\log(m/\epsilon))$ operations.
There is a classical and well-known result of John [120] which states that the number of boundary points is not too large: *The minimum-volume covering ellipsoid is determined by a subset of at most* $O(n^2 + 3n/2)$ *points.* This has motivated the design of active-set strategies for solving the problem such as the one in [181], wherein they try to make an intelligent guess of active points $\mathbf{x}_i$ at each iteration, and presumably inactive points are discarded from time to time.
Recent developments have shown that this problem can be formulated as a Semidefinite Programming Problem (SDP) which now can be solved efficiently with standard software [33][34]. It also can be considered as an instance of the more general problem of log-determinant maximization (minimization) for which several solving methods can be found [245].
In this section we aim at extending the well-studied MVCE problem. Specifically, we present an algorithm able to determine a set of ellipsoids that cover totally or partially a set of points, have minimum total volume and eventually ignore possible outliers. This problem can be formally stated as follows: find a set of centers $\mathbf{a}_e$ and matrices $\mathbf{M}_e$ for $e = 1, \ldots, |E|$ such that the following conditions are fulfilled:

1. For every data point $\mathbf{x}_j \in X$

$$(\mathbf{x}_j - \mathbf{a}_i)^T \mathbf{M}_i^{-1} (\mathbf{x}_j - \mathbf{a}_i) \leq 1, \tag{3.6}$$

   for some $i \in \{1, 2, \ldots, |E|\}$.

2. The total volume of the ellipsoids defined by $\mathbf{a}_e$ and $\mathbf{M}_e$ for $e = 1, \ldots, |E|$ is minimum.

As it can be observed, we will use an alternate formulation where $\mathbf{M}_e^{-1}$ is used in the restrictions instead of $\mathbf{M}_e$ and the objective function is changed accordingly.
In order to tackle real life situations, where noise in the data set is present, it is desirable to extend this formulation towards one that allows some data samples to fall outside the region defined by the set of ellipsoids and even ignore some potential outliers that can appear in the data set. These two issues are also covered by the proposed model.

The solution of this problem leads to a model that can be applied both in clustering applications and one class classification problems (OCC). In Section 3.1.1 the formula-

tion of the single minimum volume covering ellipsoid problem and the main challenges covered in this work are described. Moreover, the original contributions of this section are listed. In Section 3.1.2 we present the proposed method by incorporating one by one the new features and providing two theorems that justify it. In Section 3.1.3 we discuss the problem of sensitivity analysis and derive a classification rule. In Section 3.1.4 examples of covering artificial datasets and clustering problems are given. Finally, in Section 3.1.6 we summarize the main contributions and discuss future work.

3.1.1 Single minimum volume covering ellipsoid: formulation and improvements

If we are to deal with ellipsoids which are not centered at the origin and have arbitrary orientation, the boundary of one of such ellipsoids can be described as

$$(\mathbf{x}-\mathbf{a})^T\mathbf{M}^{-1}(\mathbf{x}-\mathbf{a})=1, \tag{3.7}$$

where $\mathbf{x}$ is the vector of coordinates of a point on the ellipsoid surface, $\mathbf{a}$ is the column vector with the center coordinates of the ellipsoid and $\mathbf{M}$ should be a positive definite symmetric matrix (this is an implicit constraint). In this case, the volume of the ellipsoid is directly proportional to the determinant of matrix $\mathbf{M}$. In order to deal with abnormal data points in real data sets a penalty term in the formulation which allows for some data points to lie outside the ellipsoid is added. This approach can be found in other works such as [216]. Joining all, the formulation of the problem of Single Minimum Volume Covering Ellipsoid with Direct Determinant (SCEDD) is:

$$\underset{\boldsymbol{\xi},\mathbf{a},\mathbf{M}}{\text{Minimize}} \quad det(\mathbf{M})+C\sum_i \xi_i \tag{3.8}$$

subject to

$$(\mathbf{x}_i-\mathbf{a})^T\mathbf{M}^{-1}(\mathbf{x}_i-\mathbf{a}) \;\leq\; 1+\xi_i \;:\; \alpha_i \tag{3.9}$$

$$\xi_i \;\geq\; 0 \;:\; \gamma_i, \tag{3.10}$$

where C is a constant with hypervolume dimensions which weights the error of not covered samples, ξ_i are non-negative dimensionless variables measuring the distance of the associated point i to the ellipsoid, $\mathbf{x}_i$ is the vector of coordinates of data point i, $\mathbf{a}$ is the vector of coordinates of the center of the ellipsoid, $\mathbf{M}$ is positive definite and α_i and γ_i are the associated dual variables.

The optimization problem (3.8)-(3.10) presents the following weaknesses:

1. The determinant appearing in the objective function (3.8) is difficult to deal with, even though some authors, such as [34][62] provide a solution. A method permitting an indirect and simple evaluation of this determinant would be very convenient.

2. Only one ellipsoid is considered to cover the desired region. It would be convenient to provide methods that include a set of several ellipsoids in order to obtain a better covering of the desired data set.

3. No data points can be excluded from the analysis. In practical cases, even after including the slack variables ξ_i, outliers that distort the solution of the problem can still be present. Thus, the possibility of detecting these outliers and excluding them from the analysis would be desirable.

4. The exclusion of singular points and the consideration of a set of ellipsoids involve the use of binary variables which are associated with a high computational cost. Providing a method that avoids the use of these binary variables could reduce substantially the time of computation.

5. Combining the ellipsoid identification, the outlier detection and the classification problems in a single optimization problem can lead to high memory and CPU time requirements. Thus, bilevel and multilevel methods are convenient ways to reduce these requirements that can be a burden in large size applications.

In next section SCEDD will be extended in order to tackle these limitations and to obtain a more accurate and general model.

3.1.2 Proposed model

In the following sections we describe how to modify the SCEDD problem (equations (3.8)-(3.10)) to incorporate the above five mentioned improvements.

3.1.2.1 Dealing with the determinant problem

To avoid the direct evaluation of the determinant, we use a method that is justified in the following theorem.

Theorem 3.1.1 (Single Covering Ellipsoid Indirect Determinant): The initial SCEDD problem (3.8)-(3.10) is equivalent to the following one:

$$\begin{array}{ll} \text{Minimize} \\ \boldsymbol{\xi}, \mathbf{d}, \mathbf{p}, \mathbf{a} \end{array} \quad \prod_j d_j + C \sum_i \xi_i \tag{3.11}$$

subject to

$$A_i \leq 1 + \xi_i \ : \ \alpha_i \tag{3.12}$$

$$\xi_i \geq 0 \ : \ \gamma_i \tag{3.13}$$

$$d_j > 0 \ : \ \chi_j \tag{3.14}$$

where

$$A_i = \sum_j (x_{ij} - a_j) \sum_{k \leq j} \frac{p_{jk}}{d_k} \sum_{\ell \geq k} p_{\ell k} (x_{i\ell} - a_\ell). \tag{3.15}$$

d_j are positive real numbers, x_{ij} is the jth coordinate component of point i, a_j is the j component of the center of the ellipsoid and p_{ij} is the element ij of the dimensionless unit lower triangular matrix $\mathbf{P}$ that diagonalizes matrix $\mathbf{M}$, α_i, γ_i and χ_j are dual variables. This problem will be named from now on Single Covering Ellipsoid Indirect Determinant (SCEID).

Proof: It is well known that any symmetric positive definite matrix $\mathbf{M}$ can be diagonalized by means of a lower unit triangular matrix $\mathbf{P}$, that is, we can write

$$\mathbf{M} = \mathbf{P}\mathbf{D}\mathbf{P}^T, \tag{3.16}$$

where $\mathbf{D}$ is a diagonal matrix (see [96], corollary 14.5.10).

From Equation (3.16) we have that

$$|\mathbf{M}| = |\mathbf{P}||\mathbf{D}||\mathbf{P^T}| = |\mathbf{D}| = \prod_j d_j, \tag{3.17}$$

where $|\cdot|$ refers to the determinant, and d_j is the j-th element of the diagonal matrix $\mathbf{D}$. Replacing the determinant in (3.8) by the one in (3.17), using variables p_{ij} (for $i > j$) and d_k to represent the elements of the matrices, $\mathbf{P}$ and $\mathbf{D}$, respectively, and adding the lower unit triangular constraint we get the SCEID problem (3.11)-(3.14).

Observe that although equation (3.12) has been written including the diagonal terms of matrix $\mathbf{P}$ in order not to make the notation cumbersome, we only need to consider and optimize its elements under the diagonal, as the diagonal is fixed to $\mathbf{1}$.

In the original SCEDD problem, described in (3.8)-(3.10), it is implicitly required that matrix $\mathbf{M}$ is positive definite. However, in our modified problem this is solved by imposing the positivity of the d_j values.

Theorem 3.1.1 provides an elegant form of dealing with the determinant. It employs a minimum number of variables thanks to the use of one unit lower triangular matrix and facilitates the work with positive definite matrices since this restriction reduces to impose positivity on the d_j variables. SCEID problem (3.11)-(3.14) is a convex problem because the objective function is convex in the first quadrant and the constraints include positive definite matrices.

3.1.2.2 Considering a set of covering ellipsoids

To deal with a fixed number of ellipsoids we need a binary variable η_i^e able to identify the ellipsoid associated with each data point. This variable η_i^e takes value 1 if point i belongs to ellipsoid e and 0, otherwise. However, we must guarantee that each point is associated with one and only one ellipsoid (see Equation (3.20)).

Thus, proposed problem (MCEID) (Multiple Covering Ellipsoids Indirect Determinant) can be stated as

$$\underset{\boldsymbol{\xi},\mathbf{d},\mathbf{p},\mathbf{a},\boldsymbol{\eta}}{\text{Minimize}} \quad Z = \sum_e \left(\prod_j d_j^e \right) + C \sum_i \xi_i \tag{3.18}$$

subject to

$$\sum_e \eta_i^e A_i^e \leq 1 + \xi_i : \alpha_i \tag{3.19}$$

$$\sum_e \eta_i^e = 1 : \lambda_i \tag{3.20}$$

$$\xi_i \geq 0 : \gamma_i, \tag{3.21}$$

$$d_j^e > 0 : \chi_j^e \tag{3.22}$$

$$\eta_i^e \in \{0,1\}, \tag{3.23}$$

where $\alpha_i, \lambda_i, \gamma_i, \chi_j^e; i = 1, 2, \ldots, n$ are dual variables and

$$A_i^e = \sum_j (x_{ij} - a_j^e) \sum_{k \leq j} \frac{p_{jk}^e}{d_k^e} \sum_{\ell \geq k} p_{\ell k}^e (x_{i\ell} - a_\ell^e). \tag{3.24}$$

Note that the η_i^e variables allow testing if data point i is inside the adequate ellipsoid by means of Equation (3.19). The ξ variables allow some points to be outside the ellipsoids but they are penalized in the objective function (3.18). Finally, condition (3.22) guarantees positive definiteness.

3.1.2.3 Removing outliers

In order to allow for removing outliers, we incorporate a binary variable u_i that takes value zero if the data point i is an outlier, and value one, otherwise. With this aim we modify the MCEID problem (3.18)-(3.23) to our final problem (RMCEID) (Robust Multiple Covering Ellipsoids Indirect Determinant):

$$\underset{\boldsymbol{\xi},\mathbf{d},\mathbf{p},\mathbf{a},\boldsymbol{u},\boldsymbol{\eta}}{\text{Minimize}} \quad Z = \sum_e \left(\prod_j d_j^e \right) + C \sum_i u_i \xi_i \tag{3.25}$$

subject to

$$\sum_e \eta_i^e A_i^e \leq 1 + \xi_i : \alpha_i \tag{3.26}$$

$$\sum_i u_i = n - n_{out} : \rho \tag{3.27}$$

$$\sum_e \eta_i^e = 1 : \lambda_i \tag{3.28}$$

$$\xi_i \geq 0 : \gamma_i, \tag{3.29}$$

$$d_j^e > 0 : \chi_j^e \tag{3.30}$$

$$\eta_i^e \in \{0,1\}, \tag{3.31}$$

$$u_i \in \{0,1\}, \tag{3.32}$$

where n is the number of data points, n_{out} is the number of outliers to be removed from the analysis, and $\alpha_i, \rho, \lambda_i, \gamma_i$ and χ_i^e are dual variables.

Note that with respect to other existing approaches in the literature, this problem allows to remove a given number n_{out} of outliers. This allows us to reduce substantially the total volume of the ellipsoids and provides a robust solution, because the dependence of the solution on some possible strange points is eliminated. If we are interested in the number of outliers to be chosen by the program, we can add a penalizing function to the objective function. However, decide whether or not a point is an outlier is a delicate decision that must be controlled by the user, perhaps with some additional

information (for example, revising the lab tests or data for some singularities). Thus, n_{out} is left as an input parameter determined by the user.

3.1.2.4 Avoiding binary variables

The RMCEID problem (3.25)-(3.32) can be relaxed by considering both the u_i and the η_i^e as continuous variables in the range $[0, 1]$, leading to the relaxed problem (RR-MCEID) (Relaxed Robust Multiple Covering Ellipsoids Indirect Determinant)

$$\underset{\boldsymbol{\xi}, \mathbf{d}, \mathbf{p}, \mathbf{a}, \boldsymbol{u}, \boldsymbol{\eta}}{\text{Minimize}} \quad \sum_e \left(\prod_j d_j^e \right) + C \sum_i u_i \xi_i \tag{3.33}$$

subject to

$$\sum_e \eta_i^e A_i^e \leq 1 + \xi_i : \alpha_i \tag{3.34}$$

$$\sum_i u_i = n - n_{out} : \rho \tag{3.35}$$

$$\sum_e \eta_i^e = 1 : \lambda_i \tag{3.36}$$

$$\xi_i \geq 0 : \gamma_i, \tag{3.37}$$

$$d_j^e > 0 : \chi_j^e \tag{3.38}$$

$$\eta_i^e \geq 0 : \epsilon_i^e, \tag{3.39}$$

$$\eta_i^e \leq 1 : \phi_i^e, \tag{3.40}$$

$$u_i \geq 0 : \vartheta_i, \tag{3.41}$$

$$u_i \leq 1 : \nu_i, \tag{3.42}$$

where $\epsilon_i^e, \phi_i^e, \vartheta_i$ and ν_i are added dual variables.

This means that the relaxed problem has an optimal value smaller or equal to the original problem. We will show below that in fact both problems share the same optimal value. This result has important practical implications, because if we do not need to deal with binary variables, that implies a substantial reduction in computational time. Note that we have one u-binary variable per point and one η_i^e-binary variable per point and ellipsoid. Thus, avoiding the η_i^e-binary variables is even more important than avoiding the u-binary variables.

The RMCEID problem (3.25)-(3.32), as the RR-MCEID problem (3.33)-(3.42) is a non-linear non-convex problem, so that we can expect only local optimal solutions.

However, if a good initial solution is provided (for example, by means of the k-means method), the resulting optimal seems to be satisfactory from a practical point of view. The RR-MCEID problem (3.33)-(3.42) does not require a special software, so that standard software, such as GAMS [49] can be used. In addition, also due to the not convex and combinatorial nature (due to the fact that points can be assigned to a set of several ellipsoids and several options are possible for outliers) of RR-MCEID problem (3.33)-(3.42) slow convergence is expected. A comparison of the CPU times required by different methods will be detailed in Section 3.1.4. In the following paragraphs, some rules to assign points to ellipsoids and to decide which points are the outliers are obtained from the Karush-Kuhn-Tucker (KKT) conditions. In addition, an efficient algorithm to find good solutions is provided.

3.1.2.5 Karush-Kuhn-Tucker conditions

With the aim of discovering how to avoid binary variables and to design our bilevel algorithm as an alternative to the direct solution of this problem, we analyze in this section the Karush-Kuhn-Tucker conditions of the relaxed primal RR-MCEID problem (3.33)-(3.42).

The Lagrangean function of RR-MCEID problem (3.33)-(3.42), $\mathcal{L}(\cdot) = \mathcal{L}(\boldsymbol{\xi}, \mathbf{d}, \mathbf{p}, \mathbf{a}, \mathbf{u}, \boldsymbol{\eta}; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \rho, \boldsymbol{\lambda}, \boldsymbol{\epsilon}, \boldsymbol{\phi}, \boldsymbol{\vartheta}, \boldsymbol{\nu})$, is

$$\begin{aligned}
\mathcal{L}(\cdot) \;=\; & \sum_e \left(\prod_j d_j^e \right) + C \sum_{i \in I} u_i \xi_i \\
& + \sum_i \alpha_i \left\{ \sum_e \eta_i^e A_i^e - 1 - \xi_i \right\} \\
& - \sum_i \gamma_i \xi_i - \sum_e \sum_j \chi_j^e d_j^e + \rho \left(\sum_i u_i - n + n_{out} \right) \\
& + \sum_i \lambda_i \left(\sum_e \eta_i^e - 1) \right) - \sum_{i,e} \epsilon_i^e \eta_i^e \\
& + \sum_{i,e} \phi_i^e (\eta_i^e - 1) - \sum_i \vartheta_i u_i + \sum_i \nu_i (u_i - 1)
\end{aligned} \tag{3.43}$$

and the corresponding Karush-Kuhn-Tucker conditions are:

$$
\begin{aligned}
Cu_i - \alpha_i - \gamma_i &= 0 && (3.44)\\
\prod_{j\neq k} d_j^e - \chi_k^e - \sum_i \alpha_i \sum_e \eta_i^e B_{ik}^e &= 0 && (3.45)\\
2\sum_i \alpha_i \eta_i^e \left\{ \frac{(x_{iu} - a_u^e)}{d_v^e} \sum_{\ell \geq v} p_{\ell v}^e (x_{i\ell} - a_\ell^e) \right\} &= 0 && (3.46)\\
\sum_i \alpha_i \eta_i^e \sum_{k \leq s} \frac{p_{sk}^e}{d_k^e} \sum_{\ell \geq k} p_{\ell k}^e (x_{i\ell} - a_\ell^e) &= 0 && (3.47)\\
C\xi_i + \rho - \vartheta_i + \nu_i &= 0 && (3.48)\\
\alpha_i A_i^e + \lambda_i - \epsilon_i^e + \phi_i^e &= 0 && (3.49)
\end{aligned}
$$

$$
\begin{aligned}
\sum_e \eta_i^e A_i^e &\leq 1 + \xi_i && (3.50)\\
\sum_i u_i &= n - n_{out} && (3.51)\\
\sum_e \eta_i^e &= 1 && (3.52)\\
\xi_i \geq 0, \quad \eta_i^e \geq 0, \quad \eta_i^e \leq 1, \quad u_i \geq 0, \quad u_i &\leq 1 && (3.53)
\end{aligned}
$$

$$
\begin{aligned}
\alpha_i \left\{ \sum_e \eta_i^e A_i^e - 1 - \xi_i \right\} &= 0 && (3.54)\\
\gamma_i \xi_i &= 0 && (3.55)\\
\epsilon_i^e \eta_i^e &= 0 && (3.56)\\
\phi_i^e (\eta_i^e - 1) &= 0 && (3.57)\\
\vartheta_i u_i &= 0 && (3.58)\\
\nu_i (u_i - 1) &= 0 && (3.59)\\
\alpha_i \geq 0, \ \gamma_i \geq 0, \ \epsilon_i^e &\geq 0 && (3.60)\\
\chi_i^e \geq 0, \ \phi_i^e \geq 0, \ \vartheta_i \geq 0, \ \nu_i &\geq 0 && (3.61)
\end{aligned}
$$

where

$$B_{ik}^{e} = \left\{ \sum_{j} (x_{ij} - a_j^e) p_{jk}^e / (d_k^e)^2 \sum_{\ell \geq k} p_{\ell k}^e (x_{i\ell} - a_\ell^e) \right\} \tag{3.62}$$

These conditions allow us to derive the following important theorem.

Theorem 3.1.2 (Binary and relaxed problems equivalence): The binary RMCEID problem described in equations (3.25)-(3.32) and its relaxed form RR-MCEID (3.33)-(3.42) share the same optimal value and the same solutions with the exception of the $\mathbf{u}$ values. However, the binary $\mathbf{u}$ values of the binary problem can be immediately obtained from the $\mathbf{u}$ values of the relaxed problem without changing the values of variables $\boldsymbol{\xi}, \mathbf{d}, \mathbf{p}, \mathbf{a}$ and $\boldsymbol{\eta}$.

Proof: The first part of the theorem concentrates on the u-values. From Equation (3.48) we get

$$C\xi_i + \rho - \vartheta_i + \nu_i = 0. \tag{3.63}$$

If $0 < u_i < 1$, from (3.58) and (3.59) we obtain $\vartheta_i = \nu_i = 0$, respectively, and from (3.63) we get $\xi_i = -\dfrac{\rho}{C}$, that is, this case is possible only when $\xi_i = -\dfrac{\rho}{C}$ share the same value for all i. This proves that

$$\xi_i \neq -\frac{\rho}{C} \quad \Rightarrow u_i = 0 \text{ or } u_i = 1. \tag{3.64}$$

If all the resulting values of u_i are zeros or ones, we have a binary solution and then the relaxed and the binary problems provide the same solution. Otherwise, all data points i with $0 < u_i < 1$ must share the same ξ_i value. In this case, we can reassign the u_i values to binary values by keeping its sum (see (3.35)) without changing the solution ellipsoids because we do not change ξ_i (see Equation (3.34)). Since this change in the u_i values does not alter the sum $\sum_i u_i \xi_i$ in the objective function (3.33), we obtain a feasible binary solution that provides the same value of the objective function. Since this can always be done, we have proved that the relaxed and the binary problems reach the same optimal value.

If we perform a similar analysis for the case of η_i^e, from Equation (3.49) we get[1]

$$\lambda_i = \epsilon_i^e - \phi_i^e - A_i^e. \tag{3.65}$$

and then we have:

If $0 < \eta_i^e < 1$, from (3.56) and (3.57) we obtain $\epsilon_i^e = 0$ and $\phi_i^e = 0$, respectively, and then from (3.65) we obtain $\lambda_i = -A_i^e$, that is, this case is possible only when $A_i^e = -\lambda_i$ share the same value for all e and each i. This proves that

$$A_i^e \neq -\lambda_i \quad \Rightarrow \eta_i^e = 0 \text{ or } \eta_i^e = 1 \tag{3.66}$$

If all the resulting values of η_i^e are zeros or ones, we have a binary solution and then the relaxed and the binary problems provide the same solution. Otherwise, all data points i with $0 < \eta_i^e < 1$ must share the same A_i^e value. In this case, we can reassign the η_i^e values to binary values by keeping its sum (see (3.36)) without changing the solution ellipsoids because we do not change $\sum_e \eta_i^e A_i^e$ (see Equation (3.34)). Since this change in the η_i^e values does not alter the the objective function (3.33) we obtain a feasible binary solution that provides the same value of the objective function. Since this can be always done, we have proved that the relaxed and the binary problems reach the same optimal value.

Corollary 3.1.1 (Ellipsoid and outliers assignment rules): The optimal solution of RR-MCEID problem (3.33)-(3.42) assigns points to ellipsoids using the following rule: Point i is assigned to the ellipsoid $e_i = arg\min_e A_i^e$. In addition, point i is selected as an outlier if its ξ_i is among the n_{out} largest values of ξ_i.

Proof: From the Karush-Kuhn-Tucker conditions above we have the following properties:

1. If $\eta_i^e = 0$ because of (3.57) then $\phi_i^e = 0$ and due to (3.65) we have

$$\epsilon_i^e = \lambda_i + A_i^e \geq 0 \quad \Leftrightarrow \quad A_i^e \geq -\lambda_i. \tag{3.67}$$

[1]Since α_i is common for all A_i^e it can be removed from the definition of A_i^e when assigning points to the ellipsoid.

2. If $\eta_i^e = 1$ because of (3.56) then $\epsilon_i^e = 0$ and due to (3.65) we have

$$\phi_i^e = -\lambda_i - A_i^e \geq 0 \quad \Leftrightarrow \quad A_i^e \leq -\lambda_i. \tag{3.68}$$

3. If $\epsilon_i^e > 0$ because of (3.56) then $\eta_i^e = 0$ and due to (3.57) $\phi_i^e = 0$. In this case, from (3.65) we get

$$A_i^e > -\lambda_i. \tag{3.69}$$

4. If $\phi_i^e > 0$ because of (3.57) then $\eta_i^e = 1$ and due to (3.56) $\epsilon_i^e = 0$. In this case, from (3.65) we get

$$A_i^e < -\lambda_i. \tag{3.70}$$

5. If $u_i = 0$ because of (3.59) then $\nu_i = 0$ and due to (3.63) we have

$$\vartheta_i = C\xi_i + \rho \geq 0 \quad \Leftrightarrow \quad \xi_i \geq -\frac{\rho}{C}. \tag{3.71}$$

6. If $u_i = 1$ because of (3.58) then $\vartheta_i = 0$ and due to (3.63) we have

$$\nu_i = -C\xi_i - \rho \quad \Leftrightarrow \quad \xi_i \leq -\frac{\rho}{C}. \tag{3.72}$$

7. If $\vartheta_i > 0$ because of (3.58) then $u_i = 0$ and due to (3.59) $\nu_i = 0$. In this case, from (3.63) we get

$$\vartheta_i = C\xi_i + \rho > 0 \quad \Leftrightarrow \quad \xi_i > -\frac{\rho}{C}. \tag{3.73}$$

   that implies $\xi_i > -\lambda$.

8. If $\nu_i > 0$ because of (3.59) then $u_i = 1$ and due to (3.58) $\vartheta_i = 0$. In this case, from (3.63) we get

$$\nu_i = -C\xi_i - \rho > 0 \quad \Leftrightarrow \quad \xi_i < -\frac{\rho}{C}. \tag{3.74}$$

   that implies $\xi_i < -\lambda$.

The first four properties imply the indicated point to ellipsoid assignation rule. Similarly, the last four properties imply the above outlier selection rule.

Finally, we clarify that ties can occur only by coincidence and then it is indifferent whether the point is assigned to any of the tied ellipsoids. The same is true for ties with outliers.

3.1.2.6 Proposed bilevel algorithm

The relaxed problem still incorporates the $\boldsymbol{\eta}$ and $\boldsymbol{u}$ complicating variables. Due to its combinatorial nature (it combines indices i and e), they are a problem from the point of view of memory requirements and computer time. However, the KKT conditions suggest how the points are assigned to the ellipsoids. Therefore, in addition, we can avoid the η_i^e and the u_i variables. Note that these variables decide to which ellipsoid each data point belongs to and which ones are outliers, so there are a huge number of combinations. This suggest solving the problem by using bilevel techniques. In the first level we fix the $\boldsymbol{\eta}$ and $\boldsymbol{u}$ variables and optimize the objective function with respect to the $\boldsymbol{\xi}, \mathbf{d}, \mathbf{p}, \mathbf{a}$ variables. In the second level, points are re-assigned to ellipsoids and avoided as outliers using collorary 1. More precisely, to assign the points to ellipsoids we evaluate A_i^e for all ellipsoids, and we assign a point in the ellipsoid e to the ellipsoid $e_1 = \arg\min_e(A_i^e)$ if $A_i^e > A_i^{e_1}$. On the other hand, we assign $u_i = 0$ for the n_{out} points with biggest ξ_i.

So, the resulting bilevel algorithm that we present is detailed in Algorithm 1. In the first level, ellipsoids are adjusted individually to its data subset. In the second level, data and outliers are reorganized following the aforementioned principles. The algorithm keeps executing the bi-level approach until it is not able to find a new better assignment of data points or until the new found optimum does not improve significantly the previous solution. The control variable *opt* is initialized to a large constant T in order to initiate the optimization process. This process is repeated twice for two different values of C. It is convenient to use a small value of C in a first loop and then use the desired value of C because small values of C provide initially more flexible ellipsoid choices.

Since the bilevel problem responds to the Karush-Kuhn-Tucker conditions, it provides the same solution as the original problem. Note that the bilevel structure allows us to reduce the memory requirements and reducing computer time by separating large size problems into smaller ones. In fact, the second level is a simple assignment of points to ellipses based on a simple rule obtained from the KKT conditions. Note also that the proposed algorithm divides the problem of minimum volume set of covering ellipsoids in a set of independent single minimum volume covering ellipsoid problems plus a second level re-assignment. This fact opens the possibility of (a) dividing the task in a multicore environment and (b) using any formulation available in the literature to solve the individual ellipsoids.

Algorithm 1: Solving minimum volume set of covering ellipsoids (MCSE algorithm).

- **INPUT:** The data points $\{\mathbf{x}_i; i = 1, 2, \ldots, m\}$, the number of ellipsoids E, the tolerance tol, the number n_{out} of outliers, and two values $C_{initial}$ and C_{final} of the constant C.
- **OUTPUT:** The values of $\boldsymbol{\xi}, \mathbf{d}, \mathbf{p}, \mathbf{a}, \boldsymbol{u}, \boldsymbol{\eta}$.

Initialize variables: $globalchange = true$, $opt = T$ (big const.), $lastopt = 3 * opt$, $C = C_{initial}$.

Initialize the $\boldsymbol{u}$ values: random.

Initialize the $\boldsymbol{\eta}$ values: use any clustering method (for example the k-means method) to determine the initial η_i^e values.

Use the following process to solve the problem.

loop(s=1 to 2),
  while ($globalchange$),
    if $((lastopt - opt)/opt < tol)$,
      exit and return $\boldsymbol{\xi}, \mathbf{d}, \mathbf{p}, \mathbf{a}, \boldsymbol{u}, \boldsymbol{\eta}$.
    else
      $lastopt = opt$
      $globalchange = false$
    end
    FIRST LEVEL
    SOLVE problem (3.33)-(3.42) for $\boldsymbol{\xi}, \mathbf{d}, \mathbf{p}, \mathbf{a}$ with fixed $\boldsymbol{\eta}$ and $\boldsymbol{u}$
    $opt = Z$ (objective function value (3.33))
    SECOND LEVEL
    loop ($i = 1$ *to* m),
      Evaluate $A_i^e;\ e = 1, 2, \ldots, E$.
      $e_1 = \arg min_e\ A_i^e$
      if$\left(A_i^{e1} \neq \sum_{e=1}^{E} \eta_i^e A_i^e\right)$,
        $\eta_i^{e1} = 1$
        $\eta_i^e = 0,\ \forall\ e \neq e_1$
        $globalchange = true$
      end
      if $(|\{\xi_j \geq \xi_i;\ j = 1\ to\ m\}| <= n_{out})$,
        $v_i = 0$
      else
        $v_i = 1$
      end
    end
    if $(\mathbf{u} \neq \mathbf{v})$,
      $\mathbf{u} = \mathbf{v}$
      $globalchange = true$
    end
  end
  $C = C_{final}$
end

3.1.3 Sensitivity analysis and classification rule

Sensitivity analysis is a very important technique (see [48, 51]) that permits evaluating how much the objective function, the primal or the dual variables change when some (small) changes in the data points are done. This can be used for many purposes, as for example detecting outliers or the most influential points.

In this section we perform a sensitivity analysis trying to identify which are the data points having the largest influence on the objective function. In other words, we evaluate the partial derivative of the objective function optimal value with respect to the data point coordinates.

According to [47, 50] we have that the sensitivity of the objective function with respect to a parameter is equal to the derivative of the Lagrangian with respect to that parameter. Thus, in our problem we have that:

$$\frac{\partial \mathcal{L}}{\partial x_{rs}} = 2\alpha_r \left[\sum_e \eta_i^e \left(\sum_{k \leq s} \frac{p_{sk}^e}{d_k^e} \sum_{\ell \geq k} p_{\ell k}^e (x_{r\ell} - a_\ell^e) \right) \right] \tag{3.75}$$

To evaluate the total sensitivity of the objective function Z with respect to data point r we can use the values of

$$S_r = \sqrt{\sum_s \left(\frac{\partial \mathcal{L}}{\partial x_{rs}} \right)^2}. \tag{3.76}$$

To illustrate this we have performed the analysis in the normal data example to be presented in the next section. Figure 3.1 shows the most influential data points accompanied by a number. The numbers correspond to the order of their influence, that is, we use 1 for the most influential point, 2 for the second, and so on.

When we have new data points available, we can classify then using the classification rule

$$\min_e \max\left(0, A_i^e - 1\right), \tag{3.77}$$

which corresponds to the associated ξ_i values. Ties can be broken by assigning to any of the tied ellipsoids. Figure 3.1 shows also the contours of this classification rule.

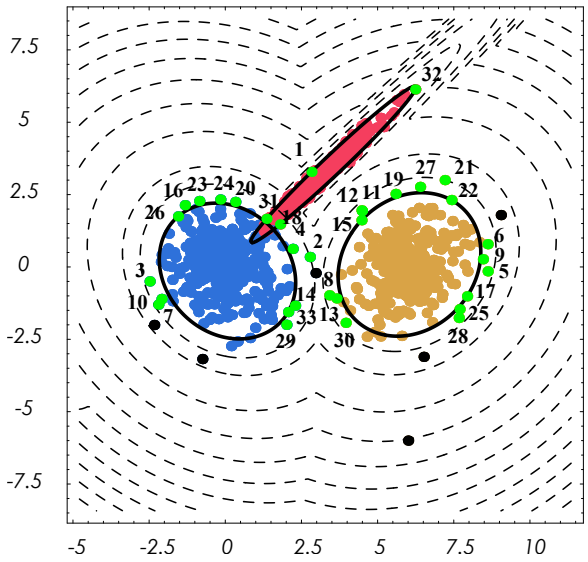


Figure 3.1: Graphic illustration of the classification rule and sensitivity analysis for the Normal data set example.

3.1.4 Experimental results

In this section we illustrate the proposed method by means of two scenarios: (a) artificial data sets, where it can be observed graphically the different properties of the proposed method and (b) its application to clustering problems. In the clustering applications section, the proposed model is compared to state of the art methods for each field of application. The method proposed in previous sections has been implemented in GAMS (see [49]).

3.1.5 Artificial Data sets

In order to analyze the behavior of the proposed method when the data shows some characteristic forms, we present below several illustrative examples.

3.1.5.1 Square example

The first is the case of data with a square ring pattern. In this case we have set the number of ellipsoids to $E = 8$. Figure 3.2 shows the resulting estimation process. We can see that the outliers (black dots in the graph) are correctly identified (and removed) from the initial stage and do not change until the end. The use of a small C value during the first step allows a more flexible estimation providing sufficient degrees of freedom to adapt to the square pattern. The increase of the C value during the second step forces the ellipsoids to include more points. The dashed ellipses provide the limits of a weak inclusion with the initial C, and the continuous line ellipses, the corresponding strong inclusion with the final C (with a higher value and thus, more restrictive). In this example, the specific values for the parameters are $C_{initial} = 0.01, C_{final} = 0.1$ and $n_{out} = 4$. The final result of the proposed method can be observed in subfigure 3.2(c). It can be observed that our bilevel approach provides the same optimal value than the binary optimization problem, which corresponds to subfigure 3.2(d), but with an improvement in computational time that will be explained at the end of this section.

3.1.5.2 Normal data

In this example we use simulated data of three bivariate normal distributions with centers $(0, 0)$, $(3, 3)$ and $(0, 6)$, respectively and different covariance matrices. These data are later rotated by different angles. In addition, we have added three outliers, one per group.

Our aim is to obtain the minimum volume ellipsoids to identify the three groups. We also want the outliers to be excluded from the analysis. We assume that we do not know the exact number of outliers so we provide a number of ellipsoids $E = 4$ value and a tentative value for n_{out} of 3. We obtain the results illustrated in Figure 3.3, where subfigures 3.3(a) and 3.3(b) show the fitted ellipsoids for the first step (for parameters $C_{initial} = 0.3, C_{final} = 3$). We have used a small value of C in order to allow for weak inclusion of some points. Finally, the full circles in black color correspond to the outliers, that is, the excluded points.

Subfigure 3.3(a) shows the initial assignment of points to the different ellipses. Subfigure 3.3(b) illustrates how the points are re-assigned by using the algorithm proposed in Section 3.1.2. Note the important improvement associated with these changes and

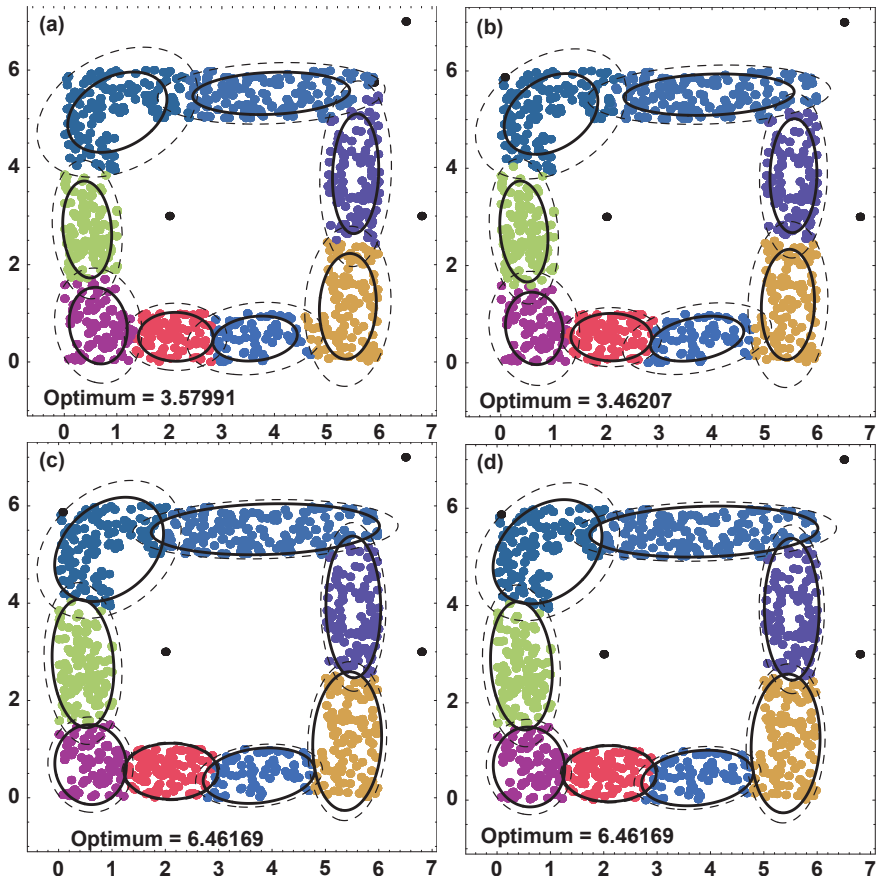


Figure 3.2: Evolution of the proposed algorithm for the Square data set: (a) initial step, (b) iteration 2, (c) final solution and (d) solution of the problem with binary variables (equations (3.25)-(3.32)).

how the outliers have changed during this process.

Subfigures 3.3(c) and 3.3(d) correspond to the second step, where the value of C has been assigned to the desired value. Note that the weak inclusion is more restrictive that the one in step one. The improvement in this step is much smaller that the one in the previous step. Subfigure 3.3(e) provides the the same local optimum as the one obtained by the binary variables program, as it can be seen by comparing both subfigures 3.3(e) and 3.3(f). We note that the subfigure 3.3(f) corresponds to the solution of the Problem (3.25)-(3.32), however this problem requires much more computation time.

In order to illustrate what happens when we overestimate the number of ellipses, we consider the case of 8 ellipses. Figure 3.4 shows the resulting process. It can be observed how the proposed model still captures the support of the data and detect outliers properly despite the overestimation of the number of necessary ellipsoids. It is interesting to see how the outliers change during the process until they stabilize in the final steps. Again, subfigure 3.4(f) corresponds to the solution of the binary problem (3.25)-(3.32).

3.1.5.3 Spiral example

Finally, we analyze the case of a highly nonlinear problem as is the spiral trend data problem. Figure 3.5 shows the data set together with the different solutions obtained in the different steps of the process (for parameters $C_{initial} = 0.01, C_{final} = 0.03$). The sequence of figures follows the same pattern than in the Normal case. It can be observed how the proposed model captures perfectly the shape of the data avoiding the use of clear outlier data points.

A comparison of the CPU times required by the different methods for the three examples in this section is given in Table 3.1, which shows how the relaxed Problem (3.33)-(3.42) requires less CPU than the initial binary Problem (3.25)-(3.32), and that the proposed algorithm allows an important improvement. This justifies the practical relevance of Theorem 3.1.2 and the ellipsoid assignment rule in equation (3.77) used in Algorithm 1.

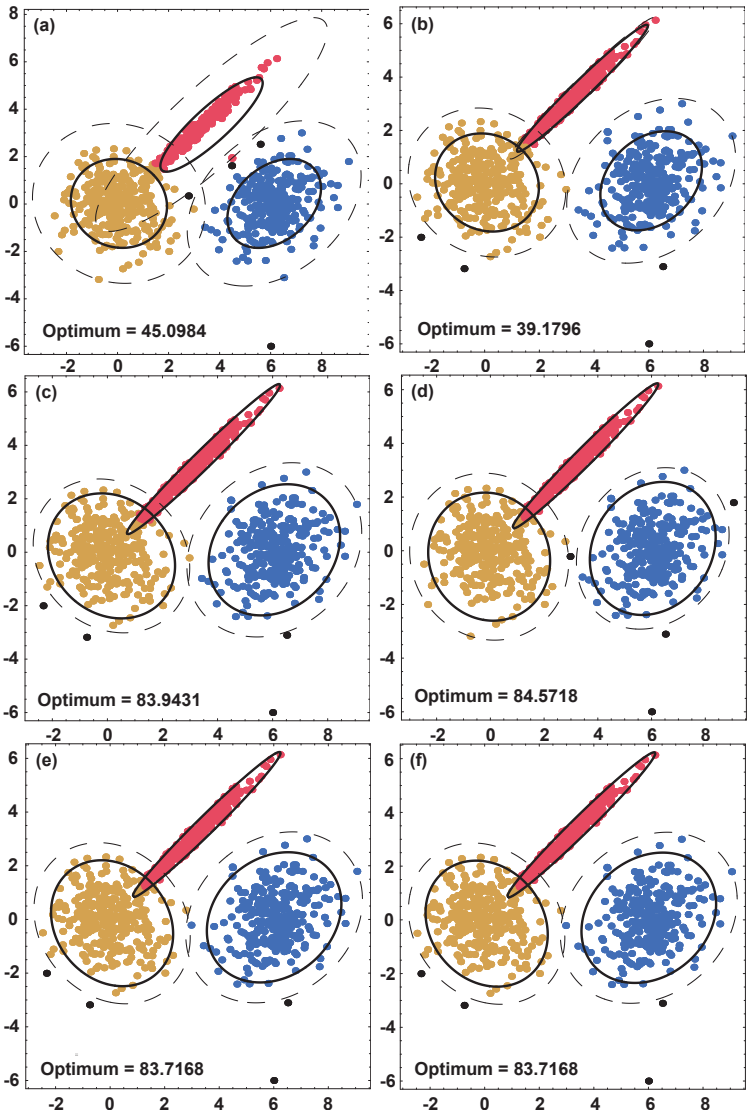


Figure 3.3: Evolution of the proposed algorithm for the Normal data set: (a) initial step, (b) iteration 2, (c) iteration 3, (d) iteration 4, (e) final solution and (f) solution of the problem with binary variables (equations (3.25)-(3.32)).

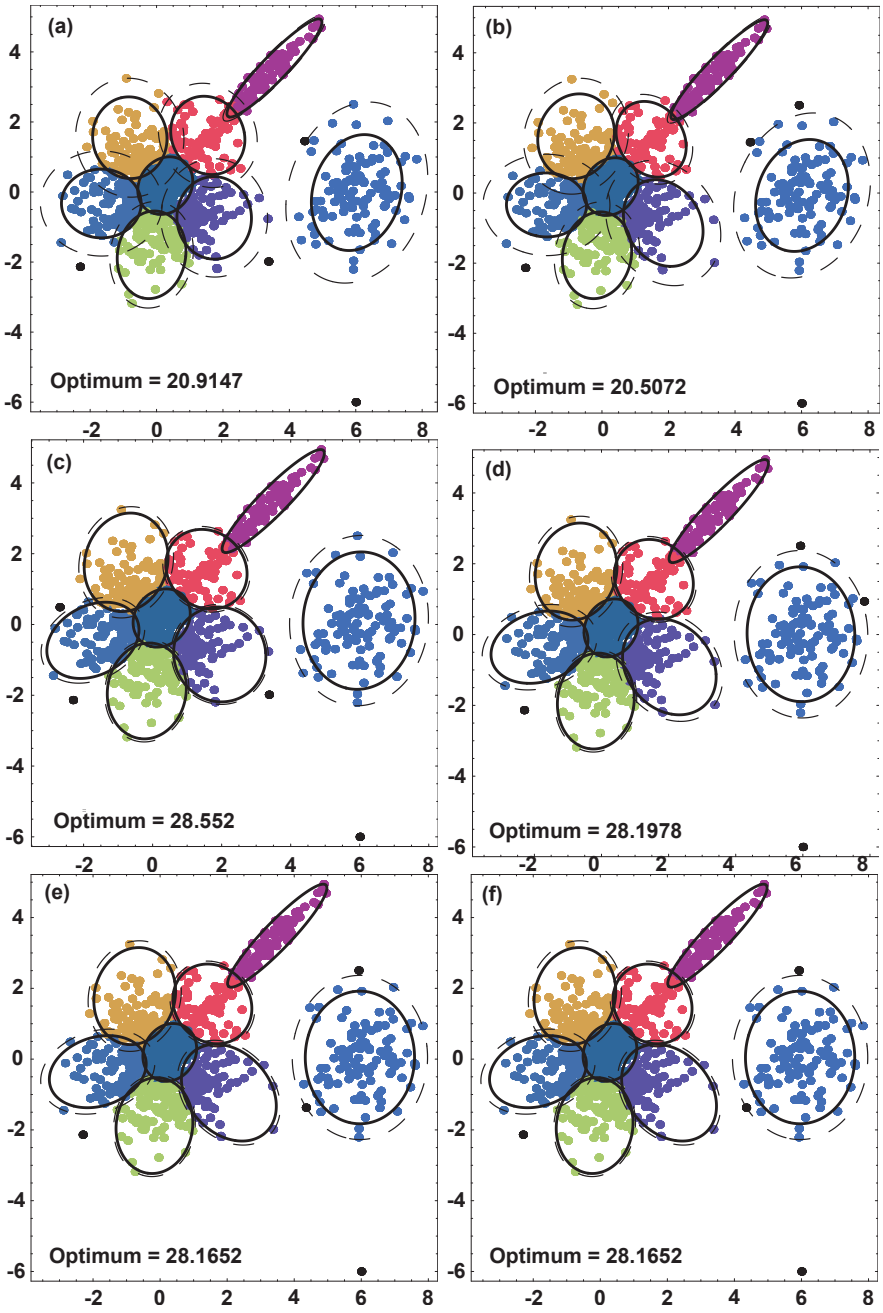


Figure 3.4: Evolution of the proposed algorithm for the Normal data set (number of ellipsoids overestimated): (a) initial step, (b) iteration 2, (c) iteration 3, (d) iteration 4, (e) final solution and (f) solution of the problem with binary variables (equations (3.25)-(3.32)).

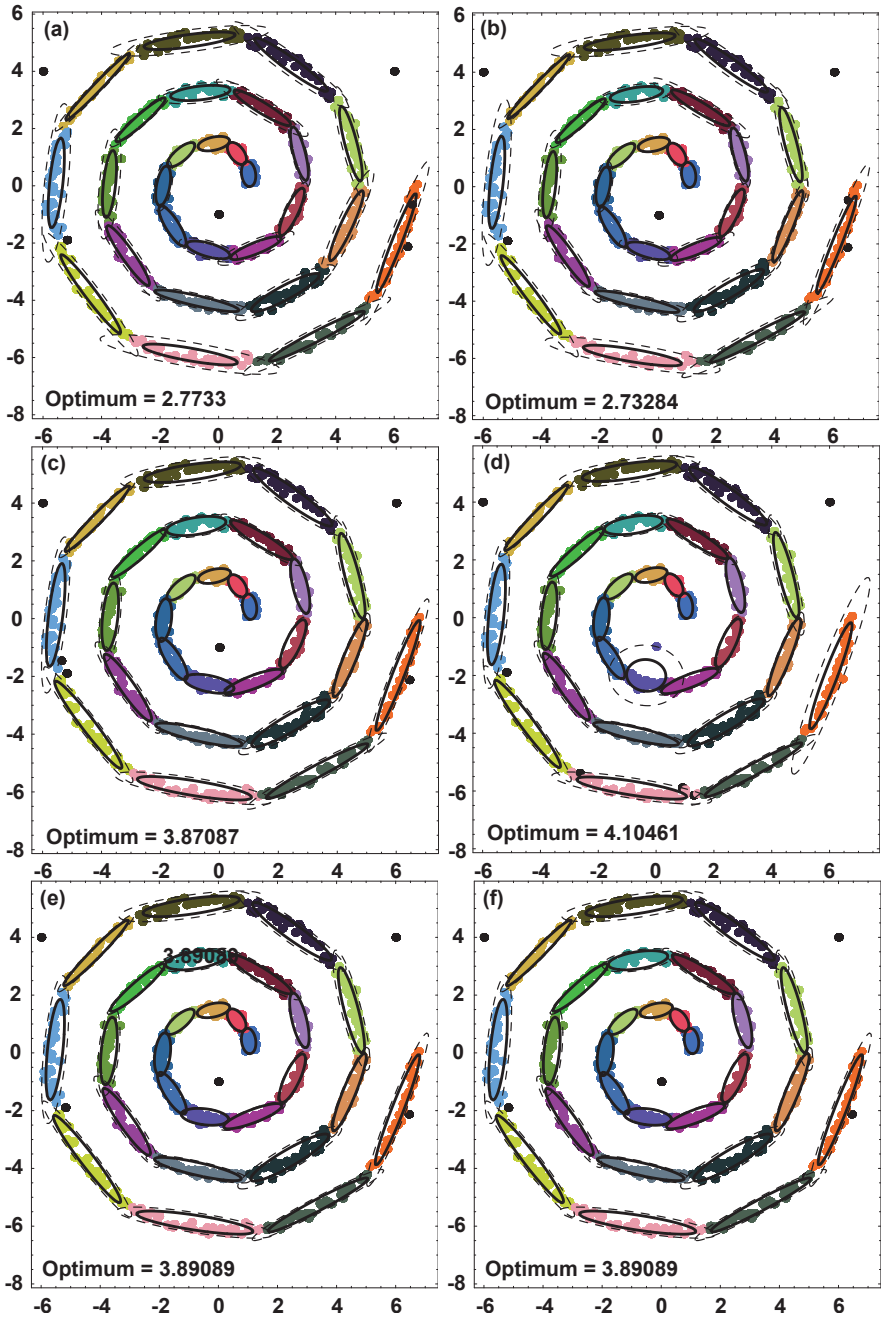


Figure 3.5: Evolution of the proposed algorithm for the Spiral data set: (a) initial step, (b) iteration 2, (c) iteration 3, (d) iteration 4, (e) final solution and (f) solution of the problem with binary variables (equations (3.25)-(3.32)).

Example	CPU times in seconds		
	RMCEID (3.25)-(3.32)	RR-MCEID (3.33)-(3.42)	Algorithm
Normal	10.575	4.431	1.918
Square	10.990	6.053	2.513
Spiral	10.689	4.743	4.181

Table 3.1: A comparison of the times required by the binary, relaxed and algorithmic approaches respectively.

3.1.5.4 Clustering applications

The aim of this section is to study the behavior of the proposed model for clustering applications. In order to accomplish this, we carried out a comparative study with the results of other state of the art methods, obtained by the experiments detailed in [174]. Four data sets are used to compare the proposed method with previous approaches. First we used two data sets (S1 and S2) from the S-dataset collection [86]. Both are 2D datasets with 5000 vectors and 15 Gaussian clusters with different degrees of complexity. In order to test the methods with real life data sets we used data gathered from meteorological stations in Grand St. Bernard (GSB) and Heron Island [174]. In both data sets we used the labeling obtained in [174], based on temporal information. We compared the proposed model with DENCLUE [106], k-means [65], Subtractive Clustering (SC) [55], Gustafson - Kessel (GK) [93] and HyCARCE [174] clustering methods.

In order to compare these algorithms, we followed the indexes of cluster quality used in [174]: (a) the normalized mutual information (NMI) and (b) the misclassification rate (MCR). NMI is a known technique in the evaluation of clustering algorithms and it is based on information theoretic concepts.

$$NMI(\Psi, \Omega) = \frac{I(\Psi, \Omega)}{(H(\Psi) + H(\Omega))/2} \tag{3.78}$$

where I represents the mutual information between Ψ and Ω and H represents the entropy. On the other hand, in order to calculate MCR we need to know which is the best pairing between the real clustering and the one obtained by the method subject of study. To tackle this problem the Hungarian algorithm [40] was used. After this preprocessing step, we compute the misclassification rate by dividing the number of

samples that are not correctly labeled by the total number of samples.

$$MCR = \frac{\#\ Missclassified}{Total\ \#\ of\ samples} \tag{3.79}$$

Table 3.2 shows the results obtained in the comparative study. The parameters of the proposed model where selected in the following way: (a) the number of ellipses where set to the actual number of clusters as in [174], (b) the number of outliers was set to 0 as we do not consider the existence of outliers in these examples, (b) the value of $C_{initial}$ was set to a 50% of the value of C_{final} and (d) the value of C_{final} was selected by the best mean result of 5 10-fold crossvalidation. The range for C_{final} was tested in the range $\{0.01, 10\}$ and the values selected were $C_{final} = 5$ for Grand St. Bernard (GSB), $C_{final} = 0.05$ for Heron Island, $C_{final} = 2$ for S1 and $C_{final} = 0.5$ for S2. It can be observed that the proposed model obtains similar or better results in both indexes compared to the ones obtained by the best method in each data set.

Table 3.2: Results for clustering problems (best results for each data set are boldfaced)

	Data sets							
	S1		S2		GSB		Heron Island	
Algorithm	NMI	MCR	NMI	MCR	NMI	MCR	NMI	MCR
K-Means	0.93	0.16	0.90	0.14	0.87	0.14	0.82	0.25
SC	0.89	0.14	0.81	0.25	0.79	0.30	0.83	0.15
GK	0.97	0.03	0.92	0.06	0.98	0.03	**0.94**	**0.02**
DENCLUE	0.98	**0.005**	0.94	**0.03**	**0.99**	**0.001**	**0.94**	**0.02**
HyCARCE	0.93	0.08	0.89	0.11	0.97	0.03	0.78	0.3
Proposed	**0.99**	**0.005**	**0.95**	**0.03**	**0.99**	0.002	**0.94**	**0.02**

3.1.6 Discussion and future work

In this section, a practical method for solving the problem of minimum volume set of covering ellipsoids was presented. Classical minumum volume covering ellipsoid problem has been generalized through relaxation and multilevel optimization and an effective training algorithm was obtained. The use of multiple ellipsoids allows to obtain better results in cases where a single ellipsoid is not a good model of the data. In

addition, the proposed formulation allows to automatically detect outliers making it suitable for practical applications. The solution to this problem can find significant applications in fields such as clustering and one class classification. Technically speaking, the development of the aforementioned algorithm faces two main challenges in order to obtain an effective method: (a) finding an equivalent formulation of the original SCEDD problem in order to avoid the use of the determinant function and obtain a convenient formulation and (b) circumvent the combinatorial explosion in the search of an optimal data point assignation to different ellipsoids and outlier selection. Both problems have been addressed using decomposition facts extracted from the realm of linear algebra and bilevel algorithms. Furthermore, a series of theoretical results have been proved in order to formally demonstrate the adequacy of the proposed method. Experimental results show that the proposed algorithm can obtain good performance both in terms of accuracy and computational time for artificial and real-word data sets. This justifies considering it as a potential solution for machinery anomaly detection presented in the second main block of this thesis. The bi-level algorithm obtained can be used to generalize the Support Vector Data Description (SVDD) [234] algorithm mentioned in previous chapter to the case of multiple ellipsoids in kernel space using the formulation in [62].

3.2 Nearest neighbor anomaly detection based on extreme value statistics

In this section a new anomaly detection technique based on nearest neighbor principle and extreme value statistics [68] is presented. It aims at exploiting distance information under normal conditions to solve one class classification problems. Specifically, the probability distribution function of the distance of a sample, under normal conditions, to its close neighbors is modeled either parametrically or non-parametrically. Afterwards, using a result from extreme value statistics, when a new sample s is presented, the probability of obtaining a more dissimilar sample than s under normal conditions is obtained and eventually used as an indicative of an anomaly. Thanks to the specific modeling of the distribution of distances to close samples under normal conditions, we shall be able to capture the support of normal data while neglecting possible spurious data in the normal state data set, as these data are typically characterized as being disperse and far from the normal state support. In the experimental section, it can be noticed that exploiting distance information in this manner leads to an accurate one class classifier.

3.2.1 Method description

In this section the principal results and rationale under the proposed Extreme Value One class Classifier (EVOC) method are presented. Firstly, a theorem rooted in Extreme Value Statistics field [68] on which the proposed method is largely based on is presented:

Theorem 3.2.1 (Distribution Function of any order statistics) *The probability distribution function $F_{r:n}$ of any order statistics r of a sample of n values of a random variable with distribution function $F(x)$ is:*

$$F_{r:n} = B_{F(x)}(r, n - r + 1) \tag{3.80}$$

where B is the regularized incomplete beta function.

This result is based on treating the sampling process as a multinomial distribution and using the probabilities extracted from the original distribution function $F(x)$. It can be derived as follows:

$$\begin{aligned} F_{r:n}(x) &= P(X_{r:n} \leq x) = 1 - F_{m_n(x)}(r-1) = \sum_{k=r}^{n} \binom{n}{r} F^k(x)[1-F(x)]^{n-k} \\ &= r\binom{n}{r} \int_0^{F(x)} u^{r-1}(1-u)^{n-r} du = B_{F(x)}(r, n-r+1) \end{aligned}$$

where $m_n(x)$ is the number of elements of the samples with a value $X_j \leq x$ and $F_{m_n(x)}(k)$ with $k \in (0, n)$ represents the probability that the number of samples below x is less than or equal to k. Based on this result, when a new sample s is presented, it is possible to model the probability of obtaining a sample more discrepant or abnormal than s. In order to do this, consider a metric space M where the data we want to classify belongs to.

First, in the training phase the probability distribution function $F_d(x)$ of the distance of each sample to its k nearest neighbors is modeled based on data drawn from only one class, which shall be called *Normal state data*. In order to do this, for each data sample in the normal state data set, we search its k nearest neighbors and use those distances to model $F_d(x)$. It is important to remark that this step relies only on the fact that the data is embedded into a metric space where we have a distance function d, so it is possible to use other data encodings apart form the Euclidean space $\mathbb{R}^n$. For the probability distribution function estimation, both parametric and non-parametric methods are available. In this work we will adopt a parametric approach. In order to set the significance level of the classification rule, we set α to a value that leaves $p\%$ samples of the available set as outliers, where p is a parameter of the method. All these steps are summarized in algorithm 2.

Subsequently, when a new data point s is to be classified, the following rule is used:

$$C(s) = I(P(D_k > d_s) - \alpha) = I\left(\prod_{i=1}^{k}(1 - F^d_{i:k}(d_s(i))) - \alpha\right) \tag{3.81}$$

where I is the heaviside step function, d_s is the set of distances to the k nearest neighbors of s in the normal state data set, $d_s(i)$ is the distance to the i-th closest pattern to s in the normal state data set, D_k is a random variable that represents the distance to the k nearest neighbors ($D_k > d_k$ if $\forall i \in [1,k], D_k(i) > d_k(i)$), $F^d_{r:k}$ is the r-th order statistics formula of theorem 3.2.1 in which the estimated distribution function $F_d(x)$ of the distance to a neighbor is plugged in, and α is a threshold that controls under which level the data sample s is considered abnormal (in this rule, logarithms can be taken to prevent underflow). Note that the classification rule is monitoring the probability of obtaining, under normal conditions, a set of k nearest neighbors more dissimilar than

the ones we have found for s. If this probability falls, it means that the normal state hypothesis for s has been violated and so it is classified as abnormal or counterexample. This classification process is detailed in the lower section of algorithm (see algorithm 2).

When applied to high dimensional data sets, negative effects due to curse of dimensionality can appear. Depending on the used metric, the notion of proximity can become meaningless degrading the contrast between sparse and close neighbors in which the proposed model is based [6]. In this situation, careful selection of distances and dimensionality reduction techniques should be considered.

Algorithm 2: Proposed EVOC classification method

Training Stage

Input: Normal State data X, number of neighbors k, estimated fraction of outliers p

Output: Classifier (X, F_d, α)

foreach *sample* $s \in X$ **do**
  Calculate the set of distances d_s of s to its k-nearest neighbors in X.
  Add the distances in d_s to the set D.

Estimate F_d based on the sample values in D.

Set α leaving a p fraction of data in X out of the support.

Classification Stage

Input: Classifier (D, F_d, α), and a new sample s

Output: Classification result $C(s)$, {1 - Normal State, 0 - Novelty}

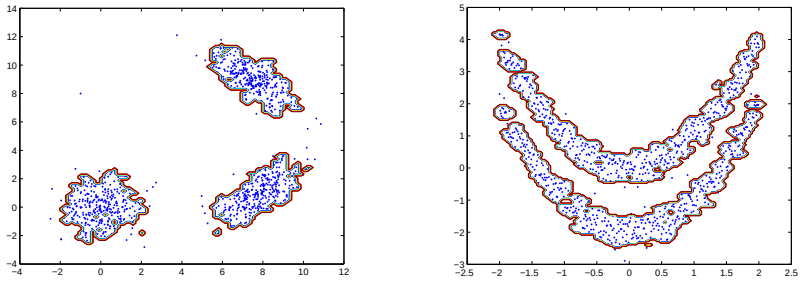
Calculate the set of distances d_s of s to its k-nearest neighbors in X.

Classify s following the rule:

$$C(s) = I(P(D_k > d_s) - \alpha) = I\left(\prod_{i=1}^{k}(1 - F_{i:k}^{d}(d_s(i))) - \alpha\right) \tag{3.82}$$

3.2.2 Experimental results

In order to explore the features of the proposed method, both artificial and real one-class classification data sets are used. For the experiments presented hereafter, we



(a) 2D Random Gaussian data support method's capture.

(b) 2D Banana data support method's capture.

Figure 3.6: Illustration of EVOC normal support capture.

adopt a parametric approach to model the distance between samples. Since distances are all positive and it was experimentally observed that they are concentrated around a mean, the lognormal distribution [121] was adopted.

3.2.2.1 Artificial data sets

In this section, the ability of the proposed method to capture the support of normal data is depicted for two 2D artificial data sets. Specifically, we tested it with a multimodal gaussian data set and a banana shaped data set. The results for number of neighbors $k = 4$ and $p = 0.06$ are depicted in figures 3.6(a) and 3.6(b). As it can be observed in the figures, the exploiting of nearest neighbors distances through order statistics, makes it possible to automatically detect far outliers while still capturing the main support of the normal data.

3.2.2.2 Real data sets

In this section we explore the applicability of the proposed method and compare it with well-established methods in the field of one class classification. Specifically, three data sets from the UCI Machine Learning Repository [85] are used. The first one, Wine, was originally proposed as a multiclass classification problem and in this case it will be casted into a one class classificacion following a one vs the rest approach (class 1 versus the rest). It is a well posed classification problem so it shall be treated as first

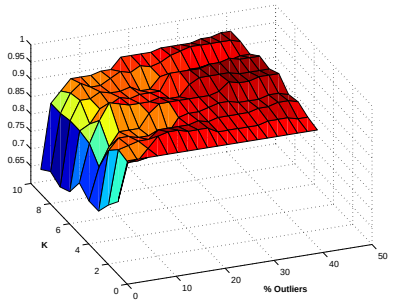


Figure 3.7: Accuracy for Wine data set when changing hyperparameters.

benchmark under good conditions. The second one, Spambase, consists of a collection of spam and non-spam e-mails. This data set is a good representation of the proposed method's applicability in abnormality detection from normal data (non-spam e-mail) in harder scenarios. The third one, Cardiotocography, exemplifies the applicability of this method to biomedical applications. This dataset consists of fetal cardiotocograms (CTGs), which were automatically processed to extract diagnostic features, and the diagnosis label of 'normal', 'suspect' and 'pathological'. For our case of one class classification, we assume both suspect and pathologic as abnormal cardiotocograms. We compare the classification accuracy obtained by the proposed method with the one obtained by two most widespread used one class classifiers: one class ν-SVM [20] and Autoassociative Multilayer Perceptron [115].

For each data set, 30 random runs using 70% of normal class data as training set were done. In table 3.3, the mean accuracy of each method and its standard deviation is shown (best combination of hyperparameters along the random runs). As it can be observed, the proposed method obtains equal or better accuracy than the other two well established one class classifiers. In addition, for the Wine data set, we tested the ability of the three methods to tackle noise samples in the normal state data set introducing in the training set a 10% of abnormal samples. It can be noticed that EVOC still maintains better accuracy than the other two tested methods. Moreover, in figure 3.7 the variability of the accuracy obtained by the proposed method is experimentally studied. It can be observed that for a large range of combinations of k and p, EVOC presents an stable behavior with accuracies above 92%.

Data set	EVOC	one class ν-SVM	Autoass-MLP
Wine	**94.70% (0.001)**	92.45% (0.004)	94.30 % (0.013)
Wine (10%)	**93.25% (0.015)**	91.70% (0.0034)	91.93 % (0.008)
Spambase	**86.33% (0.003)**	86.26% (0.002)	79.52 % (9.81e-4)
Cardiotocography	**80.38% (0.009)**	76.71% (0.007)	75.44 % (0.03)

Table 3.3: Results of EVOC method for UCI data sets.

3.2.3 Discussion and future work

In this section, a one class classifier based on extreme value statistics was presented. The proposed methodology relies only in the existence of a measure of dissimilarity or metric between samples, so it can be extended to other spaces different from $\mathbb{R}^n$. Moreover, the proposed EVOC method has a reduced set of hyperparameters and presents a good performance compared to other frequently used one class classifiers. Proposed method's performance is also explored in problems settled in the Euclidean space, obtaining good results. As it is a memory-based learning method, it can suffer when applied to large data sets. Additional effort could be made in the future aiming at reducing final model's complexity using efficient nearest neighbor methods and trimming the normal state data set.

3.3 On line anomaly detection via passive-agressive one class classification

In this section we move on to the problem of stream anomaly detection. This section is divided in two main parts having the aim of obtaining an accurate and general purpose method for solving this kind of problems. Firstly, a new on-line method for one class classification is developed. This model is based on the passive-aggressive paradigm [58][59] and introduces a new formulation more suitable for practical purposes and which can be applied in complex Hilbert spaces admitting a mapping through the *kernel trick* [211]. This on-line method has the ability of capturing complex normal data support area treating each sample in a one-pass manner. Subsequently, we present the On line Stream Data Anomaly Detector (OSDAD) algorithm. This algorithm stems from the combination of the aforementioned one class classifier with a CUSUM chart of a Bernoulli process. The rationale of this combination is as follows: under normal conditions, the learned training model has a low probability p_0 of classifying a pattern as abnormal; in case there is a change in the conditions, this proportion rises and is eventually detected by the CUSUM chart. Thanks to the ability of on-line adaptation of the classification model and its combination with the CUSUM chart, the proposed algorithm is able to detect significant anomalies with a reduced response time. The algorithm parameters can be selected in order to adapt its sensitivity to changes. Thus, it can be tuned to detect anomalies at a severity level adequate for the problem at hand.

3.3.1 Passive-Aggressive one class classifier

3.3.1.1 Background

In [58] one class classification problems are solved by covering the normal data patterns by an n-dimensional sphere with center $\mathbf{w}$ and radius ϵ. Following the passive-aggressive paradigm, when a new pattern $\mathbf{x}_t$ is presented to the model, this is adjusted balancing two criteria: a minimal perturbation of the previous model and a maximal prediction accuracy. These two criteria are considered in the original formulation which we

reproduce here:

$$\begin{aligned} \mathbf{w}_{t+1} &= \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 \\ s.t. &\quad l_\epsilon(\mathbf{w};\mathbf{x}_t)=0 \end{aligned} \tag{3.83}$$

where $\mathbf{w}_t$ is the center of the sphere at instant t, $\mathbf{x}_t$ is the new pattern presented and l_ϵ is the ϵ-insensitive error function.

$$l_\epsilon(\mathbf{w};\mathbf{x}_t) = \begin{cases} 0 & \text{if } \|\mathbf{w}-\mathbf{x}_t\| \leq \epsilon \\ \|\mathbf{w}-\mathbf{x}_t\| - \epsilon & \text{if } \|\mathbf{w}-\mathbf{x}_t\| > \epsilon \end{cases} \tag{3.84}$$

It can be observed how the optimization program penalizes a perturbation of the center of the sphere and, at the same time, it tries to include the new data point inside the sphere. Also in [58], a slack variable ξ_t to balance these two aims is introduced. This extension leads to further formulations called PA-I (Passive-Agressive I) and PA-II in [58]. The following are the corresponding optimization programs of PA-I (equation (3.85)) and PA-II (equation (3.86)).

$$\begin{aligned} \mathbf{w}_{t+1} &= \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 + C\xi_t \\ s.t. &\quad \|\mathbf{w}-\mathbf{x}_t\| \leq \epsilon + \xi_t \\ &\quad \xi_t \geq 0 \end{aligned} \tag{3.85}$$

$$\begin{aligned} \mathbf{w}_{t+1} &= \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 + C\xi_t^2 \\ s.t. &\quad \|\mathbf{w}-\mathbf{x}_t\| \leq \epsilon + \xi_t \end{aligned} \tag{3.86}$$

It can be observed that the difference between these two formulations lies in the weighting of the distance of a data point to the model represented by the slack variable ξ_t. While PA-II squares this outer error, PA-1 introduces it lineally in the objective function. The constant C is in the range $(0,\infty)$ and penalizes that a new incoming pattern is incorrectly classified.

Although in [58] the update formulas obtained from these optimization programs in equations (3.85) and (3.86) are very efficient, they present a problem concerning the radius of the sphere. It can be observed in equations (3.84), (3.85) and (3.86) that the parameter ϵ is not part of the optimization program and has to be set manually

by the practitioner, which can be an impractical decision in many real-world environments. In [58], this issue is tackled by introducing the radius as an extra dimension in the weight vector (details can be consulted in the reference). This approach has two limitations: (a) radius can only grow along the time so, in a real application, an incorrect or very low probability observation can increase it too much, leading to a suboptimal sphere which can not recover an optimal volume; (b) the interpretation of this approach when extending the algorithm to another Hilbert space using the *kernel trick* is not straightforward. In the next section, we propose an alternative formulation which tries to tackle these two issues and leads to an algorithm that: (a) can adapt its radius (volume) to the requirements of the problem thus being more robust against noise and changing conditions and (b) can be combined in a straightforward manner with the *kernel trick* in order to obtain non-linear decision boundaries.

3.3.1.2 Proposed Method

Before exposing the proposed formulation, we introduce the principles that lead to it. First of all, the original *outer error* slack variable of equation (3.85) is maintained and represented by ξ_t. This variable accounts for the distance of a data point to the projection of this point into the sphere. In addition, we add the radius of the sphere r as an extra variable in the program to be solved at each step. The idea of the objective function is the following: try to correctly classify the incoming pattern $\mathbf{x}_t$, with a minimal impact on the current model, namely the center of the sphere $\mathbf{w}_t$, and maintaining a minimum volume sphere (proportional to radius r). Figure 3.8 depicts this idea. In order to correctly classify a new incoming pattern with a minimal impact in the center of the sphere, the optimal direction of movement of the center $\mathbf{w}_t$ is to move it directly towards $\mathbf{x}_t$, so this restriction is introduced in the program to be solved a each step. Following these criteria, the proposed formulation is:

$$\begin{aligned} \mathbf{w}_{t+1}, r_{t+1} &= \underset{\mathbf{w}\in\mathbb{R}^n, r\in\mathbb{R}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 + C_r r + C\xi_t \\ s.t. \quad & \mathbf{w} = (1-\lambda)\mathbf{w}_t + \lambda\mathbf{x}_t \\ & \|\mathbf{w}-\mathbf{x}_t\| - r \leq \xi_t \\ & r \geq (1-q_i)r_t \\ & \xi_t, r, \lambda \geq 0 \end{aligned} \tag{3.87}$$

where $\mathbf{w}_{t+1}$ is the new center, r_{t+1} the new radius and $\mathbf{x}_t$ the new input pattern. The hyperparameters of the model are described in table 3.4. In the formulation, it can

Table 3.4: Hyperparameters of the proposed model.

C	Its value is in the range $(0, \infty)$. It penalizes that a new incoming pattern is incorrectly classified.
C_r	Its value is in the range $(0, \infty)$ It penalizes having a large volume sphere.
q_i	Its value is in the range $(0, 1)$. In case the new pattern falls into the sphere, the model will decrease the radius if needed. This value controls the maximum decrease of the radius as a proportion of the last radius.

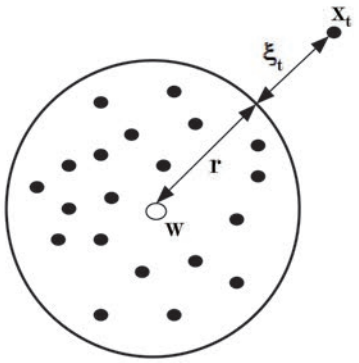


Figure 3.8: Training data patterns with $\xi_t > 0$.

be observed that the radius can increase or decrease as needed, trying to maintain a minimal volume. When a new income pattern is not covered by the current model, radius increment is penalised by the cost function. Otherwise, radius decrement is constrained by q_i to control the impact on the current model. Thus, passive-aggressive principle is maintained during learning both for $\mathbf{w}$ and r.

In order to solve the presented optimization problem, it can be proved that this can be done through a reduced quadratic program.

Lemma 3.3.1(equivalent quadratic program): The solution to the optimization problem of equation (3.87) can be found by solving the following quadratic pro-

gram.

$$\begin{aligned} \lambda_t, r_{t+1} &= \underset{\lambda,r}{\operatorname{argmin}} \frac{1}{2}P^2\lambda^2 + C_r r + C\xi_t \\ s.t. \quad & (1-\lambda)P - r \leq \xi_t \\ & r \geq (1-q_i)r_t \\ & \xi_t, r, \lambda \geq 0 \end{aligned} \tag{3.88}$$

where $P = \|\mathbf{w}_t - \mathbf{x}_t\|$ and $\mathbf{w}_{t+1} = (1-\lambda_t)\mathbf{w}_t + \lambda_t \mathbf{x}_t$.

Proof: The equivalent quadratic program is constructed by following these steps:

- Substitute the first constraint in the terms involving norms. Namely:

$$\begin{aligned} \|\mathbf{w} - \mathbf{w}_t\|^2 &= \|((1-\lambda)\mathbf{w}_t + \lambda\mathbf{x}_t) - \mathbf{w}_t\|^2 \\ &= (1-\lambda)^2\langle \mathbf{w}_t, \mathbf{w}_t\rangle + 2\lambda(1-\lambda)\langle \mathbf{w}_t, \mathbf{x}_t\rangle \\ &\quad +\lambda^2\langle \mathbf{x}_t, \mathbf{x}_t\rangle - 2(1-\lambda)\langle \mathbf{w}_t, \mathbf{w}_t\rangle \\ &\quad -2\lambda\langle \mathbf{x}_t, \mathbf{w}_t\rangle + \langle \mathbf{w}_t, \mathbf{w}_t\rangle \\ &= \lambda^2(\langle \mathbf{w}_t, \mathbf{w}_t\rangle - 2\langle \mathbf{w}_t, \mathbf{x}_t\rangle + \langle \mathbf{x}_t, \mathbf{x}_t\rangle) \\ &= \lambda^2\|\mathbf{w}_t - \mathbf{x}_t\|^2 \end{aligned}$$

$$\begin{aligned} \|\mathbf{w} - \mathbf{x}_t\| &= \|((1-\lambda)\mathbf{w}_t + \lambda\mathbf{x}_t) - \mathbf{x}_t\| \\ &= \sqrt{(1-\lambda)^2\langle \mathbf{w}_t, \mathbf{w}_t\rangle + 2\lambda(1-\lambda)\langle \mathbf{w}_t, \mathbf{x}_t\rangle} \\ &\quad \overline{+\lambda^2\langle \mathbf{x}_t, \mathbf{x}_t\rangle - 2(1-\lambda)\langle \mathbf{w}_t, \mathbf{x}_t\rangle} \\ &\quad \overline{-2\lambda\langle \mathbf{x}_t, \mathbf{x}_t\rangle + \langle \mathbf{x}_t, \mathbf{x}_t\rangle} \\ &= \sqrt{(1-\lambda)^2(\langle \mathbf{w}_t, \mathbf{w}_t\rangle - 2\langle \mathbf{w}_t, \mathbf{x}_t\rangle)} \\ &\quad \overline{+(1-\lambda)^2\langle \mathbf{x}_t, \mathbf{x}_t\rangle} \\ &= (1-\lambda)\|\mathbf{w_t} - \mathbf{x}_t\| \end{aligned}$$

  Substituting these two equations in the objective function and in the second condition, respectively, we obtain the quadratic program of lemma 3.3.1.

It can be observed that the algorithm can be applied in a feature space F via non-linear mapping of a kernel function k which approximates the inner product in F.

Namely, in equation (3.88), the only information concerning operations in F is P and so the norm induced by k can be used. This can be done because the optimal center $\mathbf{w}_t$ is always a linear combination of the past presented patterns $\mathbf{x}_i, i \leq t$, allowing the calculation of the norm induced by the inner product k using the linear property of the inner product in a Hilbert space. Several different kernel functions can be used, like RBF (Radial Basis Funtion), polynomial, sigmoid, etc [211]. In this work the following Gaussian RBF kernel is used:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \tag{3.89}$$

where σ controls the width of the distribution.

The patterns $\mathbf{x}_t$ for which $\lambda_t > 0$ are called Support Vectors (SV). The update formula of the center of the sphere $\mathbf{w}$ in equation (3.87) could lead to hard storage limitations since the number of SV could grow very quickly. Hereunder we present a lemma that states that this problem is automatically addressed by the model, discarding many patterns once a stable accurate model is obtained.

Lemma 3.3.2 (controlled growth): If $\|\mathbf{w}_t - \mathbf{x}_t\| \leq r_t$, then the optimal solution of the problem in lemma 3.3.1 has $\lambda = 0$.

Proof: First of all we construct the Langrangian and the KKT conditions of the quadratic program in lemma 3.3.1.

$$\begin{aligned} L(\cdot) &= \frac{1}{2}P^2\lambda^2 + C_r r + C\xi_t + \beta((1-\lambda)P - r - \xi_t) + \\ &+ \eta((1-q_i)r_t - r) - \theta\xi_t - \phi r - \omega\lambda \end{aligned} \tag{3.90}$$

$$\frac{\delta L}{\delta \lambda} = P^2\lambda - \beta P - \omega = 0 \tag{3.91}$$

$$\frac{\delta L}{\delta r} = C_r - \beta - \eta - \phi = 0 \tag{3.92}$$

$$\frac{\delta L}{\delta \xi_t} = C - \beta - \theta = 0 \tag{3.93}$$

$$\beta((1-\lambda)P - r - \xi_t) = 0 \tag{3.94}$$
$$\eta((1-q_i)r_t - r) = 0 \tag{3.95}$$
$$\theta\xi_t = 0 \tag{3.96}$$
$$\phi r = 0 \tag{3.97}$$
$$\omega\lambda = 0 \tag{3.98}$$

where $\beta, \eta, \theta, \phi, \omega$ are dual variables. We will prove that if $P = \|\mathbf{w}_t - \mathbf{x}_t\| \leq r$ and the solution has $\lambda > 0$ then it violates the KKT conditions of the optimal. First, using equation (3.91) we have that

$$\lambda = \frac{\beta}{P} + \frac{\omega}{P^2} \tag{3.99}$$

Thus, using equation (3.98), we have that if $\lambda > 0$ then $\omega = 0$ and $\beta > 0$. This last fact implies that in the optimal solution, using equation (3.94), necessarily

$$(1-\lambda)P - r - \xi_t = (P - r) - \lambda P - \xi_t = 0 \tag{3.100}$$

as $P \leq r$ we have that

$$-\lambda P - \xi_t \geq 0 \tag{3.101}$$

since we supposed that $\lambda > 0$ the only possibility is that ξ_t is negative, which contradicts a primal condition. This leads to the conclusion that a solution with $\lambda > 0$ can not be optimal.

This lemma has important implications for practical purposes. Basically, it states that when a new training data point is correctly labeled as normal (inside the sphere), this data point is discarded and will not be part of the final model. This fact implies: (a) this model can be considered as a member of the passive-agressive, since it minimizes the disruption of the current model when a new sample is correctly classified and (b) once the model is a stable and accurate description of the normal support of the dataset, it will automatically decelerate its growing since the probability of the next pattern $\mathbf{x}_t$ being correctly classified is high. This final aspect will be inspected in the experimental section.

3.3.2 Stream anomaly detection algorithm

In some machine learning applications, it is needed to find a method able to detect significant changes in the generative distribution of a process, while being able to adapt to low severity not significant variations that can occur during it. One example of this scenario is rotating machinery fault detection [79][125][163][170][253]. In rotating machinery working continuously, such as a wind mill or a turbine, low severity variations of the vibration signatures is normal, while the sudden degradation of them could be produced by the presence of a fault, which we would like to detect. In addition, these processes usually produce a continuous stream of data, so on-line processing could be necessary. This application field is studied in later chapters of this thesis.
In this section, a method for change or anomaly detection using the proposed one class classifier described above is presented. The method fulfills the aforementioned properties: (a) is completely on-line, (b) is able to adapt to low severity changes and, at the same time, (c) detect significant anomalies.
The proposed algorithm stems from the combination of the proposed one class classifier and the continuous application of a Bernoulli CUSUM test [196]. The rationale is as follows: under stable conditions, a trained model will maintain a stable low probability p_0 of classifying an incoming pattern as anomalous. In case the generative distribution of the process changes due to a disturbance, the model will be no longer an accurate description of the process, so this probability of misclassification will rise, giving sign of an anomaly. Detecting this change needs an accurate multipurpose change detection algorithm. In the next section, how to construct a Bernoulli CUSUM chart for this purpose is explained and we close the subsection detailing how it is integrated in the proposed algorithm.

3.3.2.1 The Bernouilli CUSUM chart

This section presents the CUmulative SUM (CUSUM) chart for monitoring a process when items from the process are inspected and classified into one of two categories, namely defective or non-defective. Specifically, the notation will be as follows: the results for the k-th item inspected can be represented as a Bernoulli observation X_k where:

$$X_k = \begin{cases} 1 & if\ k-th\ item\ is\ defective \\ 0 & otherwise \end{cases} \tag{3.102}$$

Using this data, Bernoulli CUSUM monitors whether the probability of obtaining a defective item p is p_0 or whether it has risen. The Bernoulli CUSUM chart is not the only possibility for monitoring the proportion of a Bernoulli process. Other approaches have been proposed in the literature such as the Stewart p-chart or the Binomial CUSUM [196]. However, CUSUM chart has the advantage of being based directly on each individual observations $X_1, X_2, \dots X_n$. When samples of size $n > 1$ are taken (for example in a stream of data), the Bernoulli CUSUM chart is applied individually to the items in the sample and a point is plotted on the Bernoulli CUSUM chart after each individual observation within the sample of n. This is a desirable property for an on-line processing since it is not needed to store a sample and change point detection can be more accurate. For detecting an increase in p, the Bernoulli CUSUM control statistic can be expressed as:

$$B_k = \max(0, B_{k-1}) + (X_k - \delta), k = 1, 2, 3, \dots \tag{3.103}$$

where the reference value is δ, and which calculations will be seen immediately afterwards. The starting value, B_0, for the statistic is frequently taken to be 0 but can be taken to be a positive value if a head start is desired. Before starting the test, a detection threshold h is fixed. Each time an item is inspected, the value of B_k is updated using equation 3.103. The test will signal that there has been an increase in p if $B_k > h$.

This expression of the Bernoulli CUSUM statistic is not the traditional one and it can take negative values. The reason is that some statistical properties of the Bernoulli CUSUM chart depend on knowing the value of B_k when it drops below zero (details can be consulted in [195]).

To determine the value of δ, it is necessary to specify a value $p_1 > p_0$, which represents an out-of-control value of p that should be detected quickly. For a given in-control value p_0 and a given out-of-control value p_1, constants r_1 and r_2 are defined as:

$$r_1 = ln\left(\frac{1-p_1}{1-p_0}\right) \tag{3.104}$$

$$r_2 = ln\left(\frac{p_1(1-p_0)}{p_0(1-p_1)}\right) \tag{3.105}$$

Then, starting from the basic definition of a Sequential Probability Test (SPRT), it can be shown that the appropriate choice for δ is:

$$\delta = \frac{r_1}{r_2} \tag{3.106}$$

Details and proofs of these expressions can be consulted in [195].

3.3.2.2 Proposed algorithm: OSDAD

In this section we present the algorithm On-line Stream Data Anomaly Detector (OSDAD). As previously mentioned, it stems from the combination of the on-line one class classifier proposed in section 3.3.1.2 and a Bernoulli CUSUM Chart. Inspired on the application of CUSUM charts for adaptive tracking (see [25]), the details of our approach are explained in algorithm 3.

The proposed method is designed to sequentially detect changes of concept based on the received data stream. The algorithm is able to tackle this task under the assumption that length of time intervals between concept drifts allow learning a stable model of each concept. The model under each scenario is used in the subsequent change detection phase in order to detect anomalies. The estimation and detection phases work cyclically as detailed in algorithm 3.

The algorithm takes as inputs: the hyperparameters of the proposed classifier, the initial radius, the kernel function and Average Number of Observations to Signal (ANOS), explained later in this section. On the absence of any guidance for setting the initial model, the center is set to the vector 0. When algorithm 3 starts, anomaly detection process (Bernoulli CUSUM) is disabled until a first stable model is adjusted. In order to obtain this first model, we update the classifier for N data samples (line 23). Once the model is trained, the probability p_0 of classifying a pattern as anomalous is estimated and the parameters of the Bernoulli CUSUM test are calculated (lines 14-18), see next subsections for more details). Subsequently, since anomaly detection monitoring is enabled (line 4), in case a significant change appears, the CUSUM test will signal an anomaly (lines 5-11). In this case two options are possible: (a) detection of this anomaly finishes the process or (b) tracking of changes is required. In the latter case, the process can be reinitialized in order to detect future changes.

Proportion of abnormal patterns estimation

In algorithm 3, once a model of normal data is constructed using a window S of $|S| = N$ patterns, it is necessary to calculate the probability p_0 of classifying a pattern as abnormal under normal conditions. In order to initiate the detection phase, a subwindow of the classification values for the latest M patterns in S ($M < N$) is taken as a sample to estimate p_0. It is supposed that, during this subwindow, the model has al-

Algorithm 3: OSDAD Algorithm

Require: $C, C_r, q_i, (N, M)$ (data window sizes), (k, σ) (kernel function and kernel parameters), ANOS (Average Number of Observations to Signal), r_{in} (initial radius)

$changeActivated \leftarrow false, S \leftarrow \emptyset$ (sample set)
$\mathbf{w} \leftarrow \mathbf{0}, r \leftarrow r_{in}$
for each new pattern $\mathbf{x}_t$ **do**
  if *changeActivated* **then**
    $X_t = classify(\mathbf{x}_t, \mathbf{w}, r)$
    $B_t \leftarrow \max(0, B_{t-1}) + (X_t - \delta)$
    if $B_t > h$ **then**
      Signal an alert in point t
      Reinitialize the model $\mathbf{w} \leftarrow \mathbf{0}, r \leftarrow r_{in}$
      Set $changeActivated \leftarrow false$
    end if
  else
    if $|S| = N$ **then**
      Estimate p_0 based on a subwindow
      of the last patterns M in S
      Compute p_1, δ, h based on p_0 and ANOS
      Set $B_t \leftarrow 0, S \leftarrow \emptyset$
      Set $changeActivated \leftarrow true$
    else
      Add pattern $\mathbf{x}_t$ to the set S
    end if
  end if
  Update $\mathbf{w}, r$ using pattern $\mathbf{x}_t$ and C, C_r, q_i (equation (3.88)).
end for

ready captured the support of the distribution. Thus, these classification values can be considered a sample of the behavior of the model in the subsequent detection phase. There are different estimators for this purpose in the literature. In our case, we have some limitations for this estimation, since the probability of classifying a pattern as abnormal is low and the number of patterns available for the estimation of p_0 will not be high in many real applications. In [39], a detailed study of various estimators is presented and it is demonstrated that, for low values of p_0 and sample size n, there are estimators different from the classical Wald estimator which obtain better results. Taking this study into account, we selected the Agresti-Coull estimator [10] since it presents a better behavior for low p and n:

$$\hat{p} = \frac{X+2}{n+4} \tag{3.107}$$

being X the number of successes (patterns classified as abnormal) and n the sample size used (N in algorithm 3). We refer to [39] and [10] for further details of this estimator.

Bernoulli CUSUM Limit fixation

The final aspect of the OSDAD algorithm that should be covered is how to set the limit h of the Bernoulli CUSUM chart. This value balances the false alarm rate and fast detection shifts to p_1. Since the probability of classifying a pattern as anomalous under normal conditions should be low, the target p_1 is typically set as 3-5 times the probability p_0 (see discussion in [195]). In order to set this value, a quantity called Average Number of Observations to Signal is introduced (ANOS). In practice, it is desirable to have a high ANOS when $p = p_0$ in order to have a low false alarm rate. The equations involved in this calculation are:

$$h^* = h + \epsilon(p_0)\sqrt{p_o(1-p_0)} \tag{3.108}$$

$$\epsilon(p) = \begin{cases} 0.410 - 0.0842log(p) & \\ -0.0391log(p)^3 - & \\ -0.00378log(p)^4 - & \\ -0.000008log(p)^7 & if\ 0.01 \leq p \leq 0.5 \\ \frac{1}{3}\left(\sqrt{\frac{1-p}{p}} - \sqrt{\frac{p}{1-p}}\right) & if\ 0 < p < 0.01 \end{cases} \tag{3.109}$$

$$ANOS(p_0) = \frac{exp(h^* r_2) - h^* r_2 - 1}{|r_2 p_0 - r_1|} \tag{3.110}$$

For given values of r_1 and r_2 and a desired value for the in-control ANOS, equation (3.110) can be used to find the required value of h^* and, subsequently, equations (3.108) and (3.109) can be used to find the required value for h. The value of ANOS balances the number of false alarms and the detection delay of the method. The greater the value, the lesser false alarms are expected, but on the other hand the delay of the detection can increase. The properties and derivation of these equations can be consulted in [195].

3.3.3 Experimental Results

The experiments are divided in two subsections. In the first one, the response of the on-line algorithm proposed is tested in stationary and dynamical environments. In the second one, the performance of the OSDAD algorithm is showed using for simulated anomaly detection problems. As was previously mentioned, in all experiments a Gaussian RBF kernel is used (see equation (3.89)).

3.3.3.1 One class classification in stationary and dynamical environments

Artificial datasets

Four different datasets have been used to assess the performance of the on-line one class classification method previously presented. All datasets are simulated in a two-dimensional Euclidean space and their properties are detailed in Table 3.5. Whilst Dataset #1 is an static Gaussian, Datasets #2, #3, and #4 change dynamically. Datasets #2 and #3 change the center of a Gaussian continuously along an straight line and an arc. Dataset #4 consists of 'C' shaped data generated along a semicircle and adding gaussian noise. The center of this semicircle is increased 3 units each 200 samples giving two changes for the whole dataset. These datasets have been used by other authors [43] in order to assess the performance of one class classifiers in dynamical environments.

The parameter values used for building the classifier are listed in Table 3.6. The first parameter, σ, controls the width of the RBF kernel selected to carry out the

Table 3.5: Characteristics of experimental datasets.

Dataset	#1	#2	#3	#4
Dimensionality	2	2	2	2
Size	300	200	150	600
Stationarity	YES	NO-straight movement	NO-arc movement	NO-straight movement
Geometry	Gaussian $\mu = \begin{bmatrix} 0 & 0 \end{bmatrix}$ $\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$			Letter C $radius = 8$ $center = (14, 0)$ $\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$

Table 3.6: Parameters of the model for each experiment.

Dataset Parameters	#1	#2	#3	#4
σ	2	3	3	10
C	0.1	0.1	0.1	0.1
C_r	0.005	0.03	0.014	0.02
q_i	5e-6	0.01	0.05	3e-5

experiments. The meaning of the ofther parameters has been described before (section 3.3.1.2) in Table 3.4. Since the aim of these artificial data sets is to give a good illustration of the method performance, parameters were manually tuned in order to show its main characteristics.

Figure 3.9 depicts the estimated classification boundary for dataset #1, which is used to observe the behavior of the model in stationary environments. It can be noted that the model approximates very well the high density region of the distribution. In order to check this quantitatively we approximated the following probability

$$P(\|\mathbf{w} - \mathbf{x}\| < r | \mathbf{x} \in D) = \int_S I(\|\mathbf{w} - \mathbf{x}\| - r) p(\mathbf{x}) d\mathbf{x} \tag{3.111}$$

which accounts for the probability of classifying a pattern as normal when it is indeed normal. In this formula D represents the distribution of the data under normal conditions, S for the input space and I is the indicator function for nonnegative reals. This integral has been approximated by Markov Chain Monte Carlo methods [151] giving a

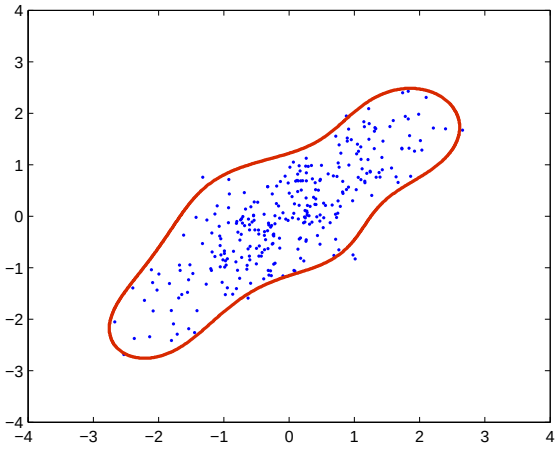


Figure 3.9: Outcome of the proposed model for stationary dataset #1.

value of $P(\|\mathbf{w} - \mathbf{x}\| < r|\mathbf{x} \sim D) = 0.96$. This value proofs that the proposed algorithm was able to capture the support of the distribution with minimum volume.

Figure 3.10 shows the evolution of the classification boundary for dataset #2 (thicker boundaries represent the final model). In this case, as part of a dynamic process, the class moves along a straight line. The boundary after every fifty new data points is shown. As can be seen, oldest data are forgotten by the model and most recent data are effectively delimited by the last boundary, thus the model captures very well the dynamic nature of the data.

In the case of dataset #3 (see figure 3.11), the class moves along the arc of a circle. As in the previous example, after every fifty new data points the classification boundary is depicted. As can be seen, the classification boundary moves in the direction of the class and approximates quite well the shape of the newest data also in more complex variations.

Finally, figure 3.12 shows the outcome of the model for dataset #4. In this case, the dataset is 'C' shaped data that move along a straight line and the boundaries are depicted every new 200 samples. Once again, the most recent data are perfectly detected, despite of the complexity of the support of the data's distribution.

In section 3.3.1.2, lemma 3.3.2 states that any time a new pattern is classified as normal, it does not increase the support vector base of the model. Figure 3.13 shows

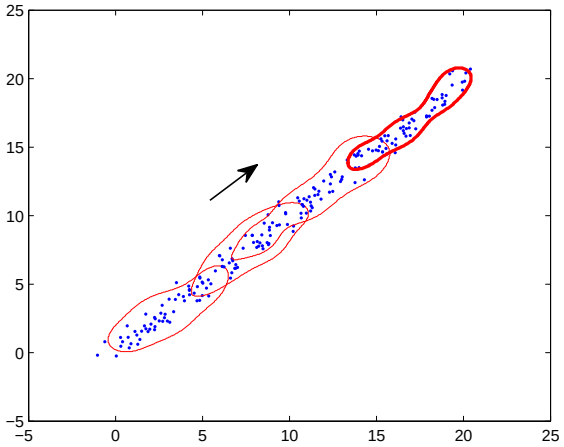


Figure 3.10: Outcome of the proposed model for non-stationary dataset #2.

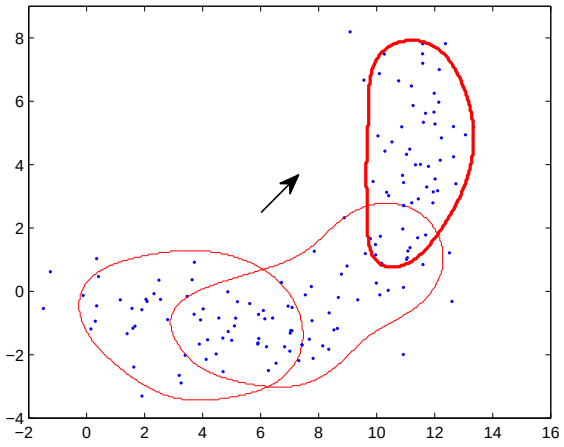


Figure 3.11: Outcome of the proposed model for non-stationary dataset #3.

the growth of the support vector base of the model for the four datasets. It can be observed that the proposed model is able to obtain an accurate representation of the support of complex datasets with a reduced support vector base regarding the total number of samples. In dataset #1, the stationarity of the problem helps to converge to a stable number of SVs in a very short time. On the other hand, the continuous change of datasets #2 and #3 makes difficult to stabilize a static model, although the proposed algorithm is able to obtain an accurate model with a small fraction of the number of patterns presented. Finally, in dataset #4 it can be observed that while

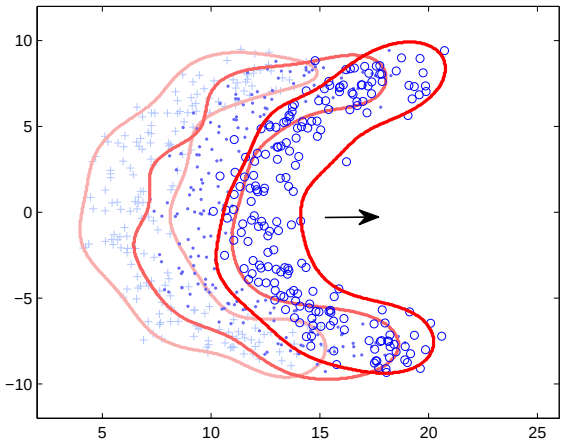


Figure 3.12: Outcome of the proposed model for non-stationary dataset #4.

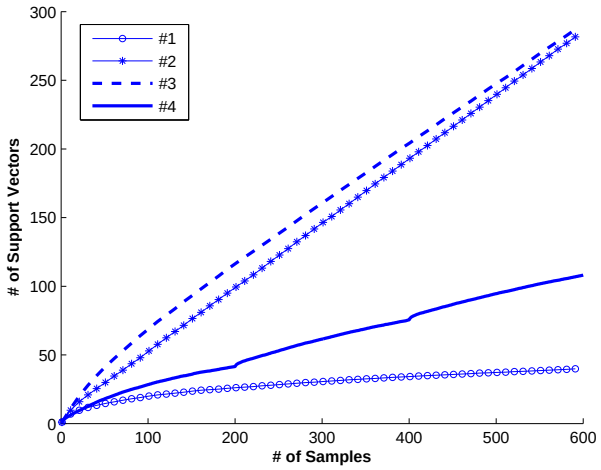


Figure 3.13: Scalability of the proposed model for datasets #1, #2, #3 and #4.

the distribution of the data remains stable, the proposed model is able to calculate its support with a reduced amount of SVs. When the changes occur, the model re-adjusts itself to the new distribution and converges again eventually. An open line of research could be how to discard useless patterns after a change has occurred leading to a further reduction of the SV base. This is a very interesting property for practical purposes in on line learning scenarios.

Real datasets

In order to assess the performance of the proposed model in real one class scenarios we have compared it with well-established kernel methods for one class classification. Namely, we have compared with: (a) the on-line PA-I described in [58] and (b) the batch one class ν-SVM [209][211] . The aim of this comparison is to confirm that the proposed model obtains better classification performance in real datasets when compared to previous methods.
We have selected four datasets in the UCI repository [85]: Iris, Balance, Ionosphere and Wine. A brief description of each dataset can be seen in Table 3.7. In order to map them to a one class problem, we have trained the three models in order to separate each one of the classes from the rest. This generates as many one classification problems as the number of classes in each dataset.
Each dataset was randomly divided into a training set and a testing set using the 70-30 ratio. In order to select the best combination of hyper-parameters, a 10-fold cross validation was performed. The range of parameters compared for each method are listed in Table 3.8. In Table 3.9 we report the mean area under the curve (AUC) on the test set for 10 runs for each of the problem setups and the mean CPU time for each dataset. In the first column the class identifier of each dataset is followed by, respectively, the number of samples of the class considered as normal and the number of samples of the class considered as pertaining to the abnormal class. It can be observed that the proposed model obtains a better mean AUC than the former PA-I in every dataset. This results confirm that an automatically selected radius can give better results in real data sets. When compared to the batch ν-SVM, the proposed model obtains a comparable or slightly worse classification accuracy. This result verifies that the proposed model can obtain a performance comparable to a batch kernel one class classifier tackling the training in an efficient one pattern at a time manner. The slightly worse results in some setups arise from the fact that the batch one class classifier has the possibility to manage the whole dataset to find the optimal model. It can be also observed that the execution times are in the order of the execution times of the PA-I and faster than the batch ν-SVM.

In order to check the impact of the parameter selection in the accuracy of the proposed model, the parameters have been varied around the optimal ones for the case of the Wine dataset. In this case, the optimal parameters were $\sigma = 8.5$, $C = 0.1$, $C_r = 0.00075$ and $q_i = 0.00001$. Figures 3.14, 3.15, 3.16 and 3.17 show the variation of the AUC when one of the parameters varies with respect to the others which remain constant. It can be observed through the AUC curves that there are many parameter

Table 3.7: UCI datasets description.

Dataset	Type	Attributes	Samples	Classes
Iris	Real	4	150	3
Balance	Categorical	4	625	3
Ionosphere	Integer, Real	34	351	2
Wine	Integer, Real	13	178	3

Table 3.8: Combinations of parameters used in cross validation.

Method	Parameters
Proposed	σ: $[1-10]$
	C: $[0.001-0.1]$
	C_r: $[0.0001-0.1]$
	q_i: [1e-7 − 1e-4]
PA-I	σ: $[1-10]$
	C: $[0.005-0.1]$
	r: $[0.9-0.99]$
ν-SVM	γ: $[0.05-0.5]$
	ν: $[0.001-0.1]$

Table 3.9: AUC results and CPU time achieved on UCI benchmarks datasets.

Dataset		Proposed	PA-I	ν-SVM
Iris	Class 1 (50,100)	100	98.46	100
	Class 2 (50,100)	95.61	94.93	96.3
	Class 3 (50,100)	93.09	92.1	93.04
	CPU time (s)	0.052	0.012	0.087
Balance	Class 1 (288,337)	89.04	87.58	88.64
	Class 2 (49,576)	68.67	65.09	69.50
	Class 3 (288,337)	88.64	87.22	89.25
	CPU time (s)	0.21	0.30	0.69
Ionosphere	Class 1 (225,126)	91.61	91.18	91.83
	Class 2 (126,225)	69.22	68.79	69.39
	CPU time (s)	0.19	0.28	1.09
Wine	Class 1 (59,119)	97.50	94.52	94.64
	Class 2 (71,107)	81.03	80.12	80.42
	Class 3 (48,130)	97.21	93.32	96.06
	CPU time (s)	0.06	0.03	0.19

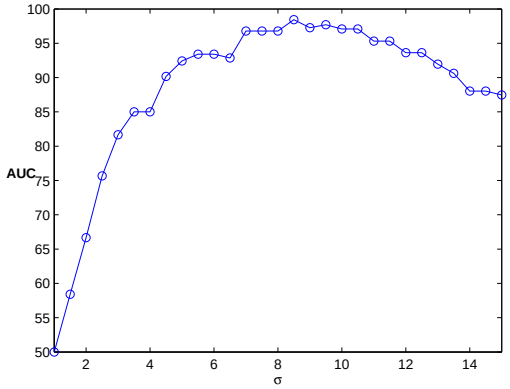


Figure 3.14: AUC obtained varying parameter σ while $C = 0.1$, $C_r = 0.00075$ and $q_i = 1e-5$.

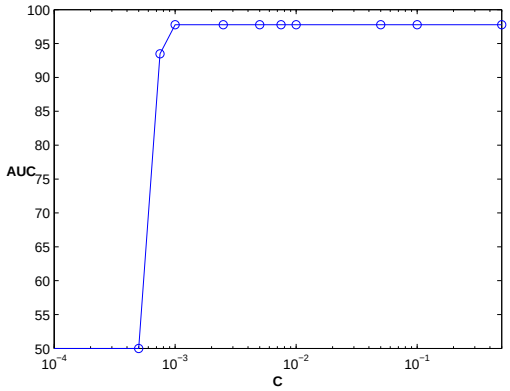


Figure 3.15: AUC obtained varying parameter C while $\sigma = 8.5$, $C_r = 0.00075$ and $q_i = 1e-5$.

combinations around the optimal which can give a good performance.

3.3.3.2 Anomaly detection scenarios

Artificial Dataset

In this section, in order to show the adequacy of the OSDAD algorithm for tracking scenarios, an artificial dataset illustrated by Yamanishi and Takeuchi in [250] was used.

This dataset is a data sequence generated according the following AR model:

$$x_t = 0.6x_{t-1} - 0.5x_{t-2} + \epsilon_t, \tag{3.112}$$

where ϵ_t is a Gaussian random variable with mean 0 and variance 1. This dataset consist of 10,000 samples and the previous mean of the data is shifted by $\Delta(x) = x$ at time $x \times 1,000$ $(x = 1, 2, ..., 9)$. The aim of this non-stationary dataset is to detect the shifts correctly with minimum false alarm rate. The detection is considered to be correct if an alarm is given within the 50 records after the true change point.
It should be noticed that the parameters of the OSDAD algorithm can be tuned in order to: (a) detect severe drifts maintaining a low false alarm rate or, (b) detect any intersected or low severity drift at the cost of a higher rate of false alarms.

In figure 3.18, it can be observed that all the drifts which are present in this dataset are abrupt and that its severity grows with time. All the changes which were considered as severe, and thus must be detected, are marked with a red mark. The parameters on this example were tuned in order to only detect abrupt severe changes while maintaining a low false alarm rate.
It can be observed in figure 3.18 how the OSDAD algorithm complies with this behaviour (change detection time points are illustrated in the figure) with absence of false alarms. In this experiment a window of 200 samples was used to stabilize the model at the beginning and every time the process was reinitialized due the detection of a significant change. Parameter values used for building the model were the following: $\sigma = 10, 2$, $C = 0, 07$, $C_r = 0, 07$ and $q_i = 1e - 6$. As we stated before, with this set of parameters the model is able to detect small changes in the data. Moreover, the values of the two parameters employed for the CUSUM chart were: ANOS = 600 and $p_1 = 5 \times p_0$. As p_0 was very small (most of samples fall inside the modeling sphere), we chose p_1 as a relative large multiple of p_0 [195].

3.3.4 Discussion an future work

In this section two classical problems are addressed from an on-line perspective: one class classification and anomaly detection. A new passive-agressive formulation for on-line one class classification is presented. From a practical perspective, the proposed formulation has the following advantages: (a) it is able to accurately fit the support

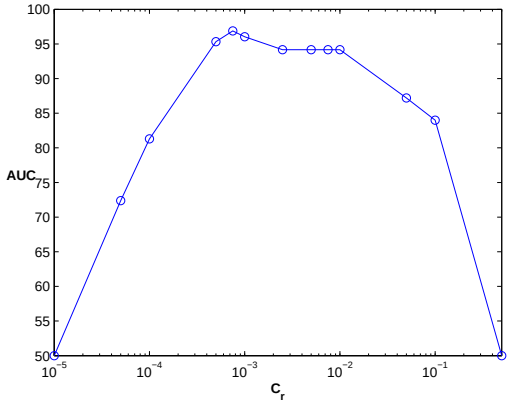


Figure 3.16: AUC obtained varying parameter C_r while $\sigma = 8.5$, $C = 0.1$ and $q_i = 1e-5$.

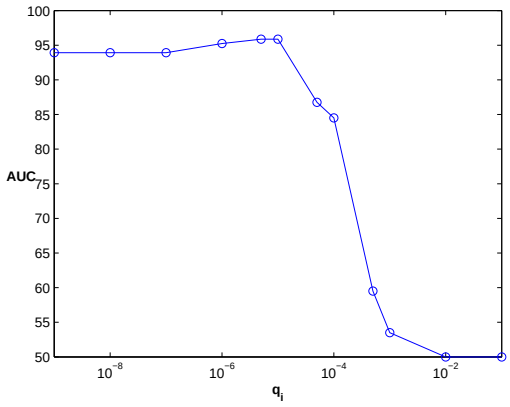


Figure 3.17: AUC obtained varying parameter q_i while $\sigma = 8.5$, $C = 0.1$, and $C_r = 0.00075$.

of normal data in an on-line fashion, (a) it is able to dynamically adapt to changes in the distribution of data, (c) it can be applied in a feature Hilbert space via kernel mapping and (d) it automatically controls the growth of the number of support vectors. Furthermore, this model is combined with a CUSUM chart of proportion of abnormal patterns giving the OSDAD algorithm, specially designed for stream anomaly detection. Experimental results on synthetic and real data sets confirm that the proposed model shows very good performance when compared to state of the art algorithms for one class classification and anomaly detection. The results of this section leave as future work the determination of the theoretical properties of this algorithm (convergence and classification error bounds), the design of a criteria for discarding not relevant past SVs

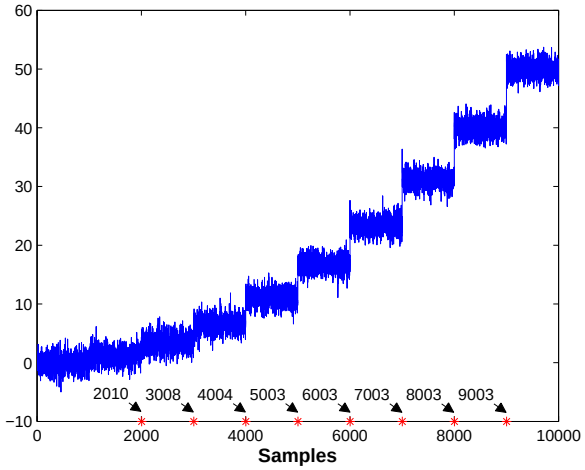


Figure 3.18: Change point detection of the proposed model for dataset by Yamanishi and Takeuchi.

and its application to problems using other feature spaces [214].

CHAPTER 4

On-line learning: incremental, non stationary and distributed scenarios

In this chapter we move on to the second main block of this thesis, on-line learning algorithms. Traditionally, most Machine Learning (ML) algorithms lay on assuming that the data being analyzed is drawn from a stationary distribution and is centrally available for training. However, in many real-life problems these assumptions are not realistic. Namely, the following eventualities can happen:

- **Non stationarity:** this happens when the joint probability distribution of the learning problem at hand $P(X, Y)$, where X are input data and Y represents the variable to predict, experiences changes between training phase and prediction phase when the learnt model is used. This can be due to the change of the conditions of the process underlying the learning problem (e.g., change of tastes in client behavior prediction, change on the physical state of the machine components in a machinery fault assessment application, etc.) or can be imposed by the way of tackling learning (e.g., changes between marginal distributions $P_{train}(X) \neq P_{test}(X)$ can be due to the way the sampling is done). Thus, in these cases what was learnt in the past may not be accurate or even significant for present data [192]. This problem can be found in the literature mainly under two names: covariate-shift [192] and concept-drift learning [243]. The former studies the case where the marginal probability distribution $P(X)$ of input data changes between training and prediction while the latter embraces more general changes in the learning problem.

- **Distributed data:** In this kind of scenario, data is distributed in several nodes. This eventuality can stem from different restrictions: (a) data is distributed in remote locations and it is not possible (or practical) to move data between nodes, and (b) learning from a large amount of data available which is so massive that

it becomes impossible to tackle learning using batch ML classical approaches in a central node. This problem has been also named in the literature as *horizontal partitioned learning* and *large scale learning* and has recently emerged as a subarea of machine learning. Currently, there is much interest in this area which continuously generates challenges and projects [207]. Recently, many classical ML methods have been studied from a distributed learning perspective [45][190][213] and even privacy preserving issues are being discussed in current research literature [7]. The latter has attracted much interest due to the expansion and exploitation of heavy enterprise databases characterized by large numbers of data points and/or high dimensions [176][227]. Incremental learning models, such as the one presented in this chapter, are promising tools in this area.

- **Incremental learning:** In this case, training data arrives continuously in a stream and a decision has to be made based on the information previously seen. Due to the continuous nature of the stream of data, learning can not be broken in training/testing phases. Training set in this case can be considered infinite in a sense (there's always new received data to add) but a decision has to be made also periodically only based on the data available so training has to be made continuously in order to have a suitable just-in-time model. In addition, this kind of scenario can be combined with non-stationarity and in many applications real-time response restrictions are imposed by the problem at hand. Thus, this kind of scenarios need to count on a model able to retrain efficiently when new data is available and which asymptotically converges to an optimal model.

Traditional batch and stationary vision of machine learning, although still being dominant and effective in many environments, has steadily yielded center stage to on-line, distributed and non-stationary learning due to their applicability in real life problems. Much research has been devoted to these areas, that had expanded ML applicability to real life scenarios. For example, and encompassing with the last part of this thesis, in a machinery fault assessment application it is desirable to anticipate a potential fault. Based on current data, an anomaly detection technique, such as any of the ones proposed in the previous chapter, can assess whether there is an anomalous behavior in the present time. But, if we can predict the value of some key parameters in the close future we could anticipate an anomalous situation. Since machine's condition changes along the time, a model able to deal with a on-line and non-stationary scenario would be worthwhile.

In this chapter we present an algorithm able to incrementally train a non linear model in all the aforementioned scenarios. The work in this chapter is underpinned by previous

work in [83] where a new convex objective function for one-layer neural networks has been presented which is able to exactly adjust (up to first order of a Taylor series) the weight matrix of a neural network without hidden layers and non linear output functions, provided that these have inverse and derivative. In that research it was pointed succinctly that the proposed approach opened the opportunity of learning incrementally (i.e., without the necessity of storing previous data). However, this incremental capacity involves the inversion of a $I \times I$ matrix for each new data point, leading in some situations to numerical instabilities and with a complexity of $O(M * I^2)$ being I the dimension of the input space and M a heavy constant. Afterwards, in [164] the incremental learning capability of the model described above [83], was explored and extended to concept drift scenarios, obtaining good results. This algorithm weights the importance of each data sample taking into account whether it is recent or not, giving exponentially more importance to recent data points. Although it demonstrates that it is an effective method for concept drift problems, it still has to solve a system of linear equations for each new data sample and needs to reset the weighting of the data samples periodically, thus leading to a cumbersome algorithm.
Although not considered in those previous works, both algorithms share its roots in the classical Recursive Least Squares (RLS) [99] algorithm, originally designed for solving least-squares problems. The RLS method is an efficient semi-second-order approach that leads to a faster convergence compared with the first-order models. It has been extensively studied and applied in the last decades to problems such as real time signal processing, control, adaptive filtering and noise cancellation, among others [91, 124]. The algorithm has the advantage of exhibiting extremely fast convergence in a few steps of learning. However, each iteration has a high computational complexity and potentially poor tracking performance when the system to be estimated changes [257]. In addition, it has the extra limitation of only considering linear output functions.
The algorithm proposed in this chapter presents the following main characteristics:

- It is able to train a single layer neural network with any non linear output function that complies with the aforementioned conditions.
- Since most of the output functions used in Artificial Neural Networks [29] comply with these requisites, it can be used as basic building block for more complex neural models.
- It generalizes previous models. Depending on the values given to the hyper-parameters and the selected output function, it includes as special cases: RLS [99] when a linear output function is fixed, the model in [83] when concept-drift capabilities are disabled and the one in [164].

- It is demonstrated the relation between its initialization scheme and regularization which can improve its generalization ability under ill-conditioned problems (high-dimensional, noise, ...).

- Due to its incremental nature, it is suitable for applications in non stationary, distributed and stream data scenarios.

Experimental results demonstrate that the proposed model can obtain accurate results and fast convergence in stable, non stationary and distributed scenarios. Its connections with machinery fault assessment will be explored in the last chapter of this work.

4.1 Background: Non linear single layer neural network learning algorithm

In this section we present the derivation of a previous algorithm that obtains the optimal weights of a single layer feedforward neural network with non linear output functions which need to have inverse and derivative. These restrictions come from the fact that we use a theorem demonstrated in [83], where an equivalent formulation for minimizing the error of a non linear single layer neural network was presented. This derivation follows a very different philosophy in comparison to previous algorithms since it *backpropagates* networks's desired output signal instead of the error committed. In figure 4.1 this process is depicted graphically. For each pattern $\mathbf{x}_s$, its desired output $\mathbf{d}_s$ is propagated backwards using the inverse of the output function for each neuron f_j^{-1} and we tackle the minimization of the error between the internal network value z_{js} and $f_j^{-1}(d_{js})$.

The theorem presented in that work is the first step in the derivation of the proposed algorithm and it states:

Theorem 1 *Let* $\mathbf{x} \in \mathbb{R}^{I+1}$ *be the input of a single-layer feedforward neural network,* $\mathbf{d}$*;* $\mathbf{y} \in \mathbb{R}^J$ *be the desired and real outputs,* $\mathbf{W} \in \mathbb{R}^{J \times (I+1)}$ *be the weight matrix, and* f *;* f^{-1}*;* $f' : \mathbb{R}^J \rightarrow \mathbb{R}^J$ *be the non linear function, its inverse and its derivative. Then, the*

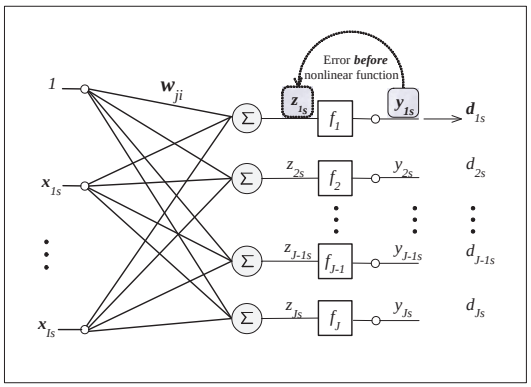


Figure 4.1: Architecture of a single-layer feedforward neural network.

minimization of the MSE between $\mathbf{d}$ *and* $\mathbf{y}$ *at the output of the non linearity*

$$\min_{\mathbf{W}} \frac{1}{S} \sum_{s=1}^{S} \|\mathbf{y}_s - \mathbf{d}_s\|^2 \tag{4.1}$$

where S *is the number of data points,* $\mathbf{y} = f(\mathbf{W}\mathbf{x})$*, is equivalent, up to first Taylor order, to minimizing the MSE before the non linearity, i.e., between* $\mathbf{z} = \mathbf{W}\mathbf{x}$ *and* $\bar{\mathbf{d}} = f^{-1}(\mathbf{d})$ *weighted according to the value of the derivative of the non linearity at the corresponding operating point. Mathematically, this property can be written as*

$$\min_{W} \; E[(\mathbf{d}-\mathbf{y})^T(\mathbf{d}-\mathbf{y})] \approx \min_{W} \; E[(f'(\bar{\mathbf{d}})\cdot\bar{\epsilon})^T(f'(\bar{\mathbf{d}})\cdot\bar{\epsilon})] \tag{4.2}$$

where $(\cdot)$ *denotes the element-wise Hadamard product of the vectors* $f'(\bar{\mathbf{d}})$ *and* $\bar{\epsilon} = \bar{\mathbf{d}} - \mathbf{z}$*.*

The details of the proof of this theorem can be consulted in [83]. In the following, we center our attention to only one neuron in order to avoid a cumbersome derivation. For solving a full layer of neurons as the one in figure 4.1, the process has to be applied identically for each neuron.
The optimal weight vector of a neural network, as the one in figure 4.1, using this theorem has to be a stationary point of the right hand side of equation (4.2). Thus, taking derivatives of this expression and equating to 0, we conclude that the optimal model $\mathbf{w}$ is the one which solves the following system of linear equations:

$$\mathbf{A}\mathbf{w} = \mathbf{b} \tag{4.3}$$

where $\mathbf{A}$ and $\mathbf{b}$ are defined as:

$$\begin{aligned} \mathbf{A} &= \sum_{t=1}^{S} \mathbf{x}_t \mathbf{x}_t^T f'^2(\bar{d}_t) \\ \mathbf{b} &= \sum_{t=1}^{S} \bar{d}_t \mathbf{x}_t f'^2(\bar{d}_t) \end{aligned} \tag{4.4}$$

With this model we have a way to tackle both batch and incremental learning scenarios, as we can save previous $\mathbf{A}_t$ and $\mathbf{b}_t$, and when new information is supplied up to time $t+p$ we can incrementally construct $\mathbf{A}_{t+p}$ and $\mathbf{b}_{t+p}$ using equation (4.4). Although mathematically correct, this approach has the following problem: it needs to solve a system of equations each time new information is provided and a much simple numerical algorithm to solve this incremental learning scenario is desirable.

4.1.1 Concept-drift learning algorithm

Taking advantage of the incremental learning capacity of the presented model, this can be extended to non stationary learning scenarios. In [164] an algorithm for tackling incremental learning with forgetting capacity based on this previous model was devised. It consisted on weighting in equation (4.4) each pattern exponentially, depending on how much time has passed since its inclusion on the learning process. This algorithm is equivalent to exponentially reducing the importance of the error committed for a past pattern x_t proportionally to the time that has passed since it appeared. If we combine this idea with the result of theorem 1, we arrive to the following error function to minimize

$$\min_{\mathbf{w}} \ (\bar{\mathbf{d}} - X^T\mathbf{w})^T F \Lambda (\bar{\mathbf{d}} - X^T\mathbf{w}) \tag{4.5}$$

where X is a matrix with data patterns $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_S\}$ as columns, $\bar{\mathbf{d}} \in \mathbb{R}^S$ (for a single neuron) complies with $\bar{\mathbf{d}} = f^{-1}(\mathbf{d})$, Λ is a diagonal matrix with diagonal elements $\Lambda_{ii} = \lambda^{S-i}$ for $i = 1, \ldots, S$ and F is a diagonal matrix with $F_{ii} = f'^2(\bar{d}_i)$. If we take derivatives with respect to $\mathbf{w}$ and equating the result to $\mathbf{0}$, we arrive to the following system of linear equations that solves the *time weighted optimal neural network*.

$$\begin{aligned} \mathbf{A}_S &= \sum_{t=1}^{S} \lambda^{S-t} \mathbf{x}_t \mathbf{x}_t^T f'^2(\bar{d}_t) \\ \mathbf{b}_S &= \sum_{t=1}^{S} \lambda^{S-t} \bar{d}_t \mathbf{x}_t f'^2(\bar{d}_t) \end{aligned} \tag{4.6}$$

The parameter λ controls the ability of the network to forget when the system under identification changes. It can be fixed in advance or it can be changed dynamically based on the error history of the network [107][117][146], controlling the length of the time window considered in order to adjust the weights.

Up to this moment, in every scenario (non stationary, incremental or distributed learning), we had to solve the system in (4.3) or (4.6) each time new information was received. This can lead to an inefficient and complex algorithm when incremental or non stationary learning scenarios are considered since in these cases the network has to be updated each time a new pattern is received.

4.2 Diminishing complexity and incrementing efficiency: proposed algorithm

In this section, we present and demonstrate two lemmas that obtain a much simpler and efficient algorithm when applied to the aforementioned scenarios. In next subsection, the new algorithm is presented and, in addition, a lemma that demonstrates the relation between its initialization and the regularization capacity of the model is detailed. Finally, in the last subsection the main differences and advantages of the proposed model compared to previous approaches are analyzed.
Algorithm 4 is able to solve the same model than the one presented in section 2, and its equivalence can be demonstrated by the following lemma:

Lemma 4.1: Algorithm 4 solves the optimal weights of a single layer neural network as the one depicted in figure 4.1 up to first order Taylor approximation.

Proof: Following theorem 1 proved in [83], the optimal weight vector of a single

layer neural network (see Figure 4.1) for a data set of inputs $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_S\}$ and desired outputs $\mathbf{d} = [d_1, d_2, \ldots, d_S]$ can be obtained solving the following system of linear equations:

$$\mathbf{A}_S \mathbf{w}_S = \mathbf{b}_S \tag{4.7}$$

where $\mathbf{A}_S$ and $\mathbf{b}_S$ are defined as:

$$\begin{aligned} \mathbf{A}_S &= \sum_{t=1}^{S} \lambda^{S-t} \mathbf{x}_t \mathbf{x}_t^T f'^2(\bar{d}_t) \\ \mathbf{b}_S &= \sum_{t=1}^{S} \lambda^{S-t} \bar{d}_t \mathbf{x}_t f'^2(\bar{d}_t) \end{aligned}$$

and this solution is given by

$$\mathbf{w}_S = \mathbf{A}_S^{-1} \mathbf{b}_S \tag{4.8}$$

If we unroll equation (4.24) for $\mathbf{A}_S$ and $\mathbf{b}_S$ we obtain:

$$\begin{aligned} \mathbf{A}_S &= \lambda \sum_{t=1}^{S-1} \lambda^{S-1-t} \mathbf{x}_t \mathbf{x}_t^T f'^2(\bar{d}_t) + \mathbf{x}_S \mathbf{x}_S^T f'^2(\bar{d}_S) \\ \mathbf{b}_S &= \lambda \sum_{t=1}^{S-1} \lambda^{S-1-t} \bar{d}_t \mathbf{x}_t f'^2(\bar{d}_t) + \mathbf{x}_S \bar{d}_S f'^2(\bar{d}_S) \end{aligned}$$

and using the Woodbury identity [90]:

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^T \tag{4.9}$$

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^T\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^T B \tag{4.10}$$

with $\mathbf{D} = 1$ in equation 4.10, $\mathbf{B}^{-1} = \lambda \mathbf{A}_{S-1}$ and $\mathbf{C} = f^{'}(\bar{d}_S)\mathbf{x}_S$ we have that

$$\mathbf{A}_S^{-1} = \lambda^{-1}\mathbf{A}_{S-1}^{-1} - \frac{\lambda^{-2} f'^2(\bar{d}_S)\mathbf{A}_{S-1}^{-1}\mathbf{x}_S\mathbf{x}_S^T\mathbf{A}_{S-1}^{-1}}{1 + \lambda^{-1}\mathbf{x}_S^T\mathbf{A}_{S-1}^{-1}\mathbf{x}_S f'^2(\bar{d}_S)} \tag{4.11}$$

if we rename

$$\begin{aligned} \mathbf{P}_S &= \mathbf{A}_S^{-1} \\ \mathbf{k}_S &= \frac{\lambda^{-1}\mathbf{P}_{S-1}\mathbf{x}_S}{1 + \lambda^{-1}\mathbf{x}_S^T\mathbf{P}_{S-1}\mathbf{x}_S f'^2(\bar{d}_S)} \end{aligned} \tag{4.12}$$

we have that

$$\mathbf{P}_S = \lambda^{-1}[\mathbf{P}_{S-1} - \mathbf{k}_S\mathbf{x}_S^T\mathbf{P}_{S-1} f'^2(\bar{d}_S)] \tag{4.13}$$

$$\mathbf{k}_S = \mathbf{P}_S\mathbf{x}_S \tag{4.14}$$

since

$$\mathbf{k}_S[1 + \lambda^{-1}\mathbf{x}_S^T\mathbf{P}_{S-1}\mathbf{x}_S f'^2(\bar{d_S})] = \lambda^{-1}\mathbf{P}_{S-1}\mathbf{x}_S \tag{4.15}$$

$$\mathbf{k}_S + \lambda^{-1}\mathbf{k}_S\mathbf{x}_S^T\mathbf{P}_{S-1}\mathbf{x}_S f'^2(\bar{d_S}) = \lambda^{-1}\mathbf{P}_{S-1}\mathbf{x}_S \tag{4.16}$$

which using (4.13) leads to

$$\mathbf{k}_S = \lambda^{-1}[\mathbf{P}_{S-1} - \mathbf{k}_S\mathbf{x}_S^T\mathbf{P}_{S-1}f'^2(\bar{d_S})]\mathbf{x}_S = \mathbf{P}_S\mathbf{x}_S \tag{4.17}$$

If we plug these results into the solution of the linear system in equation (4.8), we finally have that:

$$\begin{aligned}
\mathbf{w}_S &= \mathbf{A}_S^{-1}\mathbf{b}_S = \mathbf{P}_S[\lambda\mathbf{b}_{s-1} + \mathbf{x}_S\bar{d_S}f'^2(\bar{d_S})] \\
&= \mathbf{P}_S[\lambda\mathbf{A}_{S-1}\mathbf{w}_{S-1} + \mathbf{x}_S\bar{d_S}f'^2(\bar{d_S})] \\
&= \mathbf{P}_S[\lambda(\mathbf{A}_S - \mathbf{x}_S\mathbf{x}_S^T)\mathbf{w}_{S-1} + \mathbf{x}_S\bar{d_S}f'^2(\bar{d_S})] \\
&= \mathbf{w}_{S-1} - f'^2(\bar{d_S})\mathbf{P}_S\mathbf{x}_S\mathbf{x}_S^T\mathbf{w}_{S-1} + \mathbf{P}_S\mathbf{x}_S\bar{d_S}f'^2(\bar{d_S}) \\
&= \mathbf{w}_{S-1} - f'^2(\bar{d_S})\mathbf{P}_S\mathbf{x}_S[\bar{d_S} - \mathbf{x}_S^T\mathbf{w}_{S-1}] \\
&= \mathbf{w}_{S-1} + \mathbf{k}_s\alpha_S f'^2(\bar{d_S})
\end{aligned}$$

where

$$\alpha_S = \bar{d_S} - \mathbf{x}_S^T\mathbf{w}_{S-1} \tag{4.18}$$

These last two equations complete the algorithm 4 in conjunction with equations (4.12) and (4.13).

As it can be seen in the algorithm, we update the weight vector for each pattern through a vector $\mathbf{k}_t$ pondered by the error committed for that pattern and the derivative of the output function in $\bar{d_t}$. This vector $\mathbf{k}_t$ is proportional to a matrix $\mathbf{P}_{t-1}$ which represents $\mathbf{A}_{t-1}^{-1}$ (see appendix for details). In order to initialize the method, we have to give a value to $\mathbf{P}_0$, before any pattern is presented to the network, with the initialization term $\mathbf{P}_0 = \delta I$, where I represents the identity matrix. The meaning of the value of δ is analyzed in the next section.

4.2.1 Regularization property

It is a well known fact in statistics and machine learning that regularization schemes lead to models with a better generalization performance when ill-conditioned parameter

Algorithm 4: Non linear single layer neural network training

Input : Data set that comprises inputs $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_S]$ and desired outputs $\mathbf{d} = \{d_1, d_2, \ldots, d_S\}$, forgetting factor λ, initial value δ

Output : Optimal weight vector $\mathbf{w_S}$.

Initialize $\mathbf{P}_0 = \delta I$ and $\mathbf{w}_0 = 0$

for t =1, ..., S

$$\mathbf{k}_t = \frac{\lambda^{-1}\mathbf{P}_{t-1}\mathbf{x}_t}{1 + \lambda^{-1}\mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t f'^2(\bar{d}_t)}$$

$$\mathbf{P}_t = \lambda^{-1}[\mathbf{P}_{t-1} - \mathbf{k}_t\mathbf{x}_t^T\mathbf{P}_{t-1}f'^2(\bar{d}_t)]$$

$$\alpha_t = \bar{d}_t - \mathbf{x}_t^T\mathbf{w}_{t-1}$$

$$\mathbf{w}_t = \mathbf{w}_{t-1} + \mathbf{k}_t\alpha_t f'^2(\bar{d}_t)$$

end

estimation problems are faced [29]. One standard way of bring in regularization in linear models is by entering a penalty term into the error function that introduces in the training phase a bias towards more simple models with a better generalization ability [15]:

$$Error' = Error\ on\ data + Complexity\ penalty \tag{4.19}$$

For linear models fitted by least squares method, Tikhonov regularization [239] has demonstrated its effectiveness and has a bayesian interpretation. The Tikhonov regularization term has this form

$$Complexity\ penalty = \|\Gamma\mathbf{w}\|^2 \tag{4.20}$$

being $\mathbf{w}$ the vector of parameters of the linear model and $\Gamma = \mu I$ with I the identity matrix the most common choice. The choice for the initialization of $\mathbf{P}_0$ introduces indirectly a Tikhonov regularization term into the network error function as the following lemma states:

Lemma 4.2: The parameter δ in algorithm 4 is inversely proportional to the value μ in the following extended error function.

$$\min_{\mathbf{w}} \ (\bar{\mathbf{d}} - \mathbf{X}^T\mathbf{w})^T\mathbf{F}\mathbf{\Lambda}(\bar{\mathbf{d}} - \mathbf{X}^T\mathbf{w}) + \mu\mathbf{w}^T\mathbf{w} \tag{4.21}$$

where $\bar{\mathbf{d}} = f^{-1}(\mathbf{d})$, X is a matrix with data patterns $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_S\}$ as columns, $\mathbf{\Lambda}$ is a diagonal matrix with diagonal elements $\Lambda_{ii} = \lambda^{S-i}$ for $i = 1, \ldots, S$ and $\mathbf{F}$ is a

diagonal matrix with $F_{ii} = f'^2(\bar{d}_i)$

Proof: In order to find the minimum of expression (4.21), we take the derivative and equate it to $\mathbf{0}$. Thus we arrive to the following expression:

$$(\mu I + \mathbf{X}\mathbf{F}\mathbf{\Lambda}\mathbf{X}^T)\mathbf{w} = \mathbf{X}\mathbf{F}\mathbf{\Lambda}\bar{\mathbf{d}} \tag{4.22}$$

So, in order to obtain the optimal weights, we have to solve the following system:

$$\mathbf{A}_S\mathbf{w}_S = \mathbf{b}_S \tag{4.23}$$

where $\mathbf{A}_S$ and $\mathbf{b}_S$ are defined as:

$$\begin{aligned} \mathbf{A}_S &= \mu I + \sum_{t=1}^{S} \lambda^{S-t}\mathbf{x}_t\mathbf{x}_t^T f'^2(\bar{d}_t) \\ \mathbf{b}_S &= \sum_{t=1}^{S} \lambda^{S-t}\bar{d}_t\mathbf{x}_t f'^2(\bar{d}_t) \end{aligned}$$

Using these expressions for $\mathbf{A}_S$ and $\mathbf{b}_S$ we can use the derivation of the previous proof to derive algorithm 4. Specifically, in this case, we have an expression for $\mathbf{P}_0$,

$$\mathbf{P}_0 = \mathbf{A}_0^{-1} = (\mu I)^{-1} = \frac{1}{\mu} I \tag{4.24}$$

As we can observe, this expression corresponds to the initialization we have detailed in algorithm 4 with $\delta = \frac{1}{\mu}$.

4.2.2 Main differences and advantages

The results previously detailed in this section allow to derive an algorithm able to incrementally train a regularized single layer neural network and optionally includes the capacity of forgetting past information in presence of changes of the system under identification. Both parameters δ and λ control this behavior and can be tuned in order to obtain a quick response under changes in the model being learnt and to introduce a bias towards more simple models in noisy situations. The parameter λ takes values in the interval $(0, 1]$, and controls the ability of the network to *forget* past patterns. If it takes the value 1, this algorithm approximately obtains the same network that the one in [83]. Exact equivalence depends on the value of δ. The difference emerges from the fact that the work in [83] does not have the ability to introduce regularization into the

network. In the proposed algorithm, δ controls the importance of a Tikhonov regularization term introduced into the network error function. Both models are equivalent in the limit $\delta = \frac{1}{\mu} \to \infty$. It is important to remark that this algorithm is equivalent to the RLS algorithm when the output function is linear.

It can be also observed that the proposed algorithm is able to train a non linear model through simple matrix algebra operations, particularly by matrix-vector multiplications and vector summations, which makes it suitable for implementing single layer neural networks trained in embedded and real time systems without the need of a matrix algebra package. Since it carries all past information in $\mathbf{P}_t$ and $\mathbf{w}_t$, it is also suitable for incremental learning and horizontal partitioned distributed learning scenarios. In both scenarios, it is impossible to access the whole data set available for training, in the former case due to real time or storage restrictions, and in the latter due to data is distributed in remote nodes and it can not be collected up in a central node. In both cases, learning can be suspended and continued afterwards in a different remote node or future time thanks to the information carried by the aforementioned $\mathbf{P}_t$ and $\mathbf{w}_t$. This property is also very convenient for privacy preserving machine learning [7] since the learnt information is codified in this $\mathbf{P}_t$ and $\mathbf{w}_t$ making it possible to collaborate in the learning process without revealing actual data.

In [145], an extension of the RLS algorithm to non linear neural networks was presented. Although similar in philosophy to the present work, their derivation obtains a different algorithm through the linearization of the network's output function instead of *backpropagating* network's desired output signal. In the experimental section, we prove that the proposed derivation leads to an algorithm which obtains more accurate results and a faster convergence to the minimum in comparison to this previous work. In addition, when applied to non stationary environments, the proposed model obtains a more precise and faster system identification.

4.3 Experimental Results

In this section we study the following aspects of the proposed model: (a) its regularization capacity for some real data sets, (b) its convergence for a series of data sets in stable conditions, (c) its convergence and system change identification ability in non stationary learning problems, (d) its behavior in distributed environments and (e) the

interaction between forgetting factor and regularization.

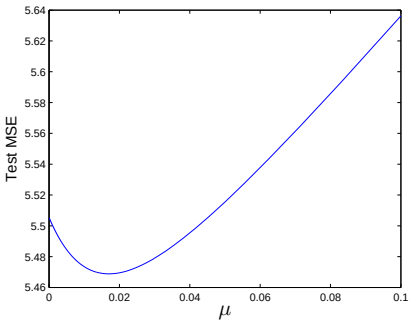
4.3.1 Regularization behavior

As we mentioned in previous sections, when a regularization term is introduced into the objective function of a model, the solution is biased towards more simple models with better generalization performance, avoiding overfitting to data. However, when this bias is too restrictive, it leads to too simple models which have not the ability to learn the underlying function. This behavior is tested experimentally in this section for both a classification and a regression problem. For the classification experiment, we divided into training and test subsets the ionosphere dataset [85]. For the regression experiment, an artificial problem was created taking an objective random weight vector of 50 dimensions and a one dimensional desired output to which we introduced random Gaussian noise. A reduced training set of 500 patterns was generated. In both situations algorithm 4 was used to train the network with the whole training partition and finally tested with a separate test set of 500 samples.

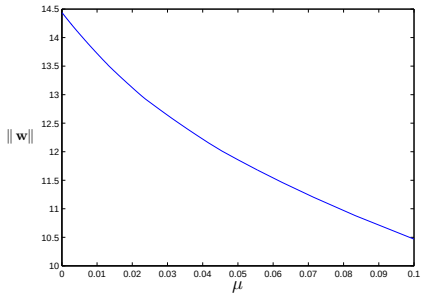
In figure 4.2 it can be observed how the initialization parameter δ controls the regularization behavior of the network. Figure 4.2(a) plots the test error and Figure 4.2(b) shows the sum (in absolute value) of the weights' network. Both graphics have in the abscissas axis the different values of the regularization term $\mu = \frac{1}{\delta}$. As it can be observed, it is experimentally confirmed what we demonstrated in section 4.2.1. In one hand, the sum of the final weights decreases as we increase the regularization term (initial δ). On the other hand, regarding the test error, there is an optimal point where the bias introduced by the regularization term into the training phase makes the model increase its generalization avoiding overfitting to training set and, from this point, this bias is too restrictive being impossible to properly learn the underlying desired function. Analogously, Figure 4.3 shows the test error and weight sum for the regression data set. As it can be seen, the behavior is the same as in the classification example.

4.3.2 Interaction between forgetting factor and regularization

In this section we explore how the forgetting factor λ and the regularization term μ interact in an ill-posed non stationary problem. On one hand we have proved how the

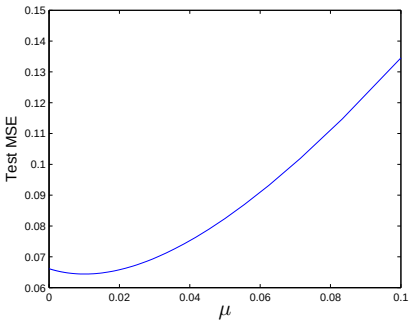


(a) Test Error for Ionosphere data set.

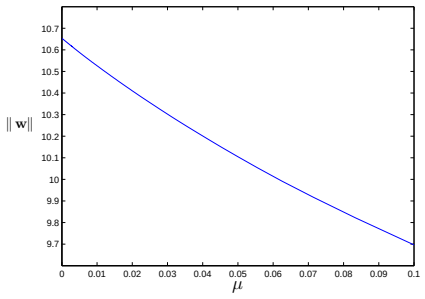


(b) Weight sum (in absolute value) of the final model.

Figure 4.2: Regularization behavior for Ionosphere data set.



(a) Test Error for artificial data set.



(b) Weight sum (in absolute value) of the final model.

Figure 4.3: Regularization behavior for linear data set.

forgetting factor controls the ability of the network to adapt to changes in the underlying function to be learnt. This is done such that if too much adaptation is allowed the network obtains an unstable accuracy in presence of noise. If on the contrary, it is too large, the model is too inflexible when it needs to adapt to changes in the data. On the other hand, the initialization introduced previously allows the network to control the complexity of the model, obtaining better performance when an adequate regularization level is introduced in the initialization of algorithm 4. Taking these two arguments into account, when we try to tackle an ill-posed non stationary learning problem, we should take into account both parameters since there should be an optimum combination of adaptation-regularization level for each specific problem depending on its particular properties. In order to test this, we generated a set of 30 dimensions random regression ill-posed problems with three changes of context in each one. For each dataset, three contexts of 30 training patterns and 100 test patterns were generated, having each context a random objective weight vector. Each problem had a single desired output with a low energy Gaussian noise added. Thus, in the generated problems it is necessary to control both the complexity of the model and its adaptation to changes in the function to be learnt. The mean error committed by the network during its adaptation is calculated for a test set. In figure 4.4 we show the mean test error obtained by the network for different combinations of regularization and adaptation levels. In the x-axis the value of μ is represented whilst in the y-axis the different values of λ can be found. As it can be derived from the obtained curves, it exists an optimum combination of μ and λ values that present an equilibrium between adaptation and regularization of the learnt model.

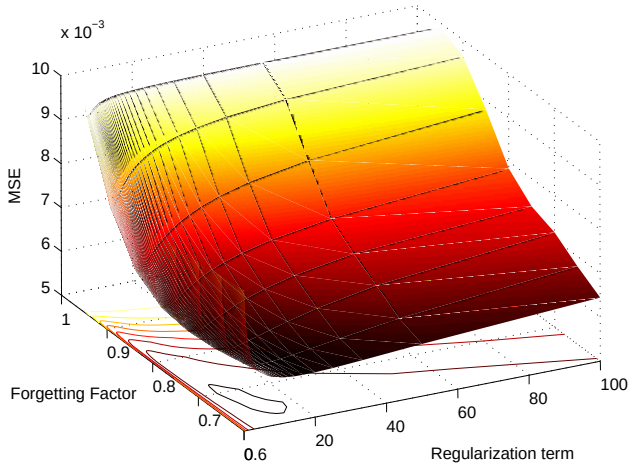


Figure 4.4: Test error for an artificial data set in function of the regularization term and the forgetting factor.

4.3.3 Stationary data sets comparison

In order to illustrate the performance of the proposed algorithm we apply it for the on-line prediction of twelve time series. Table 4.1 contains the characteristics of the data sets. The results were also compared with the RLS approach proposed in [145] to check the efficiency of the new algorithm. In all cases a initial value $\delta = 100$ was used for both algorithms.

For the comparative study 20 different simulations were performed for each data set using random permutations of the samples to construct the training and test sets. Figure 4.5 shows the average test MSE curves of the 20 simulations for the twelve time series studied respectively. As the learning is accomplished in an on-line fashion, the curves show the error for each iteration of the process until the whole data set is presented to the model (the dotted lines represent the variability between different simulations). As can be observed in this figure the proposed method obtains better results than the standard RLS, using the algorithm by Leung et al., achieving also a faster convergence speed.

Data set	Samples	Inputs
Artificial 1	20000	3
Artificial 2	20000	4
Artificial 3	20000	10
Annulus[1]	15000	6
Lorenz[1]	20000	6
Kobe Earthquake[2]	3000	6
Concrete Compressive Strength[3]	1030	9
Forest Fires[3]	517	12
Glassfurnace[4]	1247	3
pH neutralization process[4]	2001	2
Industrial dryer[4]	867	3
Industrial winding process[4]	2500	5

1. Available at Eric Weeksis Chaotic Time Series repository (http://www.physics.emory.edu/ weeks/research/tseries4.html)

2. Available at Time Series Data Library (http://robjhyndman.com/TSDL)

3. Available at UCI Machine Learning Repository (http://archive.ics.uci.edu/ml)

4. Available at DaISy: Database for the Identification of Systems (http://homes.esat.kuleuven.be/ smc/daisy)

Table 4.1: Data sets employed in the comparative study.

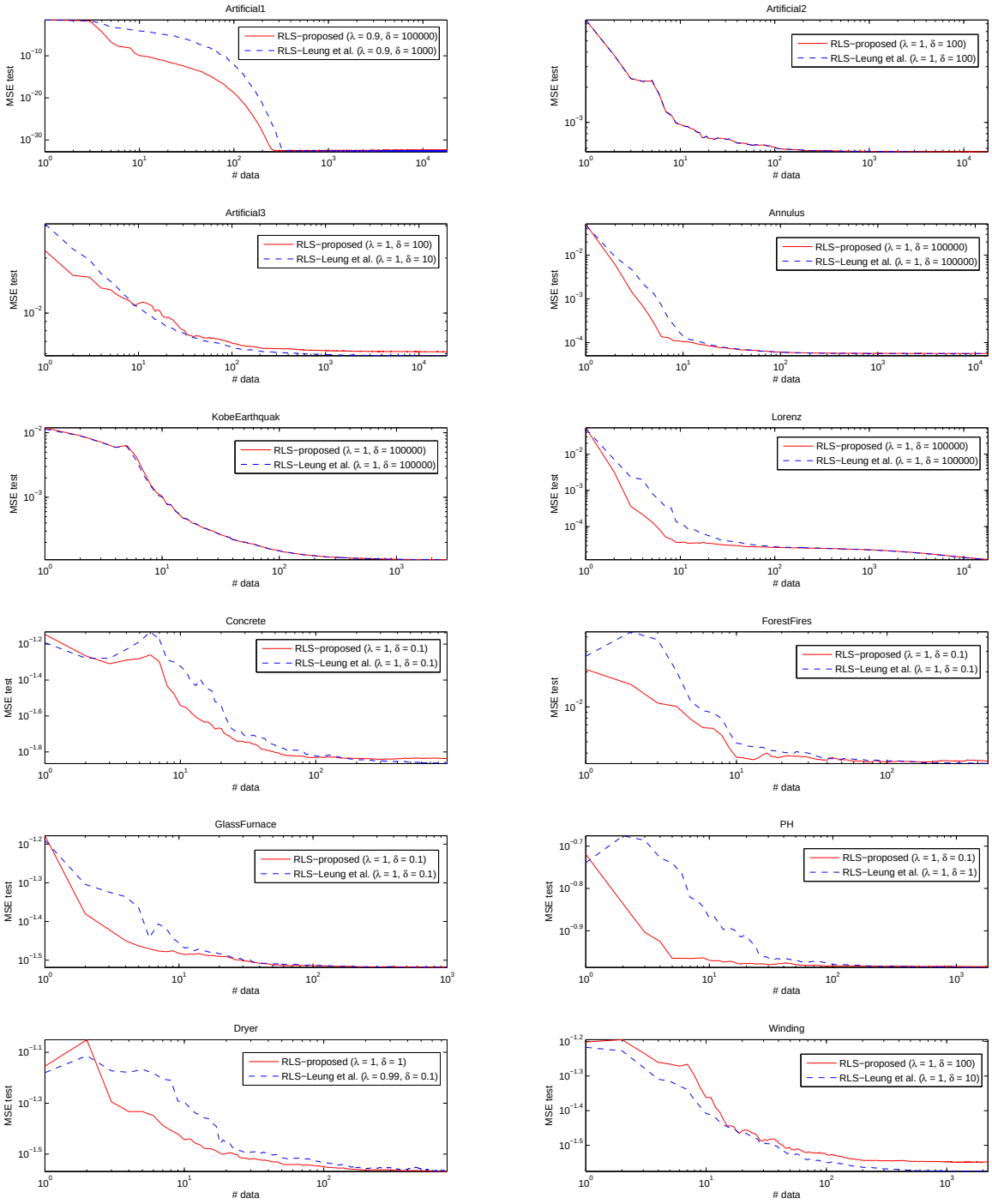


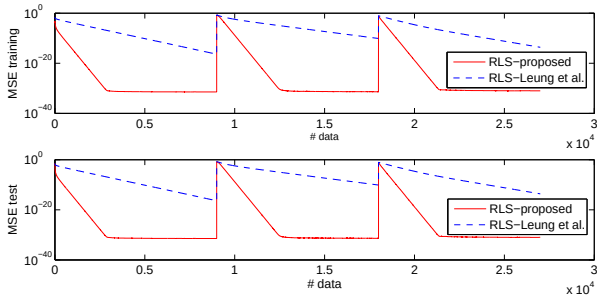
Figure 4.5: Comparative results for the twelve stationary data sets

4.3.4 Non stationary scenarios

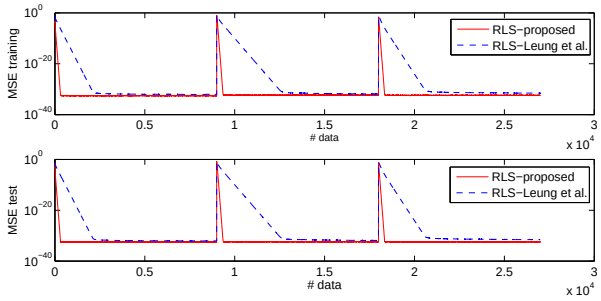
The performance of the proposed algorithm in non stationary contexts was checked both for artificial and chaotic series. In the first experiment we generated an artificial data set with 3 inputs and 1 output to be predicted using a linear combination of the inputs. This combination is changed two times during the learning process and thus three different mixtures of the inputs are obtained. The generated training set contains 27000 samples and, for each data point, a different random test set of 3000 data points was created using the associated parameters of the current context. As in previous experiments, 20 different simulations were performed and the averaged MSE curves were calculated. The RLS algorithm proposed by Leung et al. [145] was used again for comparison, however, in this case, and in order to be fair, it was modified to include a forgetting factor, not proposed by the authors in their original formulation. This term is mandatory in this experiment because we are managing non stationary data.

Figure 4.6 includes the results for these simulations. The training MSE curves for the on-line training process, and the test curves obtained using the specific test set for each sample, are shown. Each subfigure contains the results for a different forgetting factor (λ) in the algorithm. As can be observed, the proposed algorithm presents the fastest convergence speed to the optimum when a change is introduced in the function to be learnt. Specifically, in the most conservative scenario ($\lambda = 0.99$) the proposed method is able to recover its best performance, in the presence of change, in around 3300 data points while the algorithm by Leung et al. needs many more samples. In the most adaptive scenario showed in the figure ($\lambda = 0.50$) the proposed method requires approximately 60 samples to obtain the new optimal parameters whilst the other one achieves the same results but using around $[1000 - 1200]$ new samples. It is important to remark that as the forgetting factor decreases, and therefore a shorter window of relevant data is used, the variability of the error is higher.

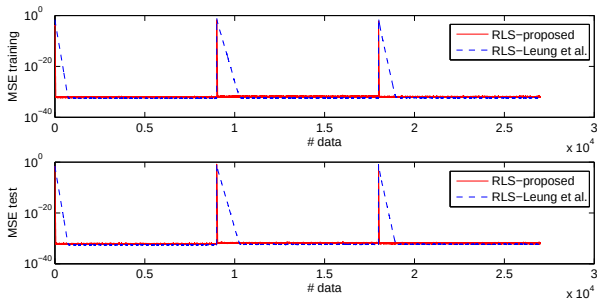
Moreover, the proposed model was tested for the prediction of the Mackey-Glass [157] and Lorenz [154] chaotic time series. In order to test the ability to adapt to changes in non stationary environments in complex scenarios, the data sets were generated in the following manner: (a) the parameters of the Mackey-Glass equations were changed each 900 data points in the following order $\tau = \{10, 15, 10, 14, 10, 13\}$ and the task was to predict the value 85 steps ahead using an embedding dimension of 8 values, and (b) the Rayleigh number ρ of the Lorenz equations was changed each 900 samples in the following order $\rho = \{13, 14, 20, 28\}$ and the task was to predict the next sample



(a) Results for a forgetting parameter $\lambda = 0.99$

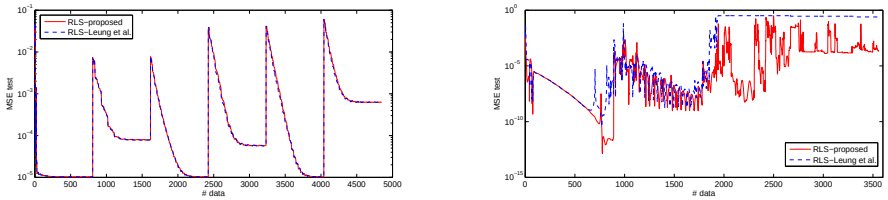


(b) Results for a forgetting parameter $\lambda = 0.90$



(c) Results for a forgetting parameter $\lambda = 0.50$

Figure 4.6: Results for the first non stationary data set.



(a) Results for the Mackey-Glass non stationary data set.

(b) Results for the Lorenz non stationary data set.

Figure 4.7: Results for the chaotic time series.

using an embedding dimension of 10 values. In both cases and for the two methods compared, λ was set to 0.99 (see figure 4.7). It can be observed how, also in complex identifications tasks, the proposed model presents a fast convergence to the optimal and a high accuracy.

4.3.5 Distributed environments

Due to the incremental nature of the proposed algorithm, learning the optimal weights of a single layer neural network in distributed environments (in which the patterns are scattered in several processing nodes and they can not be shared in a central node due to privacy or storage limitations) does not pose a further challenge. In order to fulfill a distributed learning task, the learning process has to be paused in a processing node and continued in other node after interchanging the pair $\{\mathbf{w}, \mathbf{P}\}$. Three situations can arise:

1. *Distributed batch learning*, in this scenario the nodes have all the available data from the beginning. Once all data has been processed in one node, the learning process continues in the next one until no node has any unprocessed data.

2. *Distributed on-line learning*, in this scenario the processing nodes receive available data on-line. In this case, if the current node has not any unprocessed data and other node receives new data, learning process is transferred to the latter node.

3. *Distributed concept drift learning*, in this case, in addition to the distributed nature of the learning, concept drift is present. So, whilst in the first two situations

the constant λ is set to 1, in this situation it should be set to a value in $(0, 1)$ in order to adapt to changes. This value should be shared among the processing nodes.

In this work we will deal with the distributed stairs problem. To the best of the author's knowledge, this problem has not been discussed elsewhere. Distributed stairs problem data is depicted in figure 4.8, where the numbers represent how data is distributed across the nodes. Each step of the stair is stored in a processing node and data from different stairs can not be shared between nodes. Although both the local problems and the global one are easy to solve, there is not any information in the individual nodes that guides their solutions to an optimal global solution. Only incremental methods that share global statistics, like the one proposed in this chapter can obtain an accurate global solution in an effective manner. Figures 4.8, 4.9 and 4.10 present the results of the proposed algorithm in the aforementioned three scenarios for the distributed stairs data set. The line in figure 4.8 represents the global classifier obtained in learning scenarios (1) and (2); note that they are superimposed due to the equivalence of the two situations for the case of the proposed model. In figure 4.9, learning scenario (3) is presented for this data set (the same distribution among the nodes as in figure 4.8 is used but each 2000 samples dataset is rotated) and figure 4.10 presents the classification error of the model along time for this situation with $\lambda = 0.99$. It can be observed how the presented model is able to quickly adapt in a concept drift scenario even in a distributed environment.

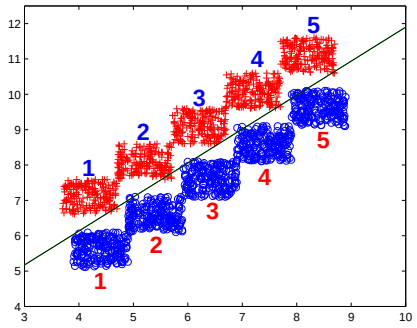


Figure 4.8: Results for the distributed stairs data set.

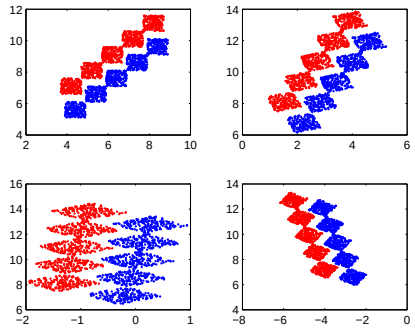


Figure 4.9: Distributed stairs data set with concept drift.

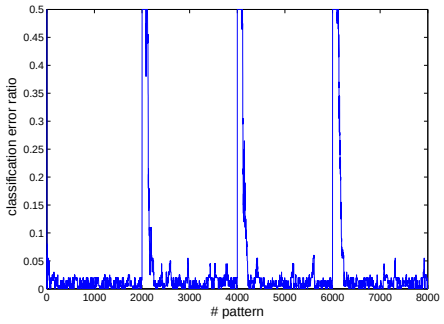


Figure 4.10: Classification error for the distributed stairs data set with concept drift.

4.4 Dicussion

In this chapter we have covered the second main block of this thesis work, on-line learning methods. Namely, a novel training algorithm for single layer neural networks with non linear output functions was presented. The derivation is theoretically underpinned by previously demonstrated results such as the one in [83]. It contains as special cases the works in [83], [164] and the classical Recursive Least Squares algorithm (RLS), which is interesting from a theoretical point of view. For practical purposes, it avoids the necessity of solving a system of linear equations each time a new network update is required, as in [83] and [164], thus making it a easier and more efficient algorithm for distributed and concept drift scenarios. For its application to large scale learning scenarios, the proposed model complies with the property of incremental learning, making

it a suitable algorithm for learning from batch data which need to be considered by parts. Finally, an initialization scheme has also been proposed, which is equivalent to introducing a Tikhonov regularization term in the training objective function, as it was demonstrated. This last property makes the proposed algorithm suitable for complex high-dimensional or noisy problems which are typically ill-posed. Experimental results show high accuracy and better performance compared with previous extensions of RLS algorithm to non linear output functions [145].

CHAPTER 5

Automatic fault detection in rotating machinery: theory and background

This chapter initiates the second main part of this work, application of Machine Learning, namely anomaly detection and on-line learning algorithms, to the problem of automatic fault detection in industrial systems. The proliferation of multi-system plants such as modern wind mill farms turns manual fault detection practices into a costly activity and pave the way for the application of the presented algorithms for its automation. We start by introducing in this chapter the main concepts and practices on industrial machinery fault detection and we leave for the last chapter the experimental results obtained with the proposed algorithms of this thesis. First, the classical process which has governed maintenance of rotating machinery in plants during the past decades is presented. Also a catalog of the existing diagnosis technologies is provided and the benefits of a vibration based process are remarked. Subsequently, basic concepts of mechanical vibrations are introduced in order to make the text self contained and further references are given. Afterwards the technologies involved in vibration data capture process and vibration signal processing are described. In addition, how fault detection in industrial environments can be tackled from an anomaly detection perspective is presented. This will be the framework that we adopt in the software described in the next chapter and in the experiments of chapter 7.

5.1 Rotating machinery fault detection based on vibration signatures

In every industrial plant where critical rotating machinery is present, it is necessary to comply with its maintenance in order to preserve its availability in optimal conditions

and guarantee plant's future production and security. The level of sophistication of this maintenance depends on the nature and criticalness of the machinery. Traditionally, the following schemes have been dominant in the industry:

- Corrective maintenance. In this case there are not maintenance actions until a breakdown occurs. Under this scheme, a machine works continuously until a breakdown makes necessary a complete stop for an overhaul. This process presents serious inconveniences since, due to the unexpectedness of the breakdown, the repair is far more costly due to the logistics and severity of the fault (which could be less severe if stopped earlier).
- Routine replacements and checking. This type of maintenance consist on a periodical replacement and checking of greases, oil, tension, hollows, etc. With this actions, it is possible to lengthen the lifetime of a machinery under normal conditions. However, if an internal fault in an inner stage is present, it will not be noticed by this kind of maintenance and will lead to an unplanned and costly stop of its functioning, similarly to the corrective maintenance. On the other hand, many replacements are made following the guidance of the manufactures. Depending on the working conditions, advised periods could be too short (leading to unnecessary costs) or too long (risking the integrity of the components).
- Preventive maintenance. This methodology may have been the most extended during the past two decades in many plants. In this scheme, each critical machinery is stopped an unmounted periodically for a complete inspection after a pre-specified period. All the possible defects are inspected and corrected. As in the previous case, if the periodicity is inaccurate (it must be taken into account that universal periods are very difficult to come up with) it can put at risk the assets or lead to high maintenance costs.

In summary, these maintenance policies are insecure, due to unnoticed inner faults which can lead to fatal breakdowns, and imprecise, due to the difficulties found to come up with accurate revision periodicities. In order to overcome these difficulties, the new trend, which has demonstrated effective if correctly implemented, is Predictive Maintenance. This kind of maintenance aims at reducing costs (due to unplanned cut downs in production and to repairs) and at improving security and availability. When generally defined, it is based on a continuous measure and analysis of health parameters of the machinery in order to detect deviations which can indicate the presence of a fault. A comprehensive predictive maintenance program utilizes a combination of the most

cost-effective tools to obtain the actual operating conditions of the equipment. On the basis of this collected data, maintenance schedules are selected. Making an analogy with medical care, the machine is the patient which may have an internal problem. If this problem exhibits a known symptom, its presence and evolution can be assessed by specific tests and suitable treatment is recommended in order to prevent further damages. In the same way, machines always exhibit known symptoms before breakdown, which can be controlled and treated before further damages produce a costly complete breakdown. The specific techniques depend on the type of plant equipment, the cost and other key parameters. Among all of them, the following are the most significant: (a) acoustic emission, mainly used to detect and locate cracks in structures and pipelines; (b) oil analysis, lubrication oil is analyzed in order to find the occurrence of certain microscopic particles in it that can be connected to the condition of bearings and gears; (c) particle analysis, in which the debris of the machinery is analyzed in order to find information of components such as gearboxes, hydraulic systems, etc.; (d) corrosion monitoring, in which ultrasonic thickness measurements are conducted in order to find any corrosive wear; (e) thermography, in which thermal images are analyzed in order to find overheating of components which can appear due to internal defects; (f) performance monitoring, in which the performance of the machinery in its duty is monitored in order to find downfalls which could indicate any problem; and (g) vibration monitoring, in which vibration pickup sensors are mounted in order to find deviations in the internal relative movements and impacts which are symptoms of many possible internal faults.

When devising and implanting a predictive maintenance program in a plant it has to be taken into account that it implies an increase in costs of personal, hardware and software. Many predictive maintenance initiatives have been stopped due to the fact that their benefits do not beat the costs compared with less sophisticated maintenance policies. Selecting an adequate technology and methodology is a matter of both detection accuracy and costs. Taking this into account, vibration analysis based predictive maintenance has emerged as the dominating technology due to the following facts: (a) it allows for a fully automatized capture, store and analysis process, (b) it can pinpoint much of the most common fault symptoms of rotating machinery and (c) there is an extensive literature already available which covers the analysis of many of the possible faults.

Ideally, predictive maintenance should follow the next curse of action: (a) an analytical model of the machine is devised and its set of fault explanatory parameters is extracted from that model, (b) this set of explanatory parameters together with their alert levels is compiled as a manual for the maintenance office and (c) the machine is periodically monitored in order to detect deviations which, if aligned with any fault parameter,

derive a machine overhaul. Unfortunately, when devising an automatic detection and diagnosis system, it has to be taken into account that although the "vibration symptoms" are qualitatively studied in the literature, the coupling of all the components in a complex machinery makes it impossible to generate accurate analytical models of each situation. Thus, there is a uncertainty in the meaning of some levels of vibration under different conditions and its significance from a diagnosis point of view. A highly trained team of human experts is always necessary in order to implant an effective predictive maintenance plan.
In the following sections, we will try to show how ML technologies can be introduced in modern vibration based predictive maintenance programs. ML techniques can save the highly valuable time of scarce vibration experts in order to expand their throughput in real demanding scenarios.

5.2 Mechanical vibration physics: an introduction

Since we will talk in the remainder of this thesis about fault detection based on vibration signatures it is convenient to clarify the basic notions of the physical phenomenon of vibration. Mechanical vibrations in rotating machinery are an example of forced vibration of a structure under an oscillatory force at a specific frequency. We will first explain the basic terminology and the model of free vibrations of the most basic mechanical system and incrementally construct the final model of forced vibrations similar to the ones that are built for mechanical systems. This section only gives a narrow view of the huge field of vibration analysis and is intended to understand the basic notions of the physical measurements on which fault assessment is based.

5.2.1 Basic concepts and terminology

Any mechanical system (machines, structures, etc.) when suffering an impact or the action of forces that vary with time, react modifying their equilibrium. When the perturbation presents some specific characteristics, its response is a vibration which can reach significant amplitudes even for weak perturbations. The amplitude of the response depends both on the variations of the perturbation along the time and on the mechanical characteristics of the system. Tangible examples of vibration are the ones of a diesel engine of a car, an earthquake on a building, etc. The effects of these

vibrations range from breakages to just annoyance. In some cases, such as the vibration of mechanical components, the maximum peak of a vibration may not cause a fault, but if the phenomenon continues along the time, it produces a fatigue which can cause a major fault eventually.
Generally speaking, any system with mass and which is able to be deformed can vibrate. Virtually all machines are subject to vibration of any kind and its design has to consider its vibrational behavior in order to avoid a vibration excess. This can be tacked through the modification of the applied forces and movements or modifying their physics.
Vibration can be defined as the variation of state (position, velocity and acceleration) of a system along the time with respect to a reference. These changes can be classified in several ways depending on the aspect considered:

- Depending on the type of movement. A vibration is periodic when the changes sequence repeats along the time. Periodic vibrations are characterizez by the period (time T between repetition of the change sequence) and frequency (number of cycles of the change sequence per time unit which is usually calculated as the inverse of the period $F = \frac{1}{T}$ and is expressed in Herzs). Aperiodic vibrations do not repeat a sequence at time intervals. They can be classified in deterministic, when we can predict its state in the future, and random, which can only be studied statistically.

- Depending on the evolution of the movement. Vibration can be: (a) damped when the maximal displacements diminish, (b) constant, when the displacements remain constant along the time or (c) amplified, when the maximal displacements increase.

- Depending on the cause of the vibration. They can be free vibrations when there are not external forces that produce the vibrations and they were just displaced from their equilibrium position. On the other hand they can be forced, when there is an external force applied during the vibration.

- Depending on the duration. Vibrations can be transitory, if their amplitude reduces with time until their complete disappearance, or permanent, which stay stable along the time. Forced vibrations are stable and remain until the excitement forces disappear.

- Considering their nature. Vibrations can be linear if the equation of the movement can be described through a linear model and non-linear otherwise. Being precise, all mechanical systems have non-linear vibrations. However, the deviation from

linearity is very small in many cases and, in practice, almost every system is studied using linear models.

- Considering the number of degrees of freedom. The number of degrees of freedom is defined as the number of parameters which are necessary to describe the configuration of a system at any time. If the number of degrees of freedom is finite then the system is discrete, otherwise it is defined as continuous. As in the previous case, almost all mechanical systems are continuous because they possess an infinite number of points for which we have to specify their configuration. In practice, they have to be treated as discrete, defining a mesh of representative points in order to be able to study their vibrational behavior.

Vibration of any mechanical system is mainly governed by four basic physical concepts:

- Mass: Is the property of the solids which causes inertial forces. This forces oppose to the changes of velocity of a solid. Thus, the inertial force is defined by the expression:

$$F_I = -ma \tag{5.1}$$

  where m is the mass of the particle and a its acceleration with respect to a reference.

- Stiffness: Is the characteristic of a solid which opposes to its deformation with a force or momentum. In the case of elastic systems, the energy absorbed by the deformation is recovered afterwards when the deformation force disappears. This reaction opposes to the previously produced deformation. The most simple example of this property is a spring. In this kind of system, the force needed to produce an increment of length δ varies linearly with the increment:

$$F = k\delta \tag{5.2}$$

  where k is called stiffness coefficient or simply stiffness. Although virtually any system has a non-linear stiffness, almost any studied system can be approximated by a linear equation like this one.

- Damping: It is defined as the process of energy dissipation while the vibration is present. If the vibration is only due to an initial perturbation, damping produces a progressive reduction of the amplitude until a final disappearance (we will see this in detail in the next section). If vibration is forced, damping reduces

the amplitude of the perturbation produced in the system. Damping can originate from diverse causes like internal friction of the materials or between moving parts (Coulomb damping), acoustic waves, the introduction of a fluid into a case through small holes like in the shock-absorvers of the cars (fluid damping). Generally, various kinds of damping are present in a system, although there is always one which is predominant

- Deformation forces: In order for a vibration to be present it is necessary an external perturbation of the equilibrium. This perturbation can present different forms and be classified into the following different kinds:

  - Initial shock. Is a modification of the position or the velocity of a system with respect to a reference equilibrium. The sudden end of the deformation allows the system to steadily recover its initial static position. System inertia prevents the system to recover immediately its initial equilibrium and a vibration phenomena appears. The example of this kind of perturbation is an impact in an object like a table, which vibrates until it recovers its initial position.

  - Short duration forces. These are characterized by a sudden initial impact followed by a slow posterior variation. Due to the initial shock, inertia of the system prevents its deformation to follow exactly the force and a transitory vibration appears. Damping produces that this vibration disappears eventually and the system finally follows the force. An example of this kind can be to put an object on a weighting scale.

  - Periodic forces. In this case the perturbation has a periodic component. The amplitude of the vibration depends on the amplitude and the frequency of the applied force and the mass, stiffness and damping of the system. Rotating mechanical systems explained in this chapter and in chapter 7 are an example of this kind of systems.

  - Irregular aperiodic forces. In this case the forces do not disappear after some time but they do not have a periodic component. This kind of variations can only be studied statistically. Example of this kind of perturbation are waves against a ship or wind on a building.

5.2.2 Fundamental equations of vibration

In this section we give a short introduction to the fundamental equations of vibration in a mechanical system which will help to understand the analysis of chapter 7. The equations that govern the vibration of any mechanical system are established through the same fundamental principles as the ones presented hereunder through discretization and linealization of the system. The most simple discrete systems which can be studied are those with only 1 Degree of Freedom (DOF). This kind of system is depicted in figure 5.1 and it is completely described by the position of the solid x. If we apply a force $f(t)$ on the solid with mass m in the positive direction on x, the basic equation of this system can be established through D'Alembert principle, introducing inertial forces:

$$mx^{''}(t) + cx^{'}(t) + kx(t) = f(t) \tag{5.3}$$

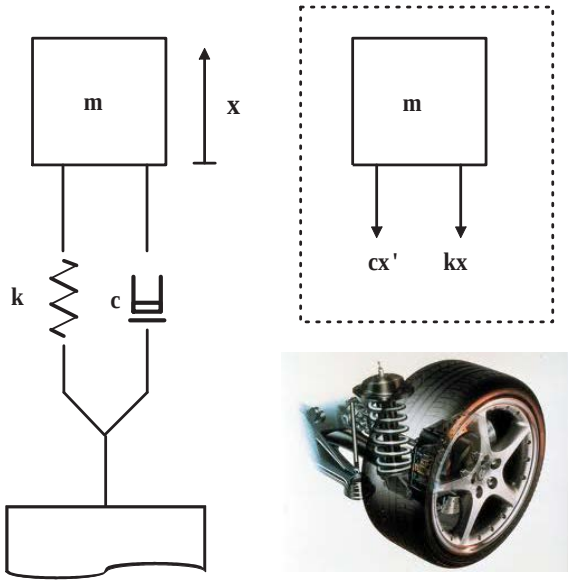


Figure 5.1: 1 Degree of Freedom System. An example of this kind of system are shock-absorbers

where m is the mass, k is the stiffness coefficient of the system and c is the damping factor, $x^{''}$ the acceleration of the particle/mass, $x^{'}$ its velocity and x its displacement. When we have a free vibration, we are in a situation where only an initial shock $x_0 = x(t_0)$, $x_0^{'} = x^{'}(t_0)$ is present and external action on the system is null $f(t) = 0$. In this case we have a linear ordinary differential equation and the equation of the vibration can be completely solved analytically. The expression of the movement has

the form

$$x(t) = Ce^{st} \tag{5.4}$$

where C is a constant to be determined from initial conditions and s has the form

$$s = -\frac{c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \frac{k}{m}} \tag{5.5}$$

Depending on the physical properties of the system, vibrations can be:

- Free vibration without damping. In this case $c = 0$ and equation for s can be reduced to

  $$s = \pm\sqrt{-w^2} = \pm iw \tag{5.6}$$

  where i is the imaginary constant and $w^2 = \frac{k}{m}$. Thus, the equation of the vibration in this case is a pure oscillatory one governed by the equation

  $$x(t) = Acos(wt) + Bsen(wt) \tag{5.7}$$

  where constants A and B can be determined by initial conditions.

  $$x(t) = x_0 cos(wt) + \frac{x_0^{'}}{w} sen(wt) \tag{5.8}$$

  The conclusion is that, without damping, vibration is an harmonic function with frequency $w = \frac{k}{m}$.

- Free vibration with damping. In this case $c \neq 0$ and the root for the equation of s can be real or complex conjugate. The constant $\xi = \frac{c}{2mw}$ is called critic damping and the expression for s can be transformed as follows

  $$s = -\xi w \pm \sqrt{\xi^2 w^2 - w^2} = -\xi w \pm w\sqrt{\xi^2 - 1} \tag{5.9}$$

  Three cases are possible:

  - Critic damping ($\xi = 1$). In this case $s = -w$ and the vibration has the form

    $$x(t) = (c_1 + c_2 t)e^{-wt} \tag{5.10}$$

    where c_1 and c_2 are constants to be calculated from the initial conditions. Thus, in this case there is not an oscillatory vibration and it has not interest for machine dynamics.

– Supercritic damping ($\xi^2 > 1$). In this case we can call $\bar{w} = w\sqrt{\xi^2 - 1}$ and the vibration has the form

$$x(t)e^{-\xi wt}(Acosh(\bar{w}t) + Bsenh(\bar{w}t)) \tag{5.11}$$

which is not oscillatory and, again, has not interest from a machine dynamics perspective.

– Subcritic damping ($\xi^2 < 1$). In this case $s = -\xi w \pm iw\sqrt{1 - \xi^2}$, we call $w_D = w\sqrt{1 - \xi^2}$ and vibration has the form

$$x(t) = e^{-\xi wt}Xcos(w_D t - \rho) \tag{5.12}$$

where X and ρ are determined from the initial conditions. The final expression has the form

$$x(t) = \sqrt{x_0^2 + \left(\frac{x_0^{'} + \xi w x_0}{w_D}\right)^2}e^{-\xi wt}cos\left(w_D t - arctg\left(\frac{x_0^{'} + \xi w x_0}{x_0 w_D}\right)\right) \tag{5.13}$$

This is the case which has interest from a machine dynamics perspective. As it can be observed in the expression, vibration in this case is a combination of a oscillatory movement and a damping component which reduces the amplitude proportionally to time (see an example in figure 5.2). The frequency w_D is called the natural frequency of the system and as it can be observed is completely determined by the mass, damping and stiffness factor of the component at hand.

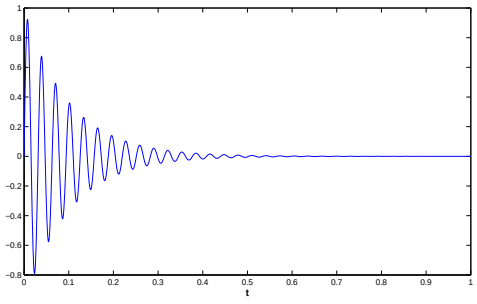


Figure 5.2: Damped Vibration example

The subcritic damping vibration is directly related to the vibrations of a defective bearing in rotating machinery which will be studied in chapter 7. In this case, the defect

produces periodically an impact in the machine which subsequently produces a subcritic damping vibration of the whole housing of the bearing. This vibration is captured by the transducers to be explained in the following sections and a peak vibration should appear at a frequency close to w_D. Unfortunately, this mathematical model is only an approximation of these continuous systems. The parameters c, k and m can only be known approximately (usually only an approximation of w_D can be known) and the vibration is affected by other components of a machinery. These difficulties makes these expressions a good qualitative model of the phenomenon but, in quantitatively terms, it is only a raw approximation of the actual vibration phenomenon.
In rotating machinery, the forces affecting the internal components vary harmonically with time. Thus, this perturbation can be expressed as a Fourier series of harmonic functions:

$$f(t) = \sum_{j=0}^{N} f_j(cos(w_j t) + i\ sin(w_j t)) \tag{5.14}$$

Physically, a complex force has not a real sense, but is a useful compact mathematical expression. Since this equation should be solved for the real and imaginary part, if the perturbation varies as a sinusoid we should consider the imaginary part of the final expression and the real part if, otherwise, the perturbation follows a cosine form. If we concede that the perturbation has a main pure oscillatory form

$$f(t) = f_0(cos(\hat{w}t) + i\ sin(\hat{w}t)) \tag{5.15}$$

we have that w_D is the natural frequency of the system, $\hat{w}$ the frequency of the perturbation and we call

$$\beta = \frac{\hat{w}}{w_D} \tag{5.16}$$

The final expression of the response of a 1 Degree of Freedom (DOF) system with an oscillatory perturbation is

$$x(t) = Xe^{-\xi wt}cos(w_D t - \rho) + \frac{f_0}{k}\frac{1}{1-\beta^2+2\xi\beta i}e^{i\hat{w}t} \tag{5.17}$$

where constants X and ρ are to be determined by the initial conditions and we consider the imaginary part of the second summand if the perturbation is sinusoidal or the real part otherwise. As we can observe, the final form of the vibration has two very different components (see an example in figure 5.3):

- Transitory vibration. Which has an oscillatory movement at the natural frequency and disappears after some time. It corresponds to the first component of the aforementioned expression.

- Stationary response. Which is present while the perturbation is active and corresponds with the second component of the final expression.

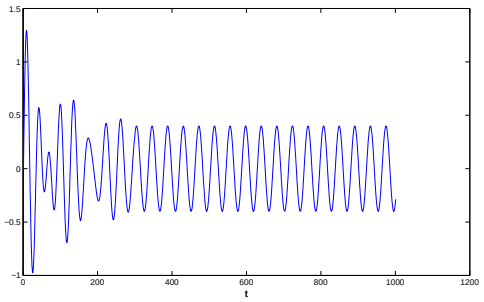


Figure 5.3: Forced vibration example (combination of a transitory and a stationary component)

The second component has an important term called transference function of the system

$$H(\hat{w}) = \frac{\frac{1}{k}}{1 - \beta^2 + 2\xi\beta i} \tag{5.18}$$

This function governs the response of a system. If a force with the expression

$$f(t) = f_0 e^{i\hat{w}t} \tag{5.19}$$

is applied, then the system produces a final stable vibration with the form

$$x(t) = f_0 H(\hat{w}) e^{i\hat{w}t} \tag{5.20}$$

This function has two effects

- $\Omega = arctg\frac{2\xi\beta}{(1-\beta^2)}$ is the delay that the system introduces in the effect of the perturbation.
- $X = \frac{f_0}{k}\frac{1}{\sqrt{(1-\beta^2)^2+(2\xi\beta)^2}}$ is the amplitude modulation introduced in the perturbation.

The most important component is the second one since, as it can be observed in the expression, even a small amplitude perturbation can produce a huge vibration depending on the β coefficient. This means, encompassing with the vibration of bearings studied in the last chapter, that a small defect in a component can produce a fatal

effect depending on the position and the rotating frequency of the machinery (in the extreme case of resonance, the vibration can become infinite, producing the breakage of the solid). Due to this effect, predictive correct design and predictive maintenance are key factors for a machinery asset to have a long life. Figure 5.4 depicts a typical transference curve for a mechanical component.

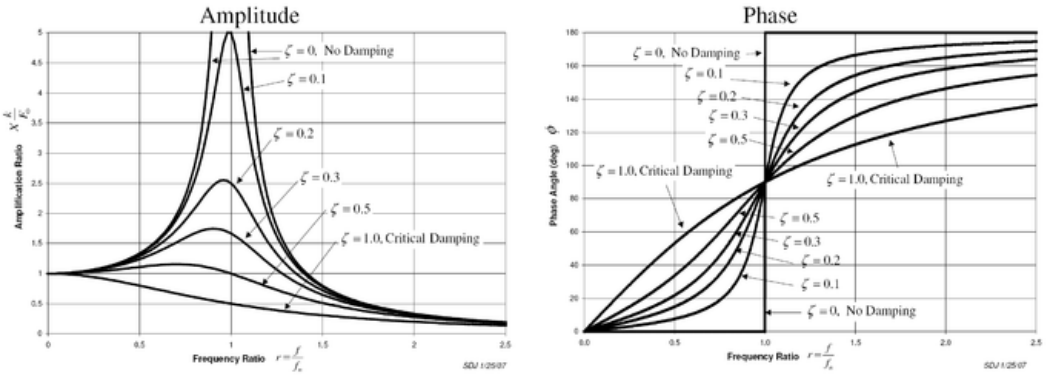


Figure 5.4: Vibration transference of a mechanical system

Of course, real systems are not of the 1 DOF type, albeit the fundamental concepts presented in this brief introduction are the basis of the vibration models of many rotational machinery systems and, specifically, the components that will be studied in the next sections. If we are to study a more general system, we will end in a system of linear ordinary differential equations and the natural frequencies at which each component will vibrate are called natural modes of vibrations. Just to give a notion of the mathematical form of a n **degrees of freedom** system consider the one in figure 5.5

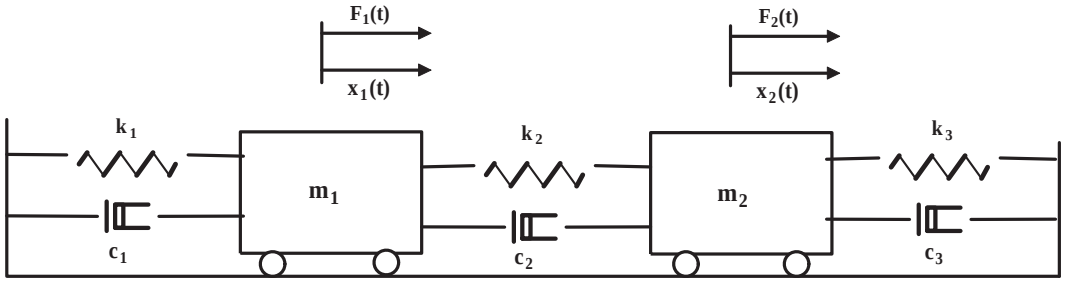


Figure 5.5: Scheme of a 2 degrees of freedom system.

The equations that govern the vibration of this system when affected by a pertur-

bation are:

$$m_1 x_1^{''} + k_1 x_1 + c_1 x_1^{'} - k_2(x_2 - x_1) - c_2(x_2^{'} - x_1^{'}) = F_1(t) \quad (5.21)$$

$$m_2 x_2^{''} + k_3 x_2 + c_3 x_2^{'} + k_2(x_2 - x_1) + c_2(x_2^{'} - x_1^{'}) = F_2(t) \quad (5.22)$$

which can be put into matrix form

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} x_1^{''} \\ x_2^{''} \end{bmatrix} + \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 + c_3 \end{bmatrix} \begin{bmatrix} x_1^{'} \\ x_2^{'} \end{bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} F_1(t) \\ F_2(t) \end{bmatrix}$$

where the first matrix is the inertial matrix, the second one is the damping matrix and the third one the stiffness matrix. In the general case, the whole system ends in a system of linear ordinary differential equations of order n in which the couplings of all the components are codified in the three matrices of the system. The analytical solution to this system of linear equations can be obtained in some cases. In many occasions, the system can be simulated in order to predict an approximation of the vibration of the whole system.

5.2.3 Physical models of vibrations of a rotating machinery

After reading the concepts of the last section it could be argued that all the vibration modes of a piece of machinery can be studied analytically. In a real scenario, this is not always the case and the models constructed are limited in the number of degrees of freedom considered and their utility is restricted to control the vibration limits of the machine in their design. As an example of this, in big structures like ships or planes, their design is conceived in order to avoid resonance frequencies due to the previously explained effects. To give an illustration of the complexity of such models of a real system, a 34 degrees of freedom model of a bearing fault simulator presented in [205] is reproduced hereunder. Figures 5.6 and 5.7 depict respectively the actual test rig and its 34-DOF dynamic model. This model will be used in chapter 7 to simulate bearing faults and illustrate the detection capabilities of the algorithms presented in this thesis. It is depicted here in order to illustrate the complexity of a complete model of a real system. This complexity makes the complete analytical complete study of many systems impractical.

From the predictive maintenance perspective, the analytical study of the vibration phenomenon carried out through the principles briefly presented in this section allows to build qualitative models of the vibration behavior like the ones presented in chapter 7

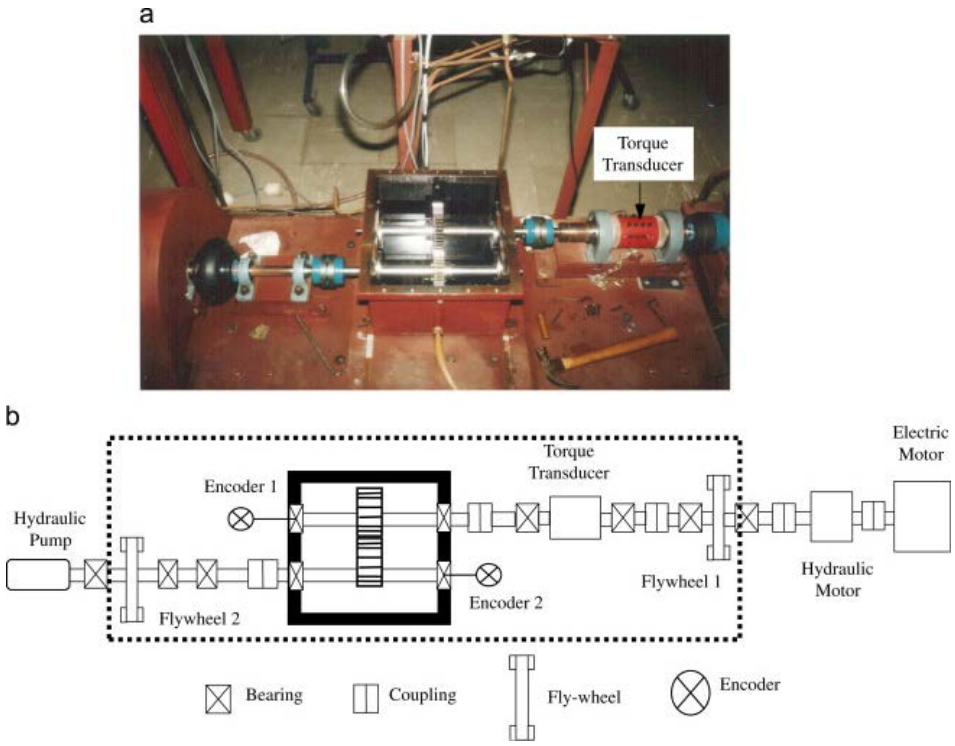


Figure 5.6: University of South Wales' fault test rig photo and scheme (courtesy of the authors)

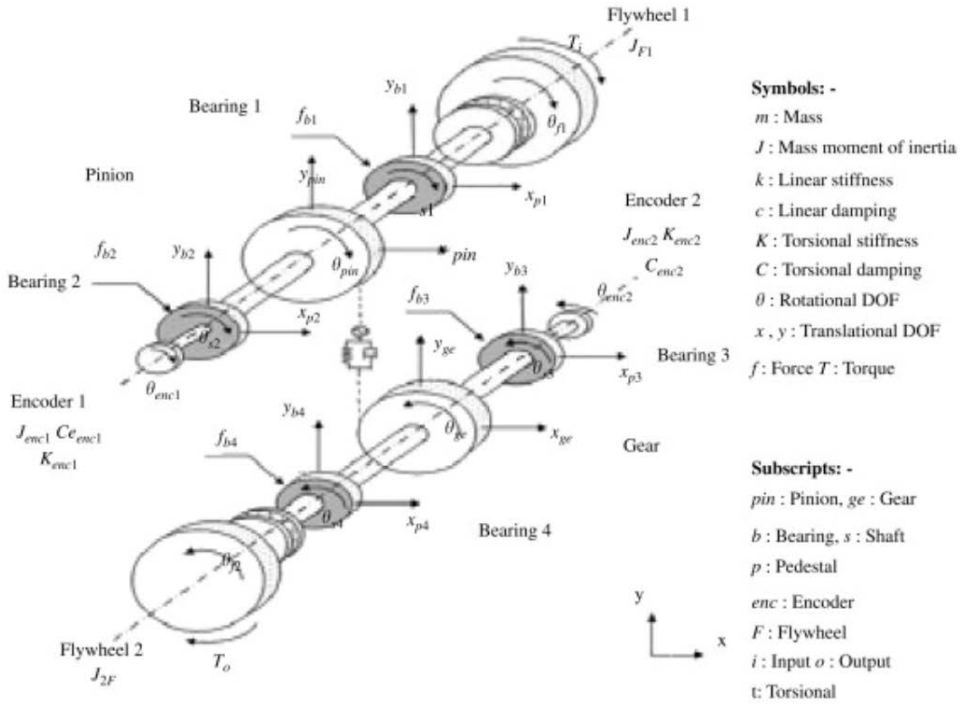


Figure 5.7: 34-DOF modeling of the gearbox (courtesy of the authors)

for bearings. Unfortunately, many assumptions have to be made during this study and many parameters such as stiffness, damping and couplings between components can not be completely determined. This leads us to the necessity of constructing *ad hoc* vibration models for each machine in order to be able to detect faults in a real situation. In chapter 7 we will see how Machine Learning (ML) techniques can be worthwhile for this task. Nevertheless, there's a synergy between analytical study of mechanical systems and ML fault detectors since the former helps in the design of ML automatic systems in decisions such as feature selection and algorithm design.

5.3 Mechanical vibration technologies

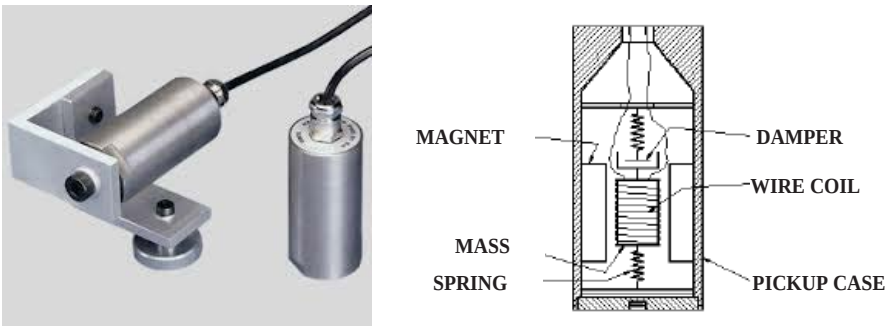
The topics discussed in the previous section were theoretical, introducing the basics of mechanical vibrations. In this section we take the first step towards practical vibration analysis. We start with the description of the most common transducers that can be found in a real scenario.

5.3.1 Vibration transducers: characteristics and installation

As in many other areas such as medicine, transports, etc. in order to measure rotating machinery vibrations a transducer, able to transform the vibration phenomenon in other type of energy, is necessary. In this section we briefly describe the three types of transducers which are standard in the field: velocity pickups or velocimeters, accelerometers and proximity probes. Each component is specialized on capturing vibration velocity, acceleration and displacement and each one has its own advantages and disadvantages which will also be commented. In the experimental section, accelerometers will be used due to the necessity of capturing a broad frequency band.

Velocity pickups or velocimeters

Velocity pickup is a very common transducer for monitoring the vibration of rotating machinery. This type of vibration transducer installs easily on most analyzers, and is rather inexpensive compared to other sensors. For these reasons, the velocity transducer is ideal for general purpose machine monitoring applications. Velocity pickups have been used as vibration transducers on rotating machines for a very long time, and these are still utilized for a variety of applications today. Velocity pickups are available in many different physical configurations and output sensitivities. Wire is wound on a hollow bobbin to form the wire coil. Some times the wire coil is counter wound (wound in one direction and then in the opposite direction) to counteract external electrical fields. The bobbin is supported by thin, flat springs to position it accurately in the permanent magnet's field. When a coil of wire is moved through a magnetic field a voltage is induced across the end wires of the coil. The transfer of energy from the flux field of the magnet to the wire coil generates the induced voltage. As the coil is forced through the magnetic field by vibratory motion, a voltage signal correlating with the vibration velocity is produced. The magnet-in-coil type of sensor (see figure 5.8(b)) is made up of three components: a permanent magnet, a coil of wire and spring supports for the magnet. The pickup is filled with oil to dampen the spring action. The relative motion between the magnet and the coil caused by the vibration motion induces a voltage signal. The coil-in-magnet contains the same components in a reciprocal mounting. The velocity pickup is a self-generating sensor and requires no external devices to produce a voltage signal. The voltage generated by the pickup is directly proportional to the velocity of the relative motion. Due to gravity forces, velocity transducers are manufactured differently for horizontal or vertical axis mounting. The velocity sensor has a sensitive axis that must be considered when applying it to rotating machinery. Velocity



(a) Example of a Velocity pick-up. (b) Velocity pick-up scheme.

Figure 5.8: Velocity pick-ups

sensors are also susceptible to cross-axis vibration, which could damage it. A velocity signal produced by vibratory motion is normally sinusoidal in nature. Thus, in one cycle of vibration, the signal reaches a maximum value twice. The second maximum value is equal in magnitude to the first, but opposite in direction. The convention is that motion towards the bottom of a velocity transducer will generate a positive output signal.

The fact that all vibration sensors measure motion along their major axis should be taken into account when choosing the number of sensors to be used. Due to the structural asymmetry of machine cases, the vibration signals in the vertical, horizontal and axial directions (with respect to the shaft) may differ. Where possible, a velocity transducer should be mounted in each of the three directions. In doing so, we will have a complete picture of the vibration signature of the machine.

Some velocity pickups are among the sensors with the highest sensitivity for rotating machine applications, which could be very convenient if electrical noise is a problem. However, there are some limitations. Generally, output values are in the range of 20–30 mV/(mm/s). The sensitivity is constant over a specified frequency range, usually between 10 Hz and 1kHz. At low frequencies of vibration, the sensitivity decreases because the pickup coil is no longer stationary with respect to the magnet, or vice versa. This decrease in pickup output drops exponentially and starts approximately at 10 Hz. Thus, velocity captures taken at frequencies below 10 Hz are inaccurate. Above that band, the velocity pickups should reach a useful response up to 2kHz. This is a limitation since the expected frequency range must be covered by the sensor. Also velocity pickups should be calibrated in a annual basis, since they are the only type of sensor with internal moving parts which are subject to fatigue failure.

Accelerometers

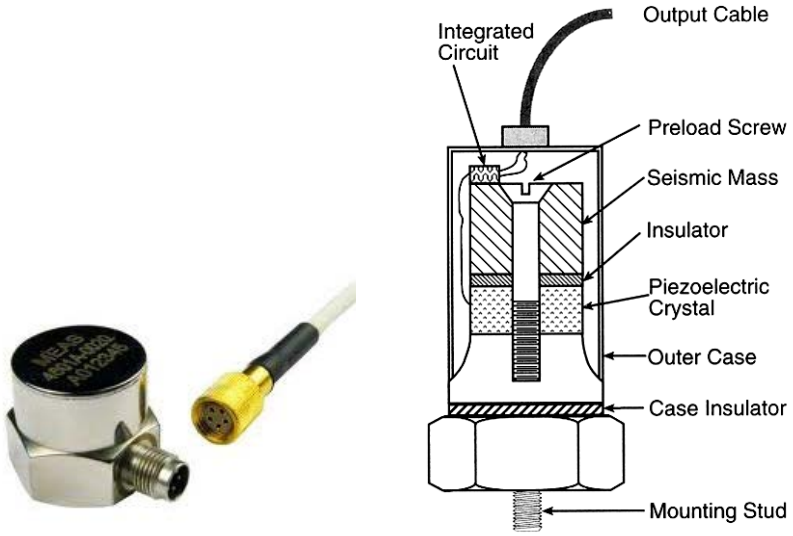
Accelerometers are the most popular transducers used for rotating machinery applications. They are compact, lightweight transducers with a wide frequency response range. Accelerometers are extensively used in many condition monitoring applications. Components such as rolling element bearings or gear sets generate vibration at high frequencies when defective (as we will observe in chapter 7). Machines with these components should be monitored with accelerometers. The installation of an accelerometer must be carefully considered for an accurate and reliable measurement. Accelerometers are designed to be mounted on machine cases and can provide continuous or periodic measure of case motion acceleration.

Accelerometers are inertial measurement devices that convert mechanical motion into a voltage signal. The signal is proportional to the vibrations's acceleration using the piezoelectric principle. Inertial measurement devices measure motion relative to a mass. This follows Newton's third law of motion: a body acting on another will result in an equal and opposite reaction of the first. Accelerometers (see figure 5.9(b)) consist of a piezoelectric crystal made of ferroelectric materials like lead zirconate titanate and barium titanate and a small mass normally enclosed in a protective metal case. When the accelerometer is subjected to vibration, the mass exerts a varying force on the piezoelectric crystal, which is directly proportional to the vibratory acceleration. The charge produced by the piezoelectric crystal is proportional to the varying vibratory force.

The charge output is measured in Pico-coulombs per g, where g is the gravitational acceleration. Current or voltage accelerometers have an internal low-output impedance amplifier and require an external power source which can be either a constant current source or a regulated voltage source. This type of accelerometers are normally a two wire transducer with one wire for the power and signal and the second wire for common. They have a lower temperature rating due to the internal amplifier circuitry and long cable lengths can reduce their effective frequency response range. There is a second family of accelerometers named charge mode, in which the amplifier is external and therefore have a higher temperature rating.

Accelerometers have usually a sensitivity of 100 mV/g but there is a wide variety of types specially designed for applications such as structural analysis, geophysical measurement, high frequency analysis, etc. Nowadays they even find applications in video games, mobile phones and robotics, being one of the standard component in many of these systems.

The main characteristic of the accelerometers is their frequency response. Once a particular frequency range of interest for a machine is known, an accelerometer can be selected. They typically have a frequency range from 1 or 2 Hz to 8 or 10 kHz. Some



(a) Example of an accelerometer. (b) Acceleration pick-up scheme.

Figure 5.9: Accelerometers

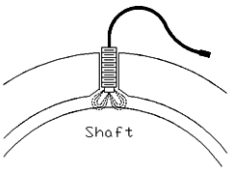
specific models can reach 20 kHz, although those are still expensive for some applications. This characteristic makes accelerometers suitable for detecting events which occur above the 2 Hz band, since this band is not well captured by velocimeters or proximity probes.

Proximity Probes

Proximity probes find their typical applications in high speed turbomachinery. They are the only transducers which provide displacement of the shaft or its relative vibration measurements. A proximity probe is a matched component system which consists of a probe, an extension cable and an oscillator/demodulator (see figure 5.10(b)). A high-frequency radio signal (RF) at 2 Mhz is generated by the oscillator/demodulator. This is sent through the extension cable and radiated from the probe tip. Eddy currents are generated in the surface of the shaft. The oscillator/demodulator demodulates the signal and provides a modulated DC voltage, where the DC portion is directly proportional to the gap (distance to the shaft) and the AC portion is directly proportional to vibration.

This vibration transducers measure motion in the mounted plane. In other words, shaft motion is either directed away from or towards the mounted Eddy current probe, and thus the radial vibration is measured in this way. This type of vibration pickups should

(a) Example of a proximity probe.



(b) Proximity probe mounting.

Figure 5.10: Proximity probe

be mounted in the plane where the largest vibrations are expected. The effective frequency range of this kind of vibration transducers is from about 0 Hz to 600 Hz.

In the experiments that we present herein afterwards, we will use always signals extracted from accelerometers measurements. This is due to the fact that we will try to detect fault in rolling element bearings, and this components usually generate fault features in the high-frequency band, where the accelerometer technology is the most adequate in terms of effectiveness and cost.

5.3.2 Analysis techniques

f To make raw vibration signatures informative from a fault detection and diagnosis perspective, they should be transformed in order to enhance relevant features. In the past years, the study of machinery fault detection has been studied thoroughly and many features for detecting the faults both manually and automatically have been established [167][235][236][237]. We give a short description of the most used features in practical fault detection and diagnosis tasks.

Time domain representations

One of the main danger indicators which can be gathered from vibration data is their amplitude. A large displacement of moving components of the machinery can lead to

fatigue and eventually breakdown. Thus, measurements extracted from the vibration raw signal proportional to the vibration amplitude can be very effective. Among them, the most common are:

1. Root mean squared energy (RMS). It measures the mean squared deviation of the raw vibration signal from the origin (equilibrium). Its expression is

$$RMS = \sqrt{\frac{1}{T}\sum_{t=0}^{T} x_t^2} \tag{5.23}$$

   where x_t are the values of the raw signal and T the signal length.

2. Peak-to-peak. It measures the distance from the most positive value of the raw vibration signal to the most negative. Recall that positive and negative values of the signal given by the transducer correspond to the maximum displacements in each direction.

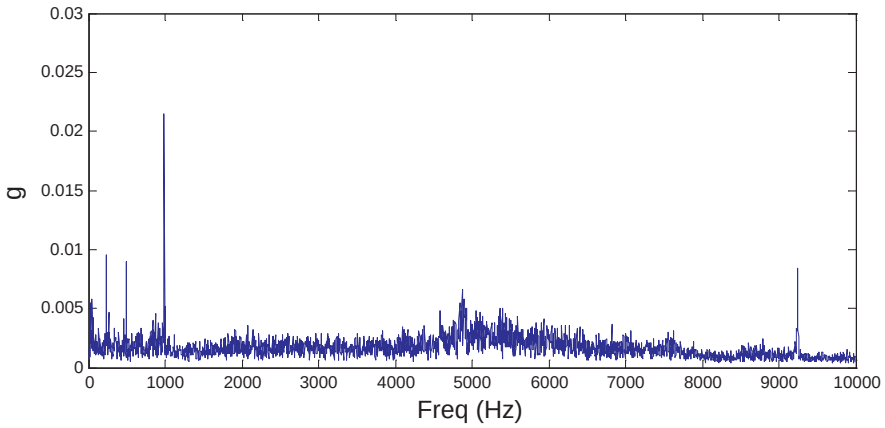
3. Zero to peak. It measures the maximum distance from the origin (equilibrium). This measure is proportional to the maximum deflection experienced by the system.

Despite their simplicity, they are the standard in many systems. Among them, RMS is the most used. These measures present limitations when incipient faults are to be detected. In inner stages, the fault is concealed into the raw vibration signal without showing any significant increment in the amplitude, thus becoming undetectable using only these indicators. On the other hand, when the fault is evident, they are simple measures to check in order to confirm it.

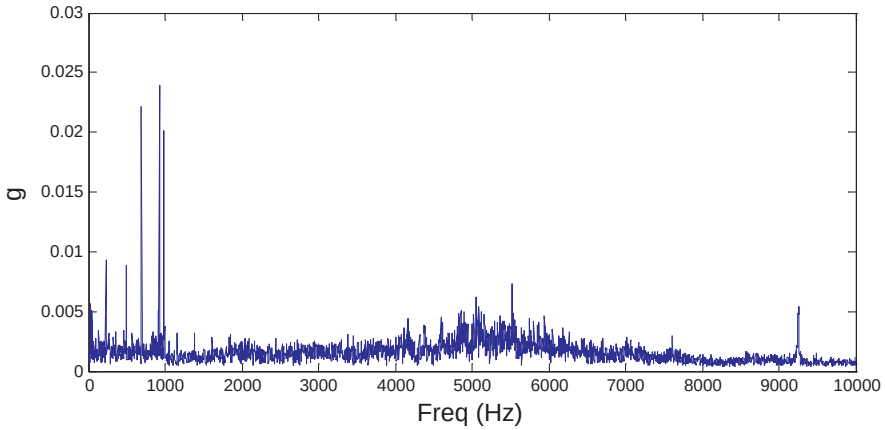
Frequency domain representation

As we have seen in previous sections, the frequency of vibration of a component is an intrinsic property which is affected by its physical characteristics. It turns out that faults in mechanical components force them to vibrate at different frequencies depending on the nature of the fault. We cite a couple of examples:

- A fault in a tooth of a gear will produce a shock each time the tooth contacts with its counterpart. If the machine runs at constant speed, these shocks produce a train of impulses in the vibration signal. Thus, a representation of this signal



(a) Fault-free state spectrum.



(b) Faulty state spectrum.

Figure 5.11: Example of the spectrum of a healthy and faulty bearing.

in the frequency domain should show a raising peak at the periodicity of the impacts, which is a more noticeable feature than the impacts in the time domain.

- When a ring of a bearing (see section 7.1.1) has a wedge (dangerous fault which could lead by fatigue to a complete breakdown), each time a rolling element passes through this wedge an impact is produced. This impact is followed by a damped vibration at the natural frequency of the system. Thus, when regarding this effect from the frequency domain perspective, a modulated peak at the natural frequency of the system appears (see band under 2KHz in figure 5.11).

These two short examples show two major properties of frequency domain trans-

formation in the context of mechanical vibrations: (a) by studying through models the ideal behavior of the components under a fault, we can establish how it will behave in the frequency domain; (b) due to noise and couplings of multiple mechanical effects, being able to notice a change in a signal in time domain could be very difficult, but these abnormal effects due to faults become more apparent in frequency domain. However, it is a well known fact that when the fault is incipient or the mechanical system is too complex, vibration features due to faults are concealed by the vibration signature of the rest of the system and noise. This fact usually makes incipient fault detection task in frequency domain still a difficult duty. So the quest of finding the most appropriate representation is still an open issue and is subject of a huge amount of literature these days.
Since we are mostly interested in the energy of the raw vibration signal at each frequency, we are interested in its power spectrum. This can be calculated though its Fast Fourier Transform (FFT) [178]. Since the signal is sampled with fixed time intervals, the signal is truncated at its start and end so the result can vary with the location of the sample with respect to the waveform's periods. Due to this fact, windowing is performed in oder to deal with this side effect [206]. There is a huge amount of available windows. Some of them used in vibration signal processing are: Rectangular, Hanning, Hamming, Barlett and Blackman [206].

Advanced representations

As we have commented above, the quest for effective feature extraction schemes for fault detection from raw vibration signals still continues. However, most of the proposed transformations are still in a research stage and have not reached a standard use among the fault detection community. Even some companies offer their own private transformations, although these are not very detailed. There are several remarkable examples of academic and industrial efforts in this area in recent years. Among them, wavelet transform [193], Hilbert transform [178] and cepstrum [206] are common technologies in vibration analysis in addition to the standard ones presented overleaf.
In the last chapter, we propose a novel transformation based on recurrence time statistics which presents promising results when analyzing the raw vibration signal in the time domain. Its effectivity will be explored in combination with the EVOC algorithm (see chapter 3) for the case of rolling element bearing fault detection.

5.4 Rotating machine fault detection from an anomaly detection perspective

As we have detailed in previous sections, fault detection is tackled ideally following the next curse of action: (a) an analytical model of the machine is devised and its set of fault explanatory parameters is extracted from that model, (b) this set of explanatory parameters together with its alert levels is compiled as a manual for the maintenance office and (c) the machine is periodically monitored in order to detect deviations which, if aligned with any fault parameter, derive a machine overhaul.
This way of managing maintenance comes from the classical plant control which has been applied since the mid past century in many areas. Unfortunately, this curse of action finds in some real situations critical obstacles to accomplish an effective maintenance. First of all, market pressure makes it impossible to derive accurate analytical models for all the models which are send to production in different plants. This means that we will not have a fully description of the scenarios that can occur in vibrational data. On the other hand, there are many situations, as it is the case of wind mill farms, in which there are simply too many components and assets in the plant which make it impossible, from the costs perspective, to manually (although based on computerized assistants) control 24/7 all the assets which the plant depends on. Maintenance professionals are a costly asset which takes a very long time to train. In these situations, a computer intelligent strategy is found as an invaluable solution which aims at lowering maintenance cost and reducing unplanned breakdowns.
In order to pose a machine intelligent solution to the problem we have to set up a framework for it. It turns out that from a ML perspective this problem can be tackled in two ways: (a) multiclass classification and (b) anomaly detection, each one with its advantages and inconveniences.

Multiclass classification: In this case, the aim is to construct a model able to distinguish between each fault state and normal state. A supervised classifier is constructed based on the vibration captures extracted both in normal and faulty states. This approach has been the most used in the area in the past decade [80][126][171][254][255] and has the advantage that very accurate models can be built relying on modern state of the art classifiers [15][31]. However, it is not a universal approach for the problem at hand due to the following facts:

1. Obtaining fault vibration captures for new models is a very costly process which involves: (a) inducing a fault in very controlled conditions and (b) running several experiments in order to cover all different possible faults and on different running conditions.

2. In some areas, such as wind mill power generators, maintenance was not the focus of the manufacturers. In the fist generations of machines, it was supposed that they were going to last for 20 years without any actuations. It turns out that this was not the case and the machines started to break down after 3 or 4 years leading to high substitution costs. Thus, there is a large pool of machines functioning for which we do not have data under faulty conditions and we still want to monitor them in an economically effective way and, evidently, it is not possible to break down them in order to obtain those faulty data records.

Anomaly detection: This perspective tries to overcome the limitations of the multiclass perspective. In this case, we only need a dataset extracted from the machine under these normal functioning conditions. We will adopt a unsupervised approach due to the fact that this data set will be not labelled (it would be very costly to label tons of captures by experts, only to purge a small percentage of abnormal captures) and could contain some abnormal captures.
In this work we will focus on the anomaly detection perspective, since it is the candidate approach which is the most universal of the two. It can be applied virtually in any situation. The procedure of an automatic fault detection software from this perspective is as follows (figure 5.12 depicts schematically the process):

1. Normal state data capture: A historic set of vibration captures under normal conditions is necessary in order to build a model of the vibration signature of the machine under normal conditions. In this stage there are two options: (a) expert assessment is available or (b) there is not any human intervention in this stage. In the fist case, an expert is available to act as a curator of the data base (i.e., he will purge the data set of abnormal captures, thus all patterns in the dataset are considered as 'healthy'). In the first case, a supervised approach can be taken. In the second case, which is more general, there is no human intervention, so a unsupervised approach (see chapter 2) is necessary since there could be a small percentage of spurious data records in the set. Usually the machines have also a set of working conditions which can affect the vibration signature. If available, these variables should also be adjoined to the data in order to be included in the model.

2. Model training: In this stage, the raw vibration signals are transformed to descriptive features able to differentiate between a normal and a faulty condition. As we described in the previous section, the Fourier Analysis of the raw vibrational signal has been found the most valuable tool in this stage. As we will see in following chapters, the availability of knowledge about components of the underlying machine can help to select the important features and focus further the detection algorithm (even to an automatic diagnosis). If this information is not available, we should consider a set of features which covers those which are descriptive. Subsequently, a personalized detector is trained. For this task, we could use an unsupervised anomaly detection algorithm. In the last chapter we will apply the proposed anomaly detection techniques to real machinery fault detection problems.

3. Monitoring: Once this model has been trained, it is included in the data flow of the machine in order to detect any anomaly which could be a potential faulty condition. In other words, the vibration captures of the machine are continuously captured and analyzed by the anomaly detection algorithm. As we pointed out in previous sections, the captures that the algorithm considers as a potential failure must be further analyzed by a human expert in order to confirm a diagnosis and counteractions. There is a huge gain in effectivity when an accurate fault detection algorithm is found since the number of monitored components per human expert increases largely. These experts do not need to revise tons of normal captures which are automatically detected as normal by the algorithm and can focus their effort in only a very reduced set of potential anomalies pinpointed by the system.

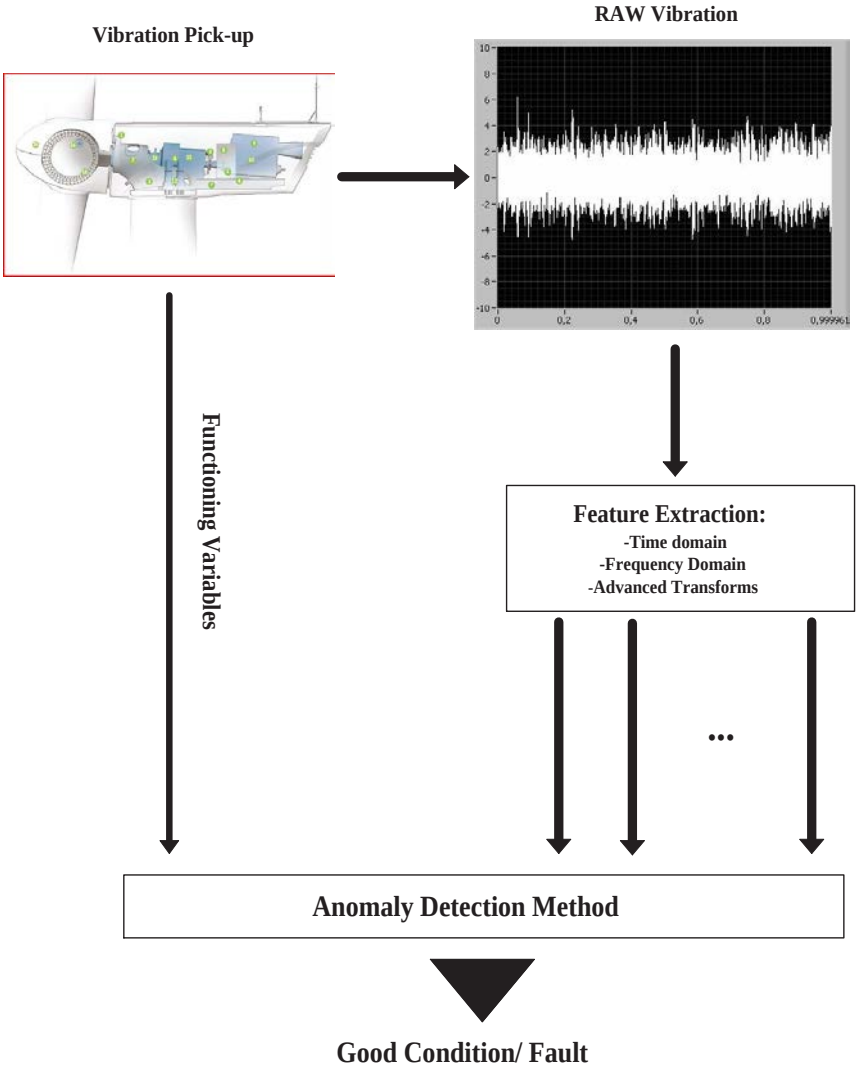


Figure 5.12: Architecture of fault detector based on anomaly detection techniques

CHAPTER 6

GIDAS®: An Automatic Fault Detection System based on Vibration Signatures

Machinery maintenance based on vibration analysis has been successfully driven by experts, following the principles of previous chapter, for years. Periodically, vibration data is gathered from the machines, mostly during periodical revisions, and subsequently analyzed by experts looking for internal fault symptoms inside data. Despite the success of this process, it presents a problem when scaled to large plants with numerous machines to be monitored. Training programs for vibration experts are costly both in investment and time and, once trained, these experts spend 80% of the time analyzing fault-free data. When we scale the figures to modern installed plants such as power wind mill farms, with tens or even hundreds of machines per farm with difficult access (p.e. offshore plants like the ones installed in the British north coastline) maintenance costs become a limitation for increasing the number of assets. In the present and the last chapter of this work we explore the transition from expert-driven predictive maintenance to a data-driven maintenance workflow aided by ML analysis. We aim to build a system able to: (a) distinguish between a healthy machine and a defective one, (b) locate where the fault is present and (c) assess if in a near future will reach an unacceptable state in order to stop it immediately. A software with these capabilities would reduce the necessity of workforce for routine data analysis, increase expert throughput and reduce final production costs. We break the results obtained in these two final chapters. In the present one, GIDAS® system main properties and its design are presented. This software was developed as part of this thesis work and is conceived as a ML driven tool for vibration analysis in power wind mill farms. The huge amount of machines that need to be monitored in a wind mill farm explains the necessity of an automatic fault detection software in this plants. This software was developed in collaboration with INDRA S.A. and renewable energy companies. Two pilot programs of GIDAS in real production plants are explained in the last section of this chapter.

In the next chapter, fault detection capabilities of the ML methods presented in chapters 3 and 4 are compared experimentally.

6.1 Aim and scope of the system

GIDAS® system is a prototype of a commercial automatic fault detection system based on ML techniques which has been developed in this thesis. This system has been used in this work as a practical workbench for the application of the algorithms presented in the first part of the thesis to the predictive maintenance problem. The benefits of its development have been twofold: (a) it has provided a software able to capture real fault cases in production (some of them are detailed in the next chapter) necessary to demonstrate the validity of the approaches proposed in this work; (b) it has provided an opportunity of testing the acceptance of the approach in real plants. As we detailed in previous sections, any computerized maintenance system must be embodied in a global maintenance program which involves much more than software and hardware components. Communication and detection abilities of the developed system must be aligned with the traditional workflow of the maintenance office in which it is deployed. The system was designed for tackling an anomaly-based fault detection system that analyzes vibrational data in power wind mill farms. This environment is specially challenging due to the numerous machines to which it should be able to adapt. Modern power wind mill farms contain many different machines of many different designs and all of them need to be monitored. On the other hand, this field is suitable for the acceptance of an automatic fault detection system since the number of turbines and the environment (off-shore, deep mountains, desserts, ...) make impossible the inspection and management of all them in an easy and cost effective way (to put this fact in numbers, in 2012 a cumulative power of 238 GW was installed worldwide) [57].
In the next section we present a brief description of the main components of a wind turbine and its main points of failure. Subsequently we present the characteristics that an automatic fault detection and monitoring system must comply in order to be accepted for certification. These requirements will be the basis of the design of the system described in this chapter. We close the chapter with a set of reflexions extracted from the pilot experience.

6.2 Modern wind turbines: concepts and design

Human inventions such as wind-powered ships, grain mills, water pumps and threshing machines all exemplify that extraction of power from wind is an ancient endeavor. With the evolution of mechanical insight and technology, the last decades of the 20th century, in particular, saw the development of machines which efficiently extract power from wind. "Wind turbines" is now being used as a generic term for machines with rotating blades that convert the kinetic energy of wind into useful power. Modern turbines evolved from the early designs and can be classified as:

- Depending on the number of blades they mount they can be two or three-bladed. The choice between both is merely a matter of a trade-off between aerodynamic efficiency, complexity, cost, noise and aesthetics. The two and one-bladed concepts have the advantage of representing a possible saving in relation to the cost and weight of the rotor. However, their use of fewer rotor blades implies that a higher rotational speed or a larger chord is needed to yield the same energy output as a three-bladed turbine of a similar size. The use of one or two blades will also result in more fluctuating loads because of the variation of the inertia. One-bladed wind turbines are less widespread than two-bladed turbines. This is due to the fact that they have, in addition to a higher rotational speed, more noise and visual intrusion problems and need a counter-weight to balance the rotor blade. The three-bladed concept is the most common concept for modern wind turbines.

- Depending on the way rotation of the transmission is allowed by the machine they can be classified as vertical axis and horizontal axis turbines. Figure 6.1 depicts two examples of each of them. Horizontal axis wind turbines constitute the most common type of wind turbine in use today. In fact all grid-connected commercial wind turbines are today designed with propeller-type rotors mounted on a horizontal axis on top of a vertical tower. In contrast to the mode of operation of the vertical axis turbines, the horizontal axis turbines need to be aligned with the direction of the wind, thereby allowing the wind to flow parallel to the axis of rotation. Insofar as concerns horizontal axis wind turbines, a distinction is made between upwind and downwind rotors. Upwind rotors face the wind in front of the vertical tower and have the advantage of somewhat avoiding the wind shade effect from the presence of the tower. Upwind rotors need a yaw mechanism to keep the rotor axis aligned with the direction of the wind. Downwind rotors are

(a) Horizontal axis design. (b) Vertical axis design.

Figure 6.1: Examples of horizontal/vertical wind mill designs.

placed on the lee side of the tower. A great disadvantage in this design are the fluctuations in the wind power due to the rotor passing through the wind shade of the tower which gives rise to more fatigue loads. The vast majority of wind turbines in operation today have upwind rotors.

Additional key turbine design considerations include wind climate, rotor type, generator type, load and noise minimization, and control approach. Moreover, current trends, driven by the operating regime and the market environment, involve development of low-cost, megawatt-scale turbines and lightweight turbine concepts. Whereas turbines operating at constant rotor speed have been dominating up to now, turbines with variable rotor speed are becoming increasingly more common in an attempt to optimize the energy capture, lower the loads, obtain better power quality, and enable more advanced power control aspects. A turbine with an upwind rotor, an asynchronous generator and an active yaw system is usually referred to as the *Danish concept*, which tends to be a standard against which other concepts are evaluated.

Wind turbines are designed to produce electricity as cheap as possible, so to yield a maximum power output at wind speeds around 15 m/s. It would not pay to design turbines to maximize their power output at stronger winds, because such strong winds are usually too rare. However, in the second case, it is necessary to waste part of the excess energy to avoid damage on the wind turbine. Thus, the wind turbine needs some sort of power control.
There are two kinds of power control in wind turbines. Stall-controlled wind turbines have their rotor blades bolted to the hub at a fixed angle. The stall phenomenon is

used to limit the power output when the wind speed becomes too high. On the other hand, pitch-controlled wind turbines have blades that can be pitched out of the wind to an angle where the blade chord is parallel to the wind direction. The power output is monitored and whenever it becomes too high, the blades will be pitched slightly out of the wind to reduce the produced power.
The ideal wind turbine design is not dictated by technology alone, but by a combination of technology and economy. Wind turbine manufacturers wish to optimize their machines, so that they deliver electricity at the lowest possible cost per unit of energy. In this context, it is not necessarily optimal to maximize the annual energy production, if that would require a very expensive wind turbine. Since the energy input (the wind) is free, the optimal turbine design is one with low production costs per produced kWh. A large generator will be very efficient at high wind speeds, but inefficient at low wind speeds. Sometimes it will be beneficial to fit a wind turbine with two generators with different rated powers.
A study of different Danish wind turbine designs shows that the specific power performance in terms of produced energy per m^2 rotor area per year (kWh/m^2/year) is almost independent of the rotor size. Hence, the main consideration in the evaluation of the cost of the turbine is the specific rotor power (kW/m^2) and the specific cost (cost/m^2 rotor) together with expected service life and cost and availability.
The power being produced by any type of wind turbine can be expressed as

$$P = \frac{1}{2}\rho V^3 A C_P \tag{6.1}$$

where P represents output power, ρ the air density, V the free wind speed, A rotor area and C_P efficiency factor. The power coefficient C_P is a product of the mechanical efficiency ν_m, the electrical efficiency ν_e and of the aerodynamic efficiency. All three factors are dependent on the wind speed and the produced power, respectively. The mechanical efficiency ν_m is mainly determined by losses in the gearbox and is typically 0.95 to 0.97 at full load. The electrical efficiency covers losses in the generator and electrical circuits. At full load $\nu_e = 0.97 - 0.98$ is common for configurations with an induction generator. It can be shown that the maximum possible value of the aerodynamic efficiency is $16/27 = 0.59$, which is achieved when the turbine reduces the wind speed to one-third of the free wind speed (Betz' law).

The produced power varies with the wind speed as can be seen from the blue graph in Figure 6.2. The form of the graph can vary slightly for different kinds of wind mills. Assuming constant efficiency (e.g., constant tip speed ratio) the graph basically consists of a third degree polynomial up to the rated wind speed at which the nominal power is

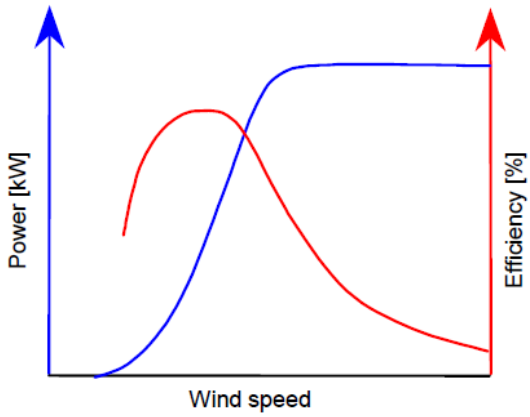


Figure 6.2: Power and efficiency curve of a wind turbine

reached. At this point the power regulation sets in, either by the blades stalling or by pitching the blades to attain an approximately constant power. The power curve and the power efficiency curve are often presented in the same graph, with the power and the efficiency scales on each side of the graph as shown in figure 6.2. Figure 6.3 illustrates the controlled power curve of a wind turbine, in the case of 1) stall controlled, fixed speed configuration, and 2) pitch controlled, variable speed configuration.

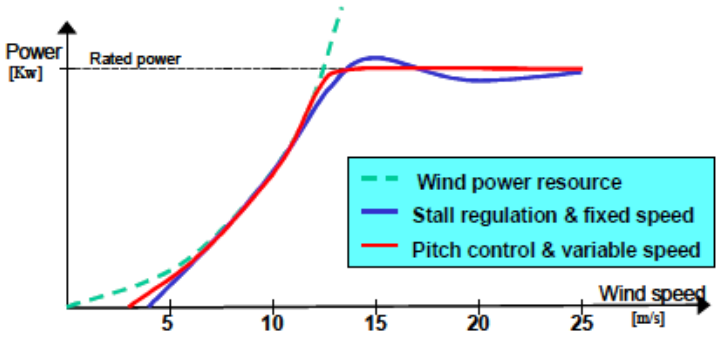


Figure 6.3: Examples of power curves for two types of wind turbines.

The efficiency factor C_P typically reaches a maximum at a wind speed of 7-9 m/sec and, normally, it does not exceed 50%. The electric power typically reaches the rated power of the turbine at a wind speed of 14- 16 m/sec.

6.3 Modern horizontal power wind mill main components

Figure 6.4 depicts the main mechanical components of a modern horizontal three blade wind turbine. This is the most common today in the market and its components are described hereunder.

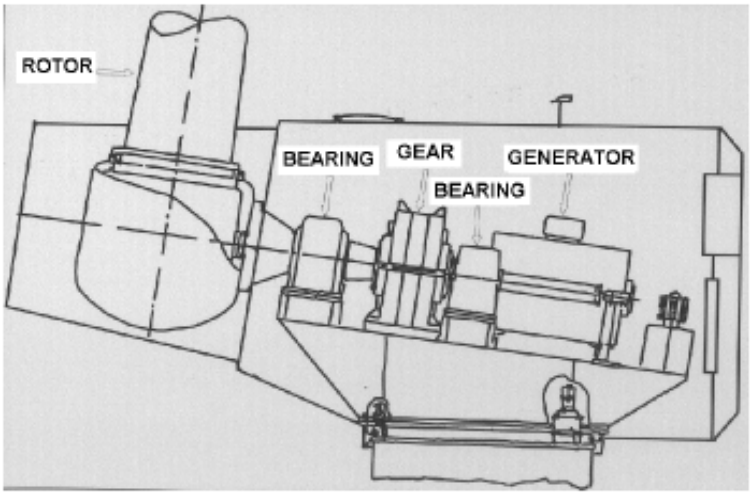


Figure 6.4: Main components of a modern wind turbine.

Rotor Blades

Rotor blades are usually made of a matrix of fibreglass mats, which are impregnated with a material such as polyester, hence the term glass fibre reinforced polyester, GRP. The polyester is hardened after it has impregnated the fibre-glass. Epoxy is sometimes used instead of polyester. The design of the outer contour of a wind turbine rotor blade is based on aerodynamic considerations. The cross-section of the blade has a streamlined asymmetrical shape, with the flattest side facing the wind. Once the aerodynamic outer contour is given, the blade is to be designed to be sufficiently strong and stiff. The blade profile is a hollow profile usually formed by two shell structures glued together, one upper shell on the suction side, and one lower shell on the pressure side. To make the blade sufficiently strong and stiff, so-called webs are glued onto the shells in the interior of the blade, thus forming a boxlike structure and crosssection (see figure 6.5). From a structural point of view, this web will act like a beam, and simple beam theory can be applied to model the blade for structural analysis in order to determine the overall strength of the blade.

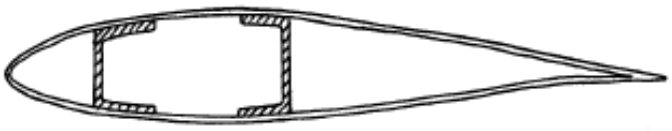


Figure 6.5: Section of a blade showing upper and lower shells and two webs, respectively.

It is important that the blade sections near the hub are able to resist forces and stresses from the rest of the blade.The blade is twisted along its axis so as to enable it to follow the change in the direction of the resulting wind along the blade, which the blade will experience when rotating. Hence, the pitch will vary along the blade. The pitch is the angle between the chord of the blade profile and the rotor plane.

Hub

The hub is the fixture for attaching the blades to the rotor shaft. It usually consists of nodular cast iron components for distribution of the blade loads to the wind support structure, i.e. ultimately to the tower. A major reason for using cast iron is the complex shape of the hub, which makes it hard to produce in any other way. In addition, it must be highly resistant to metal fatigue. Thus, any welded hub structure is regarded as less feasible.

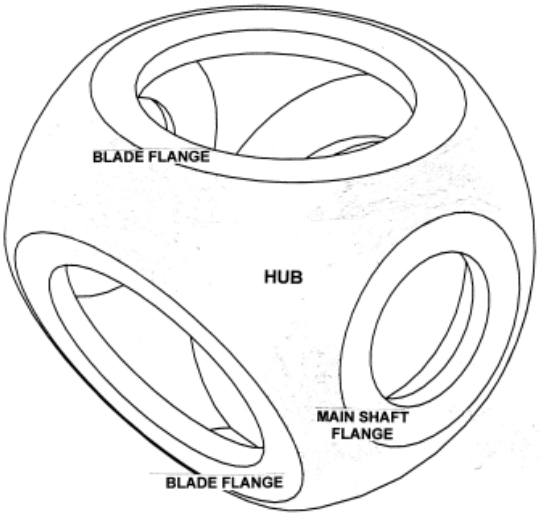


Figure 6.6: View of wind turbine hub.

The moments and forces transmitted to the hub and tower depend on the type of hub. Three types of hub are common: (a) hingeless rigid hub, which has cantilevered blades and transmits all moments to the tower; (b) teetering rotor, which has two rigidly connected blades supported by a teeter-pin joint, which can only transmit in-plane moments to the hub and (c) articulated hub, which has free hinges in flapping and lead-lag, so there is no mechanical restraint moment on the blades in either flapping or lead-lag. The hingeless hub is the most common configuration for wind turbine hubs, which is depicted in figure 6.6. Figure 6.7 and Figure 6.8 show the hub in the context of the transmission system, in which it forms part of the link between the rotor blades and the generator. Figure 6.7 and Figure 6.8 are also examples of two different bearing arrangements with one and two main bearings, respectively.

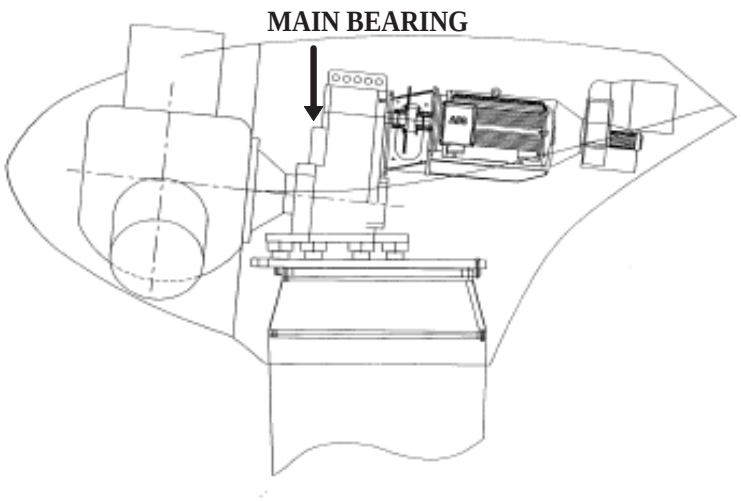


Figure 6.7: Wind turbine with main bearing integrated in the gearbox.

Main shaft

The main shaft transmits the rotational energy from the rotor hub to the gearbox or directly to the generator. Moreover, the purpose of the main shaft is to transfer loads to the fixed system of the nacelle. In addition to the aerodynamic loads from the rotor, the main shaft is exposed to gravitational loads and reactions from bearings. The main shaft is also subjected to torsional vibrations in the drive train. Such vibrations will usually be of importance to possible frictional couplings like shrink fit couplings between shaft and gear.

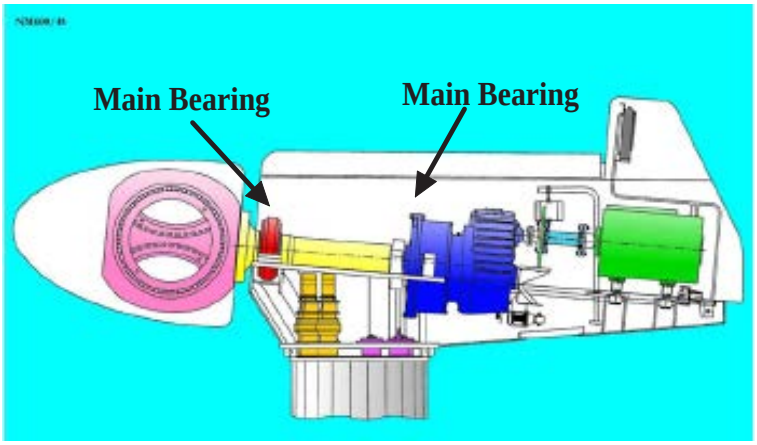


Figure 6.8: Wind turbine with two main bearings.

Main Bearing

The main bearing of a wind turbine supports the main shaft and transmits the reactions from the rotor loads to the machine frame. On account of the relatively large deformations in the main shaft and its supports, the spherical roller bearing type is often used, see Figure 6.9 for an example.

Spherical roller bearings have two rows of rollers with a common sphered raceway in the outer ring. The two inner ring raceways are inclined at an angle to the bearing axis. The bearings are self-aligning and consequently insensitive to errors in the alignment of the shaft relative to the housing and to shaft bending. In addition to high radial load capacity, the bearings can accommodate axial loads in both directions.

The main bearings are mounted in bearing housings bolted to the main frame. The quantity of bearings vary among the different types of wind turbines. Many wind turbines have two bearings, each with its own flanged bearing housing. Some turbines with two bearings use the hub as a housing. Some turbines have only one main bearing, given that the gearbox functions as a second main bearing. Each bearing arrangement has its own advantages and disadvantages.

Main gear

The purpose of the main gear is to act as a speed increaser and to transmit energy between the rotor and the generator. The most common gear types used for wind turbines can be identified and classified as follows, based on their geometrical design: (a) spur and helical gears consist of a pair of gear wheels with parallel axes (see figure 6.10). Spur gears have cylindrical gear wheels with radial teeth parallel to the axes. In

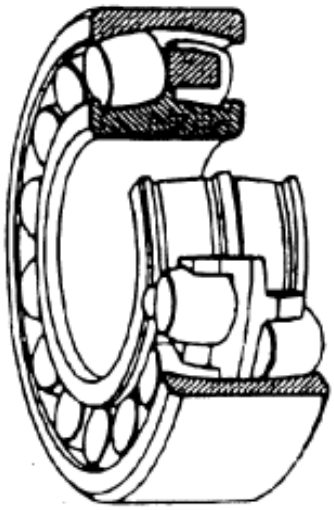


Figure 6.9: Spherical roller bearing, from Bonus (1999)

helical gears, the teeth are helical, i.e. they are aligned at an angle with the shaft axes. Double-helical gears have two sets of helical teeth on each wheel. Helical gears are sometimes referred to as spiral gears or oblique gears, (b) epicyclic or planetary gears consist of epicyclic trains of gear wheels, i.e. gears where one or more parts, so-called planets, travel around the circumference of another fixed or revolving part (see figure 6.11). Planetary gears, in combination with one or more parallel axis gears, form the most commonly applied gear type for the main gear in wind turbines. Gears in which the power is transferred from one wheel to two or more meshing wheels are referred to as gears with a split power path. Bearings for wind turbine gears should all be rolling element, anti-friction type bearings. Different bearing types applied in gears include: (a) ballbearings, (b) cylindrical roller bearings, (c) spherical roller bearings, (d) tapered roller bearings. Examples of bearings are shown in figure 6.12. Two bearings should be used to support each gear shaft, one for support of both radial and thrust forces, the other for support of only radial forces and free to allow for axial growth under thermal changes. Bearing fits should be tight to prevent damage to the bearing or the housing and to prevent spinning of inner and outer bearing races.

Mechanical brake

Mechanical brakes are usually used as a backup system for the aerodynamic braking system of the wind turbine and/or as a parking brake, once the turbine is stopped, e.g., for service purposes. Mechanical brakes are sometimes also used as part of the yaw

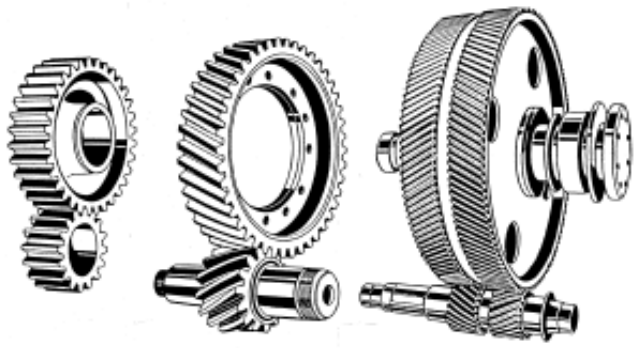


Figure 6.10: Examples of spur and helical gears.

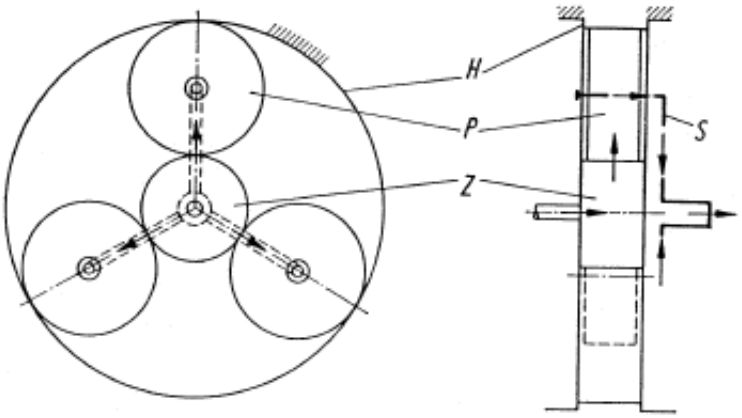


Figure 6.11: Examples of planetary gear principle with outer fixed, three revolving planets and a planet carrier in the middle.

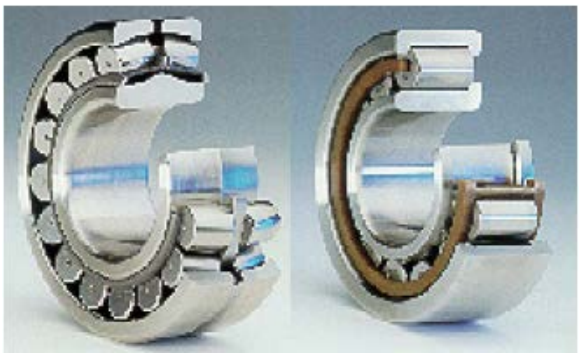


Figure 6.12: Examples of roller bearings: Spherical bearing (left) and cylindrical bearing (right). From SKF (1997).

system. In a mechanical brake, brake callipers, brake discs and brake pads form crucial parts. A hydraulic system is usually used for the actuation and release of the brake.

Generator

The generator is the unit of the wind turbine that transforms mechanical energy into electric power. While the blades transfer the kinetic energy of the wind to rotational energy in the transmission system, the generator provides the next step in the supply of energy from the wind turbine to the electrical grid. The produced alternating current, which is transmitted to the electrical grid, must match the frequency of the grid. The required rotational speed of the generator's rotor is achieved by means of the gearbox of the wind turbine, since the wind turbine rotor itself is not allowed to rotate at this high speed for physical reasons. Rotation of the wind turbine rotor at the high rotational speed of the generator rotor would cause aerodynamic problems and supersonic speeds. Noise would also be a problem, and excessive centrifugal forces would be generated.
There are two major types of generators: (a) synchronous generators and (b) asynchronous generators. A synchronous generator operates at a constant speed, dictated by the frequency of the connected grid, regardless of the magnitude of the applied torque. The speed dictated by the frequency of the grid is also known as the synchronous speed. An asynchronous generator is a generator, which allows slip, i.e. deviations from the rotational speed dictated by the frequency of the connected grid. In other words, the rotational speed is allowed to vary somewhat with the applied torque. This is the most common generator type used in wind turbines. The advantage of the variable slip comes about when the wind turbine is operated at its rated power, at which undesirable power fluctuations caused by changes in the wind appear. When a wind gust hits the wind turbine rotor, the slip enables the generator speed to increase a little in response to the gust without causing a corresponding increase in the generated power output. Thus, the slip ensures a smooth power output and at the same time contributes to keeping the loads on blades, main shaft and gearbox down. The variation of the operating speed with the applied torque for an asynchronous generator is beneficial because it implies a smaller peak torque and less wear and tear on the gearbox than for a synchronous generator. This is one of the most important reasons for using an asynchronous generator rather than a synchronous generator in a wind turbine, which is connected directly to the electrical grid. Traditionally, the active materials in a generator consist of magnetically conducting iron and electrically conducting thread arranged in a coil. Permanent magnets are becoming increasingly common, and electrical components, such as temperature sensors, are becoming integral parts of the generator.

Yaw system

Yaw denotes the rotation of the nacelle and the rotor about the vertical tower axis. By yawing the wind turbine, the rotor can be positioned such that the wind hits the rotor plane at a right angle. The yaw system provides a mechanism to yaw the turbine and to keep the rotor axis aligned with the direction of the wind. If situations occur where this alignment is not achieved, yaw errors are produced. The yaw error, or the yaw angle, is defined as the angle between the horizontal projections of the wind direction and the rotor axis. The yaw system can be either passive or active. A passive yaw system implies that the rotor plane is kept perpendicular to the direction of the wind by utilization of the surface pressure, which is set up by the wind and which produces a restoring moment about the yaw axis. For upwind turbines, this usually requires a tail vane in order to work properly. Note that a passive yaw system may pose a problem in terms of cable twisting if the turbine keeps yawing in the same direction for a long time. An active yaw system employs a mechanism of hydraulic or electrically driven motors and gearboxes to yaw the turbine and keep it turned against the wind. Such active positioning of the turbine relative to the wind is also referred to as forced yaw. Most large horizontal axis wind turbines use forced yaw to align the rotor axis with the wind.

6.4 Automatic predictive maintenance system requirements

The expansion of wind mill power generation and the inaccuracy of initial lifetime predictions made by manufacturers in design phase has pave the way for the necessity of electrical energy providers of assuring their assets in their wind farms. On the other side, insurance companies saw in the wind turbines market both an opportunity and a risk. The opportunity was clear due to the huge growth of the market impulsed by green energy policies, but the lack of clear life expectancies prevented them of being able of calculating their premiums. So, they started to impose as a condition the implementation of a predictive maintenance program which controlled the risk of the investment. In this line, they created a set of requirements that a computerized fault detection system should comply in order to be certified in the market. These requirements were collected in different norms which were very similar in nature to the *Guideline for the Certification of Condition Monitoring Systems for Wind Turbines* [152] which we will

take as a reference. Generally speaking, the most significant requirements are:

- Mechanical systems should be monitored continuously. Manual data captures (with portable vibration measurement devices) are not considered enough in order to assess the lifetime of rotating machinery.
- Automatic predictive maintenance systems (APMS) do not substitute in any case the human inspection in order to give a final diagnosis.
- The installation of an APMS should be considered an strategic step in maintenance programs.
- APMS should never substitute in any case the security systems of the machinery, in other words, they are open loop monitoring systems.
- When an APMS is installed it should never interfere with the security systems already installed in the machinery. If any modification of the standard design of the machine is needed, an approved certification by the manufacturer should be obtained.
- In case an automatic diagnosis is provided, it should be certified that all significative parameters are considered in the interpretation of the data.
- APMS should protect all data from not authorized accesses. Access policies should be adequate and approved by the person in charge.
- APMS user interface (GUI) should be intuitive, simple and free of ambiguities in order to be used by any person trained on Information Technologies.
- In order to detect failures of the APMS itself, supervision policies of the components (sensors, cabling, software, communication channels, etc) should be implemented.

In addition, the following characteristics should be taken into account during the design and documentation of the system:

- APMS designed and selected components should be able to work in extreme conditions such as off-shore, extreme temperatures, etc.
- Adequate storage devices should be used in each stage of the APMS.

- APMS system should provide backup storage of all data captured.

- Emergency energy supplies should be available for all the components of the APMS and reliability of data transmission should be assured.

- The number of vibration acquisition channels (vibration pick-ups) should be adaptable to each situation.

For the alarm notification system, the norm establishes the following:

- The APMS should be able to establish normal state levels for all the systems. Fault notifications should have at least two levels; pre-alarm and alarm.

- The APMS should provide an automatic procedure to calculate limit values, extracted from historic data and taken as base security levels. This levels should be subject to update in order to react to changes in the operation of the machinery (repairs, etc).

- In variable speed machinery, the systems should be able to adapt its interpretation to velocity variations.

- APMS system should be able to automatically notify any potential anomaly to the chief manager of the maintenance service and other related areas if necessary.

- Alarm notifications should be stored in the system as any other piece significant of information.

- All the counteractions taken for each alarm should be documented and stored in the system. APMS system should implement mechanisms that ensure that no alarm is ignored.

This norm was taken as the standard for the system design since: (a) it establishes a quality standard for all the maintenance systems in wind mill farms and (b) in the future, the application of predictive maintenance systems in any industrial environment will require certification norms similar to the one presented overleaf.

6.5 System design

In this section a brief description of the GIDAS system is given. The whole system is conceived as an information flux in which each level transforms its data input to an enriched data output, from the raw vibration capture to an elaborated report of state of each machine. Figure 6.13 depicts the different layers in which the system is divided from the capture system to the visualization of the state of each machine. We give a brief description of each level.

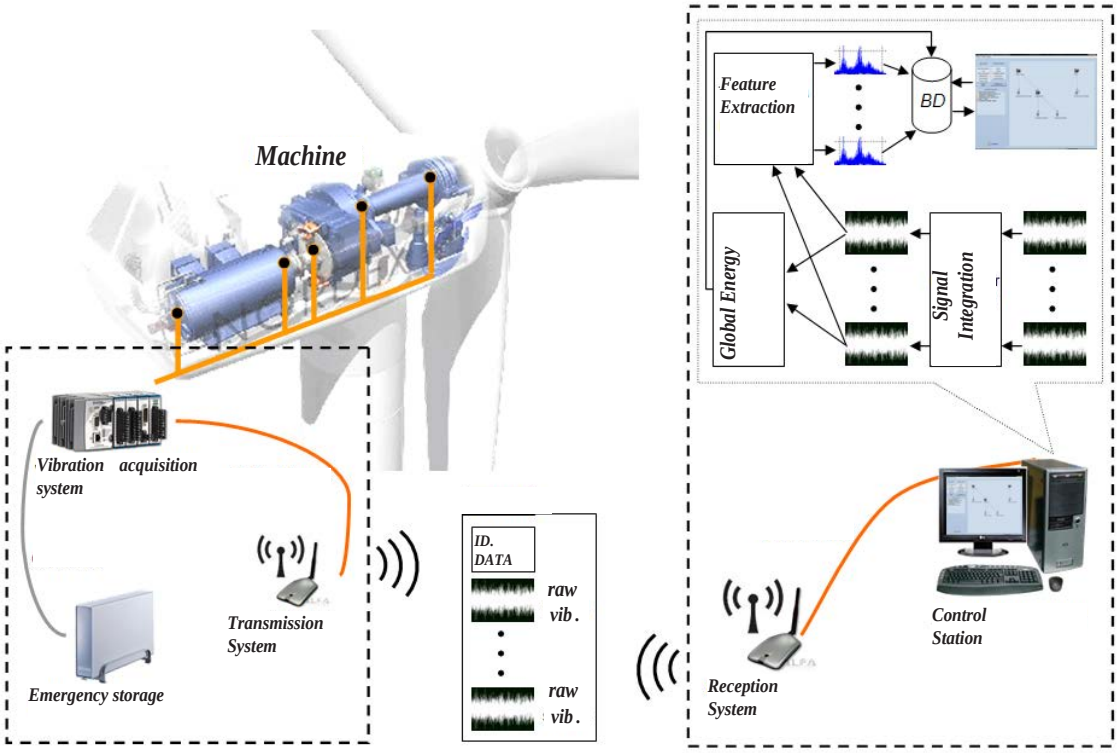


Figure 6.13: Block design of the GIDAS system.

Physic layer

This first level is responsible of capturing the vibration data using a combination of sensors and acquisition modules. Acquisition modules are real time computer systems including reading cards to which any kind of electrical transducer can be connected. These computers are specifically programmed in order to accomplish the task at hand. Usually, the selection of these computers are a concern when devising a commercial system due to their high costs. For this prototype we used a Compact RIO of National Instruments, although the commercial version would mount a specifically devised sys-

tem manufactured by INDRA Systems in order to reduce costs. Due to the fact that many internal components of the wind mill produce high frequency fault symptoms, accelerometers are the most suitable vibration pickups. In some modern machines the accelerometers are already installed. In this case it is only needed to install signal conditioners in order to adapt the signals to the acquisition system. Unfortunately, many machines in production scenarios do not mount any sensors so new instrumentation has to be added to the acquisition system. In this work, accelerometer model 4384 manufactured by Brüel & Kjaer were selected, giving a bandwidth of 20 KHz.
In addition to the vibration levels, it is desirable to have a reading of the velocity at which the machine is working, since the vibration readings can be highly influenced by the functioning regime of the machine. GIDAS mounts a tachometer in the main shaft. All the measures will be synchronized with the reading of the tachometer.
The following are some key points that have to be carefully considered when installing this level:

- Measure points selection. This task is key in order to correctly capture the vibration signals on a faulty situation. It has to be done by the maintenance team, ideally in collaboration with the manufacturers, previously to the installation.
- The varying characteristics of the physics of the machine and of the environment can turn the installation phase in a challenging task. The installation of a multichannel acquisition system and sensors has to be done by trained personnel and field test of measurement and communications have to be done in order to detect any potential difficulty.
- Once all the equipment has been correctly installed, a measurement strategy adapted to each machine has to be selected: frequency range, measure periodicity, etc. Some of these factors can greatly endanger the ability of the system to give relevant information.

In figure 6.14, mounting, on a real production wind mill, of the GIDAS' acquisition systems can be observed.

Transport layer This layer should provide a reliable communication channel between the acquisition system and the Data Processing Centre. This task is challenging due to the nature of the environment where the wind mills are installed. Modern machines include Ethernet connections which are available for the transmission of the raw vibration captures, although the bandwidth is not always wide enough. Unfortunately, this is not

(a) Acceleration pick-ups installation.

(b) Acquisition rack.

Figure 6.14: Installation of GIDAS system in a production wind mill.

always the case, and many machines have not the possibility of easily transmitting the data (think for example on off-shore machines, old models, deep mountain machines, etc). Recall that costs are a concern for AMPS, who have to compensate costs with their benefits, so an efficient solution is desired. New technologies which appeared in the last decade such as modern Wifi connections, Wimax Networks and Power Line Communications Networks (PLC) can help to solve this aspect. This layer, following the guidance of the certification norms, has to ensure the reliability of the communications, so the acquisition system and all intermediate nodes should ensure, for example through intermediate storage, that no capture is lost during the communication. It should be recalled that losing a single capture could prevent the whole system from detecting a critical fault.

Signal processing layer In this layer of the system, raw vibration captures taken by the accelerometers are transformed in order to provide relevant information for the analysis of faults. The system has been devised contemplating the possibility of providing the processing layer in-situ in the acquisition system or at reception before entering the data management level. All relevant signal processing techniques presented in previous sections would be relevant for this layer. This prototype implements the standard amplitude parameters for tendency analysis (RMS, zero-peak and peak-to-peak value) and the power spectrum of the raw vibration signal calculated through its FFT. These data is used by the analysis layer to detect deviations that pinpoint potential faults in the monitored systems.

Data management layer This layer makes use of common IT technologies to provide the system with a database where all the captures and relevant information is stored. Specifically, a service oriented platform design was adopted using Simple Object Access Protocol (SOAP) technologies [113]. Generally speaking, this layer is in charge of:

- Receiving all data from the signal processing layer and store it in the database.
- Manage all the information relative to the users of the system and access control.
- Provide services which cover all the information needs of the subsequent layers described hereunder: vibration data access, machines profiles data access, report creation, etc.

Analysis layer This layer works as an intelligent continuous real time observer of every capture that arrives from each machine after being processed. Its design is the main

topic of this work and has been introduced in previous sections. An anomaly detection approach has been taken (see section 5.4). This means that this layer is not completely autonomous since, at least, a normal state historic of each machine has to be selected and a model has to be trained. Fault detection capabilities of the algorithms presented in this work are reported in the next chapter. If an anomaly is detected in data, this layer is in charge of reporting any potential fault found to the maintenance team. In order to do this, communication channels of the next layer are used.

Presentation layer This layer communicates all relevant information to the maintenance team in real time trough two channels. The first one is a user interface devised to be installed in a main control room. Figures 6.15 and 6.16 depict some of its main information modules. The main control application had the following characteristics, mixing real time Artificial Intelligence behavior and plan management functionalities:

- Central management of the maintenance of all the machines.
- Access to all the historic vibration captures for all the machines monitored.
- Multi-plant management: multiple plants can be monitored from a central data centre.
- Manual activation/deactivation of fault detection agents for each machine.
- Scheduling of the anomaly detection process for each machine: historic gathering, model building, fault detection monitoring phase. etc.
- Access to all raw vibration signal transformations (time and frequency domain) in order to contrast automatic fault reports.
- Fault treatment process control. A fault treatment task is triggered by an automatic fault detection obtained via ML techniques (see results in the next chapter).

Following the rules of the certification norms, all the modules are designed in an intuitive way in order to be accessible. All the captures are organized in order to allow tendency analysis and real time diagnosis. In addition, following the guidance of the certification, the system maintains secondary communication channels through email and cell phone (optional) in case a potential fault appears.

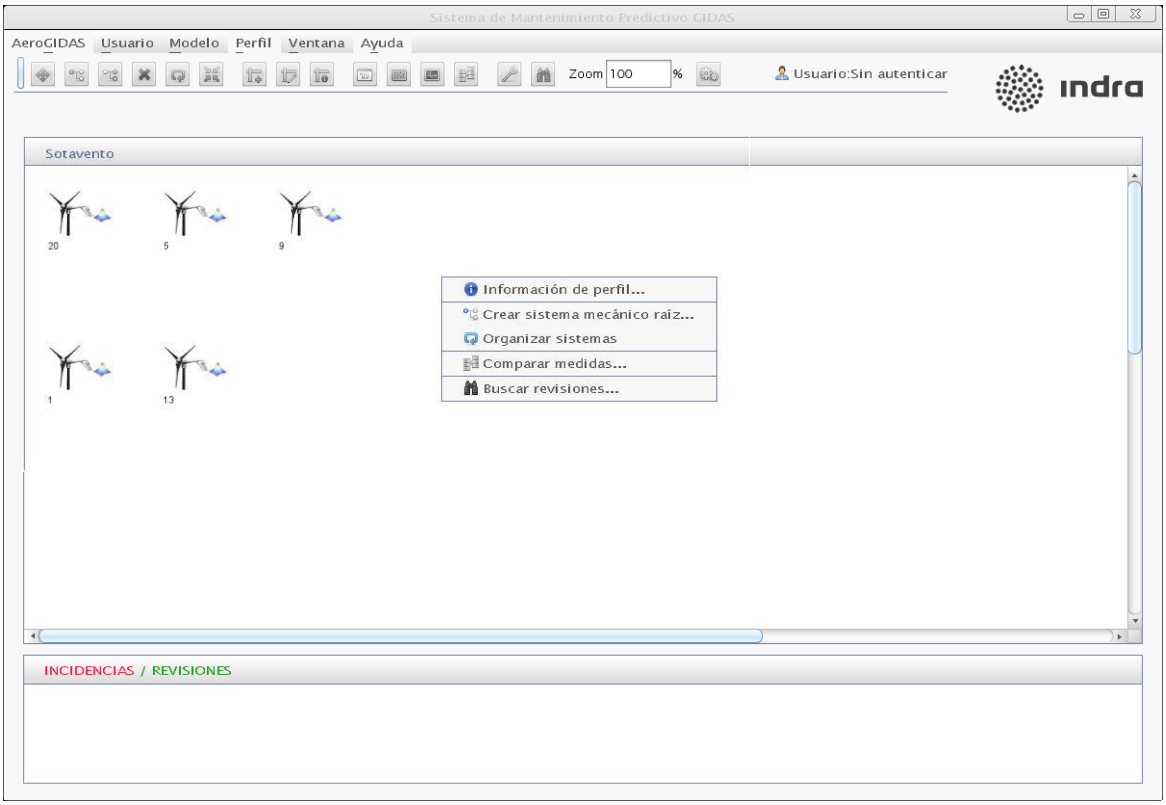


Figure 6.15: Main control module of GIDAS system.

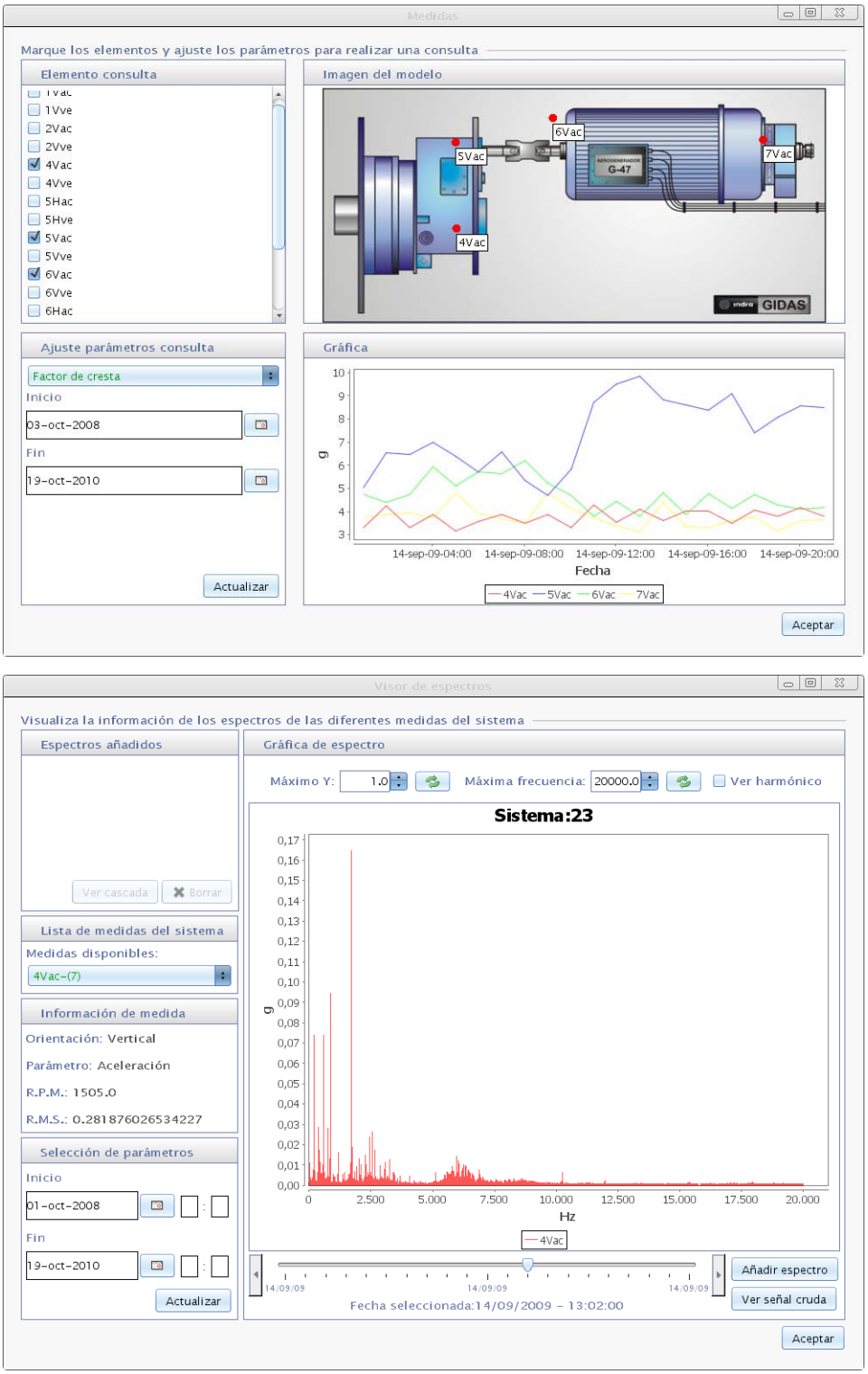


Figure 6.16: Analysis modules of GIDAS system.

6.6 Fault Detection in production environments: Sotavento and Production wind mill farm experiences

In order to assess the acceptance of the system in a real production environment and test their fault detection capabilities, two pilot experiences were carried out during this work in two wind farms in the north west of Spain:

- Sotavento: This plant is located in the county of Xermade and is made up of three public companies which constitute 51% of its shared capital, The Galician Energy Institute-INEGA, SODIGA GALICIA, S.R.C. Plc., The Energy Diversity and Saving Institute (IDAE), and four private companies representing the Galician electricity sector (Endesa, Enel Union Fenosa, Iberdrola Renewables and Energía de Galicia, Plc. Engasa). Sotavento's objective, according to the original idea is as well as the commercial exploitation, the achievement of two objectives that could be hardly raised by private enterprise: (a) being the "showcase" of the different wind technologies and (b) being a framework for the realization of I+D activities. This wind farm features a line of 24 wind turbines of the 5 different technologies. There are 9 different models of machine: Ecotecnia Model 640 (640 Kw), Gamesa Model G47 (660 Kw), Izar-Bonus Model 600 (600 Kw), Izar-Bonus Model 1.300 (1,300 Kw), Made Model AE-46/I (600 Kw), Made Series AE-800 (800 Kw), Made Model AE-61 (1,300 Kw), Neg Micon Model NM 48/750 (750 Kw) and Neg Micon Model NM 52/900 (900 Kw). Nominal power of the farm is 17,56 MW and is connected to the public power grid via a waste line of 9 Km. This is energy enough to supply electricity for 12.000 families and saves 68.000 barrels of petrol, avoiding the emission of 36.000 tons of CO_2 per year. Four machines of manufacturers Neg Micon and Izar-Bonus were monitored in the context of this work. GIDAS' main control application was used by the technicians from the main control station (see figure 6.17) and monitored the turbines for 8 months. No incidence in the machines was (correctly) detected by the system during this period.
- Production wind mill farm: This wind farm is located to the south of Sotavento in Forcarei, Lalín (Pontevedra). It is a fully commercial wind farm run by a main company in the sector. It produces a nominal power of 50 MW with 75 turbines Gamesa G47/660 (Power 660 kW, diameter 47 m). One of the machines was monitored by GIDAS and an incidence was detected by the system. The results of the ML fault detection approach for this case will be detailed in the

next chapter.

Figure 6.17: Sotavento's main control station and wind farm.

These pilot experiences were an opportunity to validate the acceptance of a ML based fault detection system in a real environment. As it was previously outlined, an APMS has worked in conjunction with a comprehensive maintenance program which involves a variety of human resources which the system has to communicate with. These human resources usually have very different backgrounds, from technicians to highly qualified engineers. The role of an APMS is twofold: (a) carry out automatically routine vibration analysis emulating an expert in order to save human resources' efforts and (b) act as a communicating agent, both reporting potential detected faults and allowing the coordination of the rest of human resources to manage any eventuality. In this sense, these pilot experiences have emerged the following further key points to take into account in the design of an automatic fault detection system:

- When reporting an alarm or potential fault, the message should be understandable by all the levels of the maintenance human resources. This implies that the selected intelligent technique should be able to provide a clear assessment of a potential fault without the necessity of understanding any of the internal details of the algorithm. Not complying with this requirement could lead to a low acceptance of the system as a valuable assistant on maintenance task.

- As we mentioned earlier, the system needs human intervention when building a normal state model of a machine. Since this has to be made on a per machine way, cumbersome models with a high number of hyper-parameters which need expert assessment should be avoided. APMS configuration should need much less effort than manual assessment itself in order to be accepted. If the algorithms need

some expertise to be applied, previous training programs or an ML assistant team should be considered if a successful deployment of the system is desired.

CHAPTER 7

Application of ML to industrial fault detection: Rolling element bearing fault assessment

This chapter is devoted to prove experimentally that fault detection, diagnosis and prognosis can be tackled through anomaly detection and on-line learning methods. Specifically, we will focus on the problem of rolling element bearing fault assessment since 80% of the problems of rotating machinery concentrate in this kind of components. In order to do that, the answer to each of the following questions will be extracted from the experimental results obtained from fault cases:

- Is vibration data adequate to detect and assess the condition of a mechanical component using an anomaly detection strategy?
- Is it possible to join traditional frequency domain transformation of vibration data and anomaly detection methods to build fault detection systems?
- Is it possible to propose alternative raw vibration feature extraction methods not based on frequency domain transformations?
- Is it possible to come up with an algorithm able to automatically diagnose faults in bearings?
- Is it possible to assess the evolution of the vibration level of a machine in order to detect when this level will become unbearable?

Experimental data in this chapter was obtained by simulation, laboratory tests and real vibration monitoring carried out by GIDAS System® (presented in the previous chapter). These data sets are described in section 7.2. By the end of the chapter, experimental results will bring light to these questions and prove worthwhile the methodologies proposed in this work. First of all, in the next section we give a description of rolling element bearing fault symptoms and how they evolve along the time.

7.1 Rolling element bearings fault detection and diagnosis

A rolling element bearing is a mechanical device that reduces the friction between a rotating shaft and two or more pieces connected to it. Its main components are: outer race, inner race, rolling elements and cage (see figure 7.1). Each time a defect on a surface of a component strikes another surface, a force impact is produced. If the rotational speed of the races is known, the impact repetition rates can be determined by the geometry of the bearing [168]. These repetition rates are called Bearing Characteristic Frequencies.
This section describes the procedure for identifying defects in anti-friction bearings by analyzing frequencies generated by the moving parts. Defects on bearing raceways, rolling elements and the cage generate different frequencies. The spectrum shape and amplitude, in addition to the time domain signal, are useful in identifying the nature, location, combination and size of the defects. Success or failure in diagnosing bearing defects often depends on the selection of the proper transducer. The discussion of transducer selection of previous chapter applies to this chapter. Roller bearings rotating at, for example, 1200 Revolutions per minute (RPM) can generate harmonics in the range of 3000 Hz or more when a fault occurs. Thus, an accelerometer should be used in this case. In the case of very low frequency machines, generated frequencies could be under the range of 100 Hz, so a displacement probe is more suitable. All the cases that will be covered in this chapter belong to the group of 1200 RPM or more, so acceleration signals are used in all cases.

7.1.1 Bearing Characteristic Frequencies

In order to understand the relationships among the different rotating elements of a bearing, the equations describing the relative speeds must first be developed. There are five main frequencies that a machine with a defective bearing can generate. Hereunder we explain the motion equations which are involved in each type of fault and the main equation that determines its descriptive frequency (see figure 7.2 for the main identities):

- Rotating unit frequency or speed (S): This is the speed at which the moving part mounted inside the bearing is spinning. Usually, the shaft inside the inner race

is the moving part and the outer race is fixed on the housing. In this case, the inner race spins at the same speed as the shaft.

- Fundamental Train Frequency (FTF): The train or cage frequency is equivalent to the angular velocity of the individual ball centers. The linear velocity of the balls can be expressed as:

$$v_c = \frac{v_i + v_o}{2} \tag{7.1}$$

The angular velocity is defined as the linear velocity v_c divided by the radius r of the trajectory. Therefore,

$$FTF = w_c = \frac{(v_i + v_o)/2}{P_d/2} \tag{7.2}$$

where w_c is the angular velocity for the ball center or cage. Since $v_i = r_i * w_i$ and $v_o = r_o * w_o$, we use these expressions to substitute v_i and v_o in the previous expression. Final expression for FTF results:

$$\begin{aligned} FTF &= \frac{w_i\left(\frac{P_d - B_d cos(\phi)}{2}\right) + w_o\left(\frac{P_d + B_d cos(\phi)}{2}\right)}{P_d} && (7.3) \\ &= \frac{1}{2}\left[w_i\left(1 - \frac{B_d cos(\phi)}{P_d}\right) + w_o\left(1 + \frac{B_d cos(\phi)}{P_d}\right)\right] && (7.4) \end{aligned}$$

It is important to note a subtlety in this equation. If the roller elements contact the races in an angle $\phi \neq 0$, then the point of the races which is spinning is not the base but the point of contact itself. That is the rationale under the $\frac{B_d cos(\phi)}{2}$ term.

- Ball pass frequency of the outer race (BPFO): This is defined as the frequency of the balls passing over a single point on the outer race. The BPFO can be described as the number of balls multiplied by the difference between cage w_c and outer race w_o frequencies or,

$$BPFO = N_b|w_c - w_o| \tag{7.5}$$

where N_b is the number of rolling elements of the bearing; this expression can be rewritten as

$$\begin{aligned} BPFO &= \left|N_b\left[\frac{1}{2}\left(w_i\left(1 - \frac{B_d cos(\phi)}{P_d}\right) + w_o\left(1 + \frac{B_d cos(\phi)}{P_d}\right)\right) - w_o\right]\right| \\ &= \left|N_b\left[\frac{w_i}{2} - \frac{w_i B_d cos(\phi)}{2P_d} + \frac{w_o}{2} + \frac{w_o B_d cos(\phi)}{2P_d} - w_o\right]\right| \\ &= \left|\frac{N_b}{2}(w_i - w_o)\left(1 - \frac{B_d cos(\phi)}{P_d}\right)\right| && (7.6) \end{aligned}$$

- Ball pass frequency of the inner race (BPFI): This frequency is defined as the frequency of the balls passing over a single point on the inner race. The BPFI can be described as the number of balls multiplied by the difference between the frequencies of the inner race w_i and the cage,

$$BPFI = N_b|w_i - w_c| \tag{7.7}$$

which can be rewritten as,

$$\begin{aligned} BPFI &= \left| N_b \left[w_i - \frac{1}{2} \left(w_i(1 - \frac{B_d cos(\phi)}{P_d}) + w_o \left(1 + \frac{B_d cos(\phi)}{P_d} \right) \right) \right] \right| \\ &= \left| N_b \left[w_i - \frac{w_i}{2} + \frac{w_i B_d cos(\phi)}{2P_d} - \frac{w_o}{2} - \frac{w_o B_d cos(\phi)}{2P_d} \right] \right| \\ &= \left| \frac{N_b}{2}(w_i - w_o) \left(1 + \frac{B_d cos(\phi)}{P_d} \right) \right| \end{aligned} \tag{7.8}$$

- Ball spin frequency (BSF): The angular velocity of a ball about its center can be expressed in two different ways. First, considering the linear velocity of a point on the inner race in contact with the ball surface, or on the other hand, considering the linear velocity of a point on the outer race in contact with the ball surface. Both lead to the same expression, thus only the first option will be detailed. The linear velocity v_b of a point on the ball surface is given by,

$$v_b = (w_i - w_c) r_i \tag{7.9}$$

where r_i is the radius of the inner race. The ball angular velocity or ball spin frequency is then,

$$BSF = \left| (w_i - w_c) \frac{r_i}{r_b} \right| \tag{7.10}$$

where r_b is the radius of the ball; using the geometrical identities of figure 7.2,

$$BSF = \left| (w_i - w_c) \frac{(P_d - B_d cos(\phi))/2}{B_d/2} \right| \tag{7.11}$$

which can be expressed as,

$$\begin{aligned} BSF &= \left| \left[w_i - \frac{1}{2} \left(w_i \left(1 - \frac{B_d cos(\phi)}{P_d} \right) + w_o \left(1 + \frac{B_d cos(\phi)}{P_d} \right) \right) \right] \right. \\ &\times \left. \frac{(P_d - B_d cos(\phi))}{B_d} \right| \\ &= \left| \left[w_i - \frac{w_i}{2} + \frac{w_i B_d cos(\phi)}{2P_d} - \frac{w_o}{2} - \frac{w_o B_d cos(\phi)}{2P_d} \right] \frac{(P_d - B_d cos(\phi))}{B_d} \right| \\ &= \frac{P_d}{2B_d}(w_i - w_o) \left(1 - \frac{B_d^2 cos(\phi)^2}{P_d^2} \right) \end{aligned}$$

The important frequency in this case is 2×BSF because a fault in the ball would hit alternately the inner and outer race in each spin. This generates two times the BSF because the timing for each strike is exact and occurs when the roller rotates half a revolution.

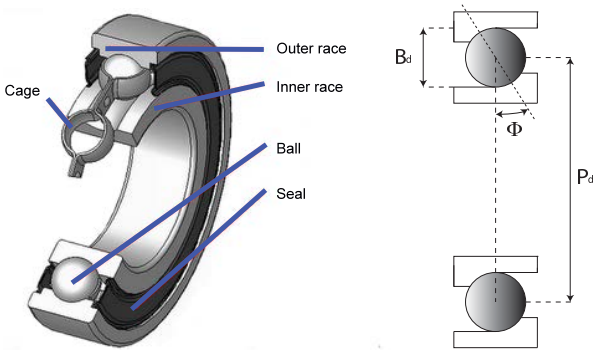


Figure 7.1: Main geometry of a rolling element bearing

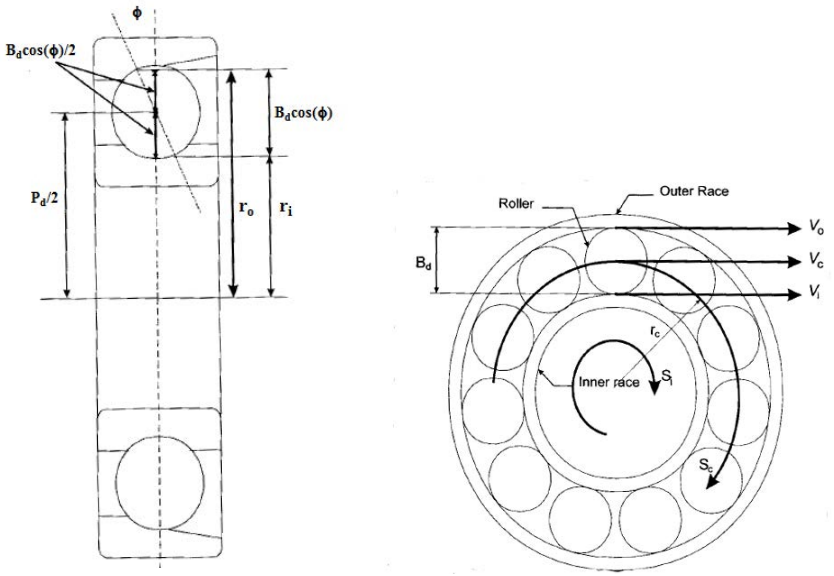


Figure 7.2: Main identities of a rolling element bearing

All of these ideal bearing frequency formulas are based on the assumption of pure rolling contact between rollers and races. Any small deviation resulting from any slipping of these surfaces would produce somewhat lower values than the ones of the above equations. When some looseness is involved, the spectral lines at the bearing frequencies can be wide-banded. The outer race frequency could not appear clear if the bearing is loose in the housing or if the fault is not aligned with the load zone. In addition, small

changes in the contact angle during functioning could make the generated frequencies vary from the theoretical ones. Despite these variations in practice, the analysis of the power spectra around these frequencies can reveal with high accuracy the nature of the defect in many phases of it. Table 7.1 summarizes the main relations between characteristic bearing vibration frequencies and fault diagnosis used in the traditional spectral analysis of vibrations [236]. In the next section, the phases which can appear for a bearing defect are described.

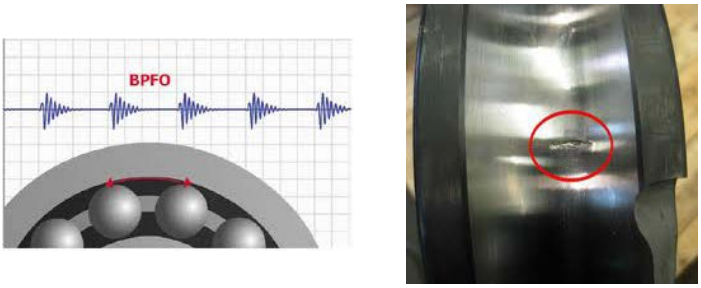


Figure 7.3: Impact of a fault for an outer-race fault (equivalent for an inner-race fault with BPFI frequency) and real wear of an inner-race.

Table 7.1: Bearing Vibration Frequency Characteristics.

Bearing faults	**Frequencies in Power Spectrum**	**Description**
Rolling elements	$2 \times BSF$, $BPFO$, $BPFI$	Modulated by $2 \times BSF$ or FTF.
Outer raceway	$BPFO$	Harmonics may be found.
Inner raceway	$BPFI$	Harmonics may be found. Inner race faults are typically modulated by F_s (frequency of the shaft).

Rolling element bearing fault phases

Defects can occur in any of the parts of the bearing and will cause both medium and high-frequency vibrations. In fact, the severity of the wear keeps changing the vibration pattern until a final breakdown occurs, as we will see further in this section. Raceways and rolling element defects are more easily detected than the ones in the cages. Though there are many techniques available to detect where defects are occurring, early detection and prediction of when the bearing defect will turn into a functional failure are

still not fully solved.
The power spectrum for bearing defects can be split into four zones, where we will notice the changes as bearing wear progresses. These zones are:

- Zone A: machine revolutions per minute (RPM) and harmonics zone (0 - 200 Hz).
- Zone B: bearing defect frequencies zone (200 - 500 Hz).
- Zone C: bearing component natural frequencies zone (500 - 5000 Hz).
- Zone D: high-frequency-detection (HDF) zone (beyond 20 KHz).

The defects in a bearing will present changes in the power spectrum in four phases (see figure 7.4):

- Stage 1: The first indications of bearing wear show up in the ultrasonic frequency band which is situated in the range of approximately 20-60 KHz. These frequencies should be evaluated by specialized high-frequency detection techniques such as gSE (Spike Energy), PeakVue, etc. since the vibration pick ups are not able to reach this band of frequencies. The high cost of these techniques motivates that this stage is usually discarded in many practical situations.
- Stage 2: In this next stage, the fatigued raceways or balls begin to develop minute pits. The contact of these pits with other surfaces (ball against a race pit, defective ball against the race, etc.) start to generate the ringing of the bearing component natural frequencies that predominantly occur in the 500-5000 Hz range. This effect consist on the periodic free vibration response of the component, as we described in the previous chapter and was determined in the seminal paper by McFadden [167] (see figure 7.3). Since the period of the spikes is equal to the characteristic frequency of the domain, the effect resembles a train of impulses which modulates in amplitude the natural frequency, which acts as the modulated frequency. Thus, the defect appears in the power spectrum sidebands around the natural frequency at a distance equal to the characteristic frequency corresponding to the present defect (see figure 7.4). This fact can be used to diagnose the defect in an early stage, as we will experimentally demonstrate in the following sections. Also, high-frequency components may also double its amplitude when compared to the readings during stage 1.

- Stage 3: As we enter the third stage, the characteristic frequencies and their harmonics are visible in the power spectrum. These may appear with a number of sidebands. Wear is usually now perfectly visible on the bearing and may expand through to the edge of the bearing raceway. The minute pits of the earlier stage are now developing into bigger pits and their numbers also increase. This stage is already dangerous and it is usually advised to replace the bearing at this stage. Some studies indicate that after the third stage, the remaining bearing life can be from 1 h to 1 % of its average life. Thus, it is desirable that a detection and diagnosis technique would be able to anticipate the fault before it reaches this stage, since the development of its defects will be exponential afterwards and in a very short period, the debris from the bearing or its own breakage can cause a fatal failure of the rest of the components.

- Stage 4: In the final phase, pits merge with each other, creating rough tracks and spalling of the bearing raceways or/and rolling elements. The bearing is in a severely damaged condition now. A generalized increase of all the vibration components, even the 1x (main speed of the shaft), characterizes this phase. Discrete bearing defect frequencies and bearing component natural frequencies actually begin to merge into a random, broadband high-frequency "noise floor". By this time, the bearing will be vibrating excessively and the whole machine is under serious danger of breakdown; it will be hot and making lots of noise. If it is allowed to run further, the cage will break and the rolling elements will go loose and run into each other until the machine trips on overload. It is very likely that, if this stage is reached, there will be serious damage to the shaft and the area around the bearing.

As it can be extracted from the description of these phases, with the transducers used (accelerometers), fault detection in stage 2 and as early as possible is desirable for both a human practitioner or an automatic fault detection system. If this behavior is obtained, monitored machinery will never, in principle, reach a dangerous stage and the component could be repaired in a cost effective way. The aim of the experiments in this chapter is to demonstrate that this behavior could be obtained applying the anomaly detection methodology described in the last section of chapter 5, combined with ML anomaly detection algorithms and signal preprocessing. In order to do this, we will use cases under different experimental settings that will be described in the next section.

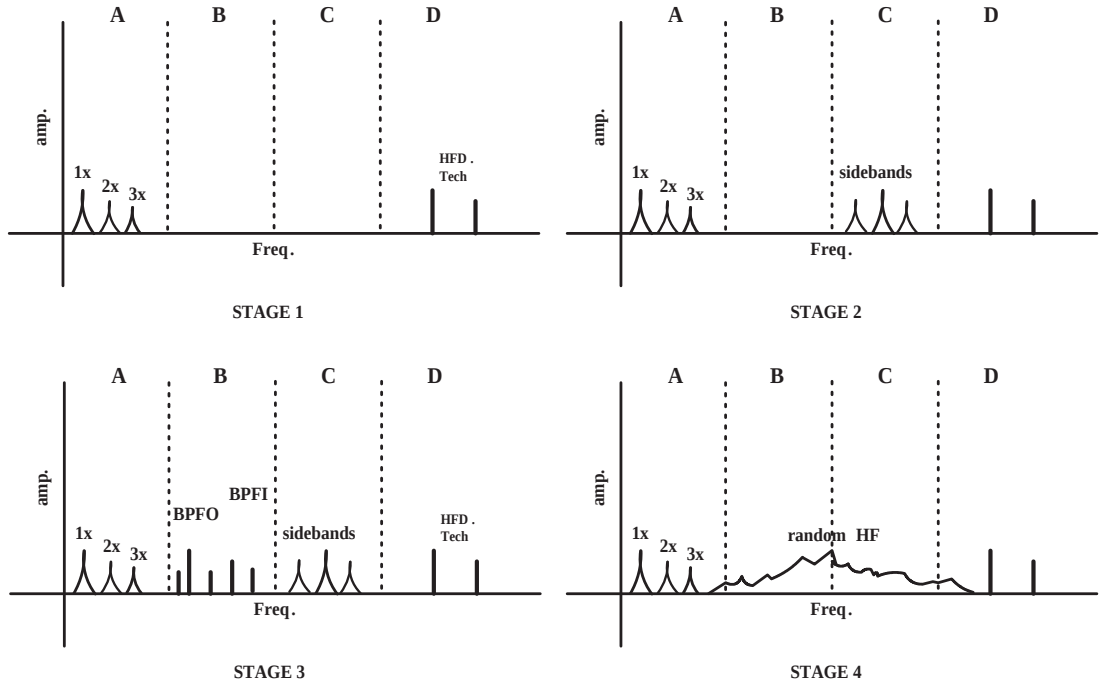


Figure 7.4: Qualitative evolution of the power spectrum of a bearing fault.

7.2 Rolling element bearing fault assessment: study cases

Obtaining vibrations captures of faulty machines is not an easy task and few datasets are available. Laboratory tests are very expensive to produce and monitoring actual production machines is not very accessible due to production or even manufacturers restrictions (manufacturer guarantee can be lost by the owner if any modification is made). As a result, few disclosed data sets are available. Firstly, common features of all the experimental cases to be used in this work are described. In all cases, the bearings will be mounted in the most common configuration, and this affects the general equations of characteristic frequencies. Namely, the outer race is always fixed to the housing and the inner race turns freely at the shaft speed. Under this conditions, with $w_i = S$ (angular speed of the shaft) and $w_o = 0$, the equations of the characteristic frequencies are (where S is the shaft speed):

$$FTF = \frac{1}{2}S\left[\left(1 - \frac{B_d cos(\phi)}{P_d}\right)\right] \tag{7.12}$$

$$BPFO = \left|\frac{N_b}{2}S\left(1 - \frac{B_d cos(\phi)}{P_d}\right)\right| \tag{7.13}$$

$$BPFI = \left| \frac{N_b}{2} S \left(1 + \frac{B_d cos(\phi)}{P_d} \right) \right| \tag{7.14}$$

$$BSF = \frac{P_d}{2B_d} S \left(1 - \frac{B_d^2 cos(\phi)^2}{P_d^2} \right) \tag{7.15}$$

When studying a specific component in the following cases, these equations will be used in order to calculate the characteristic frequencies. In all real and laboratory cases, accelerometers are used to capture the vibrations. This pickups limit our studies to a band up to 10Hz - 20 kHz depending on the instrumentation of each experiment. Mountings are described hereunder for each case. The data sets are different in their nature and they are increasingly difficul from simulated data with low noise sources to real life detection cases which present all the difficulties that can appear in a real fault detection scenario.

7.2.1 Case 1: UNSW Simulator Data

In [205] researchers from the University of New South Wales (UNSW) presented a simulation model for a gearbox test rig, in which a range of bearing faults can be implemented. This simulator was designed to facilitate the development of diagnostic and prognostic techniques for rolling element bearings in real systems. Faults can be implemented under different operating conditions rather than waiting for them to occur naturally, or alternatively having them seeded in the laboratory. Even though the modelling of the whole gearbox has to be an approximation, the simulations obtained in [205] proved to be useful for reproducing typical fault signals from gearboxes and to test new diagnostic algorithms. Simulated signals showed quite a similar pattern to that observed in their actual measured counterparts. Such fault simulation is very valuable in machine diagnostics and, for this work, it allows us to produce signals with well-defined characteristics.
Bearing faults sometimes manifest themselves by their interaction with meshing gears, and to simulate this it is necessary to model a whole system of gears and shafts supported by bearings. A model or an experimental test rig was built through the incorporation of a time-varying, non-linear stiffness bearing model into a previously developed gear model. The incorporated bearing model is based on Hertzian contact theory, which relates the raceway displacement to the bearing load, and also accounts for the slippage between the elements. It has the capacity to model localized spalls (inner race, outer race and rolling elements), though there is a further extension [204] in which the

simulation of larger faults is discussed. As the interest in our work is incipient fault detection, the version in [205] is used. Figures 7.5 and 7.6 depict respectively the actual test rig and the diagram of the 34-DOF dynamic model that simulates the system. Captures in normal state and subsequently under outer race fault were simulated. The fault was increased from 0 micrometers of depth and 0 mm width to 200 micrometers depth and 0.5 mm of width. Thanks to the capacity of controlling the fault depth and width, this simulator will be used to assess the coherence between vibration signals and fault severity and answer the first question raised at the beginning of the chapter on whether vibration data is adequate to detect and assess the condition of a mechanical component.

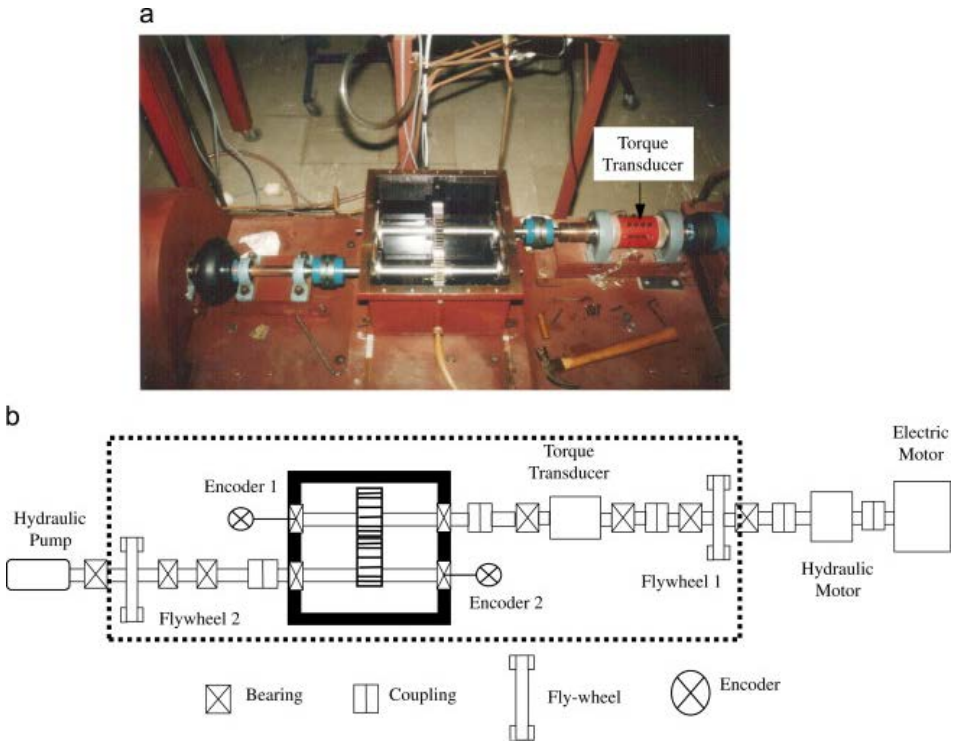


Figure 7.5: UNSW fault test rig photo and scheme (reproduced with permission of the authors)

7.2.2 Case 2: Laboratory data I

In order to show the adequacy of the proposed methodology for real life fault detection, the bearing dataset provided by the Center for Intelligent Maintenance Systems (IMS),

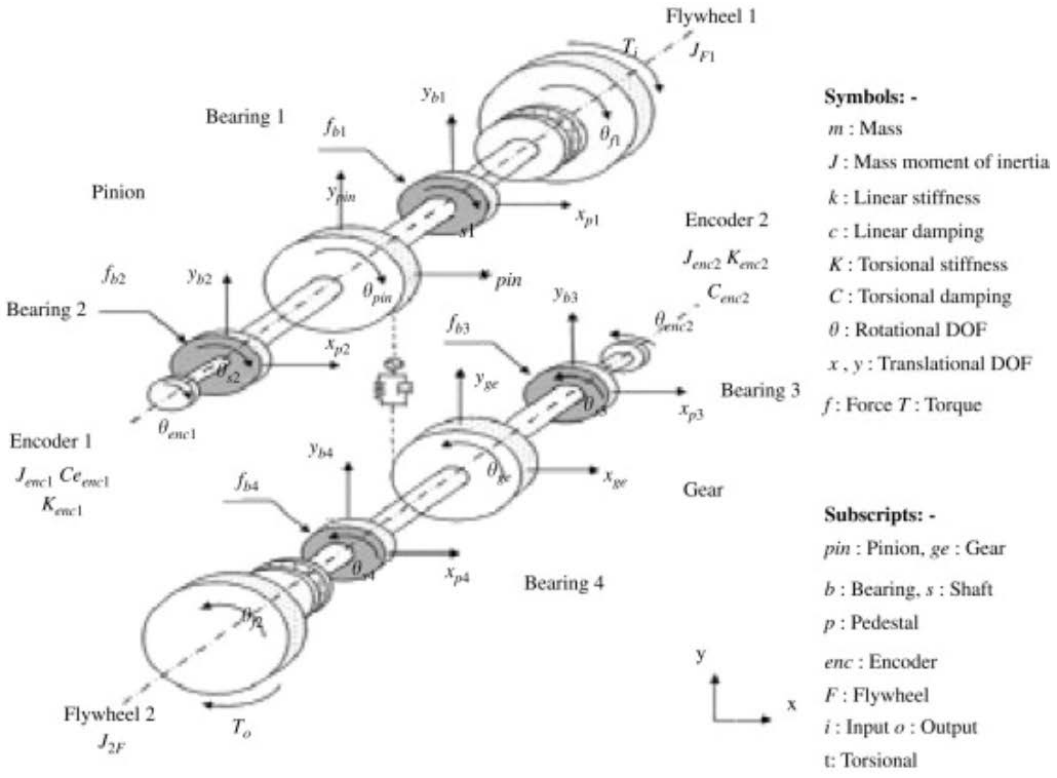


Figure 7.6: 34-DOF modeling of the gearbox (reproduced with permission of the authors)

University of Cincinnati was used [141]. For obtaining the data, four bearings were installed in one shaft. All bearings were forced, lubricated and accelerometers were installed in each of them. Rotational speed was kept constant at 2000 rpm and a 6000lb radial load was placed onto the shaft and bearing by a spring mechanism. On each bearing two PCB 353B33 High Sensitivity Quartz ICP Accelerometer were installed for a total of 8 accelerometers (one vertical Y and one horizontal X on each). All failures occurred after exceeding designed life time of the bearings that is more than 100 million revolutions. Figure 7.7 depicts the structure of the installation used for the experiment.

In this work two datasets of the database are used:

- Experiment a: Recording was carried out between 12/02/2004 10:32:39 and 02/19/2004 06:22:39 every 10 minutes. At the end of the test-to-failure experiment an outer race failure occurred on bearing 1.
- Experiment b: Recording was carried out between 04/03/2004 09:27:46 and 04/04/2004 19:01:57 also every 10 minutes. At the end of the test-to-failure experiment an outer race failure occurred on bearing 3.

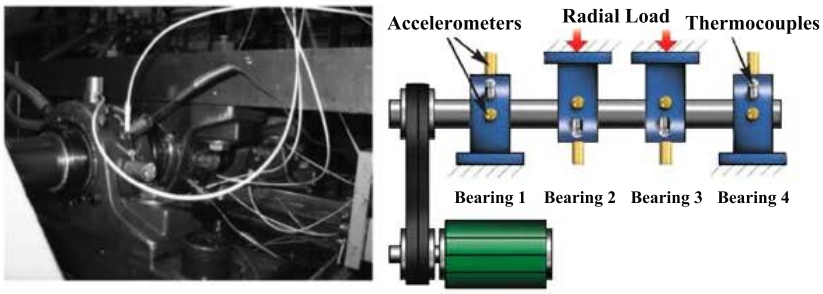


Figure 7.7: Bearing test rig for the test-to-failure experiment: (a) photo; (b) installation schema.

The nature of the fault induced in this experiment was slow so the expected behavior of the system in this case is: (a) not to give false alarms during normal operation, (b) early detection of a change of vibration signature when a fault appears and (c) qualitative indication of the exponential evolution of the induced crack until the machine stops working. In addition, the two fault cases are very similar, same type of fault on the same type of component and under the same conditions. Thus, results on any of the two datasets can be extrapolated to the other one. These datasets are used in the following sections to assess the accuracy of fault detection, diagnosis and prognosis strategies.

7.2.3 Case 3: Laboratory data II

This third example was carried out with data obtained from the data set of rolling-element bearings provided by the Case Western Reserve University [153]. The experimental setup, shown in Fig. 7.8, consisted of a Reliance Electric 2HP IQPreAlert induction motor connected to a dynamometer. Single point faults of 0.007, 0.014 and 0.021 inches in size were "seeded" only into the drive-end bearing of the motor using an electrical discharge machine. An accelerometer was placed at the drive end of the motor housing (12 o'clock position) to acquire the vibration signals from the bearing. All signals were recorded for motor loads of 0 to 3 horsepower at a sampling frequency of 48 kHz. The speed was held constant at 1740 RPM. Installed bearings have 9 balls, a pitch diameter of 1.537 in., a ball diameter of 0.3126 in. and a null contact angle. With this information and the equations presented in section 7.2, we are able to calculate the characteristic fault frequencies (shown in Table 7.2) .

Table 7.2: Characteristic fault frequencies of 6205-2RS SKF bearing.

BPFI	**BPFO**	**FTF**	**BSF**
157 Hz	104 Hz	11.6 Hz	68.5 Hz

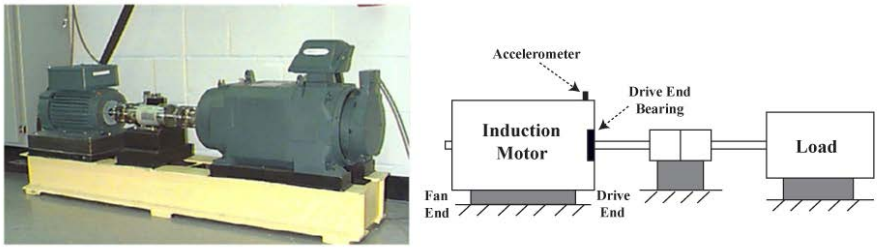


Figure 7.8: Photography and schematic description of the experimental system.

The following faults are used in the experiments: a 0.007 inches fault in the outer race, a 0.007 inches fault in the inner race and a 0.021 inches fault in a ball. We have selected the smaller size faults in the rings in order to treat the most difficult incipient fault cases. Ball faults are the ones which appear less clearly in data so bigger fault sized had to be taken. Due to the well documented faults registered in this dataset, it will be used in section 7.5 to test a proposed diagnosis strategy base on one-class ν-SVM. Unfortunately, the size of the datasets does not allow us to extract relevant conclusions for fault detection analysis, so this dataset is not used in sections devoted to fault detection performance assessment.

7.2.4 Case 4: Real Scenario, wind mill power turbines

This last dataset consist on a fault case occurred in a wind mill situated in the production wind farm during the pilot installation of GIDAS software (see previous chapter). The wind turbine was composed of a METSO PLH-310XG1 gearbox and an INDAR NCR-400-X/4 generator with 660 KW of power and which mounted two FAG 6226/C3 bearings in input and output extremes. The machine was monitored from 11th of March of 2010 until 11th of July of 2010 using GIDAS monitoring system when a breakdown occurred in the machine and a reparation was needed. In this case, we used as extra input data the revolutions per minute of the machine during the capture, as this machine was of variable speed and the vibrational signature changes significantly depending on this parameter. The nature of the failure was impossible to determine, so this dataset will only be used to assess fault detection strategies accuracy in a real production scenario.

Before continuing to the description of experimental results, we detail One Class ν-Support Vector Machines. This algorithm was included in the first prototype of GIDAS software and so will be used as the state-of-the art anomaly detection standard in the following sections.

7.3 One Class ν-Support Vector Machines

One class ν-Support Vector Machines [208] [210] are intended to solve the following problem: try to obtain a function that captures regions in input space where the probability density support lays. In doing so, the obtained function f complies with the following condition: given a probability density function P, if a previously unseen data point $\mathbf{x}$ is generated using P, with a predefined probability level α, f takes a positive value. In order to adjust f, we are given a set of normal data points $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l\}$ which have been generated i.i.d. from the normal state probability distribution P which we want to characterize. Mathematically, one-class ν-SVMs can be formulated as a

convex optimization problem as follows:

$$\begin{array}{ll} \min\limits_{\mathbf{w},\ \xi_i,\ \rho} & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu l}\sum_i \xi_i - \rho \\ \text{subject to} & \mathbf{w}\cdot\phi(\mathbf{x}_i) \geq \rho - \xi_i \\ & \xi_i \geq 0; \end{array} \tag{7.16}$$

where $\mathbf{w} \in F$ is the weight vector of the classifier hyperplane, $\mathbf{x}_i \in \mathbb{R}^n$ are the input vectors, $\xi_i \in \mathbb{R}$ are slack variables, $\rho \in \mathbb{R}$ is the bias term of the classifier, $\phi(\mathbf{x}_i) : \mathbb{R}^n \mapsto F$ represents a non linear function that maps vectors in input space to a feature space F and $\nu \in (0, 1]$ is a parameter whose meaning will become clear later. Due to the fact that nonzero slack variables ξ_i penalize the objective function, the pair $\mathbf{w}$ and ρ that solve the problem will give a decision function

$$f(\mathbf{x}) = sgn(\mathbf{w}\cdot\phi(\mathbf{x}) - \rho) \tag{7.17}$$

that will be positive for data points similar to the examples in the training set which represent the "normal" support.
Using multipliers $\alpha_i, \beta_i \geq 0$, we introduce the Lagrangian of the primal in (7.16)

$$\begin{array}{l} L(\mathbf{w}, \xi_i, \alpha_i, \beta_i) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu l}\sum_i \xi_i - \rho \\ \\ -\sum_i \alpha_i((\mathbf{w}\cdot\phi(\mathbf{x}_i)) - \rho + \xi_i) \\ \\ -\sum_i \beta_i\xi_i \end{array} \tag{7.18}$$

Using the so called Wolfe dual of convex constrained problems [82], if we take the derivative of the Lagrangian in (7.18) with respect to the primal variables and equal them to zero we obtain

$$\begin{array}{l} \mathbf{w} = \sum_i \alpha_i\phi(\mathbf{x}_i), \\ \\ \alpha_i = \frac{1}{\nu l} - \beta_i \leq \frac{1}{\nu l}, \\ \\ \sum_i \alpha_i = 1 \end{array} \tag{7.19}$$

and subtituting them in the primal (7.16), we obtain the dual problem:

$$\begin{array}{ll} \max\limits_{\alpha} & -\frac{1}{2}\sum_{ij} \alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subject to} & 0 \leq \alpha_i \leq \frac{1}{\nu l} \\ & \sum_i \alpha_i = 1 \end{array} \tag{7.20}$$

Due to the fact that normal data points in feature space $\phi(\mathbf{x}_i)$ are involved in (7.20) only in terms of their dot products, we can approximate their dot product $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ by a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$. This so called 'kernel trick' property of support vector machines (SVM) methods allows us to obtain functions that approximate complex probability density supports without the need to explicitly map normal patterns into feature space. Common choices for $k(\mathbf{x}_i, \mathbf{x}_j)$ are RBF, polynomial, sigmoid, etc (see [212]). In this work we use the RBF or gaussian kernel function which has the following form:

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \tag{7.21}$$

where σ is the width of the kernel function and must be set by the user as a hyperparameter. In the optimum, Karush-Kuhn-Tucker complementary condition and the following are fulfilled

$$\begin{aligned} &\alpha_i \left((\mathbf{w} \cdot \phi(\mathbf{x}_i)) - \rho + \xi_i \right) = 0, \forall i \\ &\beta_i \xi_i = 0, \forall i \\ &\beta_i = \tfrac{1}{\nu l} - \alpha_i \end{aligned} \tag{7.22}$$

so we can obtain ρ from any data point whose $\alpha_i \in (0, \frac{1}{\nu l})$ following

$$\rho = \mathbf{w} \cdot \phi_i(\mathbf{x}_i) = \sum_{j \in SV} \alpha_j k(\mathbf{x}_j, \mathbf{x}_i) \tag{7.23}$$

where SV is the set of data points whose corresponding $\alpha_j > 0$, which are called Support Vectors. In the optimum, only a small fraction of the input data points will have a $\alpha_j > 0$, which gives a sparse definition of the final function f. This is an advantage over other methods like Parzen Density Estimators, since we define the support of the normal distribution only in terms of a small portion of the input data set. Thus, in kernel space, the final decision formula has the following form

$$f(\mathbf{x}) = sgn \left(\sum_{x_i \in SV} \alpha_i k(\mathbf{x}_i, \mathbf{x}) - \rho \right) \tag{7.24}$$

There are additional properties of one-class ν-SVMs that play an important role in its application to fault detection problems

Proposition 1: Assume the solution of (7.16) satisfies $\rho \neq 0$. The following statements hold:

1. ν is an upper bound on the fraction of outliers.
2. ν is an lower bound on the fraction of SVs.
3. Suppose the normal data were generated i.i.d. from a distribution $P(\mathbf{x})$ which does not contain discrete components. Suppose, moreover, that the kernel is analytic and non-constant. With probability 1, asymptotically, ν equals both the fraction of SVs and outliers.

The formal proof of this statements can be consulted in [210]. This statement has practical implications for fault detection applications. The parameter ν that we have to adjust in order to estimate the "normal" model represents the estimation of the maximum number of spurious or abnormal vibration captures that are expected to appear in the training set, which can be estimated from past cases in order to help us to set this hyper-parameter.

7.4 Fault detection: experimental results

In this section we explore the anomaly fault detection strategy presented in the last section of chapter 5 when applied with both state-of-the art and anomaly detection techniques presented in chapter 3.

7.4.1 Vibration data and fault severity coherence

In this section, the first question posed at the beginning of this chapter is treated: *Is vibration data adequate to detect and assess the condition of a mechanical component using an anomaly detection strategy?*. In order to have a positive answer to this question, we should be able to observe a clear divergence between vibrations produced under normal and abnormal conditions. Furthermore, a correlation between data divergence and fault severity would also be desired. If this second property is present in vibration data, we could ensure that if that divergence is captured by our anomaly detection

strategy as soon as possible, fault severity would be indeed still incipient. Otherwise, if fault severity is not positively correlated with data divergence, an anomaly detection strategy would be invalidated for incipient fault detection.
Since we are not able to determine the conditions of machinery in laboratory and real scenarios for each point of time, simulated data is used using the simulator of case I. Specifically, the following simulation was conducted: (a) firstly 65 captures in normal state and subsequently other 65 captures under outer race fault were simulated. The fault was linearly increased from 0 micrometers of depth and 0 mm width to 200 micrometers depth and 0.5 mm width. The feature vector was constructed as follows: the Root Mean Squared (RMS) energy of the raw signal and the Power Spectrum in the 0-20kHz band (calculated using the Fast Fourier Transform (FFT)) were calculated; subsequently, the power spectrum was divided in subbands of 1000Hz and the energy of each subband (area under the curve of each subband of the power spectrum) was calculated giving a feature vector of 21 features. Figures 7.9 and 7.10 represent respectively the vibration power spectrum in normal state and with a 0.3 mm fault. It can be seen the incipient activity in 15000 Hz band.

In order to assess the divergence of faulty data we have chosen to use the one-class ν-SVM previously presented with a RBF kernel function.A normal model was trained with a subset of the captures under normal conditions and subsequently it was used to classify the remaining captures. Figure 7.11 shows the behavior of the one-class ν-SVM along the fault progress, where the dot represents the point where the fault actually started. The value of the y axis is the argument of equation 7.24. One-class ν-SVM captures the region of the input space which normal data belongs to and the more divergent the new data is the more negative this value becomes. It can be observed that the SVM has a delay in detecting the fault due to the poor signal to noise ratio at the beginning of the deviation, but once it detects the change it is able to characterize its development. In Figure 7.12 it can be seen the "normality likelihood" of the SVM versus the width of the fault. As it can be observed, the sensitivity of the SVM starts at 0.19 mm of width and from this point on, its fault characterization has a linear correlation with the fault severity up to a 0.97 Pearson's correlation coefficient.

Although based on a specific methodology, two main conclusions can be extracted from this experiment: (a) when a fault is very small there is not a clear sign of it in vibration data so a delay in the detection is expected and (b) there is a correlation between (a) fault severity and (b) the divergence of fault state vibration captures with respect to normal state ones. This conclusions justify the application of an anomaly

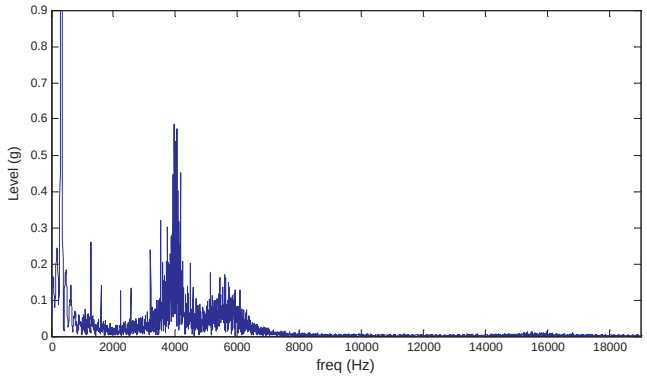


Figure 7.9: Normal state spectrum of simulated data.

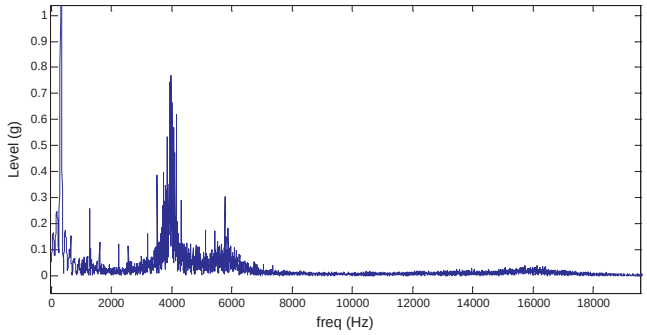


Figure 7.10: Incipient fault spectrum of simulated data.

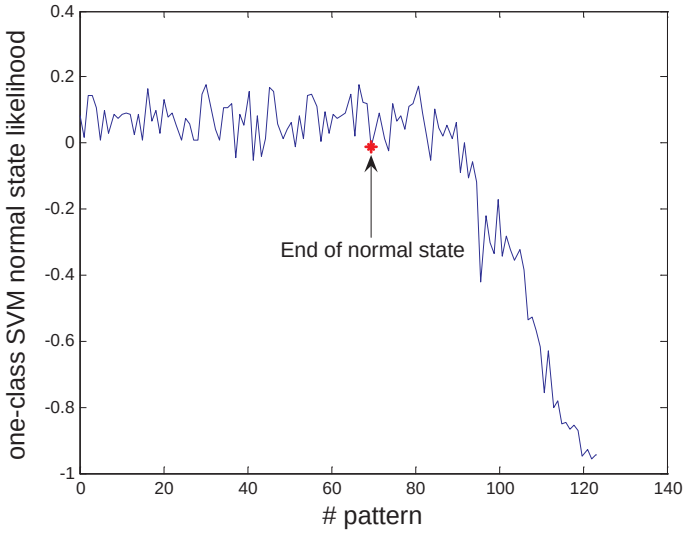


Figure 7.11: Fault detection for the simulated fault growth case.

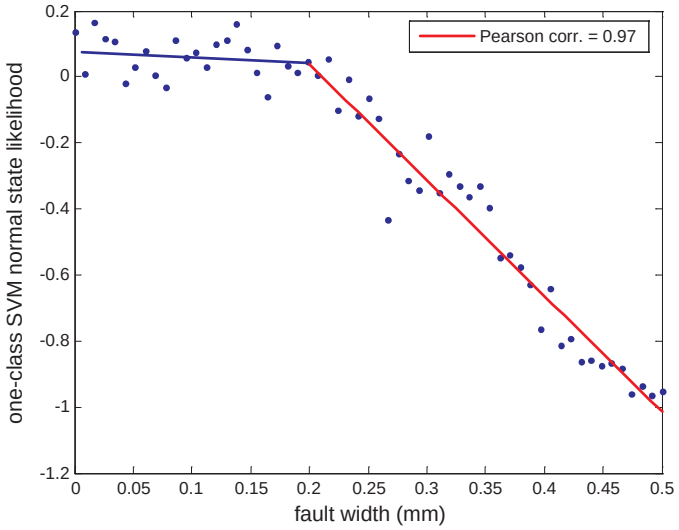


Figure 7.12: Correlation between nomal state assessment of one-class ν-SVM and fault width.

detection strategy to the fault detection problem based on vibration signatures and highlights the necessity to explore more in depth the following aspect in order to design such a system: improve both feature extraction and anomaly detection techniques in order to reduce the detection delay maintaining a reduced false positive rate. These aspects are explored in the following sections.

7.4.2 Fault detection via frequency domain transformations

Once we have brought light to the first question of this chapter, we move on to the next step: *Is it possible to join traditional frequency domain transformation of vibration data and anomaly detection methods to build fault detection systems?* In this section we test the detection accuracy of state of the art anomaly detection algorithms and the anomaly detection algorithm proposed in section 3.1.2.6. In this case, the anomaly detection strategy proposed in section 5.4 is used and the feature vector consist on: (a) the Revolutions per Minute (RPM) at which the system is working and (b) the level of the signal and each sub-band of 1000Hz width of the power spectrum. The RPM of the machine helps to discriminate its different normal states since vibration levels change depending on this parameter.
We have used two real data sets: laboratory data I (case I experiment a) and real detection scenario case (case IV). The two datasets are challenging in different ways. The first one tracks the vibration behavior of a component from brand new conditions to final breakdown. Delays in the detection can appear as we have seen in the previous section. The second one is a real complex machine case, so interference from other components and external conditions can degrade detection accuracy and increase false positive rate. Both data sets were labeled in two classes, normal and abnormal behavior. For the case of the laboratory data, captures were labelled as abnormal when their abnormality becomes evident manually analyzing data and, in the second case, we labelled the patterns as abnormal when the machine had a reported fault. These labels are not used for training but as the reference to calculate classification errors. In both cases, the classification accuracy can be interpreted as the ability to automatically detect a fault in an accurate way. All models were trained with only a portion of normal state captures and then tested with a different data set consisting on posterior normal and abnormal captures. The proportion of the test set was 50% of normal samples and 50% of fault samples. The mean test error for 20 runs of the best combination of parameters in each case is given in Table 7.3 for the proposed model and the state of the art one class ν-SVM classifier [209], Autoassociative Neural Network (AARNA)

model [115] and the Mixture of Gaussians model (MGM) [64]. It can be noticed that the proposed model is the most accurate in the gradual degradation case with a big difference in comparison with one class ν-SVM classifier and the MGM while, for the second data set, it performs slightly worse than the MGM. On average, in both data sets it is the best method among the ones compared. These results highlight that effective anomaly detection based on frequency domain features and anomaly detection algorithms is possible. In addition, MCSE algorithm proposed in section 3.1.2.6 seems to be a good candidate for this purpose.

Table 7.3: Mean test error for anomaly detection algorithms based on frequency domain feature extraction

	one class ν-SVM	AARNA	MGM	MCSE
Case II (exp. a)	15.38%	8.39 %	26.7 %	6.84%
Case IV	3.18%	4.45 %	1.98 %	2.54%

7.4.3 Fault Detection via on-line anomaly detection

In this section we explore how vibration fault detection can be tackled in a on-line manner with the OSDAD algorithm (presented in section 3.3). On-line learning algorithms with a simple update strategy and low computational requirements are appealing due to the recent interest on embedding in the mechanical components its own fault detection system.

In order to carry out the experimentation, two different datasets have been used: Laboratory Data (Case II, experiment a) and the real fault detection scenario (Case IV). As we previously mentioned, in the first case, a fault in a bearing evolves over time. In the second case, a sudden breakdown happens in the monitored system and a complete review is needed.

The OSDAD detection capabilities will be compared in this case with a state of the art non stationary change detection algorithm presented in [14]. This algorithm, named Computational Intelligence CUSUM (CI-CUSUM), aims at detecting a change in a data stream without assuming any property of the underlying probability distribution that generated the data. While in our proposed model we tackle the density support estimation through an on-line classifier, in the CI-CUSUM a new sequence is generated through a transformation of the original sequence. This transformation is tackled in two steps for multidimensional data. First, the dataset features $\mathbf{x}_t$ are mapped to a

reduced data vector $\phi(\mathbf{x}_t)$ through a Principal Component Analysis (PCA) previously calculated on an initial dataset. The number of considered eigenvectors can be empirically identified by removing those eigenvalues whose sum is below a threshold. The mean vector $\mathbf{y}_{t'} = \frac{1}{T}\sum_{i=t}^{t+T}\phi(\mathbf{x}_t)$ of a window of T patterns are periodically calculated. The size of T must be large enough to invoke the central limit theorem. This transformation produces a sequence $\mathbf{y}_{t'}$ which follows a multivariate Gaussian distribution $N(\mu, \mathbf{\Sigma})$. Afterwards, it applies a CUSUM test to this sequence. In order to do this, it is necessary to estimate the parameters μ and $\mathbf{\Sigma}$ from an initial dataset as well as the parameters of the CUSUM test. Since we are looking for increments in vibration energy, once having the test parameters, the algorithm tests the appearance of a growth in different combinations of components of the sequence $\mathbf{y}_{t'}$. Since testing all the possible combinations of the dimensions of $\mathbf{y}_{t'}$ could end in an exponential number of tests running, an equilibrium has to be found between number of combinations and detection accuracy. In this work we will apply the Configuration 2 recomended in [14], where the following hypothesis are considered: (a) all the components suffer an increase and (b) any of the components increase separately.

In this case the following feature vector was constructed for each power spectrum: (a) root mean square (RMS) energy of the whole raw vibration signal, and (b) the energy of each sub-band of 200Hz width that was extracted from the power spectrum. The parameters of the two models were selected in order to detect, in the most accurate way, the faults in the machines.

First, the results for the laboratory data of Case II are presented. A window with the first 200 captures of normal state was used to obtain a stable model of the on-line algorithm. The parameters used to adjust the model were the following: $\sigma = 0.8$, $C = 0,05$, $C_r = 0,06$ and $q_i = 1e-4$. Once the model was trained, the probability p_0 was estimated and the parameters of the CUSUM test were calculated (values of ANOS $= 500$ and $p_1 = 1,5 \times p_0$ were selected). Hereafter the methodology presented above was applied to the remaining data in order to detect possible deviations.

Figure 7.13 shows the CUSUM chart for the OSDAD algorithm. It can be observed that OSDAD raises an alarm in sample 340 (74 hours before the breakdown happens), when the CUSUM statistic exceeds the threshold h. This point coincides with the change point extracted by an expert manual analysis of the captures. Thus, the OSDAD algorithm demonstrates its capability to detect the fault in an incipient stage avoiding false alarms. In Figure 7.13 the CUSUM chart for the CI-CUSUM component with its earliest alarm is also depicted. A window size of 15 samples was used for constructing the CI-CUSUM transformed sequence and the data used to set up the test was the same that the data used for the proposed model. In this case, an alarm is raised in sample 405 (63 hours before the breakdown happens). It can be observed that, although the

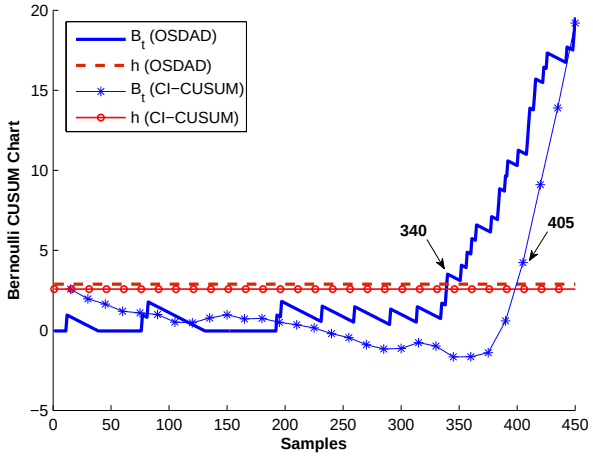


Figure 7.13: The Bernoulli CUSUM charts for Case II dataset.

CI-CUSUM also demonstrates a good performance, it needs more samples to detect the deviation.

For Case IV, the first 180 captures of normal state were used to adjust the model. The parameters used to build it were the following: $\sigma = 1$, $C = 0,09$, $C_r = 0,085$ and $q_i = 1e-5$. Once the model was trained, the probability p_0 was estimated and the parameters of the CUSUM test were calculated (values of ANOS = 500 and $p_1 = 1,5 \times p_0$ were selected).

Figure 7.14 depicts the CUSUM chart for both the OSDAD algorithm and the CI-CUSUM. Also for this case, the OSDAD algorithm demonstrates its capability to detect sudden changes in the vibrational behavior of the machine without giving any false alarm. The CUSUM chart for the CI-CUSUM component with the earliest alarm is also depicted in this figure. The set up for this case was the same that the one for Case II data. It can be observed that, also for this task, the CI-CUSUM needs more samples to detect the deviation.

In order to assess the impact of the parameter selection on the detection capabilities of the OSDAD algorithm, we have again tested its performance for different combinations of all parameters, moving them around the optimal depicted in figure 7.14. Figure 7.15 shows, for the real scenario fault detection case, the detection delay, in terms of number of samples after the break appears in sample 250 (assessed by an

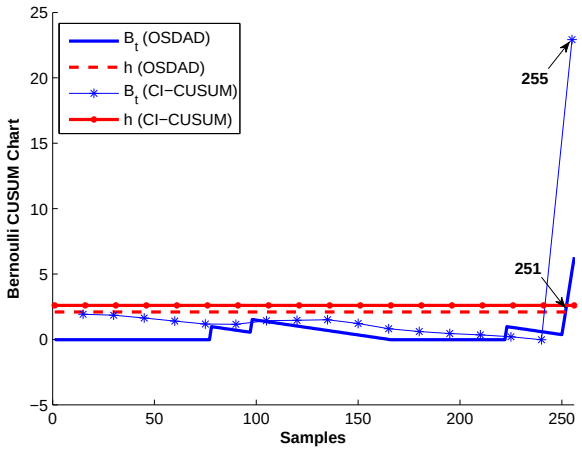


Figure 7.14: The Bernoulli CUSUM charts for Case IV data.

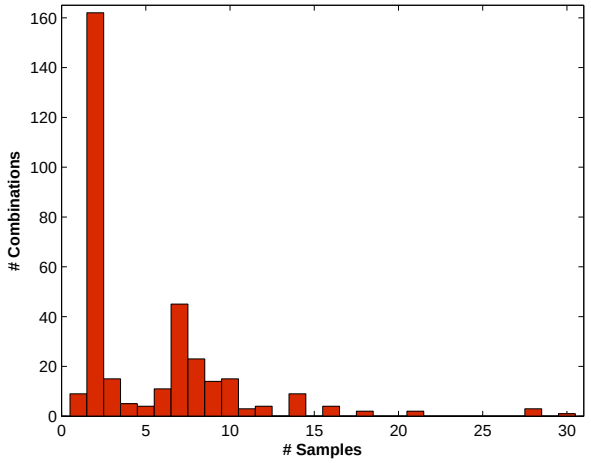


Figure 7.15: Performance of the OSDAD algorithm for the wind mill dataset with different combinations of parameters.

expert). There is a range around the optimal hyperparameters where the maximum detection delay is 10 samples, which is not a dramatic loss in detection ability.

7.4.4 Fault detection via alternative transformations

Vibration analysis has traditionally chosen to transform raw vibration signals to frequency domain in order to extract informative features. Power spectrum, cepstrum, wavelet analysis, etc. belong to this kind of transformations. In this section we aim at answering the third question posed at the beginning of the chapter: *Is it possible to propose alternative raw vibration feature extraction methods not based on frequency domain transformations?*. Namely, Recurrence Time Statistics, a transformation rooted in chaos theory [89], and which has not been explored in the past for vibration analysis, is applied to the real fault detection cases. We combine this feature extraction strategy with the EVOC method presented in section 3.2.1 and one-class ν-SVM as base classifiers. We start with the description of the aforementioned feature extraction strategy.

7.4.4.1 Recurrence Time Statistics

In recent years the interest in automatic fault detection research has moved towards studying how the information in the time domain signals can be exploited for early detection of faults. Traditional linear and nonlinear time series analysis techniques combined with other signal detection techniques have been used in the field of bearing fault detection in the past years (see for example the work in [203]). Machinery vibration generation process (and more specifically faulty bearing vibration generation [73]) is known to be a nonstationary dynamical process, so early bearing fault detection problem can be viewed as a change of dynamics and this has been studied for many decades [89]. Many indexes specially designed for characterizing the dynamics of nonlinear and chaotic systems have been devised and its applicability as features in bearing fault detection is a source of improvements that is to be fully studied yet.
Recurrence time statistics is a method rooted in chaos theory [89]. It assumes that the process under study is fully described by scalar time series $\{x(i), i = 1, 2, \ldots, M\}$, where i is the time index. According to Takens' embedding theory [228], the corresponding m dimensional phase space can be built by constructing vectors from the time series, $X_k = [x(k), x(k + L), x(k + 2L), \ldots, x(k + (m - 1)L)]$, where L is the time delay. The vector sequence $\{X_k, k = 1, 2, \ldots, N\}$ constitutes a trajectory in the phase space with $N = M - (m - 1)L$. In order to measure the time that takes the dynamical process to return to an attractor close to the initial one (Poincaré recurrence time), recurrence time statistics proceeds as follows:

1. Fix an arbitrary reference point X_0 in this constructed phase space, and consider the ball centered in that point of radius r (we will see hereunder how to set this radius value): $B_r(X_0) = \{\|X_j - X_0\| \leq r \mid j \in [1, N], j \neq 0\}$

2. Denote the ordered subset of the trajectory that belongs to $B_r(X_0)$ by $S_1 = \{X_{t_1}, X_{t_2}, \ldots, X_{t_i}, \ldots \mid t_i \in [1, N], t_{i+1} > t_i\}$. These points are called Poincaré recurrence points.

3. Calculate the Poincaré recurrence times, which are defined as $\{\mathrm{T1}(i) = t_{i+1} - t_i, i = 1, 2, \ldots\}$. The T1 index of this reference point X_0 is the mean of the above generated T1 set.

Finally, the overall T1 of the whole phase space is the average of the T1 indices of all the reference points. Figure 7.16 illustrates the T1 generation of one reference point [150].

According to Takens' embedding theory, if the attractor's dimension is D (may be non-integer), then a constructed phase space, with $m > 2D+1$ (m should be an integer) embedding dimension, is able to reveal the underlying dynamics. In the next section we present a method to define the embedding dimension for vibration generation of a rotating machinery.

Parameter Selection strategy for T1 Index: In order to apply the aforementioned recurrence time statistics to fault detection of rotational machinery in practical scenarios we need to provide an effective way of determining the three parameters that are involved in its calculation: delay L, embedding dimension m and radius of ball r. Fortunately, in the realm of non-linear dynamical systems analysis, the problem of estimating these parameters has been extensively studied and effective methods can be used.
Delay L needs to be small enough to capture the shortest change present in the data and large enough to generate the maximum possible independence between components of the phase space vectors. The autocorrelation method introduced by [1] can be used to decide L as the first zero value of the autocorrelation function.
For the case of the embedding dimensions m of the time series generated by autonomous dynamical systems in the absence of dynamical noise much work can be found in the literature. The methods developed for estimating the minimum embedding dimensions are grounded on Takens' embedding theorem [224] and most of them use the ideas

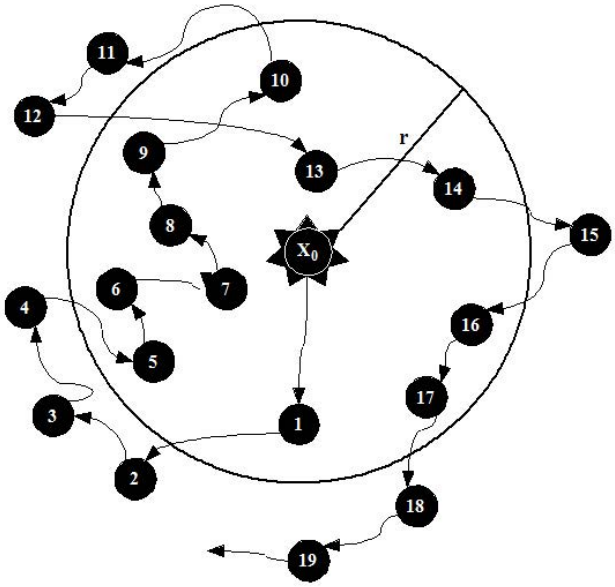


Figure 7.16: Recurrence Time Statistics calculation illustration.

of the false nearest neighbors technique [44][127]. Later a number of works discussed theoretical foundations of the delay embedding of the input-output time series [44][46]. This led to the generalization of the existing method for the case of non-autonomous dynamical systems [42][44][224]. In this work, the method proposed by He and Asada [100] is used since their strategy is based directly on measurement data and does not make any assumptions about the intended model architecture or structure. It requires only that the process behavior can be described by a smooth function, which is an assumption that must be made in black box nonlinear system identification. An explanation of this strategy's central idea follows. In general case, the task is to determine the number of relevant inputs of the function

$$y = f(\phi_1, \phi_2, ..., \phi_n) \tag{7.25}$$

from a set of potential inputs $\{\phi_1, \phi_2, ..., \phi_m\}$ that are given. If it is wrongly assumed that the function f depends on only $n-1$ inputs when it actually depends on n inputs, the data set may contain two (or more) points that are very close (in the extreme case they can be identical) in the space spanned by the $n-1$ inputs but differ significantly in the n-th input. Because the underlying function f is supposed to be smooth, if two points are close in the input space, their images must also be close. But when one (or several) relevant inputs are missing, this behavior is broken, so it is possible to conclude that those inputs are not sufficient. In [100] an index is defined based on so-called Lipschitz quotients, which is large if one or several inputs are missing (the larger the quotients, the more inputs are missing) and is small otherwise. Thus, using this Lipschitz index the correct embedding dimensions can be detected at the point where the Lipschitz index ceases to decrease. The Lipschitz quotients for the multidimensional case can be calculated using the expression:

$$l_{ij}^n = \frac{|y_i - y_j|}{\|\phi_i - \phi_j\|} \tag{7.26}$$

where n is the number of inputs, $\phi_i \in R^n$ is input i and $i \neq j$. The Lipschitz index is then defined as the maximum Lipschitz quotient

$$l^n = \max_{i \neq j} l_{ij}^n \tag{7.27}$$

As long as n is too small and thus not all relevant inputs are included, the Lipschitz index will be large because smoothness is not longer true under that situation. As soon as all relevant inputs are included, the value of equation 7.27 stays relatively constant. Once the delay L and the embedding dimension m are determined, the radius can be practically estimated in a way that the balls centered in the data samples in the T1

index have a volume proportional to the total volume of the box that contains the attractor

$$r = a \left(\frac{\Gamma\left(\frac{1}{2}n + 1\right) V}{\pi^{\frac{n}{2}}} \right)^{\frac{1}{n}} \tag{7.28}$$

where we have used the formula of the volume of an n-dimensional sphere, Γ is the gamma function, V is the volume of a box containing the attractor of the data and $a \in (0, 1)$. In the next section it will be shown that values of a around $[0.15, 0.3]$ exhibit good practical results.

7.4.4.2 Proposed methodology

In this section, the global proposed methodology based on the Recurrence Time Statistics of previous section and any anomaly detection algorithm is presented. As it can be observed in figure 7.17, the process of building a fault detector based on the T1 index is divided in three stages:

1. T1 Index hyper-parameters selection: Following the methodology in previous section, the hyper-parameters for calculating the T1 Index are selected. In order to do so, a base data set of vibration captures under healthy state are selected as representative of the vibration process under normal conditions. First, the delay L is calculated as the first null autocorrelation value of a time signal under normal conditions. Using this L value, the dimension m is calculated embedding this base set of captures into increasing dimensions until the Lipschitz index ceases to decrease, taking that dimension as m. Subsequently, using these values the volume of the attractor is estimated as the volume of the hypercube that covers the base data set embedded under L and m values, so radius r is finally calculated.

2. T1 Index feature extraction: In order to build a data set of normal patterns which subsequently will be used by the anomaly detection method, each time signal capture used in the previous section is divided into blocks, and the T1 Index is calculated for each block. This feature vector shall be called Block T1 Index (see figure 7.17). The division in blocks gives the method the ability to detect changes in the attractor localized in time.

3. Classifier training: Using the Block T1 Index feature patterns built in the previous stages, an anomaly detection model of the normal behavior of the rotating

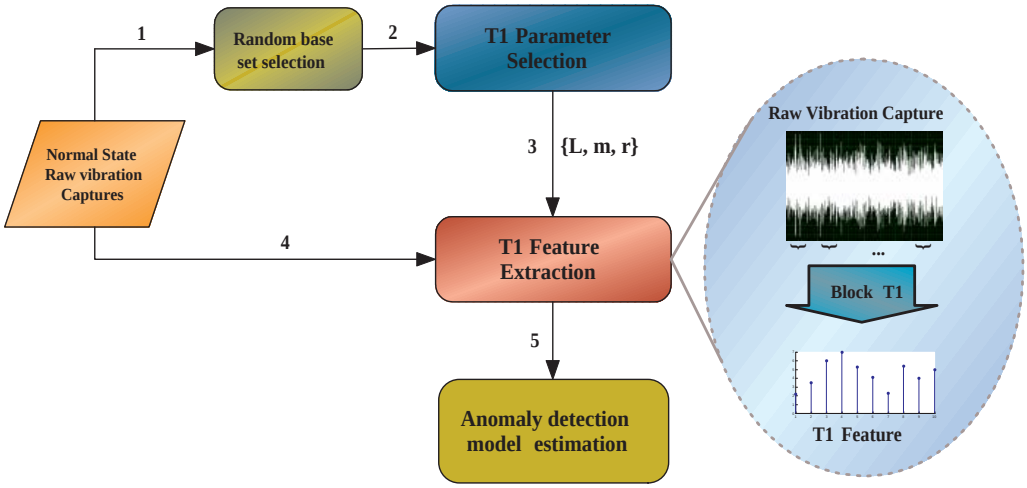


Figure 7.17: Proposed method model estimation scheme.

machinery is created.

7.4.4.3 Experimental Results

In this section the experimental results for laboratory and real fault data are explored.

Laboratory data: Case II

In order to assess the capabilities of the proposed methodology in this case, the data captures were manually inspected by an expert assessment. It was observed that there was a point during the experiment where, using these parameters manually, it was possible to detect that something had changed in the system. We fixed this point as the change time for labeling the captures as normal and faulty, so we assess the ability of the proposed model to detect a fault compared to manual human inspection using classic fault detection strategy.

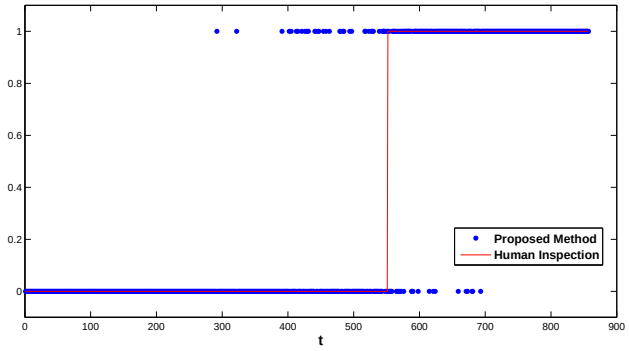
For the first data set (experiment a), eight hundred captures of normal state were used as historic (five days and a half) and were processed using the methodology presented in section 7.4.4.2 to automatically build the fault detection model. Afterwards, the last 6 days were automatically processed in order to detect a possible fault. For the second data set (experiment b), four hundred captures of normal state were used as historic (66 hours) and hereafter the trained proposed model was applied in order to detect

possible deviations.

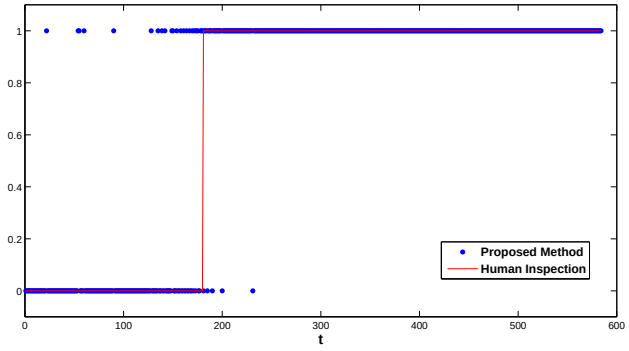
In Table 7.4, the results of EVOC classifier are compared with the classification accuracy obtained by the one-class SVM [20]. Both methods share the same human fault assessment in order to calculate the classification accuracy. For one-class SVM, RBF kernel [20] was used and optimal hyper-parameters were selected in the range $\sigma \in [1, 40]$ and $\nu = [0.01, 0.1]$, where one-class SVM showed accurate results. As it can be observed, one-class SVM obtains results less accurate than the ones obtained by EVOC (parameters were selected in the interval $k \in [1, 10]$ and $p = 1.5$ %) , which can lead to a lesser reliable detection system in practical situations.

Table 7.4 shows the classification accuracy obtained by the proposed method - using both EVOC and one-class ν-SVM as base classifiers - when compared to human inspection using RMS and FFT. A change point was set when a noticeable and stable change was detected in the vibration spectrum calculated with the FFT and the global energy (RMS) rose up from the base line under healthy conditions. It can be observed that the proposed methodology's output highly agrees with the fault assessment done manually. It is important to remark that the proposed methodology obtains the same assessment automatically and only based on a data set of captures during the operation of the machine when this is in a good state. So, the proposed methodology demonstrates its capability to obtain industry-standard results and to be highly useful in practical situations when compared to other approaches [69][255][171][254][126][80] which base their success in both: (a) using data from faulty counterexamples or (b) using a human-aided process.

It is interesting to compare the assessment made by both the proposed model and the human inspection based on the aforementioned features, in order to see whether the combination of the Block T1 Index and the EVOC classifier could be more sensitive to incipient faults than the classical approach. Figures 7.18(a) and 7.18(b) depict the state assessment of the system during the detection phase previously explained and used to build table 7.4 (in this figure an output "1" means abnormal and "0" normal state and the outputs are ordered by the time of the vibration capture until the breakdown). It can be observed that most of the discrepancies between human and automatic assessment are concentrated just before the fault was noticeable by human inspection. Due to the fact that the machine was continuously working without any other intervention, this discrepancy highlights the ability of the proposed Block T1 Index and EVOC classifier to capture more subtle differences in the vibrational behavior of a rotating machinery when an incipient fault is present. This ability can be exploited in fault detection systems in order to reduce detection delays between the detection and the presence of fatigue, which is very important for posterior fault management [112].



(a) Comparison of proposed method classification and fault detection via human inspections for Case II Data (experiment a).

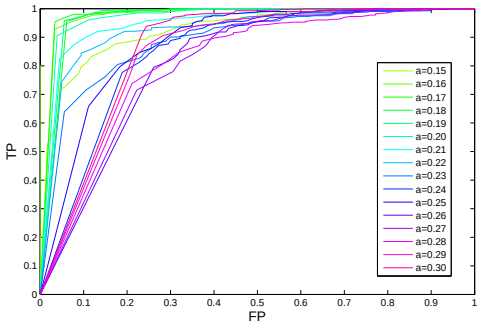


(b) Comparison of proposed method classification and fault detection via human inspections for Case II Data (experiment b).

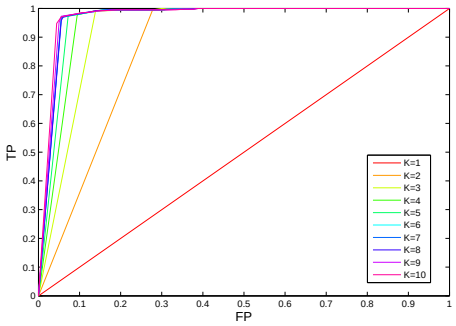
Figure 7.18: Comparison of proposed methodology vs. human detection

Data set	Case II (Experiment a)	Case II (Experiment b)	Case IV
EVOC	93.00%	96.40%	94.74%
SVM	88.21%	92.81%	94.40%

Table 7.4: Classification accuracy on the experimental data sets for the proposed methodology.



(a) ROC curve for different percentage a in T1 radius calculation. Case II Data (2nd experiment).



(b) ROC curve for different combinations of EVOC hyperparameters and fixed radius. Case II Data (2nd experiment).

Figure 7.19: ROC curves of the proposed methodology under different condition (TP: true positives, FP: false negatives)

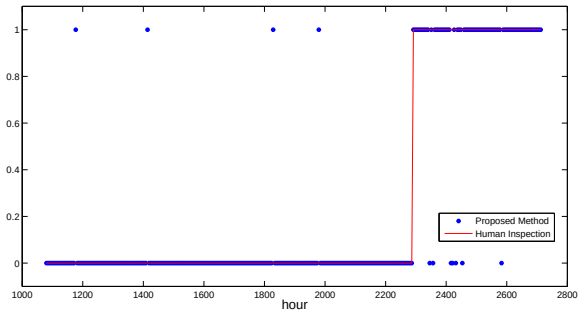


Figure 7.20: Fault detection of the proposed method in a real scenario.

The proposed methodology has three hyper-parameters that need to be fixed in order to build the fault detection model, the percentage of attractor a of T1 Block parameter (see equation 7.28), and the EVOC hyper-parameters k and percentage of outliers p which gives an output decision threshold. Figures 7.19(a) and 7.19(b) present Receiver Operating Characteristic (ROC) curves under different conditions that depict how the accuracy of the final model varies with these parameters. In figure 7.19(a), the number of neighbors k was fixed to 8 and the percentage of attractor was varied into the interval mentioned in section 7.4.4.1. It can be observed that using a value for a into that interval leads to an accurate fault detector. In the case of k, it controls the ability to avoid outliers in the final model. In 7.19(b), percentage of attractor a was fixed to its optimal value for this case. It can be observed that with a value of $k = 1$, we lose the ability to avoid outliers since areas close to any outlier will be considered as part of the *normal support*. Since the outliers are characterized for being aisled and far from *normal support*, if k is increased then EVOC starts to avoid areas in the input space where only outliers are present until it converges to a stable model.

Real Scenario: Case IV

In order to validate the proposed methodology in a production environment, it was applied in the following conditions: the data collected during the first 45 days of functioning was used to build the model using $k = 3$ and $p = 1\%$ for the EVOC classifier and $a = 0.2$ for the T1 parameter. The model obtained a 94.74% of classification accuracy and a reduced number of false positives and negatives, as can be observed in figure 7.20. If we changed the classifier to a one-class ν-SVM, classification accuracy is also high (94.40%). High accuracy of both classification methods highlights the discrimination capacity of the proposed feature extraction strategy.

7.5 Fault diagnosis strategy: methodology and experimental results

Up to this point we have treated the problem of fault detection. In this section we move on to the next question of this chapter: *Is it possible to come up with an algorithm able to automatically diagnose faults in bearings?*. In order to come up with an algorithm able to tackle this task power spectrum feature extraction and one-class ν-SVM as base classifier are used. Using the energy of different sub-bands of the normal state spectrum as training data, a one-class ν-SVM is used in order to detect a change of behavior due to the presence of a fault. Subsequently, using well known signal processing techniques, the proposed system will be capable of highlighting the source of the fault and returning a diagnosis.

7.5.1 Proposed method

The architecture of the proposed system is shown in Figure 7.21. The system performs periodically the following processing:

1. A new raw vibration signal is captured from the bearing (step 1).
2. The raw vibrational signal is transformed to frequency space via FFT in the same way as in previous sections (step 2). The energy of each sub-band is extracted from the power spectrum of the vibration signal (step 3). In this step, a decision about the size of the sub-band has to be made. This size is a compromise between two requirements: (a) it should not be sensitive to noise and (b) it should be able to accurately concentrate the diagnosis in the band where the fault is significant. If the size of the sub-band is too narrow, the method will be very sensitive to noise. On the other hand, too wide sub-bands would not allow us to accurately localize the exact band where the fault is evident. In this case, we use 200Hz sub-bands as a compromise of these two goals.
3. The sub-band energy pattern is analyzed by the one-class ν-SVM using an anomaly detection strategy (step 4). This model has been previously trained using historical data under normal conditions.

4. In the event that this analysis detects a fault, it is further analyzed to confirm the fault (step 5). Otherwise, we return to step 1.

5. The frequency band of the power spectrum where the defect appears evident is selected (step 6). This is done by selecting the most deviated sub-bands of the input pattern, using a Sensitivity Test, and concatenating them to find the deviated band of the spectrum.

6. Envelope analysis is utilized to highlight the characteristics of the abnormal signal based on the band obtained in the previous step (step 7). These will be analyzed by a knowledge-based system in order to diagnose the defective element of the bearing (step 8).

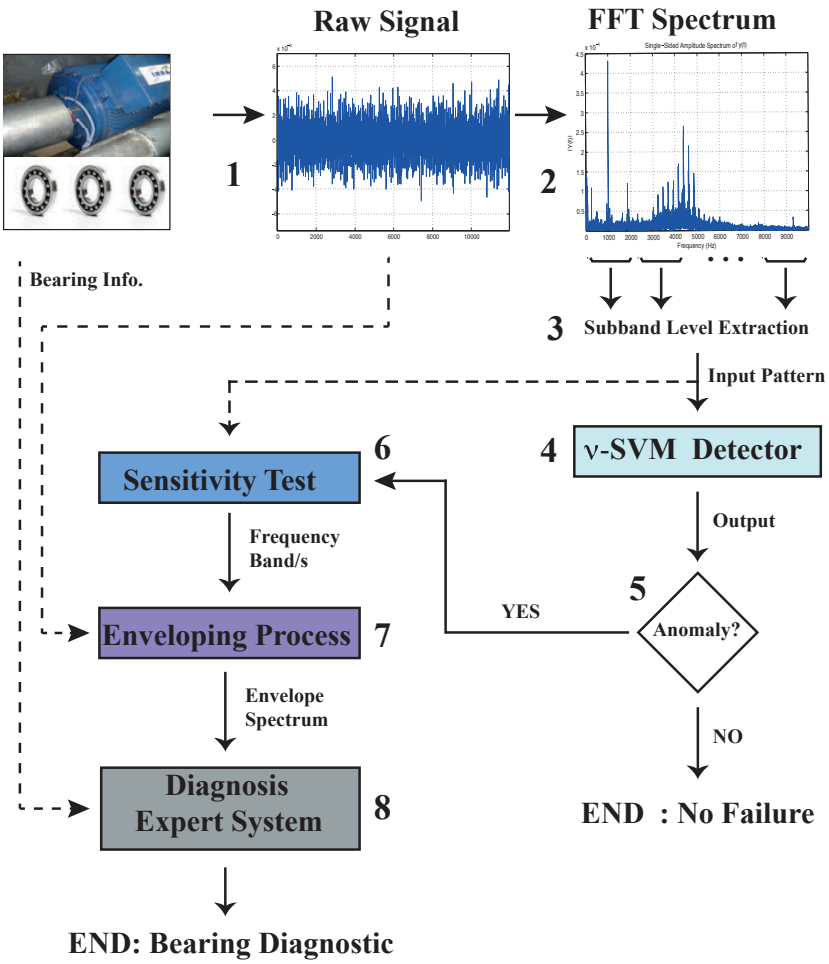


Figure 7.21: Architecture of the automatic bearing diagnosis system (numbers indicate the order of the process).

The advantages of the proposed method stem from the combination of the three analysis steps. Each step fulfills the requirements of the subsequent step (covering the necessary processing from raw vibrational signals to diagnosis). A knowledge-based system utilizes the rules extracted from the bearing characteristics frequencies presented in Table 7.1. The knowledge base needs to know which frequency is causing the anomalous behavior in the vibration of the machine in order to discern the failure mode. Envelope analysis is able to extract this frequency, but it needs to know whether there is a deviation and in which sub-band it is present. These two fundamental requirements are fulfilled by the combination of the one-class ν-SVM and the Sensitivity Test.

In the next sections, techniques used in the main steps of the model are discussed.

Sensitivity test

If the one-class ν-SVM classifies an input pattern as an anomaly, it would be interesting to determine which frequency sub-bands of the input pattern $\mathbf{x}$ are more deflected with respect to the normality represented by the support vectors of the model. In this work, a simple approximation to obtain these characteristics is devised.

Together with equation 7.21, the argument of *sgn* function in (7.17) transformed by the Support Vector (SV) expansion is the following:

$$g(\mathbf{x}) = \sum_{j \in SV} \alpha_j k(\mathbf{x}_j, \mathbf{x}) - \rho \tag{7.29}$$

where $k(x_i, x_j)$ is the kernel function and ρ the bias of the hyperplane in feature space. Taking the derivative of (7.29) with respect to each component i of $\mathbf{x}$ we obtain the following sensitivity value,

$$\mathbf{sens}^{(i)} = \left| \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}^{(i)}} \right| = 2\gamma \sum_{j \in SV} \alpha_j k(\mathbf{x}_j, \mathbf{x}) |\mathbf{x}_j^{(i)} - \mathbf{x}^{(i)}| \tag{7.30}$$

where γ is the width of the gaussian kernel ($\gamma = \frac{1}{2\sigma^2}$); this measure indicates the components or parameters of the input vector which are more deviated from the components of the support vectors. A ranking of the most influential parameters of the

input pattern can be constructed with this measure. We will refer to this process as the Sensitivity Test.

As we have already mentioned during the model description, this simple method allows us to select the most relevant sub-bands of the input pattern and obtain the frequency band of the vibration signal that will be employed for the enveloping process. This band is extracted following the next principle: select the widest band that results from concatenating the M most deviated sub-bands. In this work, we used $M = 15$ giving a maximum band length of 3000Hz.

Envelope analysis for bearing fault diagnosis

Each time a defect on a component strikes another part of the bearing, a series of force impacts are produced. These impacts may excite resonances in the bearing and in the machine. The natural resonant frequency acts as a high-frequency carrier signal that is modulated in amplitude by a low-frequency signal (i.e., the bearing defect frequency), resulting in high frequency components around the carrier frequency. This effect was detailed in previous sections (see the discussion of section 5.2 and the description of bearing's case of section 7.1.1). Envelope analysis or demodulation is able to extract the modulating signal from an amplitude modulated signal [95, 168, 206]. This technique provides an important alternative to the traditional spectral analysis. The overall process is shown in Figure 7.22.

The first step in the envelope analysis process consists on using a band-pass filter on the raw signal, with the aim of isolating the band where the natural resonant frequency excited by the impact frequency appears. Thus, effects of high amplitude, low frequency vibrations and random noise outside the band are eliminated. The Sensitivity Test explained in the previous section is responsible for selecting this frequency band.

The next step is the rectification of the filtered signal to calculate its envelope. This can be done through Hilbert Transform [169, 252].

In the last step, Fast Fourier Transform of the rectified signal is calculated in order to obtain the envelope power spectrum. This spectrum will contain peaks at the bearing characteristic frequencies of the fault and its harmonics. Furthermore, the amplitude of these peaks will increase as the fault evolves. In the last stage of the bearing failure, the noise floor will also increase blurring the peaks. This bearing vibration diagnosis principle has the advantage that bearing fault frequencies can be identified in early

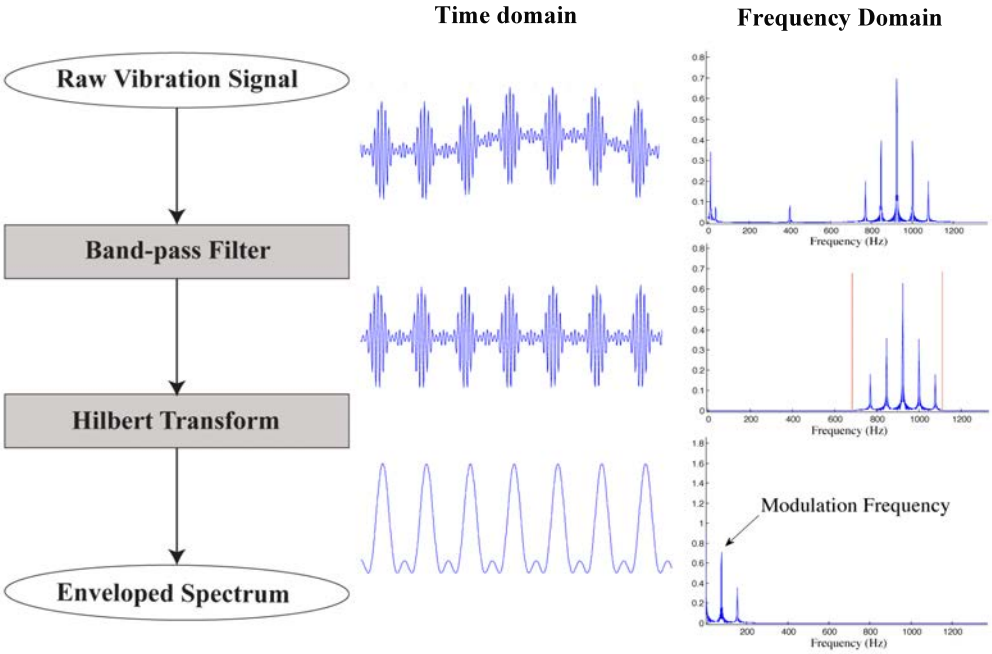


Figure 7.22: Signal processing steps for envelope analysis.

stages of the failure [123].

This analysis technique works very well for bearings and can also be used for diagnostics of other kinds of components which modulate the resonance frequency by their characteristic frequencies - such as gearboxes, turbines and induction motors.

7.5.2 Experimental results

Two different data sets have been used to check the performance of the fault diagnosis method proposed in the previous section: laboratory data of cases II (experiment a) and III. In every experiment, accurate fault detection has been achieved using the following parameters for the ν-SVM: $\nu = 0.01$ and a Gaussian kernel with $\gamma = 0.05$. The first parameter represents the estimation of spurious data (1%) in the normal state registry. The second one controls the width of the distribution and has been obtained empirically.

For case II data, 200 captures of normal state were used as the training set (33 hours) and hereafter the method described above was applied in order to detect and

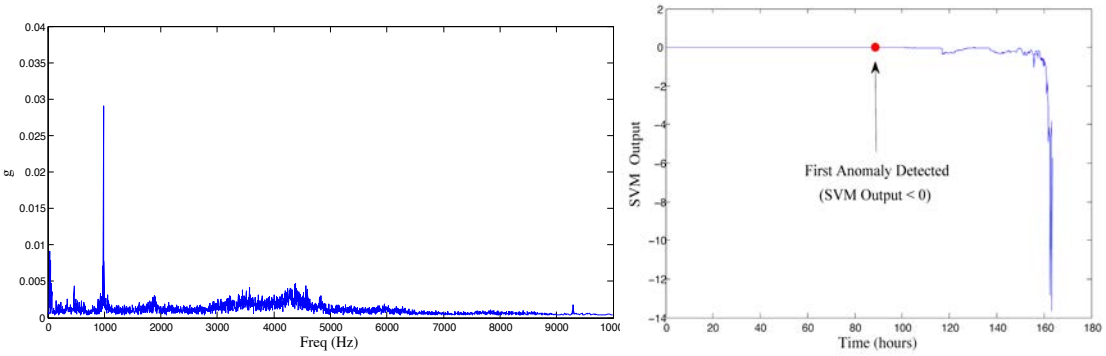


Figure 7.23: (a) Example of normal state power spectrum (b) ν-SVM detection on Case II experiment.

diagnose possible deviations. Figure 7.23 shows (a) the power spectrum of one capture from the training set and (b) the *SVM Output* (argument of equation 7.17) of the ν-SVM during this test. The dot shows the point where the system detects a change of the behavior for the first time, 75 hours before the crack forced the machine to stop working. This figure also shows the qualitative indication of the exponential evolution of the failure.

Figures 7.24 and 7.25 depict respectively an incipient fault power spectrum (89 hours after the beginning of the experiment) and an advanced fault power spectrum (120 hours) along with their corresponding frequency bands selected by the Sensitivity Test. The two corresponding envelope spectrums are also shown in Figures 7.24 and 7.25 to demonstrate how the diagnosis system works in real-time. In both figures, the characteristic fault frequency of the outer race can be seen (BPFO = 231 Hz, which is very close to the value previously calculated) along with harmonics. As the failure progresses, the amplitudes of the peaks increase, thus making the diagnosis easier and accurate.

In order to test the diagnosis methodology proposed, four sets of data from this experimental system were used: under good conditions, with a fault on the outer race (aligned with the load at 6 o'clock position), with a fault on the inner race and with a ball fault. The experimental rotating frequency is approximately 29 Hz (1740 rpm). Data under good conditions was used as a training set (see Figure 7.26) and hereafter the methodology was applied to the other data sets in order to detect and diagnose the

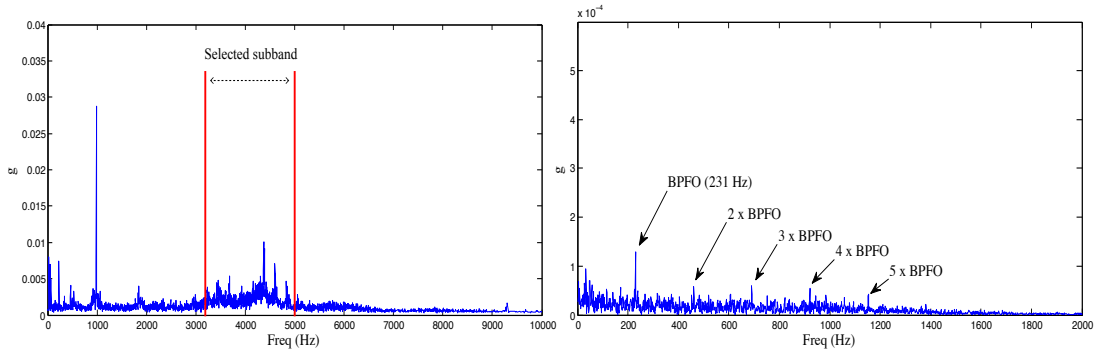


Figure 7.24: (a) Incipient fault power spectrum and selected band. (b) Incipient fault envelope spectrum.

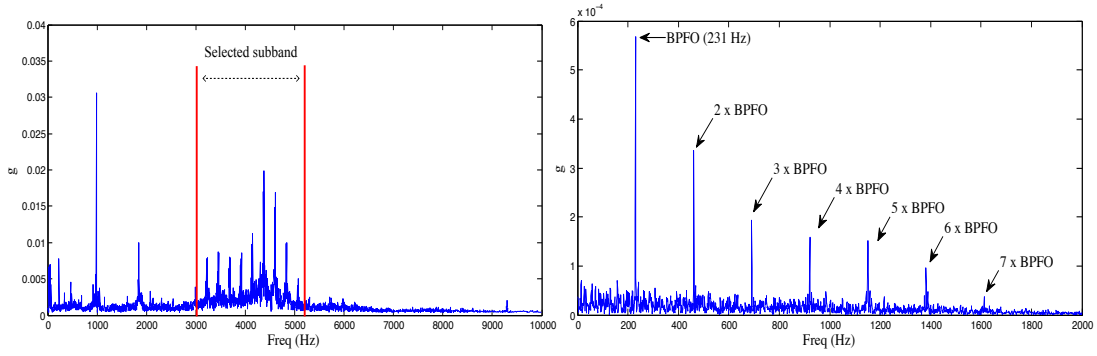


Figure 7.25: (a) Advanced fault power spectrum and selected band. (b) Advanced fault envelope spectrum.

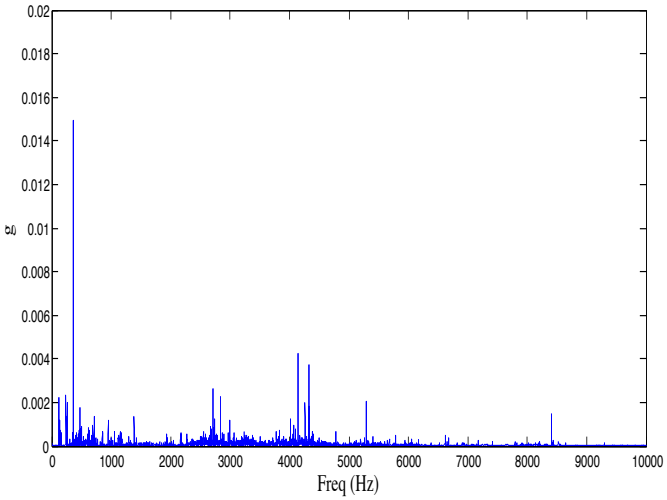


Figure 7.26: Normal state power spectrum.

possible faults. A 100% accurate detection was obtained.

Figure 7.27 depicts an inner race fault power spectrum (load 1 and 0.007 inches) along with its selected frequency band and the corresponding envelope spectrum. The inner race fault frequency is clearly identifiable (peak at 156 Hz) as well as its harmonics modulated by the shaft frequency (29 Hz). In Figure 7.28, a power spectrum of a ball fault (load 2 and 0.021 inches) and the selected frequency band are shown. The corresponding envelope spectrum is also depicted in Figure 7.28. In this case, there is a peak at 2 times ball spin frequency (137 Hz), which means that there is a fault in a rolling element. Finally, Figure 7.29 depicts (a) an outer race fault power spectrum (load 1 and 0.007 inches) along with its selected frequency band and (b) the corresponding envelope spectrum with peaks at the outer race fault frequency (peak at 106 Hz) and its harmonics. Once again, these examples demonstrate that the system can detect anomalies that do not correspond to normal behavior and diagnose the possible sources of the failure.

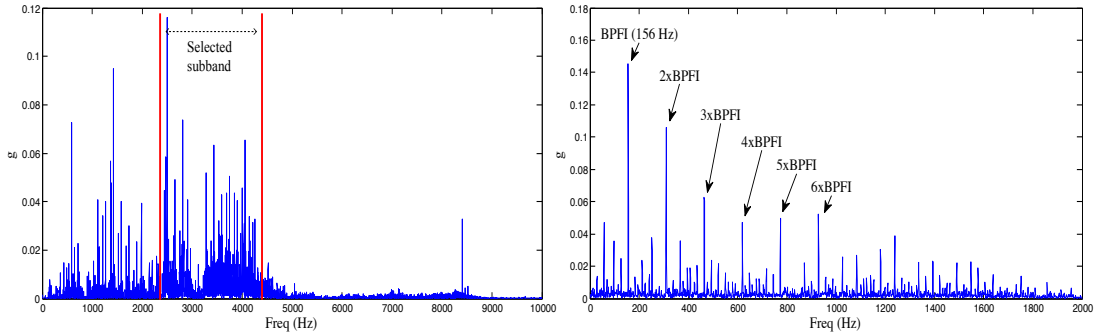


Figure 7.27: (a) Power spectrum and selected band of an inner race defect. (b) Envelope spectrum of the inner race defect.

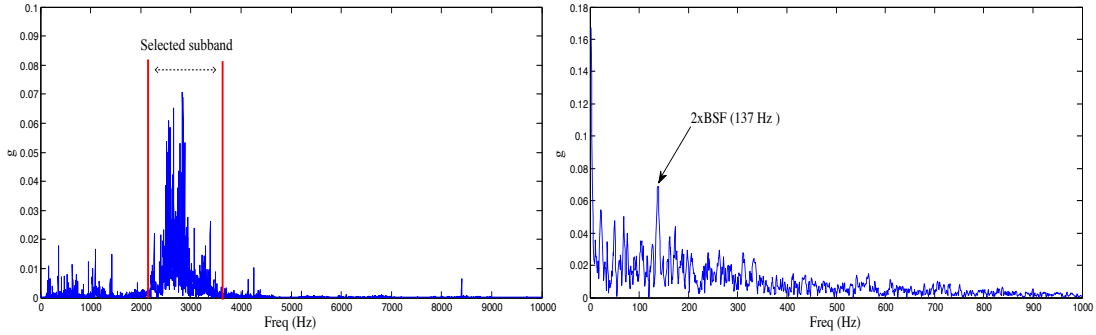


Figure 7.28: (a) Power spectrum and selected band of a ball defect. (b) Envelope spectrum of the ball defect.

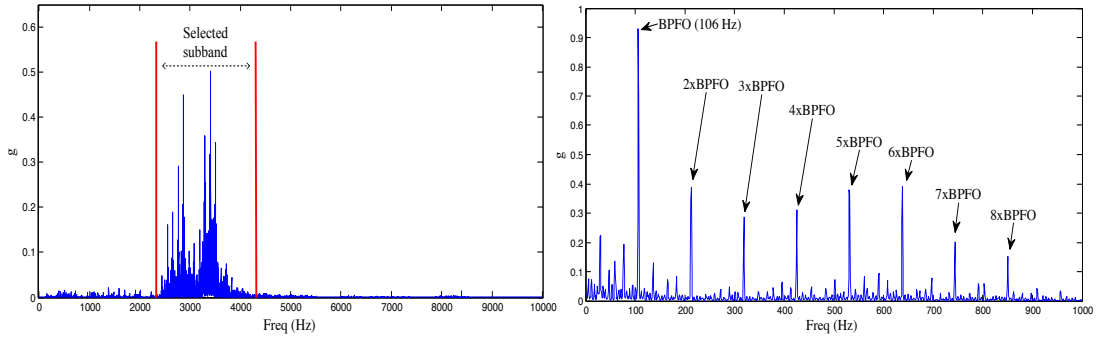


Figure 7.29: (a) Power spectrum and selected band of an outer race defect. (b) Envelope spectrum of the outer race defect.

7.6 Fault evolution assessment: methodology and experimental results

In this final section we illustrate how on-line learning methods can give an answer to the final question posed in this chapter: *Is it possible to assess the evolution of the vibration level of a machine in order to detect when this level will become unbearable?*. Vibration practitioners can select beforehand a vibration level which is considered unbearable for an specific machine. So, in order to raise an alarm based on the expected global vibration level of the system in the near future we should be able to predict it. As we have observed in previous sections, the vibration level of a machine remains more or less stable during normal conditions until a fault appears. Once a fault is present, fatigue of the materials unavoidably produce an increase in its size and so the increment of the global vibration level. If we were able to predict the tendency of vibration level and anticipate the point in the near future when this fault would become dangerous, it would be possible for the maintenance personnel (or automatically in closed loop) to stop the machine before its integrity is in danger. Since the nature of global vibration levels evolve along the time, a system able to adapt to changes is desirable.
We try to experimentally check if, using the on-line learning algorithm propose in chapter 4, it is possible to predict the global vibration level of the machinery and use this prediction to avoid dangerous situations. The following experimental setting is generated:

- Root Mean Square (RMS) energy of the raw vibration of Case II (experiment a) is calculated for each capture separately. This gives us a time series of the global vibration level of the machine. Since we are interested on the stable tendency of the vibration of the machine, RMS signal is smoothed with a moving average filter of length 10.

- Vibration level of 0.08 g is considered dangerous for this component, so when this level of vibration is predicted the machine should be stopped.

- A neural node is continuously trained with the on-line algorithm presented in chapter 4. The parameters selected for the model are $\lambda = 0.99$ (forgetting factor) and $\delta = 1e-8$ (initial regularization). This model has as inputs the previous 5 samples and tries to predict the value in time t+15. Each time a new measurement is received, the model is updated with the data available and tries to predict the future RMS level. If this prediction reaches a value above the stablished threshold

an alarm is raised.

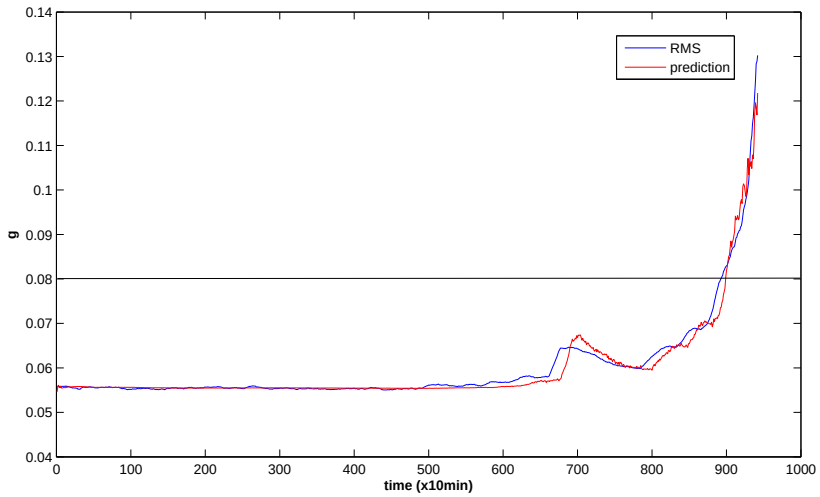


Figure 7.30: Global acceleration energy and prediction.

Figure 7.30 depicts both the actual RMS signal and its prediction made continuously by our model. It can be observed that the model is able to capture the changing tendency of the signal thanks to its continuous update. Namely, at time 884 of the experiment it determines that the value of global vibration is going to exceed the predefined level of 0.08 g. At this point of time the machine should have been stopped. Actual vibration level of the machine reaches that point at sample 895, so the proposed method was able to signal almost 2 hours in advance that the machine was going to reach a dangerous condition.

7.7 Discussion

At the beginning of this chapter we posed a set of key questions in order to evaluate the viability of applying anomaly detection techniques to fault detection problems based on vibration analysis. Experimental results have given us the following answers to these questions:

- Is vibration data adequate to detect and assess the condition of a mechanical component using an anomaly detection strategy?. In section 7.4.1 we checked the

correlation between fault severity and data divergence from normality. Thanks to the simulator data we were able to generate captures with a fault that grew linearly and could observed that vibration data presented a clear correlation with fault severity if one-class ν-SVM was used as the divergence estimator. This first experiment gives us a hint that if a divergence is soon captured, fault width should be still incipient and so anomaly detection strategy is coherent with the aim of detecting faults in an incipient phase.

- Is it possible to join traditional frequency domain transformation of vibration data and anomaly detection methods to build fault detection systems?. We have combined frequency domain feature extraction with several state-of the art and proposed algorithms. Overall, it seems that anomaly detection is able to detect faults with high accuracy. Among the tested models, it seems that MCSE is the one that gives a more accurate fault detection accuracy overall in the cases tested.

- Is it possible to propose alternative raw vibration feature extraction methods not based on frequency domain transformations?. In section 7.4.4 a new feature extraction strategy is proposed. This strategy obtained good experimental results when combined with on-class ν-SVM and EVOC algorithms. These results highlight the fact that other transformations, not based on frequency domain principles, can be also effective for fault detection problems.

- Is it possible to come up with an algorithm able to automatically diagnose faults in bearings?. In section 7.5 we have presented an algorithm able to detect and diagnose a bearing fault using one-class ν-SVM and envelope analysis. We have overcome the problem of automatically select the natural frequency band by using a sensitivity measure extracted from SVM classification rule formula when RBF kernel is used. This algorithm can save time and costs to maintenance practitioners. Fault is not only automatically detected but also adequately located so maintenance measures can be more focused without effort.

- Is it possible to assess the evolution of the vibration level of a machine in order to detect when this level will become unbearable?. In the last section of this chapter a fault prognosis case has been illustrated. From the results obtained, we could derive that an on-line learning algorithm can be used to track the tendency of the global vibration level of a piece of machinery and signal a dangerous situation beforehand.

CHAPTER 8

Conclusions, main contributions and future work

This work is devoted both to Machine Learning (ML) research and its application to the field of predictive maintenance of rotating machinery based on vibration signatures. We departed from three basic observations: (a) in predictive maintenance each machine has to be considered individually, (b) in practice, it is only possible to use data under normal condition to build a model and (c) it is desirable to assess the condition of the components both in the present and in the near future. These three observations distilled the main objective of this work: to develop novel anomaly detection and on-line learning algorithms which can tackle predictive maintenance problem in real scenarios. First, we presented general purpose ML techniques able to deal with the aforementioned conditions. The proposed anomaly detection techniques constitute the following contribution to the state-of-the art:

- Minimum Volume Set of Covering Ellipsoids (MSCE). One of the most common choices in state-of-the art algorithms to capture the region of the input space where "normal" data resides is to assume an ellipsoid shape for this region. This choice translates learning process into the classic Minimum Volume Covering Ellipsoid problem, in which a minimum volume ellipsoid which covers "normal" data samples is built. It turns out that this choice can be very restrictive and gives poor results in situations where, for example, we have to deal with multi-modal or noisy data. In addition, outliers can degrade significantly the obtained model and to purge them can be a tedious task for the practitioner. In this first proposed algorithm we extend the classic Minimum Volume Covering Ellipsoid (MVCE) problem to a robust algorithm which obtains a minimum volume set of covering ellipsoids automatically avoiding outliers. The choice of a set of ellipsoids is much more flexible for covering complex data sets and can improve anomaly detection performance in many situations. We have shown this fact using both artificial and vibration data of industrial machinery. It is important to note that the proposed algorithm divides the learning process in an ellipsoid adjustment step

and a reassignment phase. We leave open the question of extending this work to kernel spaces using kernel minimum volume ellipsoid learning algorithms.

- Extreme Value Statistics One-class Classifier (EVOC). Nearest-neighbor based anomaly detection algorithms have treated the problem of highlighting anomalous patterns in a heuristic way. In this work we have tried to give a more formal shape to the following intuition: anomalies are dissimilar to normal data and thus they should be far apart from samples obtained under normal conditions. Using extreme value statistics results, we have proposed a classification rule which does not assume any property about normal data apart from reside in a metric space. We have experimentally prove the accuracy of this rule in Euclidean space problems. We have left as future work its application to other kinds of complex structured data such as sequences, graphs, etc. making use of distance measures in these spaces. We have also left open the question of how to model distances in these metric spaces.

- On-line Stream Data Anomaly Detection (OSDAD). Most of state-of-the art anomaly detection algorithms, and also the two previous ones, deal with the case where a dataset of normal data is available beforehand and we train a model in a batch manner. When data arrives as an stream, we face a very different problem in which we want to highlight those regions of the stream where an anomaly (or change) has occurred and continuously adapt to new scenarios. The third proposed model tackled this on-line anomaly detection problem through a Passive-Agressive (PA) classification algorithm applicable when dealing with a stream of data. Two classical problems are addressed from an on line perspective: one class classification and stream anomaly detection. A new PA formulation for on-line one class classification is presented. From a practical perspective, the proposed formulation has the following advantages: (a) it is able to accurately fit the support of normal data in an on line fashion, (a) it is able to dynamically adapt to changes in the distribution of data, (c) it can be applied in a feature Hilbert space via kernel mapping and (d) it automatically controls the growth of the number of support vectors. This model is combined with a CUSUM chart of the proportion of the detected abnormal patterns giving the OSDAD algorithm, specially designed for stream anomaly detection. Experimental results confirm that the proposed model shows very good performance when compared to state of the art algorithms for one class classification and stream anomaly detection. The devised method leaves as future work the determination of the theoretical properties of the one class classification algorithm (convergence and classification error bounds), the design of a criteria for discarding not relevant past Support

Vectors and its application to problems using other feature spaces.

Second block of the work was devoted to on-line learning on stream data. On-line ML is a model of induction that learns from one instance at a time. Its goal is to predict a correct output value for each instance (a label or a real valued property) only based on the current model trained with data previously seen and the current input. An on-line learning algorithm receives periodically (after each input pattern or after a set of inputs) feedback of the correct output and, based on this information, it has to continuously update its model based only on the last received information. In this work we have reviewed the problematic of on-line learning in its different flavors (big data sets, changing contexts, distributed data) and proposed a novel on-line learning algorithm for a neural node with a non-linear output function. The proposed algorithm is able to continuously train a neuron in a one pattern at a time manner. If some conditions are hold, it analytically ensures to reach a global optimal model. Besides, it covers previous state-of-the art algorithms such as classic Recursive Least Squares (RLS) [99] as special cases and is able to tackle the aforementioned three different scenarios of stream data learning: big data sets, changing contexts and distributed data.
In the last part of this work we have tried to answer the question of wether ML-based predictive maintenance is viable in real production environments. Namely, fault detection in wind mill turbines was treated. We have developed a fault detection software, GIDAS®, which covers all processing phases from vibration measurement to alert notification. The system was installed in the maintenance offices of some pilot wind mill farms and integrated in its workflow. The techniques proposed in this thesis were applied to fault detection problems and the following conclusions were obtained:

- Overall, it seems that anomaly detection is able to detect faults with high accuracy. Among the tested models, it seems that MCSE is the one that gives a more accurate fault detection accuracy overall in the cases tested.
- A new feature extraction strategy based on recurrence statistics is proposed. This strategy obtained good experimental results when combined with on-class ν-SVM and EVOC classifier and highlights the fact that other transformations, not based on frequency domain principles, can be also effective for fault detection problems.
- We presented an algorithm able to detect and diagnose a bearing fault using one-class ν-SVM and envelope analysis. This algorithm can save time and costs to maintenance practitioners. A fault is not only automatically detected but

also located, so maintenance measures can be more focused without effort. The same principles can be applied in future work to other kind of faults with similar vibration signatures characteristics.

- Finally, a fault prognosis case was studied to illustrate that assessment of the condition of components in the near future is possible via the prediction of key parameter. For this purpose, an on-line learning algorithms can be used to track the tendency of the global vibration level of a piece of machinery and notify a dangerous situation beforehand.

Although promising results were obtained, the proposed ML methodology should be validated with more fault cases in real production scenarios. Availability of new vibration databases would open this task as a future line of work.

APPENDIX I

Resumen del trabajo

El presente trabajo de tesis se centra en la investigación en el campo del Aprendizaje Computacional (AC) y en su aplicación a entornos industriales. Más concretamente, se estudiará la aplicación de métodos de AC a la implementación de sistemas automáticos de mantenimiento predictivo de maquinaria rotativa. Por una parte, el AC intenta detectar patrones en conjuntos de datos que pueden haber sido originados por fenómenos de la más variada tipología. Por otra parte, el mantenimiento predictivo es una disciplina perteneciente al campo de la ingeniería industrial que se centra en detectar la posibilidad de aparición de roturas significativas en la maquinaria que impidan su funcionamiento en el presente o en el futuro cercano. Para llevar a cabo ésta detección, se basa en medidas físicas de las condiciones de funcionamiento de sus componentes internos. En esta tesis se pone de manifiesto que estas dos disciplinas pueden beneficiase mutuamente.

El mantenimiento predictivo de maquinaria rotativa supone un reto para el AC debido a la naturaleza de los datos generados por ésta: (a) cada máquina tiene sus propias particularidades y condiciones de montaje que la convierten en un caso individual - motivo por el cual no es viable disponer de datos de fallo para cada una de ellas, con los que construir un modelo de clasificación -, (b) las condiciones de funcionamiento de la máquina (velocidad, carga, etc.) son cambiantes en muchos casos y generan desviaciones en los datos que deben ser distinguidas de una situación de fallo.

El mantenimiento predictivo supone la revisión periódica, por parte de expertos en la materia, de medidas físicas recogidas in situ en la propia maquinaria. Este proceso genera costes de gestión de la información, recursos humanos especializados y tecnologías de adquisición. El AC puede ayudar al mantenimiento predictivo en los siguientes aspectos principales: (a) reducir los costes de producción de planta a través de la automatización de actividades de revisión periódica y (b) reducir la probabilidad de daños significativos en la maquinaria debido a la posibilidad de aumentar la frecuencia de monitorización de ésta, con un incremento de costes reducido. La construcción de sistemas automáticos de revisión de la condición supondría un ahorro de costes en

sectores tan estratégicos como la producción de energía eléctrica, transporte público, etc. La construcción de un sistema de este tipo supone un reto tanto desde el punto de vista de las TIC como desde el punto de vista de la Inteligencia Artificial.
El principal objetivo de esta tesis es desarrollar algoritmos novedosos en el campo del AC capaces de abordar el problema del mantenimiento predictivo de maquinaria rotativa en escenarios reales. Se proponen técnicas de propósito general dentro del campo del AC y algoritmos específicos para resolver la problemática del mantenimiento predictivo automático de maquinaria. Los algoritmos propuesto son capaces de tratar las restricciones en los datos antes mencionadas. La disponibilidad solamente de datos en condiciones normales, o dicho de otro modo la ausencia de contraejemplos para realizar el proceso de aprendizaje, nos obliga a la utilización de algoritmos de detección de anomalías. Éste será por tanto uno de los bloques principales del trabajo. Además, el mantenimiento predictivo también precisa valorar cuál va a ser el estado de la máquina en el futuro cercano de cara a tomar las contramedidas necesarias a tiempo. Esta necesidad nos lleva al segundo bloque principal del presente trabajo también dedicado al AC, el aprendizaje en tiempo real (*on-line*), que nos ayudará en esta tarea. A continuación se resumen las aportaciones principales de cada uno de los bloques de la tesis.

Algoritmos de detección de anomalías

El primer bloque de la tesis se dedica a la detección de anomalías o, expresado de otro modo, al aprendizaje de modelos de clasificación en ausencia de contraejemplos. Los algoritmos de detección de anomalías han ido ganado importancia en las últimas décadas debido a la proliferación de problemas que implican el aprendizaje de modelos de clasificación cuando el número de datos en condiciones normales es muy abundante, pero el número de contraejemplos es reducido o nulo. A pesar de que en esta era de la información nos vemos sobrepasados por la disponibilidad de grandes volúmenes de datos (cantidades que muchas veces son casi imposibles de tratar), en muchas ocasiones sólo un porcentaje ínfimo de esta información, las anomalías o eventos inesperados, aportan la mayor y más valiosa información. A continuación citamos algunos ejemplos de este hecho: se tramitan miles de pagos con tarjeta de crédito cada minuto, pero sólo un ínfimo porcentaje de ellos suponen un fraude que puede acarrear pérdidas para clientes y compañías; se realizan miles de operaciones de valores por segundo pero sólo un pequeño porcentaje de los corredores de bolsa tienen la posibilidad de realizar operaciones sospechosas de ser fraudulentas. No todas las anomalías tienen que poseer necesariamente connotaciones negativas: las grandes cadenas de distribución realizan ventas a miles de clientes cada día y, dentro de estas operaciones, patrones de venta anómalos pueden ser indicativos de cambios de tendencia en los gustos que pueden ser

analizados y explotados para ofrecer un mejor servicio.
Desde un punto de vista abstracto, el término de detección de anomalías se refiere al problema de encontrar patrones que no están conformes con un comportamiento esperado o normal. En términos matemáticos, contamos con un fenómeno a estudiar que genera patrones de datos descriptivos de sí mismo $\{x_1, x_2, \ldots x_n\}$ en un espacio de entrada X. Bajo condiciones normales, estos datos son generados siguiendo una distribución de probabilidad $P(x)$. Esta distribución determina qué patrones son más o menos probables bajo condiciones de normalidad y, de disponer de esta distribución, la detección de anomalías se podría reducir a la comprobación de la verosimilitud de un patrón con respecto a dicha distribución $P(x)$. Sin embargo, la distribución real que genera los datos bajo condiciones de normalidad $P(x)$ es desconocida, por lo que necesitamos construir un modelo/algoritmo, a partir de una muestra disponible, capaz de determinar qué patrones no están conformes con las condiciones esperadas. Los primeros estudios de la deteccin de valores atípicos en conjuntos de datos se remonta a estadísticos del siglo XIX. Desde ese momento, la detección de anomalías ha encontrado su aplicación en dominios tan variados como la detección de intrusiones en redes de computadoras, la detección de tumores, detección de fraudes, etc. Esta variedad de dominios ha hecho imposible encontrar una única estrategia adecuada para todos ellos, por lo que la literatura al respecto ha sido muy prolífica. Muchas de las técnicas propuestas han sido pensadas para un entorno de aplicación concreto. En este trabajo se proponen tres nuevos algoritmos de detección de anomalías:

- Envoltura de volumen mínimo a través de un conjunto de elipsoides [*Minimum Volume Set of Covering Ellipsoids* (MSCE)]. A pesar de esta amplia variedad de algoritmos de detección de anomalías, casi todos comparten una característica común: determinar la región del espacio de entrada a la que pertenecen los patrones generados bajo condiciones de normalidad. Algunos algoritmos suponen de antemano una forma abstracta para la región del espacio de entrada que recoge los patrones generados bajo condiciones normales. Una de las elecciónes más utilizadas en la literatura es la esfera o su generalización en un elipsoide. Esta elección convierte el problema del aprendizaje de un modelo de detección de anomalías en el problema clásico de determinar el elipsoide de volumen mínimo que cubre un conjunto de datos. Esta elección de forma puede ser muy restrictiva y generar un clasificador con pobres resultados en la práctica si la forma de los datos no se ajusta a la asumida (por ejemplo en el caso de datos multimodales). El primer algoritmo propuesto extiende el citado problema en los siguientes aspectos: (a) se calcula un conjunto de elipsoides de volumen mínimo que genere

una envoltura del conjunto de datos disponible y (b) se utiliza una función de coste que proporciona una estimación robusta contra datos ruidosos. Gracias a estas dos propiedades, se proporciona un algoritmo más flexible y robusto capaz de aproximar regiones más complejas de manera más precisa.

- Clasificación de una clase basada en estadísticos de orden [*Extreme Value Statistics One-class Classifier* (EVOC)]. En ciertas aplicaciones prácticas, es complejo asumir a priori una forma de la región del espacio de entrada que alberga los patrones generados bajo condiciones de normalidad. En el segundo algoritmo propuesto se establece un criterio de decisión capaz de decidir si un patrón es anómalo o no sin basarse directamente en la determinación de una región del espacio de entrada. Este algoritmo pone de manifiesto que, utilizando resultados del campo de los estadísticos de orden y modelando la distancia entre patrones bajo condiciones de normalidad, es posible abordar la tarea de detección de anomalías de un modo efectivo. Además, este algoritmo puede ser aplicado a tipos de datos más complejos no pertenecientes necesariamente a un espacio Euclídeo (secuencias, grafos, etc.), debido a que la estrategia de clasificación propuesta sólo requiere disponer de una medida de distancia entre patrones.

- Detección de anomalías en tiempo real [*On-line Stream Data Anomaly Detection* (OSDAD)]. La mayoría de los algoritmos de detección de anomalías existentes en la literatura tratan el problema considerando que un conjunto finito de patrones bajo condiciones de normalidad está disponible de antemano para construir el modelo. Una problemática diferente aparece cuando nuevos patrones son recibidos continuamente y se desea señalizar de manera automática y en tiempo real aquellas regiones del flujo de datos que presentan un comportamiento anómalo. El último algoritmo propuesto está especialmente diseñado para abordar este problema haciendo uso de la combinación de un algoritmo de clasificación de una clase pasivo-agresivo y gráficos CUSUM para distribuciones de Bernouilli. Este algoritmo es capaz de tratar en tiempo real un flujo de datos y señalizar aquellas regiones en las que es posible encontrar un comportamiento anómalo. Además de a espacios Euclídeos, el algoritmo propuesto puede aplicarse a patrones pertenecientes a espacios kernel.

Aprendizaje en tiempo real (*on-line*)

El segundo bloque de este trabajo se dedica al aprendizaje en tiempo real (*on-line*) de redes de neuronas artificiales. En el aprendizaje en tiempo real, el modelo de los datos se ajusta utilizando cada patrón en un único paso de actualización. El objetivo de realizar el aprendizaje de este modo es cumplir con las restricciones de ciertos escenarios

como pueden ser: (a) restricciones de procesamiento en tiempo real, (b) contemplar grandes bases de datos o (c) adaptarse a cambios en la distribución de los datos de entrada y la variable a predecir. Los algoritmos de aprendizaje en tiempo real están ganando protagonismo debido a su gran aplicabilidad en las grandes bases de datos modernas. El modelado de consumo de contenidos web o del comportamiento de indicadores de valores bursátiles en tiempo real son sólo dos ejemplos de los campos en los que este tipo de algoritmos de Aprendizaje Computacional pueden generar un gran impacto.
En el presente trabajo se revisa la problemática del aprendizaje en tiempo real (*on-line*) así como los distintos contextos en los que puede aparecer (grandes conjuntos de datos, cambios de contexto y datos distribuidos) y se propone un nuevo algoritmo de aprendizaje para redes de neuronas monocapa con función de salida no lineal. El algoritmo propuesto es capaz de entrenar cada neurona procesando cada patrón una única vez. De este modo, se reducen en gran medida los recursos de memoria y procesamiento necesarios para realizar el aprendizaje. Además es posible demostrar que, bajo ciertas condiciones, el algoritmo alcanza el mínimo global de la función de coste para todo el conjunto de entrenamiento disponible. También se demuestra que el algoritmo propuesto recoge implícitamente un término de regularización que puede ser explotado en problemas con un número insuficiente de patrones. Es destacable también que este algoritmo abarca como casos concretos varios algoritmos de aprendizaje que ya estaban presentes previamente en la literatura, como el algoritmo de Mínimos Cuadrados Recursivo.

Mantenimiento predictivo basado en el análisis automático de datos
En la segunda parte de esta tesis se trata el campo del mantenimiento predictivo automático de maquinaria rotativa basado en el análisis de registros de vibración. Un ejemplo gráfico que pone de manifiesto el impacto que un sistema software de este tipo puede tener en los costes de producción y la seguridad de un proceso industrial dependiente de maquinaria rotativa es el siguiente:

Un generador eólico valorado en varios millones de dólares se encuentra generando energía eléctrica de manera continua mar adentro en la costa de Gales. Debido a la corrosión y la fatiga de sus materiales, uno de sus componentes internos principales está produciendo la fatiga de todo el tren de potencia. De pararse en este momento, la reparación supondría únicamente el reemplazo del componente dañado. Pero, debido al mal tiempo, este defecto no ha podido ser detectado y en pocas horas provocará una avería que mantendrá parado al generador durante semanas o meses y cuyo coste se

eleva hasta cientos de miles de dólares.

Este ejemplo pone de manifiesto que un proceso de mantenimiento deficiente puede poner en peligro la viabilidad económica de ciertos procesos de producción, como el de la energía eólica, tanto por la indisponibilidad no programada de la maquinaria como por los altos costes de las reparaciones. El mantenimiento predictivo es una metodología que establece la viabilidad de funcionamiento de maquinaria rotativa basándose en la medida y el análisis periódico de variables que revelan el estado de sus componentes internos. Gracias a una detección temprana de los defectos (inevitables debido a la fatiga a la que se someten los materiales), se evitan roturas con posibles consecuencias catastróficas y es posible la programación de actividades de reparación de la manera menos costosa posible. El mantenimiento predictivo y las tecnologías asociadas a él se han convertido en factores clave a la hora de extender la vida de los equipos, reducir costes y aumentar la disponibilidad de los activos de producción.
La inspección continua de la maquinaria es, desafortunadamente, imposible en muchos casos debido a que el incremento de los costes en recursos humanos es prohibitivo. Uno de los factores que influyen en gran medida en dicho coste es la necesidad de inspeccionar continuamente capturas de datos que no presentan ningún problema. Ésta es la razón que hace que, en la práctica, la periodicidad de las inspecciones no sea tan frecuente como sería deseable. Es en este aspecto en donde el presente trabajo pretende dar una respuesta efectiva. El objetivo principal de la segunda parte es construir un software que sea capaz de: (a) percibir los síntomas de fallo interno de la maquinaria, (b) notificarlos antes de que puedan producir una rotura desastrosa y (c) interpretar dichos síntomas para realizar un diagnóstico. La viabilidad de inversión en un software de este tipo es fácilmente palpable en ciertos sectores como el de la energía eólica, que en 2012 acumulaba una potencia instalada de 238 GW en todo el mundo. El primer paso para construir un software de esta naturaleza es determinar las variables físicas que nos van a permitir percibir un síntoma de fallo. El análisis de vibraciones es una de las técnicas más efectivas a este respecto debido a su capacidad para revelar síntomas de fallo interno, sus reducidos costes con respecto a otros sistemas de adquisición de datos de condición (como son el análisis de aceite, ultrasonidos, etc.) y su capacidad para monitorizar continuamente y de manera automática cualquier tipo de maquinaria rotativa. En segundo lugar, de cara a automatizar el proceso de detección de fallos, un software de mantenimiento predictivo basado en vibraciones debe ser capaz de distinguir entre capturas de vibración que no revelan síntomas de defectos de aquellas en las que se revela algún tipo de fallo. El Aprendizaje Computacional puede dar una respuesta efectiva a esta problemática y será un tema central en la última parte del trabajo. Se propone la aplicación de algoritmos de detección de anomalías a la clasifi-

cación de capturas de vibración y se detalla la metodología a utilizar para analizarlas automáticamente utilizando tanto algoritmos ya disponibles en la literatura como los algoritmos propuestos en la primera parte de este trabajo. Algunos de estos algoritmos se integraron en el prototipo de software comercial GIDAS. Este software fue desarrollado por el autor en colaboración con la multinacional INDRA Sistemas S.A. Las experiencias extraídas en la aplicación del Aprendizaje Computacional al campo del mantenimiento predictivo en plantas reales con el citado software y los casos reales de fallos detectados in situ en condiciones de producción también son descritos.

Organización de los contenidos del trabajo

En este resumen se han introducido los temas principales que se tratan en este trabajo. El primer bloque (algoritmos de detección de anomalías) se trata en los capítulos 2 y 3. El capítulo 2 hace un repaso al problema de la detección de anomalías, los retos principales que plantea, los algoritmos que conforman el estado del arte así como los grupos en los que éstos se pueden clasificar. El capítulo 3 recoge los algoritmos de detección de anomalías que se proponen en este trabajo. A continuación, la parte dedicada al aprendizaje en tiempo real se recoge en el capítulo 4, donde se hace un repaso a esta casuística de aprendizaje y se detalla el algoritmo de aprendizaje neuronal propuesto. El capítulo 5 supone el punto de inflexión del trabajo entre el AC y el campo de aplicación concreto estudiado. En este capítulo se realiza una introducción al análisis de vibraciones y su aplicación al mantenimiento predictivo. Los conceptos y principios estudiados en este capítulo serán utilizado en secciones posteriores para la construcción de algoritmos de detección de fallos. En el capítulo 6 se presenta el software GIDAS, el banco de trabajo utilizado para llevar los algoritmos de detección propuestos a escenarios de producción reales, y se relata la experiencia piloto de su instalación llevada a cabo en parques eólicos. Finalmente, en el capítulo 7 se detallan los resultados obtenidos por los algoritmos propuestos en el presente trabajo para casos de detección de fallos en rodamientos, uno de los componentes mecánicos más habituales en maquinaria rotativa y que acumula un alto porcentaje de las roturas de maquinaria.

APPENDIX II

Author's key publications and mentions

The contents of the present work have been published in the following specialized journals and forums:

JCR Journals

David Martínez-Rego, Enrique Castillo, Óscar Fontenla-Romero and Amparo Alonso-Betanzos. *A minimum volume covering approach with a set of ellipsoids.* IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013 (In press).

David Martínez-Rego, Beatriz Pérez-Sánchez, Óscar Fontenla-Romero, Amparo Alonso-Betanzos. *A robust incremental learning method for non-stationary environments.* Neurocomputing 74(11), 1800-1808 (2011).

David Martínez-Rego, Óscar Fontenla-Romero, Amparo Alonso-Betanzos. *Nonlinear single layer neural network training algorithm for incremental, nonstationary and distributed learning scenarios.* Pattern Recognition 45(12): 4536-4546 (2012)

Diego Fernández-Francos, David Martínez-Rego, Óscar Fontenla-Romero, Amparo Alonso-Betanzos. *Automatic bearing fault diagnosis based on one-class ν-SVM.* Computers & Industrial Engineering 64(1): 357-365 (2013)

JCR Journals (Under review process)

David Martínez-Rego, Diego Fernández-Francos, Óscar Fontenla-Romero and Amparo Alonso-Betanzos. *On line anomaly detection via passive-agressive one-class classification.* IEEE Transactions on Systems, Man and Cybernetics (Part C: Cybernetics). 2013.

David Martínez-Rego, Óscar Fontenla-Romero, Amparo Alonso-Betanzos and José C. Principe. *Fault Detection via Recurrence Time Statistics and one-class classification.* Applied Soft Computing. 2012.

Conferences

David Martínez-Rego, Óscar Fontenla-Romero, Amparo Alonso-Betanzos, Manuel Vilaboy Jarel, Alberto Bouza Durán. *Vibration analysis and condition forecasting for rotating machinery using local modeling neural networks.* Proceedings of the IASTED International Conferemce on Artificial Intelligence and Soft Computing, pages 122-127, 2008.

David Martínez-Rego, Óscar Fontenla-Romero, Amparo Alonso-Betanzos. *Power wind mill fault detection via one-class ν-SVM vibration signal analysis.* International Joint Conference on Neural Networks (IJCNN). 2011: 511-518

David Martínez-Rego, Evan Kriminger, José C. Principe, Óscar Fontenla-Romero and Amparo Alonso-Betanzos. *One-class classifier based on extreme value statistics.* Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN). 2012, 405-410.

Software Registration

Software registration of the product *Gestión Integrada de Diagnóstico y Análisis de Sistemas (GIDAS).* Authorship shared by INDRA Systems S.A. and the author and advisors of the present PhD. dissertation.

Mentions

Best Doctoral Project for the article *Fault Diagnosis and Prognosis of Rotating Machinery based on Machine Learning and Expert Systems*. Conference of the Spanish Association of Artificial Intelligence (CAEPIA). 2009.

Bibliography

[1] Abarbanel H. "Analysis of Observed Chaotic Data". Springer-Verlag (1996).

[2] Abraham B. and Box G. E. Bayesian analysis of some outlier problems in time series. *Biometrika* **66**(2), 229–236 (1979).

[3] Abraham B. and Chuang A. Outlier detection and time series modeling. *Technometrics* **31**(2), 241–248 (1989).

[4] Agarwal D. A probabilistic resource allocating network for novelty detection. *Neural Computation* **6**(2), 270–284 (1994).

[5] Agarwal D. Detecting anomalies in cross-classified streams: A bayesian approach. *Knowledge Information Systems* **11**(1), 29–44 (2006).

[6] Aggarwal C., Hinneburg A., and Keim D. On the surprising behavior of distance metrics in high dimensional space. *Lecture Notes in Computer Science (ICDT 2001)* pages 420–434 (2001).

[7] Aggarwal C. and Yu P. "Privacy-Preserving Data Mining: Models and Algorithms". Springer Publishing Company (2008).

[8] Agovic A., Banerjee A., Ganguly A. R., and Protopopescu V. Anomaly detection in transportation corridors using manifold embedding. In "Proceedings of the 1st International Workshop on Knowledge Discovery from Sensor Data", pages 435–455 (2007).

[9] Agrawal R. and Srikant R. Mining sequential patterns. In "Proceedings of the 11th International Conference on Data Engineering. IEEE Computer Society", pages 3–14 (1995).

[10] Agresti A. and Coull B. Approximate is better than ''exact" for interval estimation of binomial proportions. *American Statistician* **52**(2), 119–126 (1998).

[11] Agyemang M., Barker K., and Alhajj R. A comprehensive survey of numeric and symbolic outlier mining techniques. *Intelligent Data Analysis* **10**(6), 521–538 (2006).

[12] Ahmed T., Coates M., and Lakhina A. Multivariate online anomaly detection using kernel recursive least squares. In "INFOCOM 2007. 26th IEEE International Conference on Computer Communications.", pages 625–633 (2007).

[13] Aleskerov E., Freisleben B., and Rao B. Cardwatch: A neural network based database mining system for credit card fraud detection. In "Proceedings of the IEEE Conference on Computational Intelligence for Financial Engineering.", pages 220–226 (1997).

[14] Alippi C. and Roveri M. Just-in-time adaptive classifiers part i: Detecting nonstationary changes. *IEEE Transactions on Neural Networks* **19**, 1145–1153 (2008).

[15] Alpaydin E. "Introduction to Machine Learning (Adaptive Computation and Machine Learning)". The MIT Press (2004).

[16] Ando S. Clustering needles in a haystack. an information theoretic analysis of minority and outlier detection. In "Proceedings of the 7th International Conference on Data Mining", pages 13–22 (2007).

[17] Angiulli F. and Pizzuti C. Fast outlier detection in high dimensional spaces. In "Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery", pages 15–26. Springer-Verlag (2002).

[18] Anscombe F. J. and Guttman I. Rejection of outliers. *Technometrics* **2**(2), 123–147 (1960).

[19] Arning A., Agrawal R., and Raghavan P. A linear method for deviation detection in large databases. In "Proceedings of the 2nd International Conference of Knowledge Discovery and Data Mining", pages 164–169 (1996).

[20] B. Scholkopf B., Platt J., Shawe-Taylor J., Smola A., and Williamson R. Estimating the support of a high-dimensional distribution. *Neural Computation* **13**(7), 1443–1471 (2001).

[21] Barbara D., Couto J., Jajodia S., and Wu N. Adam: A testbed for exploring the use of data mining in intrusion detection. In "SIGMOD Rec", volume 30, pages 15–24 (2001).

[22] Barnes E. An algorithm for separating patterns by ellipsoids. *IBM Journal on Research and Development* **26**, 759–764 (1982).

[23] Barnett V. The ordering of multivariate data (with discussion). *Journal of Royal Statistics Society Series* **139**, 318–354 (1976).

[24] Barnett V. and Lewis T. "Outliers in Statistical Data". John Wiley & Sons (1994).

[25] Basseville M., Nikiforov I., et al.. "Detection of abrupt changes: theory and application", volume 15. Prentice Hall Englewood Cliffs (1993).

[26] Basu S., Bilenko M., and Mooney R. J. A probabilistic framework for semi-supervised clustering. In "10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pages 59–68 (2004).

[27] Bay S. D. and Schwabacher M. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In "Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pages 29–38. ACM Press (2003).

[28] Beckman R. J. and Cook R. D. Outlier...s. *Technometrics* **25**(2), 119–149 (1983).

[29] Bishop C. Novelty detection and neural network validation. In "IEEE Proceedings on Vision, Image and Signal Processing. Special Issue on Applications of Neural Networks", pages 217–222 (1994).

[30] Bishop C. "Neural Networks for Pattern Recognition". Oxford University Press, USA, 1 edition (January 1996).

[31] Bishop C. "Pattern Recognition and Machine Learning". Springer, st edition (2007).

[32] Bourlard H. and Kamp Y. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics* **59**(4), 291–294 (September 1988).

[33] Boyd S., El Ghaoui L., Feron E., and Balakrishnan V. "Linear Matrix In- equalities in System and Control Theory". Society for Industrial and Applied Mathematics (1994).

[34] Boyd S. and Vandenberghe L. "Convex Optimization". Cambridge University Press (2004).

[35] Brause R., Langsdorf T., and Hepp M. Neural data mining for credit card fraud detection. In "Proceedings of the IEEE International Conference on Tools with Artificial Intelligence", pages 103–106 (1999).

[36] Breunig M. M., Kriegel H., NG R., and Sander J. Optics-of: Identifying local outliers. In "Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery", pages 262–270 (1999).

[37] Breunig M. M., Kriegel H., Ng R., and Sander J. Lof: Identifying density-based local outliers. In "Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery", pages 93–104 (2000).

[38] Brito M. R., Chavez E. L., Quiroz A. J., and Yukich J. E. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statis. Prob. Lett.* **35**(1), 33–42 (1997).

[39] Brown L., Cai T., and DasGupta A. Interval estimation for a binomial proportion. *Statistical Science* **16**, 101–117 (2001).

[40] Burkard R., Dell'Amico M., and Martello S. "Assignment Problems". SIAM (2009).

[41] Byers S. D. and Raftery A. E. Nearest neighbor clutter removal for estimating features in spatial point processes. *Journal of American Statistics Association* **93**, 577–584 (1998).

[42] C. R. and Morari M. Determining the model order of nonlinear input/output systems directly from the data. In "Proceedings of the American Control Conference", volume 3, pages 2190–2195 (1995).

[43] Camci F. and Chinnam R. General support vector representation machine for one-class classification of non-stationary classes. *Pattern Recognition* **41**(10), 3021–3034 (2008).

[44] Cao L. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica* **110**(1), 43–50 (1997).

[45] Caragea D. "Learning classifiers from distributed, semantically heterogeneous, autonomous data sources". Ph.D. thesis, Department of Computer Sciene, Iowa State University (2004).

[46] Casdagli M. Nonlinear prediction of chaotic time series. *Physical Review D* **35**(3), 35–356 (1989).

[47] Castillo E., Conejo A., Castillo C., and Mínguez R. Closed formulas in local sensitivity analysis for some classes of linear and non-linear problems. *TOP* **15**(2), 355–371 (2007).

[48] Castillo E., Conejo A., Hadi A., and Fernández-Canteli A. A general method for local sensitivity analysis with application to regression models and other optimization problems. *Technometrics* **46**(4), 430–444 (2004).

[49] Castillo E., Conejo A., Pedregal P., García R., and Alguacil N. "Building and Solving Mathematical Programming Models in Engineering and Science". Pure and Applied Mathematics: A Wiley-Interscience Series of Texts, Monographs and Tracts. John Wiley & Sons Inc., New York (2001).

[50] Castillo E., Conejo A., Castillo C., Mínguez R., and Ortigosa D. Perturbation approach to sensitivity analysis in mathematical programming. *Journal of Optimization Theory and Applications* **128**(1), 49–74 (2006).

[51] Castillo E., Conejo A., Mínguez R., and Castillo C. A closed formula for local sensitivity analysis in mathematical programming. *Engineering Optimization* **38**(1), 93–112 (2006).

[52] Chakrabarti S., Sarawagi S., and Dom B. Mining surprising patterns using temporal description length. In "Proceedings of the 24rd International Conference on Very Large Data Bases", pages 606–617 (1998).

[53] Chandola V., Banerjee A., and Kumar V. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)* **41**(3), 15 (2009).

[54] Chiu A. and Chee Fu A. W. Enhancements on local outlier detection. In "Proceedings of the 7th International Database Engineering and Applications Symposium", pages 298–307 (2003).

[55] Chiu S. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems* **2**, 267–278 (1994).

[56] Cohen W. Fast effective rule induction. In "Proceedings of the Twelfth International Conference in Machine Learning" (1995).

[57] Council G. W. E. Global wind statistics 2012. http://www.gwec.net/ global-figures/graphs/. 2010.

[58] Crammer K., Dekel O., Keshet J., Shalev-Shwartz S., and Singer Y. Online passive-aggressive algorithms. *The Journal of Machine Learning Research* **7**, 551–585 (2006).

[59] Crammer Y. "Online learning of complex categorical problems". PhD thesis, Hebrew University of Jerusalem (2004).

[60] Denning D. An intrusion detection model. *IEEE Transactions on Software Engineering* **13**(2), 222–232 (1987).

[61] Desforgues M., Jacob P., and Cooper J. Applications of probability density estimation to the detection of abnormal conditions in engineering. In "Proceedings of the Institute of the Mechanical Engineers", volume 212, pages 687–703 (1998).

[62] Dolia E., Bie T., Harris C., Shawe-taylor J., and Titterington D. The minimum volume covering ellipsoid estimation in kernel-defined feature spaces. In "European Conference on Machine Learning", pages 630–637 (2007).

[63] dos Santos Teixeira P. and Milidiú R. Data stream anomaly detection through principal subspace tracking. In "Proceedings of the 2010 ACM Symposium on Applied Computing", pages 1609–1616. ACM (2010).

[64] Duda R., Hart P. E., and Stork D. G. "Pattern Classification (2nd Edition)". Wiley-Interscience, 2 edition (November 2001).

[65] Duda R., Hart P., and Stork D. "Pattern Classification". John Wiley & Sons (2001).

[66] Dutta H., Giannella C., Borne K., and Kargupta H. Distributed top-k outlier detection in astronomy catalogs using the demac system. In "Proceedings of the 7th SIAM International Conference on Data Mining" (2007).

[67] Dyer K. and Polikar R. Semi-supervised learning in initially labeled non-stationary environments with gradual drift. *International Joint Conference on Neural Networks (IJCNN 2012)* pages 730–742 (2012).

[68] E. Castillo E., Hadi A., Balakrishnan N., and Sarabia J. "Extreme Value and Related Models with Applications in Engineering and Science". Wiley Series in Probability and Statistics (2005).

[69] Ebersbach S. and Peng Z. Expert system development for vibration analysis in machine condition monitoring. *Expert Systems with applications* **34**, 291–299 (2008).

[70] Edgeworth F. Y. On discordant observations. *Philosoph. Mag.* **23**(5), 364–375 (1887).

[71] Elahi M., Li K., Nisar W., Lv X., and Wang H. Efficient clustering-based outlier detection algorithm for dynamic data stream. In "Proceedings of the 2008

Fifth International Conference on Fuzzy Systems and Knowledge Discovery", volume 5, pages 298–304. IEEE Computer Society (2008).

[72] Elwell R. and Polikar R. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks* **22**, 1517–1531 (2011).

[73] Ericsson S., Grip N., Johansson E., Persson L., Sjoberg R., and Stromberg J. Towards automatic detection of local bearing defects in rotating machines. *Mechanical Systems and Signal Processing* **19**(3), 509–535 (2005).

[74] Ertoz L., Steinbach M., and Kumar V. Finding topics in collections of documents: A shared nearest neighbor approach. *Proceedings of Text Mine'01, First SIAM International Conference on Data Mining* (2001).

[75] Eskin E., Arnold A., Prerau M., Portnoy L., and Stolfo S. A geometric framework for unsupervised anomaly detection. In "Proceedings of the Conference on Applications of Data Mining in Computer Security", pages 78–100. Kluwer Academics (2002).

[76] Eskin E., Lee W., and Stolfo S. Modeling system call for intrusion detection using dynamic window sizes. In "Proceedings of DARPA Information Survivability Conference and Exposition (DISCEX)" (2001).

[77] Ester M., Kriegel H.-P., Sander J., and Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In "2nd International Conference on Knowledge Discovery and Data Mining", pages 226–231 (1996).

[78] Fan W., Miller M., Stolfo S. J., Lee W., and Chan P. K. Using artificial anomalies to detect unknown and known network intrusions. In "Proceedings of the IEEE International Conference on Data Mining. IEEE Computer Society", pages 123–130 (2001).

[79] Fang S. and Zijie W. Rolling bearing fault diagnosis based on wavelet packet and rbf neural network. In "IEEE Control Conference", pages 451–455. IEEE (2007).

[80] Fang S. and Zijie W. Rolling bearing fault diagnosis based on wavelet packet and RBF neural network. In "IEEE Control Conference", pages 451–455 (2007).

[81] Fawcett T. and Provost F. Activity monitoring: noticing interesting changes in behavior. In "Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pages 53–62 (1999).

[82] Fletcher R. "Practical Methods of Optimization". John Wiley and Sons (1987).

[83] Fontenla-Romero O., Guijarro-Berdiñas B., Pérez-Sánchez B., and Alonso-Betanzos A. A new convex objective function for the supervised learning of single-layer neural networks. *Pattern Recognition* **43**(5), 1984–1992 (2010).

[84] Fox J. Outliers in time series. *J. Royal Statistics Society (Series B)* **34**(3), 350–363 (1972).

[85] Frank A. and Asuncion A. UCI machine learning repository (2010).

[86] Fränti P. and Virmajoki O. Iterative shrinking method for clustering problems. *Pattern Recognition* **39**, 761–775 (May 2006).

[87] Fujimaki R., Yairi T., and Machida K. An approach to spacecraft anomaly detection problem using kernel feature space. In "Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining", pages 401–410. ACM Press (2005).

[88] Galeano P., Pea D., and Tsay R. S. Outlier detection in multivariate time series via projection pursuit. *Statistics and econometrics working articles* (2004).

[89] Gao J., Cao Y., Tung W., and Hu J. "Multiscale analysis of complex time series". Wiley (2007).

[90] Golub G. and Loan C. "Matrix Computations (3rd edition)". Johns Hopkins, Baltimore (1996).

[91] Gomes J. and Barroso V. Array-based QR-RLS multichannel lattice filtering. *IEEE Transactions on Signal Processing* **56**(8), 3510–3522 (2008).

[92] Guha S., Rastogui R., and Shim K. Rock: A robust clustering algorithm for categorical attributes. *Information Systems* **25**(5), 345–366 (2000).

[93] Gustafson D. E. and Kessel W. Fuzzy clustering with a fuzzy covariance matrix. *Chemometrics and Intelligent Laboratory Systems* **86**(2), 761–766 (1979).

[94] Guttormsson S. E., Marks R. J. I., El-Sharkawi M. A., and Kerszenbaum I. Elliptical novelty grouping for online short-turn detection of excited running rotors. *IEEE Transactions Energy Conversion* **14**(1) (1999).

[95] Harris T. "Rolling bearing analysis". Wiley, New York (1991).

[96] Harville D. "Matrix Algebra from a Statistician's perspective". Springer, 1 edition (1997).

[97] Hautamaki V., Karkkainen I., and Franti P. Outlier detection using k-nearest neighbour graph. In "Proceedings of the 17th International Conference on Pattern Recognition", volume 3, pages 430–433. IEEE Computer Society (2004).

[98] Hawkins D. "Identification of Outliers". Chapman and Hall, London and New York (1980).

[99] Hayes M. "Statistical Digital Signal Processing and Modeling". Wiley (1986).

[100] He X. and Asada H. A new method for identifying orders of input-output models for nonlinear dynamic systems. In "Proceedings of the American Control Conference", pages 2520–2523 (1993).

[101] He Z., Deng S., Xu X., and Huang J. Z. A fast greedy algorithm for outlier mining. In "Proceedings of the 10th Pacific-Asia Conference on Knowledge and Data Discovery", pages 567–576 (2006).

[102] He Z., Xu X., and Deng S. Discovering cluster-based local outliers. *Pattern recognition Letters* **24**, 1641–1650 (2003).

[103] He Z., Xu X., and Deng S. An optimization model for outlier detection in categorical data. In "Proceedings of the International Conference on Intelligent Computing" (2005).

[104] He Z., Xu X., Huang J. Z., and Deng S. "A Frequent Pattern Discovery Method for Outlier Detection". Springer (2004).

[105] Helmer G., Wong J., Honavar V., and Miller L. Intelligent agents for intrusion detection. In "Proceedings of the IEEE Information Technology Conference", pages 121–124 (1998).

[106] Hinneburg A. and Gabriel H. Denclue 2.0: Fast clustering based on kernel density estimation. In "Proceedings of the 7th International Symposium on Intelligent Data Analysis", pages 70–80 (2007).

[107] Hirayama J., Yoshimoto J., and Ishii S. Balancing placticity and stability of on-line learning based on hierarchical bayesian adaptation of forgetting factors. *Neurocomputing* **69**, 1954–1961 (2006).

[108] Hodge V. and Austin J. A survey of outlier detection methodologies. *Artificial Intelligence Reviews* **22**(2), 85–126 (2004).

[109] Hoffmann H. Kernel PCA for novelty detection. *Pattern Recognition* **40**(3), 863–874 (2006).

[110] Huber P. "Robust Statistics". John Wiley & Sons, Inc (1974).

[111] Huber P. Projection pursuit (with discussions). *Annals of statistics* **13**(2), 435–475 (1985).

[112] Iserman R. "Fault-diagnosis systems: an introduction from fault detection to fault tolerance". Springer-Verlag (2007).

[113] J. S. and Tidwell D. "Programming Web Services With SOAP". O'Reilly (2001).

[114] Jain A. K. and Dubes R. C. "Algorithms for Clustering Data". Prentice-Hall, Inc. (1988).

[115] Japkowicz N. "Concept Learning in the Absence of Counter-Examples: An Autoassociation-Based Approach to Classification". PhD thesis, The State University of New Jersey (1999).

[116] Japkowicz N. Supervised versus unsupervised binary-learning by feedforward neural networks. *Machine Learning* **42**(1/2), 97–122 (2001).

[117] Jiang J. and Cook R. Fast parameter tracking RLS algorithm with high noise inmunity. *Electronics Letters* **28**(22), 2043–2045 (1992).

[118] Jiang M. F., Tseng S. S., and Su C. M. Two-phase clustering process for outliers detection. *Pattern Recognition Letters* **22**, 691–700 (2001).

[119] Jin W., Tung A. K. H., and Han J. Mining top-n local outliers in large databases. In "Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pages 293–298 (2001).

[120] John F. Extreme problems with inequalities as subsidiary conditions. *Studies and Essays Presented to R. Courant on his 60th Birthday* pages 187–204 (1948).

[121] Johnson N., Kotz S., and Balakrishnan N. "Continuous univariate distirbutions". Wiley series in probability (1994).

[122] Jolliffe I. T. "Principal Component Analysis". Springer (2002).

[123] Jones R. Enveloping for bearing analysis. *Journal of Sound and Vibration* **30**, 10–15 (1996).

[124] Kamali C., Pashilkar A., and Raol J. Evaluation of recursive least squares algorithm for parameter estimation in aircraft real time applications. *Aerospace Science and Technology* **15**, 165–174 (2011).

[125] Kankar P., Sharma C., and Harsha S. Fault diagnosis of ball bearings using machine learning methods. *Expert Systems with applications* **38**, 1876–1886 (2011).

[126] Kankar P., Sharma C., and Harsha S. Fault diagnosis of ball bearings using machine learning methods. *Expert Systems with applications* **38**, 1876–1886 (2011).

[127] Kennel M., Brown R., and Abarbanel H. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review* **45**, 3403–3411 (1992).

[128] Keogh E., Lonardi S., and Ratanamahatana C. A. Towards parameter-free data mining. In "Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pages 206–215 (2004).

[129] Khachiyan L. Rounding of polytopes in the real number model of computation. *Mathematical Operations Research* **2**, 307–320 (1996).

[130] Khachiyan L. and Todd M. On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Mathematical Programming* **61**, 137–159 (1993).

[131] Kitagawa G. On the use of aic for the detection of outliers. *Technometrics* **21**(2), 193–199 (1979).

[132] Knorr E. M. and Ng R. T. A unified approach for mining outliers. In "Proceedings of the Conference of the Centre for Advanced Studies on Collaborative Research", volume 11. IBM Press (1997).

[133] Knorr E. M. and Ng R. T. Algorithms for mining distance-based outliers in large datasets. In "Proceedings of the 24rd International Conference on Very Large Data Bases", pages 392–403. Morgan Kaufmann Publishers Inc. (1998).

[134] Knorr E. M. and Ng R. T. Finding intensional knowledge of distance-based outliers. *International Journal on Very Large Data Bases* pages 211–222 (1999).

[135] Knorr E. M., Ng R. T., and Tucakov V. Distance-based outliers: Algorithms and applications. *International Journal on Very Large Data Bases* **8**(3-4), 237–253 (2000).

[136] Kou Y., Lu C. T., and Chen D. Spatial weighted outlier detection. In "Proceedings of the SIAM Conference on Data Mining" (2006).

[137] Kramer M. Nonlinear principal component analysis by neural networks. *AIChE Journal* **37**(2), 233–243 (1991).

[138] Kumar P. and Yildirim E. A. Minimum-volume enclosing ellipsoids and core sets. *Journal of Optimization Theory and Applications* **126**(1), 1–21 (2005).

[139] Kumar V. Parallel and distributed computing for cybersecurity. *IEEE Distributed Systems Online* **6**(10), 1–10 (2005).

[140] Lakhina A., Papagiannaki K., Crovella M., Diot C., Kolaczyk E., and Taft N. "Structural analysis of network traffic flows", volume 32. ACM (2004).

[141] Lee J., Qiu H., Yu G., and Lin J. Bearing data set. IMS center. university of cincinnati. *NASA Ames Prognostics Data Repository.[http://ti.arc.nasa.gov/project/prognostic-data-repository]* (2007).

[142] Lee W., Stolfo S., and Chan P. Learning patterns from unix process execution traces for intrusion detection. In "Proceedings of the AAAI Workshop on AI Methods in Fraud and Risk Management" (1997).

[143] Lee W., Stolfo S. J., and Mok K. W. Adaptive intrusion detection: A data mining approach. *Artif. Intell. Rev.* **14**(6), 533–567 (2000).

[144] Lee W. and Xiang D. Information-theoretic measures for anomaly detection. In "Proceedings of the IEEE Symposium on Security and Privacy", volume 130 (2001).

[145] Leung C., Wong K., Sum P., and Chan L. A pruning method for the recursive least squared algorithm. *Neurocomputing* **14**, 147–174 (2001).

[146] Leung S. and So C. Gradient-based variable forgetting factor RLS algorithm in time-varying environments. *IEEE Transactions on Signal Processing* **53**(8), 3141–3150 (2005).

[147] Li M. and Vitanyi P. M. B. "An Introduction to Kolmogorov Complexity and Its Applications". Springer-Verlag (1993).

[148] Lin J., Keogh E., Fu A., and Herle H. V. Approximations to magic: Finding unusual medical time series. In "Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems", pages 329–334 (2005).

[149] Lin S. and Brown D. E. An outlier-based data association method for linking criminal incidents. In "Proceedings of the 3rd SIAM Data Mining Conference" (2003).

[150] Liu H. "Online automatic epileptic seizure detection from electroencephalogram (EEG)". Phd. Dissertation, University of Florida (2005).

[151] Liu J. S. "Monte Carlo Strategies in Scientific Computing". Springer (2008).

[152] Lloyds G. "Guideline for the Certification of Condition Monitoring Systems for Wind Turbines, Edition 2007". Germanischer Lloyds (2010).

[153] Loparo K. Bearings vibration data set. *Journal of Case Western Reserve University* (2003).

[154] Lorenz E. Deterministic nonperiodic flow. *J. Atmospheric Sciences* **20**(2), 130–141 (1963).

[155] Ma J. and Perkins S. Online novelty detection on temporal sequences. In "Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pages 613–618. ACM Press (2003).

[156] Ma J. and Perkins S. Time series novelty detection using one-class support vector machines. In "Proceedings of the International Joint Conference on Neural Networks", volume 3, pages 1741–1745 (2003).

[157] Mackey M. and Glass L. Oscillation and chaos in physological control systems. *Science* **287** (1977).

[158] Mahoney M. V. and Chan P. K. Learning nonstationary models of normal network traffic for detecting novel attacks. In "Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pages 376–385. ACM Press (2002).

[159] Mahoney M. V. and Chan P. K. Learning rules for anomaly detection of hostile network traffic. In "Proceedings of the 3rd IEEE International Conference on Data Mining. IEEE Computer Society", volume 601. ACM Press (2003).

[160] Markou M. and Singh S. Novelty detection: A review-part 1: Statistical approaches. *Signal Processing* **83**(12), 2481–2497 (2003).

[161] Markou M. and Singh S. Novelty detection: A review-part 2: Neural network based approaches. *Signal Processing* **83**(12), 2499–2521 (2003).

[162] Martinez-Rego D., Fontenla-Romero O., and Alonso-Betanzos A. Power wind mill fault detection via one-class ν-svm vibration signal analysis. In "International Joint Conference on Neural Networks 2011 (In press)" (2011).

[163] Martinez-Rego D., Fontenla-Romero O., and Alonso-Betanzos A. Power wind mill fault detection via one-class nu-svm vibration signal analysis. In "Proceedings International Joint Conference on Neural Networks (IJCNN 2011)", pages 511–518 (2011).

[164] Martínez-Rego D., Fontenla-Romero O., Alonso-Betanzos A., and Pérez-Sánchez B. A robust incremental learning method for non-stationary environments. *Neurocomputing* **74**, 1800–1808 (2011).

[165] Martinus D. and Tax J. "One-class classification: Concept learning in the absence of counter-examples". PhD thesis, Technische Universiteit Delft (2001).

[166] Mccallum A., Nigam K., and Ungar L. H. Efficient clustering of high-dimensional data sets with application to reference matching. In "Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pages 169–178. ACM Press (2000).

[167] McFadden P. and Smith J. Model for the vibration produced by a single point defect in a rolling element bearing. *Journal of Sound and Vibration* **1**, 69–82 (1984).

[168] McFadden P. and Smith J. Model for the vibration produced by a single point defect in a rolling element bearing. *Journal of Sound and Vibration* **96**(1), 69–82 (1984).

[169] McInerny S. Basic vibration signal processing for bearing fault detection. *IEEE Transactions on Education* **46**(4-5), 149–156 (2003).

[170] Meng L., Miao W., and Chunguang W. Research on svm classification performance in rolling bearing diagnosis. *Journal of Intelligent Computation Technology and Automation* **3**, 132–135 (2010).

[171] Meng L., Miao W., and Chunguang W. Research on svm classification performance in rolling bearing diagnosis. *Intelligent Computation Technology and Automation* **3**, 132–135 (2010).

[172] Minku L., White A. P., and Yao X. Mining concept-drifting data streams using ensemble classifiers. *Proc. of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining* pages 226–235 (2003).

[173] Minku L., White A. P., and Yao X. The impact of diversity on on-line ensemble learning in the presence of concept drift. *IEEE Transactions on Knowledge and Data Engineering* **22(5)**, 730–742 (2010).

[174] Moshtaghi M., Rajasegarar S., Leckie C., and Karunasekera S. An efficient hyperellipsoidal clustering algorithm for resource-constrained environments. *Pattern Recognition* **44**, 2197–2209 (2011).

[175] Nesterov Y. and Nemirovskii A. "Interior-Point Polynomial Algorithms in Convex Programming". SIAM (1994).

[176] Network P. E. Pascal large scale learining challenge. http://largescale.ml.tu-berlin.de/about/. 2008.

[177] Noble C. C. and Cook D. J. Graph-based anomaly detection. In "Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pages 631–636 (2003).

[178] Oppenheim A. and Schafer R. "Discrete-Time Signal Processing (3rd Edition)". Prentice Hall Signal Processing (2009).

[179] Otey M., Parthasarathy S., Ghoting A., Li G., Narravula S., and Panda D. Towards nic-based intrusion detection. In "Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pages 723–728. ACM Press (2003).

[180] Otey M. E., Ghoting A., and Parthasarathy S. Fast distributed outlier detection in mixed-attribute data sets. In "Data Min. Knowledge Disc.", volume 12, pages 203–228 (2006).

[181] P. S. and Freund R. Computation of minimum-volume covering ellipsoids. *Operations Research* **52**(5), 690–706 (2004).

[182] Palshikar G. K. Distance-based outliers in sequences. *Lecture Notes in Computer Science* **3816**, 547–552 (2005).

[183] Papadimitrou H., S.and Kitagawa and Gibbons C., P. B.and Faloutsos. Loci: Fast outlier detection using the local correlation integral. *Tech. rep. IRP-TR-02-09* (2002).

[184] Parra L., Deco G., and Miesbach S. Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation* **8**(2), 260–269 (1996).

[185] Parzen E. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics* **33**(3), 1065–1076 (1962).

[186] Patcha A. and Park J. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computational Networks* **51**(12), 3448–3470 (2007).

[187] Pires A. and Santos-Pereira C. Using clustering and robust estimators to detect outliers in multivariate data. In "Proceedings of the International Conference on Robust Statistics" (2005).

[188] Pokrajac D., Lazarevic A., and Latecki L. J. Mining for outliers in sequential databases. In "Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining" (2007).

[189] Portnoy L., Eskin E., and Stolfo S. Intrusion detection with unlabeled data using clustering. In "Proceedings of the ACM Workshop on Data Mining Applied to Security" (2001).

[190] Predd J., Kulkarni S., and Poor H. A collaborative training algorithm for distributed learning. *IEEE Transactions on Information Theory* **55**, 1856–1871 (2009).

[191] Qin M. and Hwang K. Frequent episode rules for internet anomaly detection. In "Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications. IEEE Computer Society" (2004).

[192] Quionero-Candela J., Sugiyama M., Schwaighofer A., and Lawrence N. "Dataset Shift in Machine Learning". The MIT Press (2009).

[193] Raghuveer M. and Bopardikar A. "Wavelet transforms introduction to theory and applications". Prentice Hall Signal Processing (1998).

[194] Ramaswamy S., Rastogi R., and Shim K. Efficient algorithms for mining outliers from large data sets. In "Proceedings of the ACM SIGMOD International Conference on Management of Data", pages 427–438. ACM Press (2000).

[195] Reynolds M. and Stoumbos Z. A CUSUM chart for monitoring a proportion when inspecting continuously. *Journal of Quality Technology* **31**(1), 87–108 (1999).

[196] Reynolds M. and Stoumbos Z. A general approach to modeling CUSUM charts for a proportion. *IIE Transactions* **32**(6), 515–535 (2000).

[197] Roberts S. Novelty detection using extreme value statistics. *Proceedings of the IEEE Vision, Image and Signal Processing Conference* **146**, 124–129 (1999).

[198] Roberts S. Extreme value statistics for novelty detection in biomedical signal processing. In "Proceedings of the 1st International Conference on Advances in Medical Signal and Information Processing", pages 166–172 (2002).

[199] Roth V. Outlier detection with one-class kernel fisher discriminants. In "Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)" (2004).

[200] Roth V. Kernel fisher discriminants for outlier detection. *Neural Computational* **18**(4), 942–960 (2006).

[201] Rousseeuw P. and Leroy A. M. "Robust Regression and Outlier Detection". John Wiley & Sons, Inc (1987).

[202] Salvador S. and Chan P. Learning states and rules for time-series anomaly detection. *Tech. rep.CS-2003-05, Department of Computer Science, Florida Institute of Technology Melbourne* (2003).

[203] Sawalhi N. "Diagnostics, Prognostics and Fault Simulation For Rolling Element Bearings". University of New South Wales Thesis (2007).

[204] Sawalhi N. and Randal R. Simulating gear and bearing interactions in the presence of faults part ii: Simulation of the vibrations produced by extended bearing faults. *Mechanical Systems and Signal Processing* **22**, 1952–1966 (2008).

[205] Sawalhi N. and Randall R. Simulating gear and bearing interactions in the presence of faults part I. the combined gear bearing dynamic model and the simulation of localised bearing faults. *Mechanical Systems and Signal Processing* **22**, 1924–1951 (2008).

[206] Scheffer C. and Girdhar P. "Practical Machinery Vibration Analysis and Predictive Maintenance". Newnes (2004).

[207] Schlitter N. Distributed data mining project. http://www.distributeddatamining. org/Challenges. 2011.

[208] Schölkopf B., Barlett P., A.J. S., and Williamson R. New support vector algorithms. *Neural Computation* **12**(5), 11207–1245 (2000).

[209] Schölkopf B., Platt J., Shawe-Taylor J., Smola A., and Williamson R. Estimating the support of a high-dimensional distribution. *Neural Computation* **13**, 2001 (1999).

[210] Schölkopf B., Platt J., Shawe-Taylor J., Smola A., and Williamson R. Estimating the support of a high-dimensional distribution. *Neural Computation* **13**(7), 1443–1471 (2001).

[211] Schölkopf B. and Smola A. "Learning with kernels: Support vector machines, regularization, optimization, and beyond". The MIT Press (2002).

[212] Schölkopf B. and Smola A. "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)". MIT Press (2002).

[213] Shalev-Shwartz S., Singer Y., and Srebro N. Pegasos: Primal estimated sub-gradient solver for SVM. In "ICML '07 Proceedings of the 24th international conference on Machine learning", pages 807–814 (2007).

[214] Shawe-Taylor J. and Cristianini N. "Kernel Methods for Pattern Analysis". Cambridge University Press (2004).

[215] Sheikholeslami G., Chartterjee S., and Zhang A. A multi-resolution clustering approach for very large spatial databases. In "Proceedings of the 24rd International Conference on Very Large Databases.", pages 428–439 (1998).

[216] Shivaswamy P. and Jebara T. Ellipsoidal kernel machines. In "11th International Conference on Artificial Intelligence and Statistics" (2007).

[217] Shyu M., Chen S., Sarinnapakorn K., and Chang L. A novel anomaly detection scheme based on principal component classifier. In "Proceedings of the 3rd IEEE International Conference on Data Mining", pages 353–365 (2003).

[218] Smith R., Bivens A., Embrechts M., Palagiri C., and Szymanski B. Anomaly detection over noisy data using learned probability distributions. In "17th International Conference on Machine Learning", pages 255–262 (2000).

[219] Smith R., Bivens A., Embrechts M., Palagiri C., and Szymanski B. Clustering approaches for anomaly-based intrusion detection. In "Proceedings of the Intelligent Engineering Systems through Artificial Neural Networks", pages 579–584 (2002).

[220] Somervuo P. and Kohonen T. Self-organizing maps and learning vector quantization for feature sequences. *Neural Processing Letters* **10**(2), 151–159 (1999).

[221] Song Q., Hu W., and Xie W. Robust support vector machine with bullet hole image classification. *IEEE Transactions System Man Cybernetics. Part C: Applications and Reviews* **32**(4) (2002).

[222] Song X., Wu M., Jermaine C., and Ranka S. Conditional anomaly detection. *IEEE Transactions Knowledge Data Engenering.* **19**(5), 631–645 (2007).

[223] Spence C., Parra L., and Sajda P. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In "Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis.", pages 3–10 (2001).

[224] Stark J., Broomhead D., Davies M., and Huke J. Takens embedding theorems for forced and stochastic systems. *Nonlinear Analysis: Theory, Methods & Applications* **30**(8), 5303–5314 (1997).

[225] Sun P. and Chawala S. On local spatial outliers. *Proceedings of the 4th IEEE International Conference on Data Mining* pages 209–216 (2004).

[226] Sun P. and Chawala S. Slom: A new measure for local spatial outliers. *Knowledge Information Systems* **9**(4), 412–429 (2006).

[227] Sun P. and Yao X. Sparse approximation through boosting for learning large scale kernel machines. *IEEE Transactions on Neural Networks* **21**(6), 883–894 (2010).

[228] Takens F. Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence (Springer Lecture Notes in Mathematics)* **898**, 365–381 (1980).

[229] Tan P. N., Steinbach M., and Kumar V. "Introduction to Data Mining". Addison-Wesley (2005).

[230] Tan S., Ting K., and Liu F. Fast anomaly detection for streaming data. In "International Joint Conference on Artificial Intelligence (IJCAI)", pages 1511–1516 (2011).

[231] Tandon G. and Chan P. Weighting versus pruning in rule validation for detecting network and host anomalies. In "Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining". ACM Press (2007).

[232] Tang J., Chen Z., Chee Fu A. W., and W. Cheung D. Enhancing effectiveness of outlier detections for low density patterns. In "Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining", pages 535–548. ACM Press (2002).

[233] Tao Y., Xiao X., and Zhou S. Mining distance-based outliers from large databases in any metric space. In "Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pages 394–403. ACM Press (2006).

[234] Tax D. M. J. and Duin R. Support vector data description. *Machine Learning* **54**(1), 45–66 (2004).

[235] Taylor J. "The Gear Analysis Handbook". VCI (2000).

[236] Taylor J. "The Vibration Analysis Handbook". VCI (2003).

[237] Taylor J. and Wyndell Kirkland D. "The Bearing Analysis Handbook: A Practical Guide for Solving Vibration Problems in Bearings". VCI (2004).

[238] Teng H., Chen K., and Lu S. Adaptive real-time anomaly detection using inductively generated sequential patterns. In "Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy. IEEE Computer Society Press", pages 278–284 (1990).

[239] Tikhonov A. Solution of incorrectly formulated problems and the regularization method. *Soviet Math Dokl* **4**, 1035–1038 (1963).

[240] Todd M. and Yildirim E. A. On khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids. *Discrete Applied Mathematics* **155**(13), 1731–1744 (2007).

[241] Torr P. and Murray D. Outlier detection and motion segmentation. In "Proceedings of the SPIE", volume 2059, pages 432–443 (1993).

[242] Tsay R. S., Pea D., and Pankratz A. E. Outliers in multi-variate time series. *Biometrika* **87**(4), 789–804 (2000).

[243] Tsymbal A. The problem of concept drift: Definitions and related work. *Technical Report.Department of Computer Science, Trinity College, Dublin.* (2004).

[244] Turk M. and Pentland A. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience* **3**(1), 71–86 (1991).

[245] Wang C., Sun D., and Toh K. Solving log-determinant optimization problems by a newton-cg primal proximal point algorithm. *SIAM Journal of Optimization* **20**, 2994–3013 (2010).

[246] Wei L., Qian W., Zhou A., and Jin W. Hot: Hypergraph-based outlier test for categorical data. In "Proceedings of the 7th Pacific-Asia Conference on Knowledge and Data Discovery", pages 399–410 (2003).

[247] Widmer G. and Kubat M. Learning in the presence of concept drift and hidden contexts. *Machine Learning* **23**, 69–101 (1996).

[248] Williams G., Baxter R., He H., Hawkins S., and Gu L. A comparative study of rnn for outlier detection in data mining. In "Proceedings of the IEEE International Conference on Data Mining" (2002).

[249] Yairi T., Kato Y., and Hori K. Fault detection by mining association rules from housekeeping data. In "Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space" (2001).

[250] Yamanishi K. and Takeuchi J. A unifying framework for detecting outliers and change points from non-stationary time series data. In "Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining", pages 676–681. ACM (2002).

[251] Yamanishi K., Takeuchi J., Williams G., and Milne P. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In "Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining", pages 320–324. ACM (2000).

[252] Yang Y., Dejie Y., and Cheng J. Hilbert transform techniques in machine diagnostics. In "International Conference on Rotordynamics, Tokyo", pages 409–420 (1986).

[253] Yang Y., Dejie Y., and Cheng J. A roller bearing fault diagnosis method based on EMD energy entropy and ANN. *Journal of Sound and Vibration* **294**, 269–277 (2006).

[254] Yang Y., Dejie Y., and Cheng J. A roller bearing fault diagnosis method based on emd energy entropy and ann. *Journal of Sound and Vibration* **294**(1-2), 269–277 (2006).

[255] Yang Y., Dejie Y., and Cheng J. Application of artificial neural network to failure diagnosis on process industry equipments. In "IEEE International Conference on Natural Computation", volume 3, pages 1190–1193 (2010).

[256] Yu D., Sheikholeslami G., and Zhang A. Findout: Finding outliers in very large datasets. *Knowledge and Information Systems* **4**(4), 387–412 (2002).

[257] Yu W. and Shih N. Bi-loop recursive least squares algorithm with forgetting factors. *IEEE Signal Processing Letters* **13**(8), 505–508 (2006).

[258] Zhang J. and Wang H. Detecting outlying subspaces for high-dimensional data: The new task, algorithms, and performance. *Knowledge and Information Systems* **10**(3), 333–355 (2006).