Universidad Politécnica de Madrid

POLITÉCNICA

E.T.S. INGENIEROS INDUSTRIALES

Departamento de Automática, Ingeniería Eléctrica y Electrónica e
Informática Industrial

Machine Learning for Data-driven Prognostics: Methods and Applications

PhD THESIS

Alberto Diez Oliván

M.Sc. in Computer Science and Artificial Intelligence

PhD supervisors: Prof. Ricardo Sanz Bravo

Autonomous Systems Laboratory

Universidad Politécnica de Madrid (UPM)

Full Prof. Basilio Sierra Araujo

Computer Sciences and Artificial Intelligence

University of the Basque Country (UPV/EHU)

2017

Departamento de Automática, Ingeniería Eléctrica y Electrónica e
Informática Industrial

E.T.S. INGENIEROS INDUSTRIALES

Machine Learning for Data-driven Prognostics: Methods and Applications

PhD THESIS

Alberto Diez Oliván

M.Sc. in Computer Science and Artificial Intelligence

PhD supervisors: Prof. Ricardo Sanz Bravo

Autonomous Systems Laboratory

Universidad Politécnica de Madrid (UPM)

Full Prof. Basilio Sierra Araujo

Computer Sciences and Artificial Intelligence

University of the Basque Country (UPV/EHU)

2017

Thesis Committee

President: Pedro Larrañaga

External Member: Darío García

Member: Idoia Alarcón

Member: Diego Galar

Secretary: Manuel Rodríguez

Abstract

Knowledge extraction from monitoring sensor data has gained a lot of attention from many fields of research during recent years. Artificial intelligence, machine learning, advanced statistics, the Internet of things and architectures and strategies for optimal big data management are good examples of such interest. This is mainly due to the increase in the amount of data available and in the storage and speed capabilities of actual computing systems. The main motivation of this research is on providing automatic behavior modeling and intelligent decision-making strategies to prevent critical events and damages, which can imply important loss of money and safety issues.

The research activity performed also explores the difficulties derived from data modeling in several industrial sectors, figuring out specific needs and requirements. The reliability of complex assets and equipment is crucial to minimize faults and the related negative impact in terms of money and time loss, to mitigate potential risks and to successfully accomplish the task. Proactive maintenance is especially relevant to this concern. It is executed on the basis of corrective and predictive techniques that allow obtaining a diagnosis, and even anticipating potential failures and events of interest.

This PhD dissertation is also focused on the development of the Diagnosis and Impact Model 4.0, which is motivated by the fourth industrial revolution from a data science perspective. The underlying idea is to apply Machine Learning paradigms for data-driven behavior modeling and prognostics in complex systems, from imagination and innovation to real impact. Several successful cases are shown in this dissertation, covering a wide variety of challenging scenarios and assets, and targeting important industrial sectors such as maritime, renewable energy, railway, agro-food, civil structures and machine-tool.

Resumen

En los últimos años la extracción y generación de nuevo conocimiento a partir de datos ha experimentado un creciente interés por parte de la comunidad científica. La inteligencia artificial, el aprendizaje automático, la estadística avanzada, el Internet de las cosas y la gestión inteligente de grandes volúmenes de información son ejemplos representativos de este creciente interés. Todo ello viene motivado por un incremento exponencial en la cantidad de datos disponibles y en las capacidades de almacenaje y velocidad de cómputo de los sistemas de procesamiento actuales. En este contexto la motivación principal para la elaboración de esta tesis doctoral consiste en modelar comportamientos de interés a partir de datos de manera automática, y proveer estrategias óptimas que permitan anticipar fallos y eventos críticos.

La actividad investigadora realizada también explora las dificultades derivadas de la generación de modelos basados en datos en varios sectores industriales, teniendo en cuenta las necesidades y los requisitos específicos de cada sector. La fiabilidad de los activos y equipos es clave a la hora de minimizar la aparición de fallos y el impacto negativo que suponen en cuanto a pérdidas en términos de tiempo y dinero, pero también de cara a mitigar riesgos y a llevar a cabo la tarea planificada de manera satisfactoria. El mantenimiento proactivo, ejecutado en base a técnicas predictivas y correctivas y que permite obtener diagnósticos e incluso anticipar fallos potencialmente críticos y eventos de interés, resulta especialmente relevante en este sentido.

La investigación desarrollada en el marco de esta tesis doctoral está centrada en el desarrollo del Modelo de Diagnóstico e Impacto 4.0, motivado por la cuarta revolución industrial y abordado desde la perspectiva de la ciencia del dato. La idea consiste en aplicar métodos de aprendizaje automático y análisis de datos para el modelado de comportamientos y pronóstico de modos de fallo en sistemas complejos, para tener un impacto real en la sociedad desde la imaginación y la innovación aplicadas. Se presentan y discuten varios problemas y casos de éxito relativos a diferentes escenarios y activos monitorizados, y que representan importantes sectores industriales, como son el marítimo, energías renovables, ferrocarril, agroalimentario, estructuras civiles y máquina-herramienta.

Acknowledgements

It is worth and necessary to acknowledge all the people that have supported me in so many ways to reach the achievements presented in this work. From both, the technical and the personal view, which are actually related in many cases.

My both supervisors, Ricardo Sanz and Basilio Sierra have been essential part of this work, with an unconditional commitment and a highly valuable scientific guidance.

Also to my workmates at the Industry and Transport Division of TECNALIA, in which the research activity described in this dissertation has been mainly carried out. They have taught me so many valuable things during all these years, with patience and wisdom. Alberto Carrascal, Darío García and all the people that have shared so many hours of hard research work and constructive discussions with me, and some after-work drinks as well.

This work has been also carried out in a strong collaboration with the NICTA's Machine Learning Group, and my supervisors and colleagues there also deserve a special mention. Khoa Nguyen, Yang Wang, Fang Chen and many good friends I have met during my stay in Australia. It has been one of the greatest experiences in my life, which helped me grow not only professionally but also personally.

And last but not least, special thanks to my family: my parents, Hilario and Alicia, my brother Edu and to my wife, Sara. Without your help and support during all these years this would never have happened.

Thanks to all of you
Alberto Diez Oliván

From little things big things grow

List of Algorithms	vii
List of Figures	ix
List of Tables	xiii
I WORK DESCRIPTION	1
1 Context of this research activity	3
1.1 Motivation	3
1.2 Context	4
1.2.1 TECNALIA R&I	4
1.2.2 National ICT Australia, NICTA	5
1.3 Objectives	7
2 State of the Art	9
2.1 Industry 4.0	9
2.1.1 The Diagnosis and Impact Model 4.0	10
2.2 Machine Learning	11
2.2.1 Learning models from data	12
2.2.2 Knowledge representation	13
2.2.3 Feature engineering	14
2.2.4 Supervised learning	14
2.2.5 Semi-supervised learning	16
2.2.6 Unsupervised learning	17
2.2.7 Kernel methods	18
2.2.8 Deep learning	19
2.2.9 Probabilistic methods	21
2.2.10 Ensemble methods	22
2.2.11 Validation and evaluation strategies	23

2.3	Data-driven prognostics	23
2.3.1	Normality modeling	24
2.3.2	Behavior characterization	25
2.3.3	Fault detection and prediction	25
3	Related R&D projects	29
3.1	Railway industry	29
3.2	Wind industry	30
3.3	Maritime sector	30
3.4	Manufacturing sector	31
3.5	Civil structures and materials	31
3.6	Agro-food industry	32
3.7	Relationship of projects and scientific activity	32
4	Main contributions	35
4.1	ML methods for CBM and predictive maintenance	36
4.1.1	A case study on marine diesel engines	36
4.1.2	A case study on marine propulsion systems	45
4.2	ML methods for health status assessment and pattern classification	58
4.2.1	A case study on marine diesel engines	58
4.2.2	A case study on bridges: The Sydney Harbour Bridge	68
4.2.3	A case study on blind fasteners installation	80
4.3	ML methods for quality estimation and production optimization	86
4.3.1	A case study on animal farming	86
5	Conclusions and future work	97
5.1	Conclusions	97
5.1.1	Lessons learned	98
5.2	Future work	98
5.2.1	Deep learning for spatial-temporal modeling	99
	Bibliography	101
II	PUBLICATIONS	111
6	Journal articles	113
6.1	Data-driven prognostics using a combination of constrained K-means clustering, fuzzy modeling and LOF-based score	113
6.2	Deep evolutionary modeling of condition monitoring data in marine propulsion systems	125
6.3	Kernel-based Support Vector Machines for automated health status assessment in monitoring sensor data	149
6.4	A clustering approach for structural health monitoring on bridges	173
6.5	Quantile Regression Forests-based modeling and environmental indicators for decision support in animal farming	191

III	APPENDIX	221
7	Conference papers and other research work	223
7.1	Unsupervised methods for anomalies detection through intelligent monitoring systems	223
7.2	Evolutionary Generation of Fuzzy Knowledge Bases for Diagnosing Monitored Rail-way Systems	232
7.3	A Machine Learning based methodology for automated fault prediction in monitoring sensor data	241
7.4	A multiclassifier approach for drill wear prediction	243
7.5	Kernel density-based pattern classification in blind fasteners installation	258
7.6	Implementation of signal processing methods in a Structural Health Monitoring (SHM) system based on ultrasonic guided waves for defect detection in different materials and structures	271

List of Algorithms

4.1	CBM and predictive maintenance in marine diesel engines. Constrained K-means clustering for outlier detection	38
4.2	CBM and predictive maintenance in marine propulsion systems. Deep evolutionary modeling for anomaly detection	52
4.3	Health status assessment and pattern classification in marine diesel engines. Automatic σ selection in ν -SVM	62
4.4	Health status assessment and pattern classification in marine diesel engines. Kernel-based SVM algorithm for health status estimation	64
4.5	Health status assessment and pattern classification in blind fasteners installation. KDE for behavioral patterns identification	82

List of Figures

2.1	The Data Science big picture.	11
2.2	The Machine Learning process schema.	12
2.3	Illustration of the kernel trick: given a data set that is not linearly separable in the original input space, by applying a kernel function ϕ data is projected into a higher dimension, feature space where it can be divided linearly by a plane.	19
2.4	Global vs local outliers.	22
2.5	The CRISP methodology.	24
3.1	Relationship between the R&D projects and the scientific activity in ML methods and applications carried out in the context of this dissertation.	33
4.1	CBM and predictive maintenance in marine diesel engines. Proposed fuzzy partition.	39
4.2	CBM and predictive maintenance in marine diesel engines. Event score example.	41
4.3	CBM and predictive maintenance in marine diesel engines. Bar chart of resulting clusters distribution.	42
4.4	CBM and predictive maintenance in marine diesel engines. Normal engine behavior.	43
4.5	CBM and predictive maintenance in marine diesel engines. Fuel System fault detected at a normal engine load.	43
4.6	CBM and predictive maintenance in marine diesel engines. Alternator System fault detected at a normal engine load.	44
4.7	CBM and predictive maintenance in marine propulsion systems. An example of evolutionary modeling derivation tree.	47
4.8	CBM and predictive maintenance in marine propulsion systems. Evolutionary physical modeling flowchart.	48
4.9	CBM and predictive maintenance in marine propulsion systems. Illustration of (a) a typical Long Short Term Memory unit and (b) Stacked LSTM-based network architecture used in this study, indicating the number of units (dimensionality of the output space) in each layer.	49
4.10	CBM and predictive maintenance in marine propulsion systems. Influence of the smoothing window on the score values.	51

4.11	CBM and predictive maintenance in marine propulsion systems. Illustration of (a) evolutionary physical modeling predictions and (b) LSTM predictions on engine pinion bearing temperature test and validation data, respectively.	53
4.12	CBM and predictive maintenance in marine propulsion systems. The real sensor readings (above) and the resulting score values on engine pinion bearing temperature (below) over the validation set.	54
4.13	CBM and predictive maintenance in marine propulsion systems. Illustration of (a) evolutionary physical modeling predictions and (b) LSTM predictions on gas turbine thrust bearing temperature test and validation data, respectively.	55
4.14	CBM and predictive maintenance in marine propulsion systems. The real sensor readings (above) and the resulting score values on gas turbine thrust bearing temperature (below) over the validation set.	56
4.15	Health status assessment and pattern classification in marine diesel engines. Illustration of a data set with outliers.	60
4.16	Health status assessment and pattern classification in marine diesel engines. Illustration of the resulting data set after applying the kernel density-based outliers detection process.	60
4.17	Health status assessment and pattern classification in marine diesel engines. Illustration of the ν -SVM decision boundary using $\sigma = 3$	61
4.18	Health status assessment and pattern classification in marine diesel engines. Illustration of the ν -SVM decision boundary using $\sigma = 0.5$	61
4.19	Health status assessment and pattern classification in marine diesel engines. Example of kernel-based SVM health score computed over time.	63
4.20	Health status assessment and pattern classification in marine diesel engines. Kernel-based SVM normality modeling. σ selection based on training-error.	66
4.21	Health status assessment and pattern classification in marine diesel engines. kNN-based normality modeling. k selection based on 10-fold CV error.	66
4.22	Health status assessment and pattern classification in bridges. Flowchart of proposed clustering based approach for damage detection.	69
4.23	Health status assessment and pattern classification in bridges. 6 joints experiment, schematic of the evaluated joints.	72
4.24	Health status assessment and pattern classification in bridges. The Sydney Harbour Bridge schema.	73
4.25	Health status assessment and pattern classification in bridges. Illustration of the 6 joints experiment: centroid and standard deviation of joint events (above) and joints distribution (below). (a) Cluster 0 with events showing a normal behavior and (b) Cluster 1 with events from a damaged joint.	74
4.26	Health status assessment and pattern classification in bridges. Illustration of the 71 joints experiment, analysis of 5 joints located in the second bay of span 7: centroid and standard deviation of joint events (above) and joints distribution (below). (a) Cluster 0 with events showing a normal behavior and (b) Cluster 1 with events from a faulty sensor.	75
4.27	Health status assessment and pattern classification in bridges. 71 joints experiment, Cluster 4: centroid and standard deviation of joint events (above) and joints distribution (below).	76

4.28	Health status assessment and pattern classification in bridges. 6 joints, a known damage in joint 4: map of pairwise distances.	77
4.29	Health status assessment and pattern classification in bridges. 71 joints, span 6: map of pairwise distances.	78
4.30	Health status assessment and pattern classification in bridges. 71 joints, span 7: map of pairwise distances.	79
4.31	Health status assessment and pattern classification in blind fasteners installation. High density regions found in data.	83
4.32	Health status assessment and pattern classification in blind fasteners installation. KDE-based pattern classification approach. Outliers found.	83
4.33	Health status assessment and pattern classification in blind fasteners installation. Patterns found in data by (a) kernel density-based pattern classification approach and (b) K-means ($k=3$).	84
4.34	Quality estimation and production optimization in animal farming. Example of relative humidity model learned from a set of farms.	88
4.35	Quality estimation and production optimization in animal farming. Cumulative deviations in (a) temperature and (b) relative humidity.	91
4.36	Quality estimation and production optimization in animal farming. Random forests-based growth model. Illustration of real values vs. predicted values and corresponding quantile intervals in weeks 3, 5 and 6.	92
4.37	Quality estimation and production optimization in animal farming. Random forests-based welfare model. Illustration of real values vs. predicted values and corresponding quantile intervals in weeks 3, 5 and 6.	93
4.38	Quality estimation and production optimization in animal farming. Random forests-based production model. Illustration of real values vs. predicted values and corresponding quantile intervals in weeks 3, 5 and 6.	94

List of Tables

2.1	Supervised classification schema.	15
2.2	Semi-supervised classification schema.	16
2.3	Unsupervised classification schema.	17
4.1	CBM and predictive maintenance in marine diesel engines. Percentage of variance explained for each number of clusters.	41
4.2	CBM and predictive maintenance on marine diesel engines. Clusters distribution. . .	42
4.3	CBM and predictive maintenance in marine diesel engines. Results obtained per cluster.	44
4.4	CBM and predictive maintenance in marine diesel engines. Global confusion matrix.	45
4.5	CBM and predictive maintenance in marine diesel engines. Global precision, sensitivity, specificity and κ coefficient.	45
4.6	CBM and predictive maintenance in marine propulsion systems. Evolutionary modeling configuration parameters.	48
4.7	CBM and predictive maintenance in marine propulsion systems. LSTM network configuration parameters.	50
4.8	CBM and predictive maintenance in marine propulsion systems. Range of values of reduction gear parameters used in this study regarding scenario S1.	53
4.9	CBM and predictive maintenance in marine propulsion systems. Results of the deep evolutionary modeling for engine pinion bearing temperature training, testing and validation sets with diesel engine in stable operation.	54
4.10	CBM and predictive maintenance in marine propulsion systems. Range of values of reduction gear parameters used in this study regarding scenario S2.	55
4.11	CBM and predictive maintenance in marine propulsion systems. Results of the deep evolutionary modeling for gas turbine thrust bearing temperature training, testing and validation sets with gas turbine in stable operation.	56
4.12	Health status assessment and pattern classification in marine diesel engines. Results of kernel-based SVM classification. Affected systems and fraction of faults detected by each normality model.	67

4.13	Health status assessment and pattern classification in marine diesel engines. Results of kNN regression-based classification. Affected systems and fraction of faults detected by each normality model.	67
4.14	Health status assessment and pattern classification in marine diesel engines. Results of health score classification.	68
4.15	Health status assessment and pattern classification in bridges. Results obtained by kNN outlier removal process.	73
4.16	Health status assessment and pattern classification in blind fasteners installation. Confusion matrix.	84
4.17	Health status assessment and pattern classification in blind fasteners installation. Precision, recall and accuracy.	85
4.18	Quality estimation and production optimization in animal farming. Results obtained by the growth model.	93
4.19	Quality estimation and production optimization in animal farming. Results obtained by the welfare model.	94
4.20	Quality estimation and production optimization in animal farming. Results obtained by the production model.	95
4.21	Quality estimation and production optimization in animal farming. LOOCV average score results of growth, welfare and production models.	95

Acronyms

ADAM: Automation Development for Autonomous Mobility
AI: Artificial Intelligence
ANNs: Artificial Neural Networks
ANOVA: Analysis of variance
BD: Big Data
BLINDFAST: Innovative Blind Fastener Monitoring Technology for Quality Control
CAF: Construcciones y Auxiliar de Ferrocarriles
CBM: Condition Based Monitoring
CBM+: Condition Based Maintenance Plus
CCIA: Ciencias de la Computación e Inteligencia Artificial
CI: Confidence Interval
CNNs: Convolutional Neural Networks
Cobb: Cobb-Vantress
CODOG: Combined Diesel Or Gas
CRISP: Cross Industry Standard Process for Data Mining
CSHM: Clustering methods for Structural Health Monitoring
DBSCAN: Density-Based Spatial Clustering of Applications with Noise
DIPC: Donostia International Physics Center
DS: Data Science
DSSs: Decision Support Systems
DTW: Dynamic Time Warping
EP: Evolutionary Programming
ES: Evolutionary Strategies
ETSII: Escuela Técnica Superior de Ingenieros Industriales
EU: European Union
FFT: Fast Fourier Transform
FOODBASK: Food of the future
GA: Genetic Algorithms
GGGP: Grammar-Guided Genetic Programming
GP: Genetic Programming
HMMs: Hidden Markov Models
ICT: Information and Communication Technologies
IoT: Internet of Things
JPD: Joint Probability Distribution
KD-Trees: K Dimensional-Trees
KDE: Kernel Density Estimation
kNN: k-Nearest Neighbors
LDs: Linear Dynamical Systems
LS: Local Search
MA: Memetic Algorithms
MAE: Mean Absolute Error
MDI 4.0: Diagnosis and Impact Model 4.0
ML: Machine Learning
MLP: Multilayer perceptron

MLRG: Machine Learning Research Group
MS: member states
MSE: Mean Squared Error
NICTA: National ICT Australia
LOOCV: Leave-One-Out Cross-Validation
LSTM: Long Short-Term Memory
OCEAN-LIDER: Ocean Renewable Energy Leaders
RBM: Restricted Boltzmann Machines
RCM: Reliability Centered Maintenance
RIVETEST: Intelligent Riveting Monitoring
RNNs: Recurrent Neural Networks
ROC: Receiver Operating Characteristic
Ross: Aviagen
rpm: revolutions per minute
RUL: Remaining Useful Life
SDC: Signal Difference Coefficient
SHM: Structural Health Monitoring
SIDIM-ADAM: Intelligent Diagnosis System in Railway Data
SOM: Self-Organizing Maps
SPDDMS: Signal Processing methods for Defect Detection in Materials and Structures
SVMs: Support Vector Machines
SVR: Support Vector Regression
UPV/EHU: University of the Basque Country
UPM: Universidad Politécnica de Madrid

Part I

WORK DESCRIPTION

1.1 Motivation

When I entered Fatronik-TECNALIA with the Iñaki Goenaga Research Fellowship in 2006, just after finishing my Computer Science degree at the University of the Basque Country (UPV/EHU), I was really excited. It was a great opportunity to do research on Machine Learning (ML) to solve real, complex problems. At that time, the Data Science (DS) paradigm at industrial level was emerging as one of the most promising technologies and research lines to work in. This was motivated by the increase of data available and by the fact that companies were progressively conscious of the capabilities of ML, addressing their business needs by processing and modeling the information smartly and in an automatic manner. So that was the beginning, in my case. And since then I have been involved in several research projects, both public funded and directly linked to clients, from a wide variety of industrial sectors and domains: railway, wind industry, manufacturing, maritime, agro-food and civil structures. Regarding the publicly funded R&D activity, it can be framed within the 6th, 7th and H2020 Framework Programmes and Clean Sky Joint Technology Initiative from the European Commission, and nationally and locally funded projects, e.g. the CENIT-E, INNPRONTA and RETOS programmes (Spanish Government and CDTI) and the Ertortek programme (Basque Government).

From the academic point of view, I continued to study and learn gaining further qualifications in the field of data analytics. In 2009 I got a Master of Science in Computer Sciences and Artificial Intelligence by the Department of Computer Sciences and Artificial Intelligence (CCIA, Ciencias de la Computación e Inteligencia Artificial in Spanish) at the UPV/EHU. I was passionate about ML and I was enthusiastic about applying state-of-the-art algorithms to challenging problems, creating and sharing value from data. Therefore, I went a step forward and in 2012 I decided to undertake doctoral research in relation with my activity within the Industry and Transport Division at TECNALIA. The main framework was a project in which the Technical University of Madrid (UPM, Universidad Politécnica de Madrid in Spanish) was involved, so I determined to sign up for the doctoral programme of Automatic Control and Robotics, in the Escuela Técnica Superior de Ingenieros Industriales (ETSII) at that same University, and having as supervisors

Professor Ricardo Sanz (UPM) and Professor Basilio Sierra (UPV/EHU). All my research activity was supported by different projects in TECNALIA, always related to the statistical analysis of data and ML-based applications, solving convoluted real problems and addressing specific companies' needs and requirements. In 2014 I had the chance of doing an internship of 6 months as a Visiting PhD Student at NICTA's Machine Learning Research Group (MLRG), leaded by Dr. Fang Chen and supervised by Dr. Yang Wang and Dr. Nguyen Lu Dang Khoa, in Sydney, Australia. There, I was involved in the Structural Health Monitoring (SHM) project and I did some research regarding non-parametric methods for anomaly detection. At that time, the NICTA's MLRG was ranked among the top five of its kind in the world.

Since then I have continued to work in the application of ML methods, in even more challenging scenarios and involving increasingly more data from disparate sources of information.

And this is in brief the chronology and context of my research on ML methods and applications. In the following chapters and sections of this dissertation the main areas, projects and contributions are presented and discussed. This PhD is the result of the research work performed in the context of R&D projects developed in TECNALIA and NICTA, in collaboration with the CCIA Department (UPV/EHU), the Autonomous Systems Laboratory (UPM) and the NICTA's MLRG.

1.2 Context

1.2.1 TECNALIA R&I

The research activity described in this dissertation has been mainly carried out within the applied research perspective given by R&D projects developed in TECNALIA. TECNALIA Research & Innovation¹ is a private, independent, non profit applied research centre of international excellence. Legally a Foundation, Tecnalia is the leading private and independent research and technology organisation in Spain and one of the largest in Europe, employing 1,336 people (198 PhDs) and with income of 102 Million € in 2013.

The main goal is to transform knowledge into GDP, meaning wealth to improve peoples quality of life by generating business opportunities for industry. TECNALIA is committed to generate major impacts in economic terms, by means of innovation and technological development, addressed by 7 business divisions, covering economic sectors of Energy, Industry, Transportation, Construction, Health and ICT. TECNALIA has been granted over 250 patents and promoted more than 30 spin-off companies, which highlights the knowledge-to-application-oriented activity in which I have developed my research activity.

TECNALIA is a key agent in the ERA (European Research Area), holding position 12th among RECs and 26th overall in ECs 6th FP7 Monitoring Report 2012. It also actively participates in the governing bodies of several European Technology Platforms and partners in 377 FP7 projects, coordinating 81 of them; in H2020 it participates in 37 projects, coordinating 6 of them, up to the end of 2014. TECNALIA is a member of EARTO and of EUROTECH, linking together the most important research centres in Europe. This research network represents an excellent opportunity to learn from scientific experts of different nationalities, which allows to enrich the research performed in the context of this PhD.

The EHU/UPV, Tecnalia and Donostia International Physics Center (DIPC) are the founding

¹www.tecnalia.com

members of Euskampus², a project with attained the qualification of International Campus of Excellence from the Spanish Ministry of Education (2010). Euskampus partners with the PRES of the Universities of Bordeaux in the Euskampus-Bordeaux Transborder Campus.

Industry and Transport Division

The research described in this dissertation has been carried out within the Industry and Transport Division, which responds to the needs of sustainable mobility and efficient manufacturing in a globalised environment. Having a wide knowledge of the transport, machine-tool, foundry and steel-works industrial sectors it allows the strategic positioning of the research activity in the production chain, providing comprehensive solutions to the industry through R&D and applied innovation.

Given the fact that the R&D investment towards a smarter and more sustainable business is beneficial, the first model capable of diagnosing and discovering the Industry 4.0 level has been designed, filtering, prioritising and determining the R&D projects that are really worth investing in because they will have greater impact on the business.

The Diagnosis and Impact Model 4.0 (MDI 4.0 in Spanish) 'powered by TECNALIA' has been designed to have a vision of all those aspects and characteristics that will influence the fourth industrial revolution: Industry 4.0. The main benefit for the company consists of identifying the priority improvement opportunities on the way to becoming an Industry 4.0 company. This is one of the main motivations and core elements for the research activity carried out within this PhD.

1.2.2 National ICT Australia, NICTA

The scientific insights obtained in the context of this dissertation have been possible as a result of a close collaboration with the NICTA's MLRG. NICTA (National ICT Australia³) is Australia's Information Communications Technology (ICT) Research Centre of Excellence and the nations largest organisation dedicated to ICT research. NICTA's primary goal is to pursue high-impact research excellence and, through application of this research, to create national benefit and wealth for Australia. Their aim is to be one of the worlds top ICT R&D centres.

NICTA's research addresses the technology challenges facing industry, the community and the whole nation. They seek to improve the international competitiveness of both academic ICT research and industry innovation by tightly linking the two to achieve greater economic and social impact. NICTA provides decision support for owners and maintainers of civil and industrial assets. Sensing, continuous monitoring and advanced data analysis techniques enable asset managers to make more informed maintenance decisions. NICTA has developed technology to enable more informed maintenance decision making. There are three main technology components:

- Sensing and Data acquisition: sensors and distributed processing capabilities to suit large and small structures.
- Data Analytics: analytical techniques developed by NICTA's world-class MLRG provide information for specific situations such as damage detection, condition assessment, loading assessment and maintenance prioritisation. Data sources can be from NICTA or other sensing systems, and other sources of data such as environmental data, inspection and maintenance records.

²2010, <http://euskampus.ehu.es/en/>

³NICTA formally merged with CSIRO to form a new entity called Data61 (<https://www.data61.csiro.au/>) on 2015

- A continuous monitoring service: the service applies data management and the analytical techniques to provide asset managers and engineers with situational awareness and the information they need to make decisions. The service is hosted from NICTA data centres and available to users via web and mobile applications and database services.

All these elements play a key role in the research performed within this PhD dissertation.

NICTA's Machine Learning Research Group

ML is becoming a pervasive and disruptive technology. Its algorithms are making their ways in our devices, appliances, transports, and changing the way we invest, buy, search, drive, record, write, etc.

In order to understand the world we always work with models. The essence of Science is to construct models or theories. Simple models can be understood as mathematical equations: $Force = Mass \times Acceleration$. This allows the inference of one quantity from another. But many things one would like to understand are intrinsically complex, and there is no hope of such simple formulas or they may not be adequately understood. The technology of machine learning allows one nevertheless to build models (in a computer) that can be used for accurate prediction and reliable decision making. The way this works is conceptually the same: the computer needs to infer a mathematical relationship between the inputs (say the pixels or points of colour in a photograph) and the outputs, a categorisation of what those pixels represent. The difference with simple models is both of scale, the amounts of data, and of complexity, the structure of data and the facets of the model.

In the MLRG, the research activity ranges from core theory to wide ranges applications, with connections to numerous fields outside ML. The directions of research performed pertain to the composability and servicisation of ML, making the field secure, transparent and efficient, and improving its scalability. Even when a strong emphasis is put on text and spatio-temporal data, all kinds of data are susceptible to be analyzed.

The focus is on important and challenging problems such as:

- Detecting and monitoring in real-time incidents using social media
- Predicting failures of widespread infrastructure
- Making machine learning transparent
- Understanding the processes underpinning ecosystem diversity
- Learning from private data
- Predicting the output of rooftop solar photovoltaic systems
- Building predictive tools for the EPA Air Quality Prediction

To do so, new technologies are developed to solve these problems and make them freely available or commercially deploy them. And this is the knowledge that has supported the research activity presented in this dissertation, from a more algorithmic and methodological point of view. But focusing on the end application and on the usability of the proposed approaches and solutions, to be discussed in further chapters and sections of this document.

The deep knowledge of the NICTA's MLRG on ML paradigms has provided the scientific foundations for the more technological work developed in TECNALIA.

1.3 Objectives

This PhD dissertation is focused on the application of ML and data analysis strategies and methods to real, new monitoring data coming from complex industrial assets and systems. Special focus is on marine diesel engines, bridges, animal farming and manufacturing processes.

The main goals of this research are summarized as follows:

1. To automatically model normality and characterize behaviors from monitoring data
2. To infer and classify new patterns and knowledge of interest regarding assets' operational behavior
3. To early detect potential deviations from normality and normal operation, e.g. failures, degradations, wear and malfunctions
4. To estimate the health status of the assets and its temporal evolution
5. To provide intelligent monitoring strategies to assure the optimal operation of the assets and to improve their reliability
6. To optimise production and product quality in a sustainable way

All these objectives are included in the novel MDI 4.0 concept, which is the first step towards the smart company and smart manufacturing. The work described in this dissertation indicates how ML science and technologies can strongly support the development of the MDI 4.0, providing a high potential context for scientific research.

2.1 Industry 4.0

The Industry 4.0 is a global modernization movement of the manufacturing industry towards the adaptation of the recent advances in Information and Communication Technologies (ICT). New communication systems and protocols, cyber security standards, multi-device displays, mobile and compact communication devices with computational capabilities, Artificial Intelligence algorithms, remote and distributed applications and information technologies resources have become a reality during the last decade. Internet has grown rapidly and exponentially, and nowadays it is used in any economical and social aspect of our lives. The industrial manufacturing sector is clearly concerned by this situation. What is more, the adoption of all these technologies by the companies is leading to a new industrial revolution and a paradigm shift. It must be noted that all mentioned technologies are essentially digital, defining cybernetic and virtual environments, and industry works in the physical world. The merging of the physical and digital worlds is in fact the core of this new revolution, establishing the basis for the smart factories of the future.

This paradigm shift has been defined as the fourth industrial revolution or Industry 4.0 (Industrie 4.0, in German [1] [2] [3]), on the basis of the full deployment of the new ICT paradigms in the production processes from the product design phase, then the manufacturing and related logistics and, finally, to the product life-cycle management. Similarly, in the United States there exists what is known as Industrial Internet, merging the physical manufacturing systems and the recent advances in the ICT domain.

Production systems are typically fixed, hierarchical processes that implies important changes and costs when adapting the production and resulting products to market requirements. Current market needs require more flexible solutions, with a customization of the production while assuring the profitability of smaller runs of products. Furthermore, and in order to continue monetizing the product once it has been produced, the after-sales services are mainly focused on maintenance.

Based on the definition of the concept Industry 4.0 previously introduced, and from a production perspective, three main levels of implementation, application and integration of technologies can be found. They are the following:

- **Vertical integration:** in the context of production and automation, it is referred to the integration of diverse ICT systems into different hierarchical levels, from the very basic levels, e.g. sensors and actuators, to the highest levels that are related to the production management, execution, planning and scheduling. This level of integration highly supports the manufacturing processes, making them more flexible.
- **Horizontal integration:** this level implies the integration of ICT technologies among the mechanisms and agents involved in the different stages of the manufacturing processes and business planning, which means exchanging energy and information within the company (e.g. input and output logistics, production and commercialization), and between distinct companies and entities (value networks).
- **Circular integration:** it tries to unify both vertical and horizontal integrations to link the end user and the product life cycle. This integration ends the production loop and, therefore, a whole end-to-end digitalization is fully achieved, from the initial design stages, the planning and manufacturing, the logistics and resources management mechanisms and, finally, even reaching to the level of the end user and the product related services.

All of the above mentioned concepts are being increasingly adopted by the strategic plans of entities and companies around Europe, America and Asia, focusing on specific Industry 4.0 elements [4]. In the United Kingdom, for instance, a technological demonstrator named Digital Factory or Industry 4.0 Demonstrator has been created. Conceptually, it is a living laboratory in which other companies of the sector can learn and experiment the potential of the technology [5]. The demonstrator consists of a real production line connected to a 3D virtual factory, which is designed to demonstrate the capabilities of the customization and personalization. Spain is also incorporating the Industry 4.0 concept in the science, technology and innovation plan. One of the priorities of this plan is the advanced manufacturing for the Horizon 2020, as it is specified in the Basque Industry 4.0 strategy [6].

2.1.1 The Diagnosis and Impact Model 4.0

The Diagnosis and Impact Model 4.0 (MDI 4.0 in Spanish) tries to address all the aspects and characteristics that will influence the fourth industrial revolution, or Industry 4.0. The MDI concept turns to be a broad range of technological solutions to assess the health status of the monitoring processes and assets, to optimize production and to mitigate the possible impact and risks arising from faulty conditions or unexpected breakdowns. In order to achieve this, monitoring data is the key, and the main goal is to know exactly what to do with them. This is mainly due to the apparition of the Big Data (BD) paradigm, which is driven by the Internet of Things (IoT) explosion. The IoT consists of an increased amount of smart devices connected and sending huge volumes of data [7].

Having a set of data related to the condition of the process and the involved assets, and by means of mathematical algorithms and advanced statistical analysis, new, previously unseen, relevant knowledge can be extracted and modeled. Prediction models can be thus obtained to anticipate events that lead to a negative impact in the business. This is the main objective and motivation of this research. And in this context, ML paradigms are the tool to successfully analyse and model monitoring data.

2.2 Machine Learning

When talking about data analytics it is difficult to clearly identify the boundaries between ML, statistics, Data Mining (DM) and Artificial Intelligence (AI). A new concept is becoming popular to roughly cover all these disciplines: DS. The data scientist is thus supposed to have some good notions of all these paradigms, and, additionally, to apply BD strategies and techniques to efficiently and smartly manage big amounts of data. A global schema of the main concepts related to data science and their interrelationships is shown in Figure 2.1.

Figure 2.1: The Data Science big picture.

The main concepts presented in the diagram are introduced as follows:

- **Big Data:** it is a concept that encompasses high-volume, high-velocity and high-variety information and the technologies and techniques needed to collect, store, manage and analyse such amount of data [8]. The focus is on data and how to optimally handle it, carefully considering the 5 V's: Volume, Velocity, Variety, Veracity and Value. Regarding assets management, the goal would be more oriented to provide cost-effective, innovative forms of information processing for enhanced insight and decision-making from monitoring data, specially when dealing with the analysis of high-frequency data streams in an online fashion. The BD concept implies a true revolution and a promising business opportunity for companies.
- **Data Mining:** it refers to the science of collecting and smartly handling all the historical data coming from many disparate sources, and then searching for patterns and trends [9]. The main goal is to transform data in useful information or insights. This knowledge mainly consists on consistent patterns and relationships between variables. It is usually applied in the business intelligence domain and it mainly involves tasks related to statistical analysis of data, summarization, classification, association and data preparation.
- **Machine Learning:** it is the paradigm of creating algorithms and programs which learn on their own [10]. Once designed, they do not need a human to perform better and more accurately. Some of the most well known applications of ML include the following: Web search, spam filters, recommender systems, ad placement, credit scoring, fraud detection,

stock trading, computer vision and drug design. It is humanly impossible to create models for every possible search or spam, so the idea is to make the machine intelligent enough to learn by itself. When the later part of the DM process is automated, it is known as ML.

- **Artificial Intelligence:** it is the study of intelligent agents or machines, able to replicate cognitive functions, e.g. natural language processing, artificial vision, reasoning, learning or perception [11]. It is usually applied in robotics, character recognition and self-driving cars, emulating typical complex human behaviors and tasks. It makes use of all the terms showed above, from mining of data coming from sensors and devices used as source of information to modeling the extracted knowledge for its use in a learning framework.
- **Data Science:** it refers to a broad umbrella term that aims to extract knowledge and insights from data, from a scientific and creative point of view [12]. This includes scientific and mathematical methods, statistics and any other tool able to tackle BD. The focus is on data itself and its intelligent analysis for a large variety of tasks. Similar to data science, data analytics is the process of combining and defining analytical methods over data but in a more focused way, often with a specific goal already in mind.

All these paradigms are undoubtedly interconnected. Nevertheless, if the final goal is to learn a model from data in order to be further used as an expert system, a recommender or an event detection and classification tool, for instance, ML becomes crucial.

By means of mathematical algorithms, knowledge can be automatically represented and modelled for a wide variety of purposes: prediction, classification, anomaly detection, etc. This task is specially challenging nowadays, since it involves processing and analyzing huge amounts of data and additional information coming from different, diverse, monitoring systems and smart devices. A graphical representation of the ML process is presented in Figure 2.2. The core of the process has to do with fitting data to a model, given a hypothesis and a performance criteria.

Figure 2.2: The Machine Learning process schema.

2.2.1 Learning models from data

Many science areas need to find a model that fits a data set. Model to be inferred is usually a mathematical expression with a set of coefficients (parameters) that have to be determined. In the best scenario the model is known and the problem can be reduced to find this set of coefficients that fit the data in an optimal way.

Curve fitting is considered a mathematical problem in which a curve fits a set of points. This process is carried out by applying mathematical methods, such as least squares, or alternatively it can be reduced to a classical interpolation problem when the function has to perfectly fit the points.

When analysing big amounts of heterogeneous data coming from disparate sources it is crucial to structure the information in order to succeed in the knowledge extraction process. Learning process is otherwise rough and imprecise, likely to yield irrelevant and misleading conclusions. Data Science community is aware of this important challenge, adopting automatic data driven modeling as a key added value to efficiently perform learning process from large data streams [13]. The role of ML in artificial systems tries to emulate the human-like learning process, carefully considering the following key aspects when learning models from data that represent behaviors of interest:

- what model to learn,
- how to learn and
- when to learn the model.

Adaptive behavior modeling aims to integrate these principles in the models that represent the knowledge, learnt from data, and that will be used to solve a specific task, e.g. failure mode classification, optimization, energy consumption estimation or pattern extraction. Knowledge models such as Artificial Neural Networks (ANNs) [14] can adapt its structure (add new resources), or update the network parameters as soon as new samples that define new behaviors of interest with relation to the problem and to the task to be solved arise, and thus improving the generalization and adaptability capabilities of the model. It is important to establish the correct mechanisms that trigger the learning process according to the data available and their relevance.

There is a wide variety of methods and algorithms to train and learn a model from data. Depending on the nature of the data and on the problem to be solved, different strategies are adopted.

2.2.2 Knowledge representation

In general terms, the modeling process inputs are the instances that contains a set of m features, $X = \{X_1, \dots, X_m\}$ where each feature X_i can take a value from its own set of possible values χ_i , and n feature vectors or instances, $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$. More precisely, the model to be learned will process this information to produce a knowledge representation of the data set. Depending on the method to be applied this output can be obtained in the form of groups of similar data (clusters that represent behaviors), trees, tables, classification and association rules or functions.

Expert's knowledge must be captured and modelled to be further exploited in a generic and automatic fashion. The most commonly used strategy when modeling knowledge deals with the definition of expert rules. When possible, precise rules can be easily defined with the help of experts in the domain, but this is not the usual case. Automatic data-driven rules extraction approaches are then usually applied, aiming to characterize behaviors of interest as a combination of parameters contained in the data set whose values satisfy the apparition of such behaviors (a given condition).

Rules-based diagnosis systems are a very common and reliable knowledge representation when detecting failures and anomalies, associated to abnormal system working conditions. Thus, there

exist different methods to extract rules from data. The most popular algorithms when modeling knowledge as a set of rules are the following:

- **Decision trees:** nodes in a decision tree involve testing a particular attribute. It typically compares an attribute value with a constant. Leaf nodes give a classification that applies to all instances that reach the leaf. Top-down induction of decision trees is probably the most extensively used research method in DM [15] [16].
- **Classification rules:** a popular alternative to decision trees where the antecedent of a rule, or precondition, is a series of tests just like the tests at nodes in decision trees. The consequent, or conclusions, gives the class or classes that apply to instances covered by that rule [17].
- **Sequential patterns:** sequential pattern mining aims to extract sets of items commonly associated over time [18], as an extension of the concept of frequent itemset by handling timestamps associated to items [19]. Contextual information can be also considered to enrich patterns found and, additionally, sequential patterns can also be directly transformed into association rules, with some support and confidence associated to them [20].
- **Fuzzy models:** models based on Fuzzy Logic, which is a truth-functional system for reasoning with logical expressions describing memberships to fuzzy sets. In fuzzy control, mapping between real-valued input and output parameters is represented by fuzzy rules [21].

A rule-based expert system also requires a rule engine to be applied in order to check existing rules, previously defined by any of the above strategies. Basic conditions can be easily processed, but in the case of fuzzy rules, a more complex fuzzy inference procedure must be accomplished by firstly fuzzifying parameter values, then calculating the membership function to each fuzzy domain for every parameter involved, and eventually obtaining the resulting crisp or belief value.

2.2.3 Feature engineering

Feature engineering is an important process when learning a model from data [22]. Feature extractors, problem descriptors and key performance indicators are created to reduce the complexity of the raw data, making patterns related to domain knowledge more visible to algorithms to be applied. Therefore, resulting models are more meaningful and accurate. Nevertheless, this process is difficult and very time consuming because it usually requires the support of domain experts to correctly define the features of interest based on their expertise.

Automatic selection of features is also applied when the support of domain experts cannot be put in practise. A ranking of most relevant features can be thus obtained given their importance when solving a problem, e.g. variance explanation or impurity decrease given a target feature and a set of input features. Some ML algorithms are also able to deal with feature transformation and extraction in an automatic manner, as part of their learning framework. That is the case of deep learning and kernel methods, for instance. They perform data transformations and learn high-level features to operate in a high-dimensional, implicit feature space.

2.2.4 Supervised learning

In supervised learning the data set consists of a set of m features, $X = \{X_1, \dots, X_m\}$ where each feature X_i can take a value from its own set of possible values χ_i , and n feature vectors or instances,

$\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, and a target feature, $\mathbf{y} = (y_1, \dots, y_n)$, containing a value for each feature vector [23]. The target feature can be continuous or discrete. A label can be also estimated by tracking events of interest or based on maintenance operations carried out in the past. From such kind of event on, data are supposed to correspond to normality. Similarly, when a corrective action is performed, preceding data can potentially represent abnormal behaviors.

The data matrix schema of a typical supervised learning problem is presented in Table 2.1.

Table 2.1: Supervised classification schema.

	X_1	X_2	$\dots$	X_i	$\dots$	X_m	$\mathbf{y}$
$\mathbf{x}_1$	x_1^1	x_2^1	$\dots$	x_i^1	$\dots$	x_m^1	y_1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{x}_i$	x_1^i	x_2^i	$\dots$	x_i^i	$\dots$	x_m^i	y_i
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{x}_n$	x_1^n	x_2^n	$\dots$	x_i^n	$\dots$	x_m^n	y_n

In terms of data-driven prognostics, and in order to anticipate an event of interest in time, the problem can be addressed in different ways depending on the nature of the target feature to predict. When dealing with discrete features, it is interesting to predict or classify symptoms that can lead to the event under study. The frequency analysis of the apparition of symptoms also can provide a good indicator of system wear or degradation.

In the case a continuous target feature is available representing, for instance, the normal status of the system given other input conditions, the problem can be addressed by learning a regressor that best fits a set of data clearly representing a specific behavior to be modelled (e.g. normality, optimal conditions or faults). Some remarkable examples of regression methods are:

- **Symbolic and parametric regression:** by defining a simple mathematical grammar the creation of a great number of physical models can be accomplished [24]. The technique allows the searching of both the coefficients that best fit the data and the model itself. It is considered as a parametric approach, since the goal is to estimate parameters whose structure is assumed or known a priori. By means of evolutionary computation, and especially Grammar-Guided Genetic Programming (GGGP), it is possible to define the model searching space and the mathematical expressions to be used. The search problem, inherent in symbolic regression, can be faced avoiding the existing constraints in other approaches. Efficient search space exploration skills together with local optima avoidance are two main virtues of this approach.
- **Non-Symbolic regression:** non-symbolic models refer to models that are not directly human-understandable. Instead of using human-understandable symbols, they use other knowledge formats such as weights, connections, etc. The best known data-driven model generation methods based on non-symbolic approach are ANNs [25]. ANNs are inspired by human brain cognitive ability, interconnecting and grouping neurons (process units) in layers. Input layer receives input data while output layer provides output data.
- **Non-parametric regression:** non-parametric methods are statistical techniques that allow learning a model from data without making any assumption of normality distribution. There-

fore, there is no dependence on the distribution of the data under analysis and, consequently, expert knowledge is not required and it is not necessary to build in features of anomalous behavior [14]. The key advantage of non-parametric approaches is that they are able to identify highly non-linear dependencies in data. However, one of the main practical problems of non-parametric regression estimation is the curse of dimensionality, or the computational complexity of making predictions as the training data set grows. Regarding anomaly detection and fault diagnosis, two techniques have been widely used in several application domains and problems: k-Nearest Neighbors (kNN) [26] and Support Vector Regression (SVR) [27].

Similarly, when dealing with discrete or categorical target features, kernel methods and Support Vector Machines (SVMs) or ANNs are also applied for classification due to their high approximation capability based on non-linear relationships among the input features. kNN, as the core of many other approaches, has been also largely used for classification, predicting the label of a new feature vector as the majority of labels or values among its k closest neighbors.

2.2.5 Semi-supervised learning

Semi-supervised learning is a kind of learning framework in which the training data combines labelled and unlabelled instances [28]. Given a data set containing a set of m features, $X = \{X_1, \dots, X_m\}$ where each feature X_i can take a value from its own set of possible values χ_i , and n feature vectors or instances, $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, the target feature can be in the form $\mathbf{y} = ((y_i, y_j, y_k), \dots, ?, \dots, (y_j \neq y_{j+1}), \dots, ?)$. This ML field is also known as weak supervision. The instance-label relationship determines the problem to be addressed having, for instance, multi-label frameworks in which the instances are categorized with one or more labels, i.e. several symptoms arising at the same time instant (e.g. $y_1 = (y_i, y_j, y_k) \in \mathbf{y}$), instances with unknown categories (e.g. $y_i = ? \in \mathbf{y}$), or groups of instances that are known to belong to different categories, i.e. several unknown symptoms that occur at different time instants (e.g. $y_j = (y_j \neq y_{j+1}) \in \mathbf{y}$).

In the following table (see Table 2.2), the data matrix schema presenting the most typical semi-supervised learning problems is shown.

Table 2.2: Semi-supervised classification schema.

	X_1	X_2	$\dots$	X_i	$\dots$	X_m	$\mathbf{y}$
$\mathbf{x}_1$	x_1^1	x_2^1	$\dots$	x_i^1	$\dots$	x_m^1	(y_i, y_j, y_k)
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{x}_i$	x_1^i	x_2^i	$\dots$	x_i^i	$\dots$	x_m^i	$?$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{x}_j$	x_1^j	x_2^j	$\dots$	x_i^j	$\dots$	x_m^j	$(y_j \neq y_{j+1})$
$\mathbf{x}_{j+1}$	x_1^{j+1}	x_2^{j+1}	$\dots$	x_i^{j+1}	$\dots$	x_m^{j+1}	$(y_{j+1} \neq y_j)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{x}_n$	x_1^n	x_2^n	$\dots$	x_i^n	$\dots$	x_m^n	$?$

In this kind of scenarios, the available labels are usually related to critical events that occurred in the past. The strategy most commonly used when dealing with time series data consists of

selecting a set of data before and after the registered label (e.g. in a time window of one or several months). Then, each set of data can be contextualized on the basis of the corresponding event. For instance, if a maintenance operation or an overhaul occurred at a given time instant, the set of data selected after that event in time can be considered to categorize normality. Whereas the set of data selected before such event must be further analysed in order to try to infer and model the trend that lead to the event of interest. One-class SVM, or ν -SVM, for example, allows controlling the false positive rate given by ν and therefore it can be used to model normality on the basis of a small percentage of anomalies that are assumed to be present in data [29].

2.2.6 Unsupervised learning

In unsupervised learning the data set does not contain a target feature to be predicted [30]. Having a set of m features, $X = \{X_1, \dots, X_m\}$ where each feature X_i can take a value from its own set of possible values χ_i , and n feature vectors or instances, $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, the most typically used approach is to group instances by their similarity given a distance metric. Unfortunately, the scenarios involving data-driven prognostics problems do not always provide a proper track of past abnormal behaviors or maintenance operations performed to prevent or correct a faulty condition. Therefore, they must be addressed from an unsupervised perspective.

Table 2.3 shows the data matrix schema of a typical unsupervised learning problem.

Table 2.3: Unsupervised classification schema.

	X_1	X_2	$\dots$	X_i	$\dots$	X_m
$\mathbf{x}_1$	x_1^1	x_2^1	$\dots$	x_i^1	$\dots$	x_m^1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{x}_i$	x_1^i	x_2^i	$\dots$	x_i^i	$\dots$	x_m^i
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{x}_n$	x_1^n	x_2^n	$\dots$	x_i^n	$\dots$	x_m^n

When dealing with unlabelled data, the most direct way of obtaining a first look at the set of features has to do with simple statistical approaches. Whenever a normal distribution is present, the mean, μ , and standard deviation, σ , can be computed to detect outliers and extreme values. The most extended rule says that any value greater or lower than $\mu \pm 2\sigma$ is likely to be abnormal, corresponding to the 0.5% of the data. However, this value must be carefully selected and adapted to the problem under study.

Aiming to approximate behaviors or groups of similar values in data, clustering methods are usually applied [31]. They commonly use density-based algorithms (e.g. density reachability and density connectivity), mathematical models (e.g. a mixture of underlying probability distributions), or proximity and distance metrics in order to estimate the similarity between feature vectors, $\mathbf{x}_i$ and $\mathbf{x}_j$. Different distance metrics can be used given a data set, typically on the basis of the statistical dependence among features and their joint distribution, e.g. Euclidean, cosine, correlation coefficient or Kullback-Leibler [32].

From groups formed, behavioral patterns can be inferred, e.g. probability density functions and distributions, cluster representatives, centroids or average values. Some remarkable examples of this

genre of methods are the following:

- **K-means:** it performs a partition of the data space into K clusters or groups of similar instances in an orderly and non-linear manner [33]. Groups are formed automatically and strictly, classifying new instances based on their similarity to the cluster representatives or centroids, which are the average of all feature values of the instances belonging to a cluster. During the learning process feature vectors are iteratively assigned to the cluster whose centroid has the smallest distance, and centroids are recalculated accordingly. The learning process converges to a solution with a linear complexity, $O(n)$. It shows a good performance and robustness in presence of slight variations in data, such as peaks or outliers, in comparison to other similar methods.
- **kNN and extensions:** it learns groups of similar data based on their proximity, given a distance metric, and density, given a radius of k closest neighbors [34]. Neighbors-based methods are known as non-generalizing ML methods, and due to its good performance it is the foundation of many other learning methods, e.g. spectral clustering, in which clustering is applied to a projection to the normalized laplacian, based on a kernel function and on a distance metric used to compute the kNN connectivity matrix or affinity matrix. Therefore, it can be also seen as a kernel K-means.
- **Self-Organizing Maps (SOM):** also known as Kohonen neural networks, it allows representing into a low-dimensional map a high-dimensional data set [35]. A SOM map is composed by neurons grouped according to a topology (e.g. hexagonal or rectangular). Each neuron has associated a weighted vector that allows mapping input data into each neuron on the basis of a given measure. During the learning process, the feature vectors are presented to the network iteratively, in such a way that for each feature vector the winning neuron, or the neuron that has the weighted vector most similar to the feature vector, modifies its associated vector to increase its similarity with it. The vector associated to the winning neuron and the neighboring neuron vectors according to the topology used, are modified by means of a decreasing function of the distance between nodes on the map grid. This method provides a non-linear, ordered, smooth classification of high-dimensional input data, preserving neighborhood relations.
- **Density-Based Spatial Clustering of Applications with Noise (DBSCAN):** it automatically finds high density, core feature vectors and expands clusters from them [36]. It is mainly suitable for data sets in which groups of feature vectors of similar density can be formed, also properly managing noise and outliers present in data. It requires defining two free parameters: ϵ , which controls the maximum distance between two feature vectors for them to be considered as in the same neighborhood, and the minimum number of points required to form a dense region.

This algorithms learn groups or patterns incrementally, until a convergence criteria is met. Some of them requires computing an affinity matrix or to carefully choose the free parameters that configure the algorithm and that can dramatically affect the model to be learned.

2.2.7 Kernel methods

Real world scenarios involving complex monitoring industrial systems usually requires methods able to detect nonlinear relationships and dependencies among features. A positive definite kernel

basically consists of a dot product in a high-dimensional feature space, depending on the complexity of the problem to be solved [37]. This operation, also known as kernel trick, is often computationally cheaper than the explicit computation of the coordinates. Therefore, it allows making linear estimations transparently through a formulation based on kernel evaluations, without the need of explicitly computing the high-dimensional feature space. The main advantage is that more complex functions can be implicitly approximated, since the similarity measure given by a kernel K allows constructing algorithms in dot product spaces. An illustration of the kernel trick is presented in Figure 2.3.

Figure 2.3: Illustration of the kernel trick: given a data set that is not linearly separable in the original input space, by applying a kernel function ϕ data is projected into a higher dimension, feature space where it can be divided linearly by a plane.

Following the same formulation as in previous cases, given a set of m features, $X = \{X_1, \dots, X_m\}$ where each feature X_i can take a value from its own set of possible values χ_i , and n feature vectors or instances, $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, a kernel is a function $K : \chi \times \chi \rightarrow \mathbb{R}$, $(\mathbf{x}_i, \mathbf{x}_j) \mapsto K(\mathbf{x}_i, \mathbf{x}_j)$. It satisfies for all $\mathbf{x}_i, \mathbf{x}_j \in \chi$: $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$; where ϕ is the feature map of kernel K because it maps into some dot product space $\mathcal{H}$.

There exist several types of kernel functions, e.g. linear, polynomial, sigmoid, convolution, analysis of variance (ANOVA), splines, Fisher, graph, tree or cosine, but the most widely used is the radial basis function, also known as Gaussian kernel. The Gaussian kernel on feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$.

The most representative kernel methods are the SVMs-based algorithms, for classification (C-SVM and ν -SVM) and regression (ϵ -SVM and ν -SVM) [38] [39]. They mainly perform classification and regression tasks by constructing hyperplanes in a multidimensional space that separates feature vectors of different class labels or continuous target feature values. The main advantage of these methods is their computational efficiency and versatility. But on the contrary they perform poorly when the number of features in the projected space exceeds the number of training data samples.

2.2.8 Deep learning

Motivated by the recent significant increase in the computational capacity of processing machines and the amount of data available nowadays, deep learning offers a flexible and powerful approach to solve complex problems. Deep learning methods can be seen as a cascade of many layers of processing units that combine the predictor features to approximate the target feature, in a similar

way to which it is done in ANNs [40]. Therefore, deep learning can be seen as a particular kind of ML paradigm. Nowadays deep learning algorithms are very popular because they are able to achieve remarkable results for complex problems spanning natural language processing, image captioning, handwriting recognition and genomic analysis.

From the algorithmic point of view, deep learning models are similar to those obtained by regular ANNs. The main difference between them lies in the complexity of the network architecture. In regular ANNs each neuron is fully connected to all neurons in adjacent layers and few layers are usually employed, whereas in deep learning models more layers are used, hierarchically learning the features that best fit the data and internally performing abstract representations and transformations. Therefore, feature engineering step is automatically achieved by the algorithm itself and very complex problems can be successfully addressed, with the drawback of being rather computationally expensive. Some remarkable examples of deep learning-based methods are the following:

- **Convolutional Neural Networks (CNNs):** in CNNs neurons are arranged in 3 dimensions (width, height and depth, which corresponds to the third dimension of an activation volume) [41]. This key aspect allows connecting neurons in a layer to a small region of interest of the previous layer, instead of fully connecting all the neurons. The convolutional layer is the core building block of CNNs, which mainly consists of a set of learnable filters (or kernels) that compute dot products between the input at any position of the data matrix and the entries of the filter, usually extended through the full depth of the input volume. Each filter at each convolutional layer will thus produce a separate 2-dimensional activation map that gives the response of that filter at every spatial position, in order to detect patterns or events of interest. Other types of layers can be also used when defining the architecture of the CNNs, i.e. fully connected layers and pooling layers, to progressively reduce the spatial size of the representation and thus reducing the amount of parameters in the network, and hence controlling overfitting.
- **Deep-belief networks:** in deep-belief networks the building blocks that constitute the network architecture are called Restricted Boltzmann Machines (RBM) [42]. They are two-layer neural nets, one being the input, or visible layer, and the second being the hidden layer. The restriction relates to the lack of communication intra-layer. Each node stochastically estimates whether to transmit its input or not. The outputs of a hidden layer are passed as inputs to the next RBM node (multiplied by their corresponding weights, summed and added to a bias) and so on, until a final classifying layer is reached. In addition, there is a reconstruction phase in which the activations of a hidden layer become the input in a backward pass (again multiplied by their corresponding weights, summed and added to a bias). The sum of those products is added to the visible layer. The outputs of these operations are approximations of the original input. The error between the reconstructions and the original input is iteratively minimized by backpropagation against the node's weights, which is known as generative learning. Therefore, the RBM learns to approximate the training data by making guesses about the probability distribution of the original input.
- **Recurrent Neural Networks (RNNs):** they are networks with loops in them, allowing information to persist [43]. The idea is to let every step of a RNN pick information to look at from some larger collection of information. Therefore, they can be thought of as multiple copies of the same network, each passing a message to a successor. This chain-like nature reveals that RNNs are intimately related to sequences and lists. They are powerful and increasingly

popular models for learning from varying-length sequence data, particularly those using Long Short-Term Memory (LSTM) hidden units [44]. The key to LSTMs are the memory cells, whose states are carefully regulated by structures called gates. They are composed out of a sigmoid neural net layer, which outputs numbers between zero and one to describe how much of each component should be let through, and a pointwise multiplication operation. An LSTM has three of these gates to protect and control the cell state, deciding what information is going to be thrown away, what new information is stored in the cell state and what will be the output.

2.2.9 Probabilistic methods

A probabilistic method draws probability distributions over a target feature. By using Bayesian inference, the probability for a hypothesis can be computed given a set of observations. The posterior probability is usually computed by using the Bayes' rule [45]. Given a set of evidences or observed values in the predictor features, the conditional distribution is calculated as follows:

$$P(\mathbf{y} = (y_1, \dots, y_n) | \mathbf{x}_i = (x_1, \dots, x_m)) = \frac{P(\mathbf{x}_i = (x_1, \dots, x_m) | \mathbf{y} = (y_1, \dots, y_n)) P(\mathbf{y} = (y_1, \dots, y_n))}{P(\mathbf{x}_i = (x_1, \dots, x_m))}$$

Probabilistic graphical models can be drawn based on this concept, where nodes of the graph represent random variables connected by arcs if they are conditionally dependent [46]. In Markov networks, for instance, the arcs are undirected. In contrast, Bayesian or Belief networks present directed graphical models, showing a more complicated notion of independence. From this basic idea, Dynamic Bayesian Networks can be defined as directed graphical models of stochastic processes that generalize Hidden Markov Models (HMMs) and Linear Dynamical Systems (LDSs) [47]. They are temporal models with discrete hidden nodes and discrete or continuous observed nodes, which represent the hidden and observed state in terms of state variables and can have complex interdependencies. Although a LDS present the same topology as an HMM, in a LDS all the nodes are assumed to have linear-Gaussian distributions.

The Joint Probability Distribution (JPD) specified by a graphical model over all the variables allows answering all possible inference queries by means of marginalization. Due to the exponential time needed to compute the JPD for all the nodes (size $O(2^n)$, where n is the number of nodes), other more efficient approaches are commonly employed, e.g. variable elimination, sampling (Monte Carlo), parametric approximation, variational methods, bounded cutset conditioning or dynamic programming.

When describing a Bayesian network the network structure, or graph topology, and the conditional probability distribution parameters to be used must be defined. Both can be automatically learned from data (e.g. K2, Expectation Maximization or Maximum Likelihood Estimation algorithms), but it is sometimes difficult to approximate hidden nodes or missing data.

Probability density estimation methods also provide useful information regarding the similarity between feature vectors. To that concern, the strategy used to find high density regions in feature space will determine the kind of low density samples to be isolated. The more samples are considered to estimate the probability density function, the more global outliers will be found. In contrast, local anomalies, corresponding to samples that can happen in any region or neighborhood in the feature space, will be hidden [48]. In Figure 2.4 an example of local and global outliers in a bivariate problem is shown. X_1 and X_2 are global anomalies, whereas X_3 is a local outlier. c_i , $i = (1, 2, 3)$,

are the clusters or groups of similar data found. Interestingly, c_3 could be considered as an anomaly given the small percentage of instances grouped on it.

Figure 2.4: Global vs local outliers.

2.2.10 Ensemble methods

Ensemble methods are combinations of single methods to improve the predictive accuracy and control overfitting [49] [50]. The combination strategy is usually performed on the basis of averaging or voting criteria. This is possible nowadays thanks to the exponentially increasing computational capacity over time. There exist several approaches that make use of an ensemble strategy to solve complex problems. A subset of the most relevant ones are the following:

- **Random forests:** this technique grows an ensemble of trees and it averages their outputs to produce a final prediction [51]. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement (bootstrap). A large number of trees is therefore grown.
- **Bagging (bootstrap aggregation):** in this case additional data is generated from the original dataset for training [52]. To do so, combinations with repetitions are used to produce multisets of the same size as the original data. The idea is to decrease the variance of the prediction by narrowly tuning the prediction to expected outcome, and thus improving the model prediction accuracy.
- **Boosting:** this is a two-step approach in which subsets of the original data are first used to produce a series of averagely performing models, and then their performance are boosted by combining them together using a particular cost function [53]. Unlike bagging method, in the classical boosting the subset creation is not random and depends upon the performance of the previous models, meaning that every new subsets contains the elements that were most likely to be misclassified by previous models.

- **AdaBoost:** it is mainly used to boost the performance of decision trees on binary classification problems (i.e. decision stumps). Therefore, it is used for classification rather than regression. AdaBoost can be used to boost the performance of any ML algorithm but it is more commonly used with weak learners or models that achieve accuracy just above random chance on a classification problem [54].
- **Stacking:** similarly to boosting, several models are applied to the original data. The difference is that instead of just having an empirical formula for the weight function, a meta-level is introduced and another approach is used to estimate the input together with outputs of every model to estimate the weights or, more precisely, to determine what models perform better given the input data [55].

2.2.11 Validation and evaluation strategies

Once the model that best fits the data under study is generated, some good indicators regarding its validity and generalization capacity must be provided. Otherwise there are no guarantees that the model will keep good enough accuracy rates when analysing new, unseen data and therefore, misleading predictions and conclusions can be produced.

The validation of a model is based on the estimation of the error rate it produces. Such error rate is computed by comparing the prediction made by the model and the real value, either a continuous target feature or a discrete label. In order to provide the model with generalization capacity, a test error is calculated, which checks the model accuracy with unseen data. The model is trained with a subset of available information and tested with the remaining data, normally a small percentage. This operation can be done several times, by random subsampling or by cross validation [56]. In this last case, samples are divided into f exclusive subsamples of (approximately) the same size. Training set will contain $f - 1$ data sets and model learned will be tested with the other samples. This process is repeated for every fold and an average of the accuracy obtained is provided. The most commonly used number of folds is $f = 10$, but depending on the number of samples available other number of folds could be more appropriate, even $f = n$ (where n is the total number of feature vectors), which corresponds to the so-called leave-one-out cross validation. It produces an error estimate with low bias but high variance.

Other validation methods can be also applied, e.g. bootstrapping, loss function or Receiver Operating Characteristic (ROC) curves [57] [58].

In addition, some metrics and quality measures are widely used to evaluate the accuracy and performance of the resulting model, e.g. precision, sensitivity, F1-score and specificity. They are mainly computed on the basis of the true positive, true negative, false positive and false negative rates of predictions made.

2.3 Data-driven prognostics

Data-driven approaches for prognostics aims at predicting when an abnormal behavior is likely to arise. For this reason it becomes specially interesting to find a degradation pattern or trend in data. The idea is to model the degradation process of the monitoring system under study and to apply an extrapolation method able to anticipate a fault. Unfortunately, in many cases the degradation process is not well established, and therefore other additional strategies have to be envisaged for characterizing behaviors from data. The key assumption is that the majority of the monitoring data

implies a normal behavior. The basic approach consists on modeling normality from data and then to learn the mechanisms (e.g. physical or structural), implicit in data, that lead to an anomaly. Then, deviations from normality can be detected and further classified and studied. But this is a very challenging problem, since it involves carefully and automatically eliminating outliers and abnormal behaviors from training data.

The Cross Industry Standard Process for DM (CRISP) consists of a set of steps that can be seen as a workflow, from the business and data understanding, then the data preparation and modeling phase, which is focused on the data under study, and finally the evaluation and deployment of generated models. It is a key issue to fully understand the problem we are dealing with, in order to properly address the business requirements and thus improving the current situation in an optimal manner. Once the models are deployed in an online monitoring platform (e.g. a Condition Based Monitoring, CBM, or a Decision Support System, DSS), the output can be a useful recommendation, a warning or a critical alarm or even an optimal planning and scheduling of maintenance operations. A brief description of the main steps of the CRISP methodology for data-driven prognostics can be seen in Figure 2.5.

Figure 2.5: The CRISP methodology.

2.3.1 Normality modeling

When initial data are experimentally obtained an inevitable error in the measurement process has to be assumed. The way in which the impact produced by this error can be mitigated is by carrying out measurement repetitions, so that a statistical analysis can be performed. Thus, there exist statistical techniques, like regression analysis, which provide the model parameters taking into account the error that can be present in data. Besides, other statistical techniques like lack-of-fit test provide and estimate the suitability of the evaluated model through a fitness value which considers measurement errors.

Whatever the employed technique to find the coefficients with the best fitting is, in the classical regression methods the initial mathematical function is known, both because the physical law that supports the experiment is known or because it is an hypothesis that has to be evaluated. In the second case, the mathematical expression search process can be extremely complex, since the functions search space is infinity.

2.3.2 Behavior characterization

In order to understand real-world complexity, science tries to formulate world by means of a mathematical language able to model each of the observed physical processes. The mathematical representation of a physical process allows to quantify magnitudes and establish relations between involved variables. Once the physical model is built, it can be employed to predict the system state in the future whenever the initial conditions are known.

There is no physical model construction methodology beyond the scientist intuition, experience and intelligence [59]. The more complex the physical process to describe is, the more difficult to learn the corresponding model will be. Once a model is proposed, it has to be experimentally evaluated. When the obtained deviation between the model prediction and the experimental data is within reasonable error limits (arbitrarily determined), the model is considered as valid. This inductive process assumes that the model is valid while there are no contradictory experimental cases found.

Another approach when building physical models consists of learning models from monitoring data. By applying any of the ML techniques presented in previous sections, carefully considering the problem and the related data to be addressed, behaviors of interest characterized by sets of data can be automatically modeled [60] [61]. For instance, if a regressor is applied, once the model that best fits the data has been learned, warning thresholds can be established and thus trends can be estimated to anticipate failures and abnormal behaviors. Another popular strategy is based on unsupervised learning and clustering methods, by grouping data by their similarity so they are supposed to have the same pattern. Such pattern could be very significant in order to classify or to identify behaviors linked to the data, or in order to detect or to infer possible failures or anomalous conditions. Big groups, or groups that are close together, usually imply normal behaviors. Whereas small groups or events that are far from the pattern (of the same group or regarding a big group), imply anomalies or outliers (e.g. noise or transient data).

2.3.3 Fault detection and prediction

Fault detection can be performed in a straightforward manner, by simply comparing new, incoming data to previously modeled behaviors. To do so, several approaches can be applied, e.g. proximity-based, matching or inference [62]. Prognostics can be done under these circumstances by detecting the apparition of symptoms and their frequency in a time window, which are likely to occur before the real faulty event arises [63].

Data-driven approaches are more practical than physics-based approaches because physical degradation models are rare in practice. Even though they turn out to be more accurate in prediction results using the same data, since they provide more information (e.g. physical models or loading conditions), they require a deep knowledge of the involved physics and the tuning of the model parameters is costly [64]. Having monitoring, historical data, a data-driven model can highly support the knowledge modeling process. As new, previously unseen conditions are envisaged, and consequently the model accuracy decreases in terms of the apparition of higher rates of false positives and false negatives, the model needs to be retrained. Some self-learning strategies are put in practice, e.g. triggering the learning process when new external conditions are envisaged or with some periodic frequency, but the final validation by an expert is advisable before deploying the model in a condition monitoring platform.

Remaining Useful Life prediction

When the goal is to truly anticipate the anomaly, another strategy must be adopted. Data-driven approaches rely on the availability of run-to-failure data [65]. The temporal dimension is crucial to this concern, since it can provide the warning trend that leads to a fault condition. If it is well defined and characterized, degradation must show a monotonically increasing or decreasing behavior. The challenge lies in correctly modeling the normal condition and then establishing the mechanisms that leads to different anomalies. Given the monotonically property of the warning trend, using linear or quadratic polynomials could be enough to accurately anticipate an event of interest.

The Remaining Useful Life (RUL) prediction can be accomplished by different strategies, e.g. by applying a multivariate pattern matching process from the data to the remaining life or by first estimating damage and then extrapolating its progression over time until it intersects the failure criterion. The future degradation state is predicted based on the model and the identified parameters, including the uncertainties inherent to predictions made in future monitoring system states. As we get further in time from the current state, the uncertainty increases and, consequently, the prediction accuracy decreases.

From the wide variety of algorithms available, Gaussian process regression and ANNs are two very popular examples of prognostics models. They are briefly introduced in the following subsections.

Gaussian process regression

A Gaussian process is a stochastic process that generates normally distributed samples with multivariate normal joint distributions [66]. Gaussian process regression models typically assume that residuals are Gaussian and have the same variance for all observations. However, applications with input-dependent noise (heteroscedastic residuals) frequently arise in practice, as do applications in which the residuals do not have a Gaussian distribution.

A Gaussian process regression model can be expressed as a function f on a set of covariates X , which is a zero-mean Gaussian process with covariance function $K = (X, X')$ that approximates a regression target variable $\mathbf{y}$ such that $\mathbf{y} = f(x) + \epsilon$; with ϵ being independent normally distributed noise. In order to predict the value of new test point y_i given a training set $D = \{X_i, y_i\}_{i=1}^N$ and an input vector $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, with $i = 1, \dots, n$, the following posterior distribution must be estimated:

$$p(f_i | X_i, D) N(f'_i, \sigma_i^2) \quad (2.1)$$

where the mean and variance are defined as:

$$\begin{aligned} f &= K^T M^{-1} y \\ \sigma_i^2 &= K(X_i, X'_i) - k_i^T M^{-1} k_i \end{aligned} \quad (2.2)$$

given $M = (K + \sigma^2 I)$ and $k_i = [K(X_i, X_1), \dots, K(X_i, X_N)]^T$.

Gaussian process regression approach allows to accurately interpolate and predict new values when few observations are available over time. Therefore, from data collected at specific dates, the mean of each distribution of values is computed and a model that statistically fit observations can be learned. A confidence interval can be also calculated, given the standard deviation of each set of

samples to the mean. The degree of uncertainty is much higher as the differences on observations increases over time.

Artificial Neural Networks

ANNs are non-symbolic models inspired by human brain cognitive ability [25]. Instead of using human-understandable symbols, they use other knowledge formats such as weights, connections, etc. Neurons, represented by process units, are interconnected with each other and grouped into layers forming a network. Input layer receives input data while output layer provides output data. Input-output patterns are used to train the ANN. During the training phase ANN error is iteratively reduced by using different algorithms such as the back-propagation method or genetic algorithms. From a mathematical point of view an ANN can be considered as a complex non-linear mathematical expression regressor. By means of a set of training data the ANN is able to extrapolate or generalize its behavior. This is a very interesting feature in order to reduce the data set cardinality as well as the number of experiments. Moreover, the nature of the ANN computational process makes ANN less sensitive to noise in input data than other data-driven model generators.

There are no precise rules to determine an ANN architecture: number of layers, neurons distribution, topology, etc. A small neural network will provide limited learning capabilities, whereas a large one will overfit the training data, inducing generalization loss. Heuristic-based approaches are used in order to define the most promising ANN architecture regarding data to be fitted.

They are not directly human-understandable since resulting model is not explicit. This black-box format may imply an important drawback in many domain applications. Despite this fact, ANNs have been used in engineering and industrial applications for many years due to their learning and generalization capabilities and noise-tolerance [67]. Time series prediction [68], classification [69] and regression [70] are most common applications of ANNs.

Multilayer perceptron (MLP) networks for regression are networks with only one or more hidden layers and a sigmoid activation function. It is trained using back-propagation with no activation function in the output layer, which can also be seen as using the identity (linear) function as activation function. Therefore, the output is a set of continuous values $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_{n_1})$. Given a target feature, $\mathbf{y} = (y_1, \dots, y_n)$ and a set of m features, $X = \{X_1, \dots, X_m\}$ where each feature X_i can take a value from its own set of possible values χ_i , and n feature vectors or instances, $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, with $i = 1, \dots, n$, the MLP for regression fits training data as follows:

$$\hat{\mathbf{y}} = \sum_{j=1}^d \mathbf{w}_j \mathbf{z}_j \quad (2.3)$$

where $\hat{\mathbf{y}}$ is the target feature values estimated by the MLP, d is the number of neurons in the hidden layer, $\mathbf{w}_j$ is the weights vector and $\mathbf{z}_j = S(v_{i0} \sum_{k=1}^m x_k v_{ik})$, being $S(t) = \frac{1}{1+e^{-t}}$ the sigmoid activation function.

The parameters of the equation (weights) are computed by back propagation, as it can be seen in Equation 2.4, and using a loss function based on the gradient descent method. Network error is usually computed as the mean squared error, $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$.

$$\mathbf{w}_j \leftarrow \mathbf{w}_j - \eta \frac{\delta E}{\delta \mathbf{w}_j} + \alpha \mathbf{w}_j \quad (2.4)$$

where η is the learning rate, E is the network error, δ is the error gradient and α is the learning momentum.

By means of a set of training data the ANN can extrapolate or generalize its behavior. Deviations from normal behaviors can be drawn from residuals obtained when predicting new values over time and then establish warning trends able to anticipate anomalies.

In the following sections the application of ML paradigms to solve different problems related to a wide variety of industrial sectors are presented. They correspond to R&D projects with common underlying technical and scientific needs materialized into a variety of customer specific requirements. The knowledge and experience acquired during the implementation of these projects has driven the development of several learning frameworks, towards the MDI 4.0, which is presented as one of the main contributions of this work.

3.1 Railway industry

- Title and acronyms: Intelligent Diagnosis System in Railway Data (SIDIM, AURA) [71] [72]
- Project typology: Basic research project and industrial project
- Company: NEM Solutions and Construcciones y Auxiliar de Ferrocarriles (CAF)
- Period: 2006-2009
- Keywords: Unsupervised anomaly detection, unsupervised classification, intelligent monitoring systems, clustering, monitoring and diagnosis systems, grammar guided genetic programming, fuzzy logic, rule-based system, evolutionary computation, knowledge discovery

The goal of this project was to identify and model abnormal behaviors in the monitoring system under study, the train bogies, one of the most critical parts of the high-speed fleet. In one of the projects this problem was addressed from an unsupervised perspective, since there was limited or no information regarding anomalies and system faults that occurred in the past. Different unsupervised methods were used to accurately identify abnormal behaviors from data, namely hierarchical clustering, k-means and SOM or Kohonen maps. The K-Means method improved the detection of slight data variations related to bogies behaviors.

Another approach based on Grammar-Guided Genetic Programming (GGGP) was also implemented, in order to automatically generate fuzzy knowledge bases. The resulting fuzzy rules set is

able to optimally represent the expert knowledge for the detection of abnormal behaviors. In this case, the proposed evolutionary system successfully supported the definition of the fuzzy rules set that represents the knowledge to be modeled regarding train bogies data.

3.2 Wind industry

- Title and acronym: Ocean Renewable Energy Leaders (OCEAN-LIDER)
- Project typology: Basic research project and industrial project partially supported by the CENIT-E Programme (Spanish Government and CDTI)
- Company: NEM Solutions and IBERDROLA
- Period: 2010-2013
- Keywords: Condition based monitoring, artificial intelligence, system modeling

The goal of this project was to model and to early detect failures in the wind energy sector. The main motivation was to avoid important costs derived from unexpected maintenance actions and operations, potentially offshore. To that aim, a novel CBM system that provides an efficient behavior modeling and symptoms analysis was proposed. It automatically models the normal behavior of the system and then it detects deviations from it. This includes both sudden deviations that need to be detected as close to real-time as possible and slow, progressive degradations that must be taken into account to optimally schedule the maintenance operations, and to accurately estimate the remaining useful life of the asset. Historical data gathered from a set of wind turbines were analysed and the resulting models were applied and tested in real time, detecting deviation patterns from the normality models before an anomaly really occurs.

3.3 Maritime sector

- Title and acronym: Automation Development for Autonomous Mobility (ADAM) [73] [74]
- Project typology: Industrial project partially supported by the INNPRONTA Programme (Spanish Government and CDTI)
- Company: NAVANTIA
- Period: 2011-2014 and 2016-2018
- Keywords: Behavior characterization, condition monitoring, constrained k-means clustering, fuzzy modeling, local outlier factor, support vector machines, kernel density estimator, bandwidth selection, normality modeling, fault prediction, health status assessment, evolutionary modeling, genetic programming

This project is another example of condition monitoring of complex assets, but in this case the main contribution is a novel methodology based on a workflow that combines a set of ML-based methods. It can be efficiently used to generate normality and behavior models from data that are able to predict potential failures in an online fashion, preventing costly corrective interventions. The

proposed methodology was integrated in a Condition Based Maintenance Plus (CBM+) platform, integrating Reliability Centered Maintenance (RCM) strategies and AI algorithms. A set of marine propulsion systems were analysed, namely auxiliary diesel engines and reduction gears. Experimental results show promising advantages over traditional strategies, detecting deviation patterns and degradation symptoms at an early stage. Therefore, critical faults can be anticipated and serious damages can be avoided, improving reliability and availability of the assets.

3.4 Manufacturing sector

- Titles and acronyms: Intelligent Riveting Monitoring (RIVETEST) [75], and Innovative Blind Fastener Monitoring Technology for Quality Control (BLINDFAST)
- Projects typology: Basic research project and Clean Sky 2 Joint Technical Programme (European Commission)
- Company: Airbus
- Period: 2008-2011 and 2015-2018
- Keywords: Classification, multiclassifier, drill wear prediction, pattern identification, kernel density estimator, behavioral patterns, outlier detection, unsupervised classification, blind fasteners installation

In the RIVETEST research project a multiclassifier approach that combines the output of some of the most popular data mining algorithms was designed, addressing the drill wear detection in the manufacturing sector. The accuracy obtained by each isolated classifier was compared with the performance of the multiclassifier when characterizing the patterns of interest involved in the drilling process, predicting the drill wear. The approach is based on voting criteria, by estimating the confidence distributions of each algorithm individually and combining them according to three different methods: confidence voting, weighted voting and majority voting. Experimental results showed that, in general, false positives obtained by the classifiers can be slightly reduced by using the multiclassifier approach.

The other approach, BLINDFAST, aims at identifying behavioral patterns from monitoring data related to blind fasteners installation. To do so, a kernel density-based pattern classification method was proposed, which analyzes the fastener features representing the quality of the installation. Patterns are computed as the average of related monitoring torque-rotation diagrams, on the basis of densities and distances between samples. New fastening installations can be thus automatically classified in an online fashion.

3.5 Civil structures and materials

- Titles and acronyms: Clustering methods for Structural Health Monitoring (CSHM) [76], and Signal Processing methods for Defect Detection in Materials and Structures (SPDDMS) [77]
- Projects typology: Basic research projects
- Period: 2014-2016

- Keywords: Structural health monitoring, damage detection, novelty detection, unsupervised learning, K-means clustering, signal processing methods

The overall aim of the CSHM project was to apply unsupervised and clustering methods for damage detection, as part of the Structural Health Monitoring (SHM) to the Sydney Harbour Bridge. The motivation of this work was to identify clusters of bridge parts with similar behaviors based on sensor data and other information, to understand bridge global behavior and to, finally, complement local damage identification techniques.

In addition, with relation to the SPDDMS study, a SHM system based on guided waves was designed. The proposed approach consists of comparing the signals to each other (signal related to non-damaged components compared to damaged signal), in order to measure their differences as a distance that can be used to estimate the damage level. To do so, different mathematical methods and distance metrics were applied, e.g. Signal Difference Coefficient (SDC), Dynamic Time Warping (DTW), Euclidean, Manhattan and Chebyshev. The accuracy obtained by each of them when detecting damage was analyzed and discussed.

3.6 Agro-food industry

- Titles and acronyms: Food of the future (FOODBASK), and Intelligent software to support sustainable strategies and decisions in the meat chicken production chain (iBOSP)
- Projects typology: Basic research project and Industrial project partially supported by the Ertortek and RETOS programmes (Basque and Spanish Governments)
- Companies: NEIKER-Tecnalia
- Period: 2009-2011 and 2013-2016
- Keywords: Decision Trees, Decision support system, Quantile regression forests, Environmental indicators, Efficient production, Animal welfare, Machine learning

Two different Decision Support Systems (DSSs) were provided within this research line. One of the two projects was based on decision trees, and the resulting DSS was able to classify the sex and the age of the mackerel (*scomber japonicus*) on the basis of colorimeter data. The other DSS aimed at estimating weights, leg problems and mortality rates in animal farming, assuring efficient and sustainable production according to animal welfare and social responsibility. The proposed quantile regression forests-based growth, welfare and production models turned to be robust and comprehensive, yet accurate decision support tools based on deviations from optimal environmental conditions, automatically collected by a set of sensors.

3.7 Relationship of projects and scientific activity

Figure 3.1 shows a summary of the main scientific activity around the aforementioned projects. It can be noticed that different projects share a common research line and scientific activity, related to the application of ML methods to solve different complex problems and towards the MDI 4.0. In most of the cases the goal was to generate one or several prediction models from data, then applying them in an online fashion for a real-time intelligent monitoring and predictive maintenance of the

involved assets. But also to provide a decision support tool or recommender to detect deviations from optimal conditions, for instance in the case of the agro-food industry. Or to study the effect of applying different unsupervised methods and distance metrics to optimally detect structural damage.

Figure 3.1: Relationship between the R&D projects and the scientific activity in ML methods and applications carried out in the context of this dissertation.

In the next chapter some remarkable contributions to the state-of-the art methods and applications in ML are further presented and discussed. They are directly related to some of the above mentioned industrial sectors and R&D projects, addressing the MDI 4.0 challenge from a DS perspective.

Main contributions

In this chapter the main contributions to the current methods and applications in ML for data-driven prognostics, towards the MDI 4.0, are presented. They are related to the research activity driven by specific problems and needs raised in different application fields. The proposed methods and resulting applications address complex and challenging industrial scenarios that involve data in different formats, acquisition frequencies, size and nature.

Several methods are proposed, depending on the data to be explored and modeled and on the problem to be solved. They are listed as follows:

- A combination of constrained K-means, fuzzy modeling and LOF-based score.
- Kernel-based SVMs with automatic bandwidth selection.
- Deep evolutionary modeling of condition monitoring data.
- A clustering based approach for SHM.
- A kernel density-based pattern classification approach.
- Quantile regression forests-based modeling and environmental indicators.

The above mentioned methods were motivated by a set of practical problems, which correspond to different MDI 4.0 key aspects. They are listed below:

- CBM and predictive maintenance in marine diesel engines and marine propulsion systems.
- Health status assessment and pattern classification in marine diesel engines, bridges and blind fasteners installation.
- Quality estimation and production optimization in animal farming.

These real problems involve challenging industrial scenarios that must be dealt with from a DS perspective. They are further described in the following sections, and the proposed methods to solve them are also presented and discussed.

4.1 ML methods for CBM and predictive maintenance

4.1.1 A case study on marine diesel engines

Motivation

Unexpected incidental failures imply an important impact in terms of risks, costs, resources and service loss that should be minimized [78]. The growing complexity of industrial equipment, systems and installations results in an ever-increasing amount of health monitoring information, which eventually exceed the capacity of most fault detection systems and makes the design of successful maintenance methodologies more challenging. Moreover, it is important to provide a better understanding of monitored systems and to efficiently characterize the normal behaviors from a huge amount of historical data. The lack of knowledge about the behavior of complex assets makes the problem of maintenance very difficult.

Naval sector, for instance, is traditionally focused on preventive strategies, usually divided by 3-4 maintenance difficulty level groups: vessel crew, vessel base, shipyard, and manufacturer [79]. Crew and base level maintenance tasks are planned and carried out during vessel operating time; however, shipyard and manufacturer tasks are done in programmed dock periods. The whole vessel life cycle is divided by long operating periods separated by short maintenance periods, some of them dry-dock. When an important unexpected breakdown occurs during operation (at both shipyard or manufacturer level), vessel activity stops and planned missions have to be cancelled. Additionally, important repair costs must be envisaged. In such cases it is often necessary to open dismantling routes aiming to remove defective parts. Sometimes a cesarean in the vessel hull or even a dry-dock have to be performed to extract the involved equipment. Therefore, total costs may include: defective parts, reparation manpower, dismantling route procedures and a cesarean or a dry-dock. Indirect tasks are usually more expensive than normal component reparation processes. Moreover, when a failure arises while a mission is in process, objectives could not be fulfilled or the mission might be cancelled and vessel is returned to base. But the worst-case scenario is that in which vessel or crew safety are threaten.

This work is an effort to implement a novel ML-based approach that aims to minimise the negative effects of unexpected breakdowns, providing a reliable fault detection and prediction strategy. This approach has been applied over real operational data acquired from an auxiliary diesel engine during real vessel operation, since it is one of the most critical vessel components: it supplies propulsion and energy to the vessel and its behavior is complex, as it is a reciprocating engine matched with a turbocharger. A study in-depth was carried out into the possibility of improvement through the use of data-driven ML techniques to statistically model the normal behavior of the engine, in a fully automated unsupervised fashion. To do so, behavior characterization and fuzzy modeling are applied to monitoring sensor data. Moreover, knowledge models generated are comprehensive, yet accurate methods to anticipate potential critical faults. Resulting models and all available information are integrated in a specific CBM+ system, which combines CBM, RCM and AI capabilities. Although the study is focused on the exploitation of operational parameters, it must be mentioned that the proposed approach can be also applied to other types of operational parameters that can be used as failure indicators, such as vibrations, fluids analysis, thermography information, in-cylinder pressure or ultrasonic information.

Proposed approach

In the proposed constrained K-means approach, K value is automatically provided based on cluster distribution and cluster data variance as established by [80, 31]. It is computed as a previous step of the clustering process. Variance explained of resulting classification model, or clusters compactness, Cmp , is calculated until $Cmp_K - Cmp_{K-1} \leq 0.5$. As Euclidean distance is applied [81], it becomes coherent to average cluster scattering index [82]. The member of each cluster should be as close to each other as possible. The clusters compactness is computed as it can be seen in Equation 4.1.

$$Cmp = \frac{1}{K} \sum_{i=1}^K \frac{\|\sigma(C_i)\|}{\|\sigma(X)\|}$$

$$\sigma(C_i) = \begin{bmatrix} \sigma_{C_1}^1 \\ \vdots \\ \sigma_{C_K}^m \end{bmatrix} \quad \sigma(X) = \begin{bmatrix} \sigma_X^1 \\ \vdots \\ \sigma_X^m \end{bmatrix} \quad (4.1)$$

where K is the number of clusters; $\sigma(C_i)$ is the variance of cluster C_i with $\sigma_{C_i}^p = \frac{1}{|C_i|} \sum_{j=1}^n (x_j^p - C_i^p)^2$ and $\sigma(X)$ is the data variance with $\sigma_X^p = \frac{1}{n} \sum_{i,j=1}^n (x_i^p - x_j^p)^2$, $i \neq j$.

When knowledge regarding the system behavior to be modeled is available in addition to the data instances themselves, the algorithm can be modified to make use of this knowledge [83]. A constrained that consists on specifying an asset main input feature, X_f , is thus considered so that instances are grouped on the basis of X_f non-parametric distribution. Given the number of clusters, K , their width in terms of X_f values are estimated as $w = \frac{\max(X_f) - \min(X_f)}{K}$. Then, for each cluster C_i , $i = (1, \dots, K)$, $u_K = \min(X_f) + (i+1) * w$ and $l_K = u_{K-1}$, with $u_0 = \min(X_f)$, are set as upper and lower limits on X_f values, respectively. Then, a local distance-based outlier detection can be accurately performed considering the asset status, determined by its main input feature values.

Given a set of m features, $X = \{X_1, \dots, X_m\}$ where each feature X_i can take a value from its own set of possible values χ_i , and n feature vectors or instances, $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, with $i = 1, \dots, n$, $L2$ normalization is calculated for each instance $\mathbf{x}_i$ in the data set in order to minimise the impact of having different range of values of raw data in the resulting classification model. It is computed as the root of the sum of its squared elements, as it can be seen in Equation 4.2.

$$L2(\mathbf{x}_i) = \sqrt{\sum_{l=1}^m |x_{il}^2|} \quad (4.2)$$

Once the normalization of data samples is performed using $L2$, and in order to calculate the distance between instances, $\mathbf{x}_i$ and $\mathbf{x}_j$, Euclidean metric is computed (see Equation 4.3).

$$D(\mathbf{x}'_i, \mathbf{x}'_j) = \|\mathbf{x}'_i - \mathbf{x}'_j\| = \sqrt{\sum_{l=1}^m (x'_{il} - x'_{jl})^2} \quad (4.3)$$

where $\mathbf{x}'_i = \frac{\mathbf{x}_i}{L2(\mathbf{x}_i)}$ and $\mathbf{x}'_j = \frac{\mathbf{x}_j}{L2(\mathbf{x}_j)}$ are the $L2$ -normalized instances $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively.

The convergence criteria is established as a maximum number of iterations and a stability threshold, which checks the clusters compactness variation of resulting classification model from i -th iteration to iteration $i+1$ (see Equation 4.1).

An iterative outlier detection loop is then performed, so that from groups of instances formed during the constrained-learning stage, outliers are detected and isolated and behaviors of interest given a certain asset main input feature values are characterized. They are represented by instances that belong to a certain group but differs notoriously from the pattern. For each iteration in the outlier detection process and for each cluster, C_i , an anomaly threshold is calculated based on the distances of instances grouped in C_i to the centroid, as it can be seen in Equation 4.4.

$$Th(C_i) = \frac{1}{|C_i|} \sum_{\mathbf{x}_j \in C_i} D(\mathbf{x}'_j, \mathbf{c}_i) + 3 \sqrt{\frac{\sum_{\mathbf{x}_j \in C_i} D(\mathbf{x}'_j, \mathbf{c}_i)^2}{|C_i| - 1}} \quad (4.4)$$

where $\mathbf{c}_i = \frac{1}{|C_i|} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}'_j$ is the centroid of cluster C_i and $\mathbf{x}_j$, $j = \{1, \dots, |C_i|\}$, is the j -th instance grouped in cluster C_i .

Events whose distance to the centroid are over the anomaly threshold are set as outliers. Consequently, for each iteration during the outlier detection process and for every cluster, centroids are recalculated filtering out detected outliers. The process stops when no more outliers are detected. Outliers detected within a cluster and small clusters could imply abnormal asset behaviors and operational faults.

The overall algorithm can be seen in Algorithm 4.1.

Algorithm 4.1 CBM and predictive maintenance in marine diesel engines. Constrained K-means clustering for outlier detection

Input: a set of m features, $X = \{X_1, \dots, X_m\}$

```

1: Compute  $K$  using Equation 4.1
2: Select an asset main input feature  $X_f \in X = \{X_1, \dots, X_m\}$ 
3: Compute  $w = \frac{\max(X_f) - \min(X_f)}{K}$ 
4: for all  $C_i$ ,  $i = (1, \dots, K)$  do
5:   Find  $u_K = \min(X_f) + (i + 1) * w$  and  $l_K = u_{K-1}$ , with  $u_0 = \min(X_f)$ 
6:   for all  $\mathbf{x}_j \in \{l_K, u_K\}$  do
7:      $\mathbf{x}_j \rightarrow \{C_i\}$ 
8:   end for
9: end for
10:  $outliers = \{\}$ 
11: while  $outliers$  found do
12:   for all  $C_i$ ,  $i = (1, \dots, K)$  do
13:     Compute  $Th(C_i)$  using Equation 4.4
14:     if  $D(\mathbf{x}'_j, \mathbf{c}_i) > Th(C_i)$  then
15:        $\mathbf{x}_j \rightarrow outliers$ 
16:       Remove  $\mathbf{x}_j$  from  $C_i$ 
17:     end if
18:   end for
19: end while

```

Although constrained K-means clustering is a known algorithm in the literature [84, 85], the main novelty in this work lies in successfully applying the proposed approach in practice, to a complex industrial scenario.

Anomaly detection

The anomaly detection process is performed on the basis of behaviors characterized from data by applying constrained K-means clustering and on outliers found. The most important behavior to be considered is normality.

Fuzzy rules generation process and inference engine are based on the work proposed by Cingolani et al. [86]. Fuzzy controllers are currently considered to be one of the most important applications of the fuzzy set theory proposed by Zadeh [21]. This theory is based on the notion of the fuzzy set as a generalization of the ordinary set characterized by a membership function μ that takes values from the interval $[0, 1]$ representing degrees of membership in the set. Fuzzy controllers typically define a non-linear mapping from the systems state space to the control space. Thus, it is possible to consider the output of a fuzzy controller as a non-linear control surface reflecting the process of the operators prior knowledge. A fuzzy controller is a kind of fuzzy rule-based system that is composed by the following parts:

- a knowledge base that comprises the information used by the expert operator in the form of linguistic control rules,
- a fuzzification interface, which transforms the crisp values of the input variables into fuzzy sets that will be used in the fuzzy inference process,
- an inference system, which uses the fuzzy values from the fuzzification interface and the information from the knowledge base to perform the reasoning process, and
- the defuzzification interface, which takes the fuzzy action from the inference process and translates it into crisp values for the control variables.

The knowledge base encodes the expert knowledge by means of a set of fuzzy control rules.

Figure 4.1: CBM and predictive maintenance in marine diesel engines. Proposed fuzzy partition.

In order to consider each group of similar events, normal and outliers, in a relevant way, a fuzzy partition in two fuzzy sets is defined over the universe U_i of the Euclidean distances to centroid in cluster C_i . Let d_i be the Euclidean distance of event $\mathbf{x}_i$ to centroid of cluster C_i , computed as it can be seen in Equation 4.3. The membership function of these fuzzy sets, respectively denoted as μ_n and μ_o are defined in Equation 4.5 and Equation 4.6.

$$\mu_n(d_i) = \begin{cases} \frac{Th(C_i) - d_i}{Th(C_i) - min_n} & \text{if } d_i \in [min_n, max_n] \\ 0 & \text{otherwise} \end{cases} \quad (4.5)$$

$$\mu_o(d_i) = \begin{cases} \frac{d_i - Th(C_i)}{max_o - Th(C_i)} & \text{if } d_i \in [min_o, max_o] \\ 0 & \text{otherwise} \end{cases} \quad (4.6)$$

where min_n and max_n and min_o and max_o are the minimum and maximum values in fuzzy sets *normal* and *outlier*, respectively, $\mu_n(d_i) : d_i \rightarrow [0, 1]$ quantifies the degree of membership of d_i to *normal* and $\mu_o(d_i) : d_i \rightarrow [0, 1]$ quantifies the degree of membership of d_i to *outlier*. Obtained fuzzy partition is described in Figure 4.1.

Note that considering a membership degree allows to provide experts with more interpretable information about the real status of the asset.

Event score

In order to distinguish between outliers and real faults, a local outlier factor is computed for each event distance, as it can be seen in Equation 4.7. It is based on the work proposed by [87] and it measures the degree of isolation of a point with respect to its neighbors. Thus, the local density is also considered when determining if an outlier is an actual anomaly.

$$LOF(d_i) = \frac{\sum_{d_j \in N(d_i)} \frac{LRD(d_j)}{LRD(d_i)}}{|N(d_i)|} \quad (4.7)$$

where $LRD(d_i)$ is the local reachability distances of d_i , computed for the closest subset of distances $N(d_i)$ of size $max\left\{\frac{|C_i|}{10}, 1\right\}$, as it is shown in Equation 4.8. $LRD(d_j)$ is calculated likewise.

$$LRD(d_i) = \left(\frac{\sum_{d_j \in N(d_i)} reachDist(d_i, d_j)}{|N(d_i)|} \right)^{-1} \quad (4.8)$$

where $reachDist(d_i, d_j) = max\{k - dist(d_i), d_j\}$ and $k - dist(d_i)$ is the k -distance neighborhood of d_i , with $k = |\{outliers\}|$ for each cluster, being $k = max\left\{\frac{|C_i|}{100}, 1\right\}$ in the case that no outliers are found in cluster C_i .

The *score* of event $\mathbf{x}_i$ is then defined as a combination of the membership function to fuzzy sets *normal* and *outlier* and the local density of distances to the normal behavior.

$$score(\mathbf{x}_i) = \begin{cases} \frac{\mu_n(d_i) * LOF(d_i)}{max\{\mathbf{LOF}\}} & \text{if } d_i \in [min_n, max_n] \\ -\frac{\mu_o(d_i) * LOF(d_i)}{max\{\mathbf{LOF}\}} & \text{if } d_i \in [min_o, max_o] \end{cases} \quad (4.9)$$

where $\mathbf{LOF} = (LOF_1, ..., LOF_n)$, calculated for the whole set of distances $\mathbf{d} = (d_1, ..., d_n)$.

An event will be considered as anomaly if its *score* falls below -0.5 . Figure 4.10 shows an example of the evolution of the resulting event score calculated over time. The proposed event score is an optimal method for reducing false positives in anomaly detection process. It also allows users to access results quickly and efficiently.

Figure 4.2: CBM and predictive maintenance in marine diesel engines. Event score example.

Experimental results

The proposed methodology has been tested on two months' time operational data, from January to February 2015, of an auxiliary diesel engine. A total of 17,377 events are analyzed. The number of clusters, K , is set to 8 given the percentage of explained variance, as it can be seen in Table 4.1. Although other complementary tests were performed with different K values (e.g. $K = 6$ and $K = 10$), results obtained were less accurate in terms of the number of false negatives and false positives.

Table 4.1: CBM and predictive maintenance in marine diesel engines. Percentage of variance explained for each number of clusters.

Number of clusters	% of explained variance
2	76.03
3	88.23
4	93.46
5	95.32
6	95.91
7	96.52
8	96.96

As a result of the constrained K-means clustering-based outlier detection process, a total of 78 events are isolated from normal patterns. The resulting cluster distribution can be seen in Table 4.2 and in Figure 4.3.

Table 4.2: CBM and predictive maintenance on marine diesel engines. Clusters distribution.

Cluster	0	1	2	3	4	5	6	7
Total number of events	330	13	751	2098	10550	3265	286	6
Outliers Found	9	0	13	7	33	15	1	0

Figure 4.3: CBM and predictive maintenance in marine diesel engines. Bar chart of resulting clusters distribution.

A cluster is graphically represented by a dashed line, which corresponds to the cluster centroid. The operational parameters values are drawn as dots, being each event a set of dots (one value per operational parameter) at a specific time instant. Operational parameters values are normalized between 0 and 1. As it can be expected, clusters containing stable engine load conditions grouped the majority of events. That is the case, for instance, of Cluster 3, 4 and 5. Outliers detected in such clusters are more likely to imply real system faults. However, and depending on nature and type of system fault, a problem can also occur when the engine is not in stable operating conditions. That behavior can be observed in relation to Cluster 0, when the engine is starting up. In Figure 4.4 a typical normal engine behavior under stable operation is shown. It corresponds to normal events grouped within Cluster 3.

Among outliers found, some events correspond to abnormally low exhaust temperatures in cylinders, probably due to a scavenge fire and/or a defective fuel valve, both of which are caused by a fuel system fault. This fault was present in a total of 13 events distributed in different clusters. In Figure 4.5 an example of such behavior in three events of one of the clusters formed can be appreciated. An alternator system symptom was also detected in 2 events in two different clusters. It is characterized by extremely high alternator intensity and reactive power at a normal engine load, as it can be seen in Figure 4.6. It is usually produced during docking manoeuvres.

Figure 4.4: CBM and predictive maintenance in marine diesel engines. Normal engine behavior.

Figure 4.5: CBM and predictive maintenance in marine diesel engines. Fuel System fault detected at a normal engine load.

Figure 4.6: CBM and predictive maintenance in marine diesel engines. Alternator System fault detected at a normal engine load.

The obtained confusion matrices are presented in Table 4.3. As it can be seen, only very few events correspond to the real fault under study: 15 in total, distributed throughout different engine load groups. This is one of the main difficulties in the anomaly detection process: how to distinguish between outliers and real faults. To quantify the number of anomalous events in each cluster, an event is considered as a real fault if its score is below -0.5 . Given that the test case scenarios showing the fuel system and alternator faults were deliberately chosen to be difficult to detect, it is still encouraging that the classification of faulty events rises above the false positive rate, accurately distinguishing real faults among outliers found.

Table 4.3: CBM and predictive maintenance in marine diesel engines. Results obtained per cluster.

Cluster		0	1	2	3	4	5	6	7
Real Normal		326	13	750	2094	10547	3263	285	6
Predicted Normal		328	13	750	2094	10547	3263	285	6
Real Fault	Fuel System	4	0	1	3	3	2	0	0
	Alternator System	0	0	0	1	0	0	1	0
Predicted Fault		2	0	1	4	3	2	1	0

Then the approach for anomaly detection was tested on a 10-fold cross-validation basis for each cluster, by segmenting the total set of cluster events into 10 equal parts. Thus, the confusion matrix presented in Table 4.4 is obtained, containing the average results of the 10 folds. Note that the constrained K-means clustering step is performed on the whole data set only once, in order to establish the different engine load operational ranges that will be used to isolate the outliers and

predict the real faults.

Table 4.4: CBM and predictive maintenance in marine diesel engines. Global confusion matrix.

78 outliers found	Predicted Normal	Predicted Fault
Real Normal	TN=17362	FP=0
Real Fault	FN=2	TP=13

In order to evaluate the results, the precision, sensitivity and specificity of the detection process are calculated. They are three widely used quality measures in this kind of processes. As it is shown in Table 4.5, precision, sensitivity and specificity are globally above 93%, so the approach accurately limits false anomalies and undetected faults. The inter-rater agreement statistic coefficient (Cohen's kappa, κ) is also computed aiming to evaluate the agreement between normal and fault events [88]. The resulting κ coefficient is 0.93, therefore a high strength of agreement is achieved.

Table 4.5: CBM and predictive maintenance in marine diesel engines. Global precision, sensitivity, specificity and κ coefficient.

	Real Normal	Real Fault	Global Results
Precision	99.99%	100%	99.98%
Sensitivity	100%	86.67%	93.34%
Specificity			100%
κ			0.93

By taking into account estimations made by this approach and next asset maintenance periods, usage planning, spare parts availability and manpower resources, optimal maintenance strategies can be suggested.

4.1.2 A case study on marine propulsion systems

Motivation

Physical formulation and simulation of systems and processes is usually accomplished when no data is available, e.g. to support system design [89]. When condition monitoring data is available, data-driven models can be used to automatically establish linear and non-linear relations between variables involved in the physical process under study by means of a mathematical language [90]. Once the empirical model is built, it can be employed to predict the dependent variable that characterizes the asset state in the future whenever the initial conditions are known. There are several techniques based on this approach: finite-element method, regression, interpolation, etc. [91]. However, this kind of techniques usually provides black-box solutions, lacking in conciseness and clarity, or it can only be applied to relatively simple physical processes.

Advanced mathematical modeling from operational data can be accomplished in many different ways, usually applying ML algorithms and statistical analysis [92] [93], which allow, for instance, managing knowledge from data in industrial scenarios by means of predictive models [94]. From existing methods, evolutionary-based algorithms provide a flexible, efficient and robust optimization and search strategy to infer physical models from data [60]. There are four main types of evolutionary-based algorithms: Genetic Algorithms (GA), Genetic Programming (GP), Evolutionary Programming (EP) and Evolutionary Strategies (ES) [95] [96]. In addition to these methods,

some combinations and extensions of previous ones can be found in the literature. Memetic Algorithms (MA), for instance, emerged as a combination of GA and Local Search (LS) methods and are now becoming very popular to solve a broad range of different problems. Another interesting example is GGGP, which is an extension of traditional GP systems and it always generates valid solutions, represented as individuals of the population or points that belong to the search space.

Time series prediction is another big issue when detecting anomalies in condition monitoring data. Traditional strategies use statistical measures such as moving average over a time window, ARIMA, Kalman filter and cumulative sum [97]. Regression models fitted to nonstationary data can better represent more complex, nonlinear dependencies with other related features. To that concern, Gaussian process regression [66] and Multilayer Perceptron (MLP) networks for regression [25] are two very popular examples of prognostics models. In a similar way to which it is done in MLP networks, deep learning methods can be seen as a cascade of many layers of processing units that combine the predictor features to approximate the target feature [40]. Recurrent Neural Networks (RNNs) present some interesting properties for time series forecasting like loops in them, allowing information to persist [43]. They are powerful and increasingly popular models for learning from varying-length sequence data, particularly those using LSTM hidden units [44]. LSTM networks for anomaly/fault detection in time series have demonstrated very good accuracy [98] [99].

Temporal anomaly detection approaches usually learn models that best fit time series to compute errors when comparing new, incoming data to predicted values. Some recent works have dealt with LSTM networks for anomaly detection in time series [100] [101]. However, they have not yet been combined with an understandable physical modeling of condition monitoring data for prognostics, anticipating anomalous data sequences over time.

This work is an effort to provide a deep evolutionary method able to efficiently and accurately model physical behaviors and to predict anomalous sequences of data. Models learned can be used for condition monitoring of the asset and to optimize its performance. Moreover, they can highly improve knowledge about the behavior and physics of the process or asset under study, providing mathematical expressions easy to interpret and to apply. The proposed approach is applied over real operational data acquired from a reduction gear of a CODOG (Combined Diesel Or Gas) marine propulsion system.

Proposed approach

Evolutionary physical modeling

By defining a simple mathematical grammar a great number of physical models can be inferred from data [102]. Thanks to both linear and non linear mathematical functions, a wide variety of industrial processes can be modeled from operational data: thermodynamic, kinematics, dynamics, chemical models, etc. Furthermore, by means of exponential functions the representation of several linear differential equation solutions can be accomplished.

The technique that allows searching both the coefficients that best fit data and the model itself is known as symbolic regression. Symbolic regression is based on concepts known some decades ago, although the symbolic regression concept is relatively recent [24]. Technological revolution experimented by computational systems made it possible to envisage mathematical expressions search and fitting, which are computationally expensive, with a progressive success.

The symbolic regression method for modeling different behaviors of complex systems by means of their operational parameters is based on the work proposed by Carrascal et al. [103], addressing the

automatic physical model design problem by an evolutionary approach. Evolution-based algorithms are considered to be the most flexible, efficient and robust of all optimization and search algorithms known to computer science. Therefore, they are becoming widely used to solve a broad range of problems of different nature and characteristics [104].

The overlying idea consists on inferring mathematical models that characterize behaviors of interest. Given a target feature, $\mathbf{y} = (y_1, \dots, y_n)$, a set of operands, $\{sine, tangent, cosine, exp, log, abs, pow, sqrt, *, /, +, -\}$, a set of m input features, $X = \{X_1, \dots, X_m\}$ where each feature X_i can take a value from its own set of possible values χ_i , and n feature vectors or instances, $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, with $i = 1, \dots, n$, the evolutionary process finds the derivation tree that best fits the model, $f(X) = \hat{\mathbf{y}} \approx \mathbf{y}$, by optimally combining features and operands and considering a fitness value that is the mean squared error (MSE) computed by the formula that is being tested. Each individual in the population codifies a valid derivation tree, which represents a mathematical expression. It is important for two individuals representing similar solutions to have similar codifications. This principle avoids a strictly random searching process. Consider, for instance, the formula $f(X) = X_1 + \sqrt{X_2} - \frac{X_3}{\sin X_4}$, which can be represented by the derivation tree shown in Figure 4.7.

During the evolutionary process, previously selected individuals are crossed with probability 0.9. Data is split into training (60% of the total data size) and test (30% of the total data size, holding the other 10% for validating the LSTM-based model) sets, aiming to avoid a possible overfitting problem. A maximum number of nodes to be used in mathematical expressions is also considered. From a mathematical point of view, this restriction avoids obtaining too complex expressions.

Figure 4.7: CBM and predictive maintenance in marine propulsion systems. An example of evolutionary modeling derivation tree.

The mathematical context-free grammar defined allows combining a great number of mathematical operands to create behavioral models from data. Thus, the automatic generation of complex mathematical expressions that represent behaviors of interest can be accomplished. The evolutionary modeling process outcome is a mathematical model in which all the constants have been solved. The corresponding flowchart can be seen in Figure 4.8.

The GP Module starts generating an initial population of mathematical expressions that are evaluated by the GA Module. The evaluation of candidate expressions is performed by computing their fitness value, which measures how accurately the expression solves the problem. After selecting the best candidates of the current population, a new population is generated applying the following operations:

- Reproduction: an existing expression is copied to next population.
- Crossover: different expressions are combined by crossover to create a new expression.
- Mutation: a component in an existing expression is changed to create a new expression.

The evolutionary process finishes when the the maximum number of evolutions is reached.

Figure 4.8: CBM and predictive maintenance in marine propulsion systems. Evolutionary physical modeling flowchart.

The details of the configuration parameters of the modeling process are presented in Table 4.6.

Table 4.6: CBM and predictive maintenance in marine propulsion systems. Evolutionary modeling configuration parameters.

Parameter	Value
Population size	500
Number of evolutions	1,000
Constant range	0-100
Operands	$\{sine, tangent, cosine, exp, log, abs, pow, sqrt, *, /, +, -\}$
Maximum nodes	21
Reproduction rate	0.1
Mutation rate	0.1
Crossover rate	0.9

LSTM-based anomaly prediction

RNNs are networks with loops in them, allowing information to persist [43]. The idea is to let every step of a RNN pick information to look at from some larger collection of information. Therefore, they

can be thought of as multiple copies of the same network, each passing a message to a successor. This chain-like nature reveals that RNNs are intimately related to sequences and lists. They are powerful and increasingly popular models for learning from varying-length sequence data, particularly those using Long Short-Term Memory (LSTM) hidden units [44]. LSTM neural networks overcome the vanishing gradient problem that is present in RNNs.

The key to LSTMs are the memory cells, whose states are carefully regulated by structures called gates. They are composed out of a sigmoid neural net layer, which outputs numbers between zero and one to describe how much of each component should be let through, and a pointwise multiplication operation. An LSTM has three of these gates to protect and control the cell state, deciding what information is going to be thrown away, what new information is stored in the cell state and what will be the output. More precisely, the input, I_G , output, O_G , and forget, F_G , gates prevent memory contents from being perturbed by irrelevant inputs and outputs and thus allowing for long term memory storage.

The LSTM units in a hidden layer are fully connected through recurrent connections. Layers are stacked so each unit in a lower LSTM hidden layer is fully connected to each unit in the LSTM hidden layer above it through feedforward connections. An schema of a typical LSTM unit is presented in Figure 4.9a. In the schema, σ and $\tanh$ refer to sigmoid and tangent hyperbolic neural network layers, respectively, whereas $\times$ and Σ correspond to product and addition pointwise operations, respectively, over vectors.

Figure 4.9: CBM and predictive maintenance in marine propulsion systems. Illustration of (a) a typical Long Short Term Memory unit and (b) Stacked LSTM-based network architecture used in this study, indicating the number of units (dimensionality of the output space) in each layer.

In this study a stacked LSTM network with two hidden layers and sigmoid activation function is used. Considering the time series that results from the evolutionary physical model predictions, $f(X) = \hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_n)$, data standardisation is applied to $\hat{\mathbf{y}}$ as $\frac{\hat{\mathbf{y}} - \mu(\hat{\mathbf{y}})}{\sigma(\hat{\mathbf{y}})}$, being $\mu(\hat{\mathbf{y}})$ and $\sigma(\hat{\mathbf{y}})$ the mean and standard deviation of $\hat{\mathbf{y}}$, respectively. Then data sequences are defined as m -dimensional

vectors whose elements correspond to the input variables. The LSTM-based prediction model learns to predict the next l values for d of the input values s.t. $1 \leq d \leq m$. Given the asset under study and the monitoring time frequency (one value per minute), d is set to 30 time steps and l is set to 10 time steps, which means that using the data corresponding to the last 30 minutes of data the LSTM network will estimate the next 10 minutes of data. Although other complementary tests were performed with different l values (e.g. $l = 20$ and $l = 30$), results obtained were less accurate in terms of error rates and established time parameters were considered as good enough by the domain experts in order to anticipate behaviors of interest.

The network architecture that serves as the basis to model the performance of the reduction gear can be seen in Figure 4.9b. 50 process units (dimensionality of the output space) are considered in the first layer and 20 – 30 in the second layer, which indicates the number of units regarding the scenarios under consideration within this study, S1 and S2, respectively. Different candidate network architectures were tested against validation data (the last 10% of the data set), and the resulting model error (mean absolute error, MAE) was calculated in each trial. All of the networks used in this study were trained with an epoch (number of iterations) of 100, given the fact that from that iteration the loss function (MSE) and the network error (MAE) did not improve. The configuration parameters and the architecture of the model with lowest error were considered as the optimal in each scenario [105].

All details regarding the configuration parameters used in this study are presented in Table 4.7.

Table 4.7: CBM and predictive maintenance in marine propulsion systems. LSTM network configuration parameters.

Parameter	Value
Type of network	<i>StackedLSTM</i>
Number of layers	2
Units in the first hidden layer	50
Units in the second hidden layer	20 – 30
Activation function in hidden layer	<i>sigmoid</i>
Activation function in output layer	<i>linear</i>
Training method	<i>feedforward</i>
Number of iterations	100
Time sequence size used as input values, d	30
Prediction length, l	10

The training process finishes when the error of the model (MAE) for all training data in an epoch is less than 0.01 or when the maximum number of iterations is reached.

Having a prediction length of l , each of the selected d dimensions of $\hat{y}_t \in \hat{\mathbf{y}}$ for $l < t \leq n-l$ is predicted l times. The sequences of residuals are computed for every $\hat{y}_i$ as $\mathbf{r}_i^t = (r_{i,11}^t, \dots, r_{i,1l}^t, \dots, r_{i,d1}^t, \dots, r_{i,dl}^t)$, being $r_{i,kj}^t$ the difference between $\hat{y}_t$ and the predicted value at time $t-j$, h_t , and $k = 1, \dots, d$. Then the log likelihood of each sequence is calculated as it is shown in Equation 4.10.

$$p_i = f(\bar{\mathbf{r}}_i | \mu(\log \mathbf{r}), \sigma(\log \mathbf{r})) = \frac{1}{\bar{\mathbf{r}}_i \sigma(\log \mathbf{r}) \sqrt{2\pi}} \exp \frac{-(\ln \bar{\mathbf{r}}_i - \mu(\log \mathbf{r}))^2}{2\sigma(\log \mathbf{r})^2} \quad (4.10)$$

where $\bar{\mathbf{r}}_i$ is the average value of sequence of residuals $\mathbf{r}_i$ and parameters $\mu(\log \mathbf{r})$ and $\sigma(\log \mathbf{r})$ stand for the mean and standard deviation of $\log \mathbf{r}$, respectively, being $\mathbf{r} = (\bar{\mathbf{r}}_1, \dots, \bar{\mathbf{r}}_n)$ the list of average values of sequences of residuals over time.

Finally, a moving average filter of time window size $d = 30$ is applied to $\mathbf{p} = (p_1, \dots, p_n)$ in order to avoid detecting transient behaviors that concern noise and false alarms. The smoothed score of p_i , denoted by $score_{p_i}$, is computed as follows:

$$score_{p_i} = \frac{\sum_{j=i-d}^{i+d} p_j}{d} \quad (4.11)$$

Figure 4.10 shows the evolution of the log likelihood score over time, with and without applying the moving average smoothing (with $d = 30$). Without smoothing, it can be seen a decrease of the value that could be interpreted as an anomalous behavior. However, the immediate increasing of the score indicates that it is not a real anomaly. Therefore, the smoothed score is an efficient method for restricting this phenomenon.

Figure 4.10: CBM and predictive maintenance in marine propulsion systems. Influence of the smoothing window on the score values.

An anomaly threshold is calculated over the training data as the lowest score value, $\min(score_{\mathbf{p}})$, after smoothing the resulting set of log likelihoods for every sequence of residuals in the training data set, which corresponds to the predictions made by the evolutionary physical model. Then, when checking new data, $\hat{y}_{new}$, if the corresponding score value is below this threshold the sequence will be considered as a potential anomaly to arise in l time steps. Similarly, a threshold to predict changes in the operating condition of the asset is also provided, by computing $\mu(score_{\mathbf{p}}) - 3\sigma(score_{\mathbf{p}})$.

The definition of these thresholds allows performing an intelligent online monitoring of the asset, anticipating possible faulty behaviors and changes in the operating condition of the asset by considering the predictions made by the LSTM network.

Overview of the algorithm

The overall algorithm combines the evolutionary physical modeling of condition monitoring data with the prediction of time sequences given by the LSTM network. The overall algorithm can be seen in 4.2.

Algorithm 4.2 CBM and predictive maintenance in marine propulsion systems. Deep evolutionary modeling for anomaly detection

Inputs: a target feature $\mathbf{y} = (y_1, \dots, y_n)$, a set of operands, $\{sine, tangent, cosine, exp, log, abs, pow, sqrt, *, /, +, -\}$, a set of m input features, $X = \{X_1, \dots, X_m\}$, the time sequence size used as input values d and the prediction length l

```

1: Compute  $f(X) = \hat{\mathbf{y}} \approx \mathbf{y}$  using the evolutionary physical modeling presented in 4.1.2
2: Fit standardize  $\hat{\mathbf{y}}$  using the LSTM network presented in 4.1.2
3: Compute  $\mathbf{p}$  using Equation 4.10
4: Compute  $score_{\mathbf{p}}$  using Equation 4.11
5: anomaly threshold =  $\min(score_{\mathbf{p}})$ 
6: operational threshold =  $\mu(score_{\mathbf{p}}) - 3\sigma(score_{\mathbf{p}})$ 
7: for all  $\hat{\mathbf{y}}_{new}$  do
8:   Compute  $score_{p_{new}}$  using Equation 4.11
9:   if  $score_{p_{new}} < \text{anomaly threshold}$  then
10:     $\hat{\mathbf{y}}_{new} \rightarrow \text{anomaly}$ 
11:   end if
12:   if  $score_{p_{new}} < \text{operational threshold}$  then
13:     $\hat{\mathbf{y}}_{new} \rightarrow \text{operating condition change}$ 
14:   end if
15: end for

```

Besides the interpretability of the obtained physical model and the prediction accuracy given by the deep neural network, one main advantage of this approach is that the target variable to be modelled and predicted, normally corresponding to the control parameter of the asset, can be fully approximated by other operational parameters. Once the modeling phase is completed, in a test bench under laboratory conditions, for instance, the deep evolutionary model can be deployed for online condition monitoring with no need of measuring the asset control parameter.

Experimental results

The proposed Evolutionary Modeling approach has been tested on one year's time operational data, from January to December 2016, of a port side marine vessel reduction gear. A total of 142,702 events are analyzed. An event is collected approximately every minute and it is composed by values of all monitored parameters at that specific time instant. Events in time regions where the engine is not in stable operating conditions are not considered, since they have no significance regarding physical models. Data were used as training (60%), testing (30%) and validation (10%) sets for the modeling step to achieve generalization. Training set was used for learning the evolutionary physical model, whereas testing set was used for testing it. Predictions made by the evolutionary physical model were used for training the LSTM network, which was tested over the validation set. In each case, test data, validation data and the estimations made by generated models regarding each step of the learning process are graphically shown. Finally, the resulting score is provided and the capability of the approach to anticipate potential anomalies is demonstrated.

The overall performance of the method is quantified by computing the mean absolute error, $MAE = \frac{1}{n} \sum_{i=1}^n abs(y_i - \hat{y}_i)$, mean squared error, $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$, and coefficient of determination, $R^2 = 1 - (\sum_{i=1}^n (y_i - \hat{y}_i)^2 / \sum_{i=1}^n (y_i - \bar{y})^2)$, being $\bar{y}$ the average value of the observed data, $\mathbf{y} = (y_1, \dots, y_n)$, and $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_n)$ the predicted values for each $i = 1, \dots, n$ by the model. In

the case of the LSTM network predictions, the produced sequences of size l are compared to the estimations made by the evolutionary model learned and the validation data during training and testing phases, respectively.

Scenario S1: diesel engine in stable operation

The details of the experimental parameters including range, mean (μ) and standard deviation (σ) used in this study with diesel engine in stable operation are presented in Table 4.8.

Table 4.8: CBM and predictive maintenance in marine propulsion systems. Range of values of reduction gear parameters used in this study regarding scenario S1.

Target variable	Range	$\mu \pm \sigma$	Input variable	Range	$\mu \pm \sigma$
y_{S1}	17-63°C	54.61 ± 4.17	$x_{1,S1}$	22-59°C	53.91 ± 2.27
			$x_{2,S1}$	17-46°C	43.13 ± 1.11

Formula obtained regarding the engine pinion bearing temperature in scenario S1 can be seen in Equation 4.12.

$$y_{S1} = \cos(|x_{2,S1}|) + |x_{2,S1}| - x_{1,S1} - 43.0 \quad (4.12)$$

where y_{S1} is the engine pinion bearing temperature, $x_{1,S1}$ is the 1st reduction pinion axial bearing temperature and $x_{2,S1}$ is the 1st reduction bull gear temperature.

Figure 4.11: CBM and predictive maintenance in marine propulsion systems. Illustration of (a) evolutionary physical modeling predictions and (b) LSTM predictions on engine pinion bearing temperature test and validation data, respectively.

The prediction undertaken by the evolutionary model learned, together with sensor readings for the same input test data are shown in Figure 4.11a. Then the LSTM network is trained over evolutionary physical modeling predictions. The resulting LSTM network estimations and the real engine pinion bearing temperature values on validation data set can be seen in Figure 4.11b.

The statistical performance of the method regarding each learning phase of the modeling process in scenario S1 and with relation to training, testing and validation data sets is given in Table 4.9.

Table 4.9: CBM and predictive maintenance in marine propulsion systems. Results of the deep evolutionary modeling for engine pinion bearing temperature training, testing and validation sets with diesel engine in stable operation.

Parameter	Data used	Learning stage	MAE	MSE	R^2
y_{S1}	Training set	Evolutionary physical modeling (training)	0.935	1.385	0.917
	Testing set	Evolutionary physical modeling (testing)	0.804	0.898	0.938
	$f(X_{S1}) = \hat{y}_{S1} \approx y_{S1}$	LSTM network (training)	0.723	0.908	0.931
	Validation set	LSTM network (testing)	0.573	0.759	0.917

Figure 4.12: CBM and predictive maintenance in marine propulsion systems. The real sensor readings (above) and the resulting score values on engine pinion bearing temperature (below) over the validation set.

The training results proved that the proposed evolutionary bearing model in scenario S1 obtained good accuracy ($R^2_{Evol-train} = 0.917$), and low error rates when fitting the training data ($MAE_{Evol-train} = 0.935$ and $MSE_{Evol-train} = 1.385$). A high generalization can be also observed when comparing the evolutionary model predictions with the experimental data used for the test stage ($R^2_{Evol-test} = 0.938$), and once again rather low error rates ($MAE_{Evol-test} = 0.804$ and $MSE_{Evol-test} = 0.898$). LSTM network also performed quite satisfactorily, giving $R^2_{LSTM-train} = 0.931$ and $R^2_{LSTM-test} = 0.917$, and keeping low error rates when approximating the predictions made by the evolutionary bearing model and predicting the validation data values ($MAE_{LSTM-train} = 0.723$, $MSE_{LSTM-train} = 0.908$ and $MAE_{LSTM-test} = 0.573$, $MSE_{LSTM-test} = 0.759$, respectively). Error rates decreased over time, which could be due to the fact that data set used for

testing and validation presented a more stable behavior regarding the propulsion system operating conditions.

The anomaly threshold computed over the smoothed log likelihoods of the residuals, r_{S1} , and the resulting score over the validation data are presented in Figure 4.12. The real sensor readings are also shown in the same figure, above. The sudden drop in engine pinion bearing temperature at the end of the time series is successfully anticipated by the resulting score, which is significantly lower than the anomaly threshold at time $a_{t,S1} - l$. In this case the problem could have been caused by a faulty sensor. Additionally, several operational changes are predicted at times $b_{t,S2} - l$, $t = (1, \dots, 9)$.

Scenario S2: gas turbine in stable operation

The details of the experimental parameters including range, mean (μ) and standard deviation (σ) used in this study with gas turbine in stable operation can be seen in Table 4.10.

Table 4.10: CBM and predictive maintenance in marine propulsion systems. Range of values of reduction gear parameters used in this study regarding scenario S2.

Target variable	Range	$\mu \pm \sigma$	Input variable	Range	$\mu \pm \sigma$
y_{S2}	33-80°C	60.42 $\pm$ 6.56	$x_{1,S2}$	30-48°C	43.53 $\pm$ 1.75
			$x_{2,S2}$	34-70°C	57.78 $\pm$ 5.60
			$x_{3,S2}$	36-74°C	56.34 $\pm$ 5.12

Formula obtained regarding the gas turbine thrust bearing temperature in scenario S2 is presented in Equation 4.13.

$$y_{S2} = \log x_{1,S2} + 2x_{2,S2} - \frac{x_{3,S2}}{\log(2x_{1,S2})} \quad (4.13)$$

where y_{S2} is the gas turbine thrust bearing temperature, $x_{1,S2}$ is the main back thrust bearing temperature, $x_{2,S2}$ is the engine pinion bearing temperature and $x_{3,S2}$ is the 1st reduction pinion bearing temperature.

Figure 4.13: CBM and predictive maintenance in marine propulsion systems. Illustration of (a) evolutionary physical modeling predictions and (b) LSTM predictions on gas turbine thrust bearing temperature test and validation data, respectively.

The values predicted by the formula given by the evolutionary physical modeling process are presented in Figure 4.13a. The real gas turbine thrust bearing temperature values are also provided. The LSTM network predictions made with relation to scenario S2, in addition to the real engine pinion bearing temperature values on validation data set, are shown in Figure 4.13b. As in the previous case, the resulting LSTM network was trained using the values obtained by the evolutionary physical model, $f(X_{S2}) = \hat{y}_{S2} \approx y_{S2}$.

The results achieved by the proposed approach with relation to each learning phase of the modeling process in scenario S2, in terms of error rates and R^2 , are shown in Table 4.11.

Table 4.11: CBM and predictive maintenance in marine propulsion systems. Results of the deep evolutionary modeling for gas turbine thrust bearing temperature training, testing and validation sets with gas turbine in stable operation.

Parameter	Data used	Learning stage	MAE	MSE	R^2
y_{S2}	Training set	Evolutionary physical modeling (training)	0.97	1.717	0.949
	Testing set	Evolutionary physical modeling (testing)	1.01	1.731	0.959
	$f(X_{S2}) = \hat{y}_{S2} \approx y_{S2}$	LSTM-network (training)	1.44	6.690	0.843
	Validation set	LSTM network (testing)	1.075	4.354	0.88

Figure 4.14: CBM and predictive maintenance in marine propulsion systems. The real sensor readings (above) and the resulting score values on gas turbine thrust bearing temperature (below) over the validation set.

As it was found with relation to the bearing model in scenario S1 when the diesel engine was working in stable operation, the training results regarding scenario S2 also proved that the

proposed deep evolutionary bearing model obtained good accuracy ($R_{Evol-train}^2 = 0.949$ and $R_{LSTM-train}^2 = 0.843$), and low error values ($MAE_{Evol-train} = 0.97$, $MSE_{Evol-train} = 1.717$ and $MAE_{LSTM-train} = 1.44$, $MSE_{LSTM-train} = 6.690$) when fitting the training data. A high generalization can be also observed when comparing the evolutionary model predictions with the experimental data used for the test stage ($R_{Evol-test}^2 = 0.959$), and once again rather low error values ($MAE_{Evol-test} = 1.01$ and $MSE_{Evol-test} = 1.731$). LSTM-network was also quite accurate when predicting the sequences of data on validation set, giving $R_{LSTM-test}^2 = 0.88$, $MAE_{LSTM-test} = 1.075$ and $MSE_{LSTM-test} = 4.354$. However, a slight decrease on performance can be appreciated when fitting the deep network to the evolutionary physical modeling values, also producing worse results than those obtained in scenario S1. This can be due to the more variable nature of the control parameter under study, the gas turbine thrust bearing temperature.

The resulting score over the validation set, along with the anomaly threshold computed as the min of the smoothed likelihood of the differences between the predictions made by the evolutionary physical model learned and the LSTM network estimations, are shown in Figure 4.14. The real gas turbine thrust bearing temperature values over the validation set are also provided, above. No anomalous sequences are detected. However, heavy changes in the computed score over time can be appreciated. They mainly correspond to changes in the operating behavior of the propulsion system, as it can be appreciated in the sensor readings. They can be also anticipated by the proposed approach, namely at times $b_{t,S2} - l$, $t = (1, \dots, 7)$, by means of the operational threshold calculated to predict changes in the operating conditions of the asset. The corresponding warnings can be used for obtaining higher efficiency, optimizing fuel consumption and reducing emissions.

4.2 ML methods for health status assessment and pattern classification

4.2.1 A case study on marine diesel engines

Motivation

The recent development of new technologies for intelligent control and monitoring, and the advances in computational capacity of inspection devices and data processing methods, are highly improving the life cycle management and maintenance of industrial systems [106] [107] [108]. Depending on the nature of the machine and the corresponding monitoring data to be processed, which is determined by the application domain and by the particular problem to be solved, these techniques have demonstrated a good accuracy when accomplishing fault detection tasks [109] [110] [111].

Failure modes analysis is usually performed to identify most probable causes of detected abnormal behaviors in order to avoid them in the future [112]. The core of the whole process is executed on the basis of ML-based methods [113]. They are employed to learn models from historical, monitoring data (e.g. operational parameters, contextual information and maintenance operations performed), which are then applied to detect faults by checking real-time data in an online fashion [114]. The problem is that in many industrial scenarios the limited information of real faults makes it specially challenging to obtain accurate fault prediction models [115].

In this work this issue is addressed by proposing a novel ML approach. The modeling process is performed automatically and in a fully unsupervised fashion. A nonparametric density estimation technique is first applied to remove outliers from data [29][116], avoiding prior assumptions about their distribution. Then, due to their strong regularisation and generalisation properties, and its good accuracy and flexibility, SVM technique is used as the baseline of the proposed algorithm to model normality [117]. Normality models learned for every involved subsystem are finally combined to generate real-time system health scores. The obtained models are deployed in a control monitoring platform in order to accurately assess the health status of the system. Therefore, a reliable method to predict critical faults is obtained overcoming the curse of normality modeling from data without clear evidences of real faults.

In order to illustrate the usefulness and benefits of the proposed approach, the normal behavior of a complex industrial system is statistically modeled from monitoring data during real operation: an auxiliary marine diesel engine. Models learned are tested to analyse the propagation of a critical fault over time in a set of subsystems of the marine diesel engine, demonstrating the validity and the improvement achieved through the use of the proposed approach. This scenario is of special interest, due to the fact that conservative strategies are commonly adopted and limited information of real faults is available, as it was mentioned in previous section (see Section 4.1).

Proposed approach

The algorithm relies on the assumption that all normal samples share some common properties that differ from the outliers, which can have very different properties without any commonness. Therefore, they may imply abnormal behaviors. The steps for the proposed automated health status assessment are as follows:

1. Remove outliers from data iteratively using kernel density estimation, until the convergence criteria is met.

2. Estimate the optimal bandwidth for resulting normal data.
3. Given the parameters ν and σ , model normality from data using ν -SVM.
4. Compute the health score of the monitoring system, based on the computed decision function, and on the log-normalization of the distance of a given set of values of all monitored features at a specific time instant to the separating hyperplane:
 - if the health score of the given set of values is below the normality threshold, it implies a faulty behavior.
 - otherwise, it corresponds to a normal behavior.

The details for the steps of the algorithm are given in the next subsections.

Kernel density-based outliers detection

Multivariate Kernel Density Estimation (KDE) is a nonparametric technique that allows estimating the density of the data [118], [119]. Probability density functions (pdf) are inferred in order to establish the underlying density function and the overall structure of the data. Given a set of m features, $X = \{X_1, \dots, X_m\}$ where each feature X_i can take a value from its own set of possible values χ_i , and n feature vectors or instances, $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, with $i = 1, \dots, n$, the multivariate joint pdf $\hat{f}_h(X)$ can be computed as follows:

$$\hat{f}_h(X) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_1 \dots h_m} K \left(\frac{X_1 - x_{i1}}{h_1}, \dots, \frac{X_m - x_{im}}{h_m} \right) \quad (4.14)$$

being $\mathbf{h} = (h_1 \dots h_m)^T$ the vector of bandwidths calculated by the rule of thumb using Scotts Rule [120] [121] and G is the multivariate Gaussian kernel function [122] operating on the extracted features:

$$G(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2} \quad (4.15)$$

Assuming that the majority of the data represent a normal behavior, instances that are isolated on feature space also show a density lower than those of the normality regions. An iterative process that removes instances with low density is defined, until the following convergence criteria is met: $\min(\mathbf{d}_{i-1}) \geq \min(\mathbf{d}_i)$, with $\mathbf{d}_{i-1} = (d_1, \dots, d_j)$ and $\mathbf{d}_i = (d_1, \dots, d_{<j})$ the density computed for every instance at iterations $i - 1$ and i , removing the outliers found, which corresponds to the feature vectors with the minimum density at each iteration. Once the process is finished, normality is modeled by means of the ν -SVM.

An example of the use of the kernel density-based outliers detection process can be seen in figures 4.15 and 4.16.

Figure 4.15: Health status assessment and pattern classification in marine diesel engines. Illustration of a data set with outliers.

Figure 4.16: Health status assessment and pattern classification in marine diesel engines. Illustration of the resulting data set after applying the kernel density-based outliers detection process.

Bandwidth selection

In order to build-up valid normality models, parameters ν and σ for bandwidth selection must be carefully chosen. ν controls the false positive rate and σ is the Gaussian kernel parameter and

can dramatically affect the performance of the ν -SVM, since it is used to compute the parameter $\gamma = 1/2\sigma^2$. It is a measure of how well the model generalizes to unseen data.

Figure 4.17: Health status assessment and pattern classification in marine diesel engines. Illustration of the ν -SVM decision boundary using $\sigma = 3$.

Figure 4.18: Health status assessment and pattern classification in marine diesel engines. Illustration of the ν -SVM decision boundary using $\sigma = 0.5$.

By varying the scale parameter σ , the ν -SVM can determine multiple regions of support for a

dataset. This allows modeling multimodal distributions, as it can be seen in figures 4.17 and 4.18. Note how using a smaller value for σ leads to a tighter decision boundary. For anomaly detection and normality modeling, the optimal selection of σ implies a significant reduction in the number of false alarms.

In the proposed method, σ is calculated on the basis of the false positive rate, ν , by a training-error based approach as established by [123]. ν is often set to very small values based on the low rate of instances assumed to be outliers (transient data, sensor errors, etc.).

Aiming to optimally fitting this parameter to each data set, $\nu = 1/\max(d_{ij})^2$ is calculated, with $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{i=1}^n (\mathbf{x}_i - \mathbf{x}_j)^2}$ being the Euclidean distance between instances $\mathbf{x}_i$ and $\mathbf{x}_j$ in the feature space [124], after removing instances with low density. For a range of potential values of σ , the fraction of the training data that is classified as an outlier is evaluated. Because ν is a theoretical upper bound on this fraction, the lowest value of σ that gives a classification error equal to ν is selected as the optimal.

The algorithm for optimally fitting σ to each data set X is described in Algorithm 4.3.

Algorithm 4.3 Health status assessment and pattern classification in marine diesel engines. Automatic σ selection in ν -SVM

Input: a set of m features, $X = \{X_1, \dots, X_m\}$ and n feature vectors or instances, $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, with $i = 1, \dots, n$

- 1: $\nu = \frac{1}{\max(d_{ij})^2}$ with $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$
 - 2: **for all** $\sigma \in (1, \dots, l)$ **do**
 - 3: Find σ such that $|f(X) \leq 0| \geq \nu$, being $|f(X) \leq 0|$ the number of instances classified as anomalies by the ν -SVM model (see Equation 4.19)
 - 4: **end for**
-

ν -SVM normality modeling

SVM for novelty detection is an unsupervised approach proposed by Schölkopf et al. that finds a normal region containing most of data samples and the anomalies elsewhere. The technique separates all the training data samples from the origin with maximum margin. The resulting function, $f : R^d \rightarrow \{-1, +1\}$, finds a hyperplane that separates the positive or normal instances, denoted as $\{+1\}$, from outliers or negative instances, denoted as $\{-1\}$. The learning process of such function solves the following minimization problem (see Equation 4.16):

$$\min_{\omega, \xi_i, \rho} \frac{1}{2} \|\omega\|^2 + \frac{1}{\nu n} \sum_{i=1}^n \xi_i - \rho \quad (4.16)$$

subject to $(\omega \cdot \phi(\mathbf{x}_i) \geq \rho - \xi_i)$, and $\xi_i \geq 0$.

A hyperplane characterized by ω and ρ that has maximal distance from the origin in feature space, X , and separates all the data points from the origin is then created.

Parameter ν sets an upper bound on the fraction of outliers and a lower bound on the number of training samples, and also controls the balance between ξ_i , which is the training error, and ω , which is the margin. It has a similar function to C for supervised SVM. In many cases the decision boundary is non-linear in the input space and, therefore, a non-linear kernel function is employed to fit the hyperplane in a transformed high-dimensional feature space [125]. The problem can be transformed to the dual form by using Lagrange multipliers, α_i , and quadratic programming, as can be seen in Equation 4.17.

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha \quad (4.17)$$

subject to $0 \leq \alpha_i \leq 1/(\nu n)$, $i = \{1, \dots, n\}$, $e^T \alpha = 1$ and being Q_{ij} the Gaussian kernel, computed in this case as it can be seen in Equation 4.18.

$$G(X_i, X_j) = \exp(-\gamma \|X_i - X_j\|^2) \quad (4.18)$$

with $\gamma = \frac{1}{2\sigma^2}$ and σ the Gaussian parameter, set in previous step.

When the minimisation problem is satisfied, the minimum enclosing hypersphere is found. Then the decision function f is used to determine whether or not a feature vector, $\mathbf{x}_i$, lies within or outside the hypersphere (see Equation 4.19).

$$f(\mathbf{x}_i) = \text{sign}((\omega \cdot \phi(\mathbf{x}_i) - \rho)) \quad (4.19)$$

Instances considered as anomalies will produce negative values. Therefore the decision rule specified in Equation 4.20 can be easily applied to a new feature vector $\mathbf{x}_{new}$, in such a way that when $\sum_{i=1}^n \alpha_i K(X, \mathbf{x}_{new}) - \rho \leq 0$ holds true, $\mathbf{x}_{new}$ is labelled as an anomaly.

$$\mathbf{x}_{new} = \begin{cases} normal & \text{if } \left(\sum_{i=1}^n \alpha_i G(X, \mathbf{x}_{new}) - \rho \right) \geq 0 \\ anomaly & \text{otherwise} \end{cases} \quad (4.20)$$

Health score computation

Figure 4.19: Health status assessment and pattern classification in marine diesel engines. Example of kernel-based SVM health score computed over time.

A health score is computed to establish the health status of the monitored system as a continuous value. The idea is to early detect deviations from normal behavior when checking a new instance, $\mathbf{x}_{new}$. The health score is obtained by applying the log normalisation to the distance of $\mathbf{x}_{new}$ to the separating hyperplane so that a normalised value between 0 and 1 is obtained, as it is shown in Equation 4.21.

$$hs(\mathbf{x}_{new}) = \frac{1}{1 + e^{-\beta}} \quad (4.21)$$

where $\beta = \left(\sum_{i=1}^n \alpha_i K(X, \mathbf{x}_{new}) - \rho \right)$.

The normality threshold is then set to 0.5. The closer the health score is to 0, the farther the system is to the normality. An example of resulting health score over time can be seen in Figure 4.19.

The overall algorithm can be seen in Algorithm 4.4.

Algorithm 4.4 Health status assessment and pattern classification in marine diesel engines. Kernel-based SVM algorithm for health status estimation

Input: a set of m features, $X = \{X_1, \dots, X_m\}$ and n feature vectors or instances, $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, with $i = 1, \dots, n$

- 1: Compute $density = \hat{f}_h(X)$ using Equation 4.14
 - 2: Set $min_d = \max(density)$
 - 3: $outliers = \{\}$
 - 4: **while** $\min(density) < min_d$ **do**
 - 5: $min_d = \min(density)$
 - 6: $\mathbf{x}_{min_d} \rightarrow outliers$
 - 7: Remove $\mathbf{x}_{min_d}$ from X
 - 8: Compute $density = \hat{f}_h(X_{|<n|})$ using Equation 4.14
 - 9: **end while**
 - 10: Compute σ applying Algorithm 4.3
 - 11: Obtain the hyperplane characterized by ω and ρ that has maximal distance from the origin in X using Equation 4.16
 - 12: Calculate $f(\mathbf{x}_{new})$ using Equation 4.19 and $\gamma = \frac{1}{2\sigma^2}$
 - 13: Compute $hs(\mathbf{x}_{new})$ using Equation 4.21
 - 14: **if** $hs(\mathbf{x}_{new}) \leq 0.5$ **then**
 - 15: $\mathbf{x}_{new} \rightarrow anomaly$
 - 16: **end if**
-

Having several interrelated monitored systems, health scores given by the corresponding normality models can be then combined by applying a weighting based on criticality level associated to each subsystem. Finally, an overall system health status at a specific time instant, t , is obtained by computing Equation 4.22.

$$score_t = \sum_s hs(\mathbf{x}_t^s) * c_s \quad (4.22)$$

subject to $\sum_s c_s = 1$ and being $\mathbf{x}_t^s$ the new instance feature values related to subsystem s and c_s the subsystem s criticality.

Experimental results

The auxiliary marine diesel engine under study was monitored from January 2015 to February 2016. An event is collected every minute and it is composed by values of all monitored features (e.g.

temperatures, intensity, pressure, power or voltage) at a specific time instant. After the normality models are fitted to the training data they are tested on unseen test data. Training set covers the data corresponding to the whole 2015 (323,437 samples in total). Test set comprises two months' time series data, from January to February 2016 (17,267 samples in total). It is known that a seawater circulation pump fault arose at the end of February 2016 (in last 10 to 18 events of the test set). Events in time regions in which the engine is not working are not considered, since they are meaningless for discovering critical failures. Therefore, training set finally consists of 88,409 events and test set consist of 16,948 events. Five different engine subsystems (Charge air, Combustion, Alternator, Cooling and Lubricating) and their corresponding scenarios are considered, given the fault propagation.

Proposed kernel-based SVM models of marine diesel engine subsystems were compared with kNN method [126] [127]. It is largely used for classification and regression, predicting the label or the value of a new data sample as the majority of labels or values among its k closest Neighbors. In this case regression-based normality models are learned, but following the same learning schema. The engine load, used as reference value (primary input), and the other subsystem parameters are considered as model inputs. The seawater temperature, which is the most influential environmental parameter, is the output or target variable to be estimated.

A different classifier is trained for each subsystem, hence obtaining several classifiers independent among them. Localisation of the anomaly and its propagation is done as follows: the subsystems where the highest percentage of test points are farther to the normal regions are the most likely to be affected by the anomaly. Results are expressed by the fraction of test points classified as anomaly at each subsystem, for the considered test scenario. Finally, a global health score is defined straightforward: every single health score is combined by performing a weighting based on the criticality level associated to each subsystem. Results achieved using the proposed methods and their ability to anticipate potential faults are discussed.

Regarding the kernel-based SVM approach and for every subsystem, the multivariate joint pdf $\hat{f}_h(X)$ introduced in Equation 4.14 becomes:

$$\hat{f}_h = \frac{1}{16948} \sum_{i=1}^{16948} \frac{1}{h_1} \frac{1}{h_2} \frac{1}{h_3} k \left(\frac{X_1 - x_{i1}}{h_1}, \frac{X_2 - x_{i2}}{h_2}, \frac{X_3 - x_{i3}}{h_3} \right)$$

being h_1 , h_2 and h_3 the bandwidths for the corresponding subsystem features, X_1 , X_2 and X_3 , and x_{i1} , x_{i2} and x_{i3} the values at the i th event.

After having removed low-density events from training data, σ is calculated for each subsystem as it was presented in Algorithm 4.3. The range of σ values tested was (1, ..., 100). Similarly, in the case of k the range of values tested was (1, ..., 100). Representative plots of these processes are shown in Figures 4.20 and 4.21.

Normality models generated for every subsystem are then evaluated using test data, which included the real failure symptoms related to the seawater circulation pump problem. In Tables 4.12 and 4.13 the accuracies obtained by the proposed approaches are provided.

The false positive rate achieved by the kernel-based SVM detection is low for all normality models when healthy data is considered. For the major scenarios with the propagation of the seawater circulation problem, it is clear that the anomaly detection approach is functioning as an early warning of potential damage in the system, as soon as the very first symptoms appear in the charge air subsystem. Only combustion and alternator normality models obtained a lower rate of true positives. Anomaly localisation is not so obvious in that cases.

Figure 4.20: Health status assessment and pattern classification in marine diesel engines. Kernel-based SVM normality modeling. σ selection based on training-error.

Figure 4.21: Health status assessment and pattern classification in marine diesel engines. kNN-based normality modeling. k selection based on 10-fold CV error.

Table 4.12: Health status assessment and pattern classification in marine diesel engines. Results of kernel-based SVM classification. Affected systems and fraction of faults detected by each normality model.

Model	No faults	System fault				
		Charge air	Combustion	Alternator	Cooling	Lubricating
Charge air	0.036	0.721	-	-	-	-
Combustion	0.000	-	0.267	-	-	-
Alternator	0.006	-	-	0.537	-	-
Cooling	0.043	-	-	-	1.000	-
Lubricating	0.045	-	-	-	-	1.000

Table 4.13: Health status assessment and pattern classification in marine diesel engines. Results of kNN regression-based classification. Affected systems and fraction of faults detected by each normality model.

Model	No faults	System fault				
		Charge air	Combustion	Alternator	Cooling	Lubricating
Charge air	0.007	0.556	-	-	-	-
Combustion	0.001	-	0.667	-	-	-
Alternator	0.000	-	-	1.000	-	-
Cooling	0.003	-	-	-	0.832	-
Lubricating	0.005	-	-	-	-	1.000

For kNN regression-based detection, the false positive rate is also significantly low. The critical failure of the lubricating system is detected extremely reliably, but with less anticipation accuracy at the charge air stage. Results are largely similar to those obtained by kernel-based SVM detection. The faulty condition in the charge air and combustion subsystems is detected less reliably. Given that the test case scenarios showing the propagation of the anomaly were deliberately chosen to be difficult to detect, it is still encouraging that the classification of faulty events rises above the false positive rate.

Differences in performance achieved by both approaches could be contributed by the joint data distribution of the data sets under study. Both kernel-based SVM and kNN regression are nonparametric methods but, although Gaussian kernel used for ν -SVM also estimates the density of probability of the training data, the method is based on the support vectors or data samples that lie closest to the separating hyperplane. kNN regression is completely based on local distances, fixing the number of similar samples, k , and determining the local region that contains those samples. Having identified a target variable for every subsystem and learning a regressor that best fits the normal data, a local neighbourhood-based outlier detection approach can be adopted. Notice that it is strongly dependent on the number of Neighbors and on how locally distributed is the system failure.

A global health score is then calculated taking as input parameter the score given by each model and for all samples in the test set, $\mathbf{x}_t$. The subsystems health scores are weighted as follows, based

on their criticality: $score_t = hs(x_t^{s_1}) * 0.1 + hs(x_t^{s_2}) * 0.1 + hs(x_t^{s_3}) * 0.2 + hs(x_t^{s_4}) * 0.3 + hs(x_t^{s_5}) * 0.3$; being s_1 the charge air system, s_2 the combustion system, s_3 the alternator system, s_4 the cooling system and s_5 the lubricating system. The health scores are weighted on the basis of the criticality analysis of the subsystems under study, and also considering the recommendations made by the domain experts.

Time filters are applied to detect when a symptom occurred in isolation, with no continuity over time. They represent less than 0.2% and 0.02% of the test set in the kernel-based SVM approach and kNN regression-based approach, respectively. Therefore, and since they are meaningless, they are filtered out. They are due to operational failures that occur when starting pump operation while engine warms up with the gate valve closed. Nevertheless, in some cases they might also imply evidences of minor symptoms related to other failures that could arise in the near future. Additionally, and bearing in mind the conservative policies usually adopted by the naval sector, focused on reliability, this issue should be carefully addressed with the support of domain experts.

Table 4.14: Health status assessment and pattern classification in marine diesel engines. Results of health score classification.

Health score	No faults	System fault				
		Charge air	Combustion	Alternator	Cooling	Lubricating
Kernel-based SVM	0.000	0.667	0.800	0.922	1.000	1.000
kNN	0.000	0.556	0.667	0.768	0.832	1.000

The seawater circulation pump failure is clearly identified in the last phases of its propagation over the engine subsystems by both methods, as it can be seen in Table 4.14. The results demonstrate good performance and accuracy, which were generally achieved through the proposed normality learning framework that performs an iterative outliers removal process incrementally. However, a lower predictability was obtained in the case of kNN estimations in terms of an early detection of the degradation of the computed health score. This is due to the decision function defined in kernel-based SVM approach, which considers the distance of the samples to the optimal hyperplane that separates the normal region. kNN method obtains the health score without requiring the training of support vectors. The computational cost is therefore lower than the kernel-based SVM, but the identification of a target variable is required.

4.2.2 A case study on bridges: The Sydney Harbour Bridge

Motivation

Ageing and damage effects in transport infrastructures, such as roads, bridges and tunnels, are becoming a big issue nowadays. In order to improve safety and reduce costs derived from ageing and damage effects, several technological challenges must be carefully considered. The most important are those related to damage assessment methods and decision making procedures, employing advanced monitoring methods and novel resilient materials. Such challenges should support an extension of the useful life-time of aged infrastructures. SHM-based approaches have been increasingly used during recent years to address this problem [128] [129].

With regard to bridges, and despite the advances in abstract analysis and controlled testing,

failures have the most conspicuous influence on their design, construction and management. Many failures are mainly caused by inappropriate design and poor maintenance (corrosion, scour, etc.) [130]. For this reason maintenance should be considered as a fundamental pillar to face ageing and damage effects in bridges [131]. Maintenance strategies are essentially based on nonintrusive sensing, monitoring, and analysis techniques to provide flexible decision support, normally in the form of inspection recommendations (when, where and why to act). To this concern, data-driven or ML-based analysis [132]) and model-driven or finite element analysis [133]) approaches are most commonly used to generate data (from an analytical perspective) and physical (from a numerical or mechanical perspective) models that aims to represent the structure. Generated models are then used to identify damage in the modeled structures based on resulting health scores when checking new events [134].

This work is an effort to contribute to the SHM applied to the Sydney Harbour Bridge, one of iconic structures in Australia, by a ML approach. It presents a clustering-based methodology to group joints bridge with similar behavior and then detect abnormal or damaged ones. Vibration events caused by passing vehicles are acquired by accelerometers located in several joints of the structure. Since there are not clear evidences of anomalies, events are analysed in an unsupervised fashion. A combination of feature extraction and outlier removal is performed, and then similar events and joints are grouped by the clustering technique. This will allow isolating possible damaged joints and finding evidences of cracking.

Proposed approach

The proposed algorithm is based on an unsupervised classification of vibration events and joints. The main steps of the methodology can be seen in Figure 4.22, which has been applied for the presented work on the Sydney Harbour Bridge.

Figure 4.22: Health status assessment and pattern classification in bridges. Flowchart of proposed clustering based approach for damage detection.

The vibration responses of the structure excited by passing vehicles, also named as events, are measured from different joints within the structure using the installed accelerometers. The measured time histories of vibration responses, corresponding to each event, are stored for further analysis. For every event, raw acceleration data are transformed into a unique feature in time domain. Outliers are then removed based on density estimation of bridge joints and the energy of the signals, allowing reducing dimension of the training set. Having removed the outliers, the remaining events are then transformed into the frequency domain using the Fast Fourier Transform (FFT). Finally, clustering techniques are employed to train models that are able to characterize

behaviors of interest from data, focusing on normality. Thus, events and joints classification for damage detection can be automatically performed.

For online damage detection, previously generated models from historical data can be used. When new excitation events are obtained, feature extraction and signal processing steps are applied as explained above. Then, in the case of event-based classification, distances to behavior models are computed and similarities are established. Events far from the normality can be further studied and compared to the models representing any abnormal behavior, e.g. cracking, sensor failure, etc. Instant warning can be also performed by defining membership rules based upon patterns of event-based models. Regarding joint-based classification, joint-based models are updated so that any change in the behavior of the structure over time can be determined. Therefore, the overall status of the structure can be estimated and any deviation from the normality can be detected.

Data preprocessing for feature extraction

An event is formally defined as a time period in which a vehicle is driving across an instrumented joint. Vibrations caused by passing vehicles are recorded by tri-axial accelerometers positioned at the joints located in different parts of the structure. For every event, accelerometer data in the three axis (x, y, z) are transformed in a unique feature, $V = |A_i| - |A_r|$, where A_i is the instantaneous acceleration at i th sample and A_r is the rest vector or the average of the three readings (x, y, z) in the first 100 samples. They are collected before the event is triggered for ensuring that all the events have the same wave form. Events that are triggered within the first 100 samples are filtered out since they can lead to misleading conclusions when detecting outliers and characterizing damage. These events are caused by vehicles that were driving close to each other. Data standardisation is then applied to resulting set of events (see Equation 4.23).

$$X = \frac{w - \mu(W)}{\sigma(W)} \quad (4.23)$$

where $W = (w_1, \dots, w_m)$ is the data set containing m events, being each event $\mathbf{w}_i = (v_1, \dots, v_n)$, with v_t a value of feature V at time t .

kNN for outliers removal

kNN-based approach proposed within this study performs an iterative process that allows removing outliers and noisy signals incrementally, until the convergence criteria is met. Outliers are joints signals that are far from their joint representative, calculated as the mean of all joint events. This also allows resampling the number of events for every joint in order to balance the whole data set, and thus preventing a bias towards the majority joint. K Dimensional-Trees (KD-Trees) are used for optimising the k -nearest neighbors searching process [135].

For every signal, $X = (x_1, \dots, x_n)$, the sum of the energy in time domain is calculated, as it can be seen in Equation 4.24.

$$E(X) = \sum_{i=1}^n |x_i|^2 \quad (4.24)$$

Then, for each iteration, the k closest neighbors to the mean of the energy of the joint signals, μ_{joint} , in the KD-Tree are taken. The distance of the k closest signals, $d(E(X_j), \mu_{joint})$, $j = 1, \dots, k$,

for every joint is calculated. If condition in Equation 4.25 is met, X_j is marked as outlier and it is removed from current joint.

$$d(E(X_j), \mu_{joint}) > \mu(\mathbf{d}) + (2 * \sigma(D)) \quad (4.25)$$

where $d(E(X_j), \mu_{joint}) = |E(X_j) - \mu_{joint}|$, $\mu(\mathbf{d}) = (1/k) * \sum_{j=1}^k d(E(X_j), \mu_{joint})$ is the mean of the distance of every joint signal energy to the joint mean and $\sigma(\mathbf{d})$ is the standard deviation of such distances:

$$\sigma(\mathbf{d}) = \sqrt{\frac{\sum_{j=1}^k (d(E(X_j), \mu_{joint}) - \mu(\mathbf{d}))^2}{k}} \quad (4.26)$$

This process is repeated until any of the previously established stopping criteria is met. Namely, a maximum number of iterations and a distance threshold, calculated during the first iteration as it can be seen in Equation 4.27.

$$threshold = \mu(\mathbf{d}) + (0.5 * \sigma(\mathbf{d})) \quad (4.27)$$

If the maximum distance obtained from the k nearest points to the mean of the joint at any iteration is below this threshold, the process is stopped.

Fourier transform for vibration signals processing

Fourier transform is a signal processing technique that decomposes a time domain signal into the series of frequencies (amplitudes and frequencies) that composed the time domain signal. It was first discussed by Joseph Fourier [136], and since then it has been further developed becoming a robust frequency domain method in modal analysis [137]. The basic idea of spectral analysis is to represent the original vibration signal as a new sequence, which determines the importance of each frequency component in the dynamics of the signal. This is achieved by using the discrete version of the Fourier transform as specified in Equation 4.28.

$$X(f) = \sum_{-\infty}^{\infty} x(t) e^{-2\pi i f t} \quad (4.28)$$

where f denotes the frequency at which $X(f)$ is evaluated.

Within proposed approach, FFT is applied. The frequency spectrum of each vibration signal in the time domain, $X = (x_1, \dots, x_n)$, is thus computed as it is shown in Equation 4.29.

$$A(f_x) = \sum_{j=1}^{j=n} x_j \omega_n^{(j-1)(f_x-1)} \quad (4.29)$$

where f_x represents frequency, $|A(f_x)|$ is signal amplitude in frequency domain, x_j is one of the n time domain sampling points of the vibration signal and $\omega_n = e^{\frac{-2\pi i}{n}}$.

This signal analysis technique provides a powerful spectral based diagnostic method in stationary conditions, when there is no transient signals involved.

K-means clustering for behavior characterization

Once the FFT is applied over remaining events after the outlier removal process, Euclidean metric is computed in order to determine the distance between events. It is calculated similarly to Equation 4.3 as $D(X, Y)$, being $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_n)$ vibration signals after applying FFT.

K-means clustering for damage detection allows detecting anomalies, e.g. ageing symptoms, damage and cracking effects, which are usually isolated in small clusters. They can be removed and further analysed. Therefore, remaining data set can be used for training accurate normality models. Classification of new events is then performed by computing their distances to previously learned cluster centroids that represent behaviors of interest.

The required computational time for online warning is considerably small, only requiring calculating the Euclidean distance between a new event and a the cluster representatives, representing behaviors of interest. The combination of off-line learning, which includes the required time for feature extraction, signal processing and the construction of the clustering models with a linear complexity $O(n)$, and online monitoring using the proposed approach can be therefore applied for real-time damage identification.

Experimental results

The Sydney Harbour Bridge was opened in 1932. It is a steel through arch bridge that carries lanes of 8-road traffic and two railway lines. Traffic lane 7 is a dedicated bus and taxi lane on the eastern side of the bridge. Lane 7 consists of an asphalt road surface on a concrete deck supported by concrete and steel jack arches. There are approximately 800 jack arches over a total distance of 1.2 km. The jack arches are difficult to access and are inspected typically at two yearly intervals according to standard visual inspection practices.

Two different test case scenarios are presented. In the first case study, 6 joints from North pylon and North main span were monitored during the first week of August 2012, namely Joints 1 to 6 as shown in Figure 4.23. It is known that a crack was present in the 4th joint at that time. Vibration signals were sampled at a frequency of 375 Hz and during 1.5 seconds, resulting in 600 samples per event. Frequency range was set from 0 to 300 Hz.

Figure 4.23: Health status assessment and pattern classification in bridges. 6 joints experiment, schematic of the evaluated joints.

In a separate case study, 71 joints were monitored from the 1st of October to the 7th of October 2014. Joints monitored belong to the following bridge zones: span 6, span 7, span 8, North pylon and North main span (see Figure 4.24 for locations of these monitored areas). It is known that one of the sensors mounted on joint 135, located in the second bay of span 7, was faulty at that time. Vibration signals were sampled at a frequency of 250 Hz during 2 seconds, resulting in 500 samples per event. Frequency range is set from 0 to 250 Hz.

Figure 4.24: Health status assessment and pattern classification in bridges. The Sydney Harbour Bridge schema.

kNN-based outlier removal process was executed specifying a different value of k for each experiment. This is due to the fact that the number of available events per joint in each case varied significantly. k must be specified according to the least amount of joint events that are being analysed. This allows keeping a balance between joints cardinality, avoiding any of them to dominate the clustering process. In relation with first experiment involving 6 joints, k was set to 5,000. Regarding the 71 joints data set, k was set to 500, since less events per joint were available. At every iteration and for each joint, k is the number of closest events (in terms of mathematical distance) to the mean of the energy of all joint events. In case $|events_i| < k$, k is set to the total number of events in joint i . The maximum number of iterations was set to 10.

Results obtained in terms of the number of events filtered for each experiment are summarized in Table 4.15.

Table 4.15: Health status assessment and pattern classification in bridges. Results obtained by kNN outlier removal process.

	6 joints 36,947 events	71 joints 45,818 events
Number of events for training	28,511	27,407
Number of filtered events	8,436	18,411

Event-based clustering

Event-based clustering is performed aiming to group similar vibration signals, and thus capturing the structural behavior and reducing event variance. Clusters formed are shown as graphs containing the pattern (centroid or mean values) and the variance of grouped events for each FFT value. Clusters distribution is also provided as the percentage of events belonging to each joint that are grouped in a cluster.

6 joints experiment

In this experiment, comprising joints 1, 2 and 3 in North main span and 4, 5 and 6 in North pylon, the main motivation was to characterize normality from joints events.

When there is no previous knowledge about the presence of any kind of abnormal behavior, the goal is to try to isolate outliers from high-density regions, containing the majority of the data. At first with a lower number of clusters and increasing K depending on the percentage of data variance explained and on the number of behaviors to be represented. In this case, notably, with

$K = 2$ normal events of all joints were grouped in a big cluster, containing a total of 23,849 events, whereas 4,662 events related to the damage effect, mostly located in joint 4, were isolated in a small cluster. Clusters formed can be seen in Figure 4.25. Events belonging to the same joint are equally colored.

Figure 4.25: Health status assessment and pattern classification in bridges. Illustration of the 6 joints experiment: centroid and standard deviation of joint events (above) and joints distribution (below). (a) Cluster 0 with events showing a normal behavior and (b) Cluster 1 with events from a damaged joint.

71 joints experiment

A quick test was first conducted only considering 5 joints located in the second bay of span 7. As in previous case K-means was executed with $K = 2$, in order to find a small cluster defining abnormal behaviors in data.

Figure 4.26: Health status assessment and pattern classification in bridges. Illustration of the 71 joints experiment, analysis of 5 joints located in the second bay of span 7: centroid and standard deviation of joint events (above) and joints distribution (below). (a) Cluster 0 with events showing a normal behavior and (b) Cluster 1 with events from a faulty sensor.

As it can be seen in Figure 4.26, several events from joint 135 were located in a small cluster (together with a few events of joint 131, located in the same bay and span), showing an abnormal

pattern characterized by centroid values. The other cluster grouped the majority of the events, representing a normal behavior.

Within this same experiment, and examining all available joints, K-means clustering was executed with $K = 5$. The goal was to group similar behaviors in joints located in similar relative positions along the bridge, and 5 different bridge areas were involved. They are the following:

- Span 6: 6 bays, 33 joints.
- Span 7: 5 bays, 21 joints.
- Span 8: 4 bays, 12 joints.
- Span North pylon: 2 bays, 7 joints.
- Span North main span: 1 joint.

Joints belonging to the same bay are equally colored. From resulting clusters, as it can be seen in Figure 4.27 for the case of four clusters, no useful information can be acquired given the fact that events from one joint can appear in several clusters. It is therefore not straightforward to group the same joint together using event-based clustering.

Figure 4.27: Health status assessment and pattern classification in bridges. 71 joints experiment, Cluster 4: centroid and standard deviation of joint events (above) and joints distribution (below).

Joint-based clustering

To overcome the weakness of event-based clustering mentioned above, joint-based clustering was utilized. In joint-based clustering, a map of pairwise distances among representatives of all joints

was generated. A joint representative is calculated as the mean values of all events of each joint, after outliers removal phase. As in the case of event-based clustering, the distance metric used was Euclidean distance. The distances obtained among joints representatives were characterized as a map of pairwise distances to easily interpret results. Moreover, correlations between similarly located joints in the different bridge parts under consideration can be discovered.

6 joints experiment

For the experiment related to the analysis of 6 joints, it can be appreciated that the distance between the representative of joint 4 and the others is significantly high (see Figure 4.28). The more red the colour the higher the dissimilarity between joints representatives. Therefore, the damage effect in this joint is detected.

Figure 4.28: Health status assessment and pattern classification in bridges. 6 joints, a known damage in joint 4: map of pairwise distances.

71 joints experiment

Regarding the analysis of 71 joints, since only spans 6 and 7 had sufficient instrumented joints covering most of the span, the map of pairwise distances was only calculated for these two spans. Missing data in spans 6 and 7 are related to the following joints:

- In span 6: joint 32 in bay 4; joints 36, 38 and 40 in bay 5 and joints 44 and 46 in bay 6.
- In span 7: joint 137 in bay 1; joint 132 in bay 2; joints 130 and 127 in bay 3; joints 124 and 122 in bay 4; all joints in bay 5 and joints 112 and 108 in bay 6.

Figure 4.29: Health status assessment and pattern classification in bridges. 71 joints, span 6: map of pairwise distances.

Resulting maps of pairwise distances for spans 6 and 7 can be seen in Figures 4.29 and 4.30. The black lines within the map in each figure delimit regions that correspond to different bays of

the span. Each small cell with a color shows a pairwise distance between two corresponding joints in the span. The higher the distance, the less similarity between two related joints.

Figure 4.30: Health status assessment and pattern classification in bridges. 71 joints, span 7: map of pairwise distances.

Overall, the similarities between joints located in different bridge parts are found out. In span 6, it is shown that joints at middle of a bay/span behave similarly and are different from joints at the two ends. It may suggest an indication of bridge global behavior in a span. In span 7, and although a global view is not available due to the missing data, joint 135 located in bay 2 of span 7 obtained considerably large distances to other joints. Joint 131 was also behaving differently with comparison to the others, which implies a possible problem that should be checked in order to prevent further damages. It is known that the sensor in joint 135 was faulty at the time the data were collected. It must be also noted that the distance from the damaged joint to the other joints representatives increased to 18,000. By contrast, the experiment covering 71 joints showed a distance among joints

representatives below 250. Such differences of distance scale to normality may relate to severity of involved failure.

Evaluating the performance of the approach for damage detection, it can be easily deployed to provide a real-time health score of the structure. It allows detecting any potential damage on the basis of the distances between new events and the clusters centroids and joints representatives.

4.2.3 A case study on blind fasteners installation

Motivation

Intelligent monitoring of complex industrial processes is a big issue nowadays. The Industry 4.0 revolution is acting as a great driver of the development of new methodologies and technological improvements in the manufacturing industry [138]. One of the big challenges deals with how to optimally and automatically characterize behaviors of interest from monitoring data, and how to use them in an online fashion for fault detection and diagnostics purposes [139] [140]. Current techniques and procedures are still based on manual inspections and basic control systems, neither fully exploiting data available nor considering last advantages on data analytics methods and processing capabilities [141].

Benefits derived from adopting such technological advances and methodologies are clear in terms of knowledge management enhancement and time and cost reduction. However, there is still room for improvement in the development and integration of such advanced algorithms. To this concern special attention must be paid to the potential of hybrid methods for fault detection, pattern identification and process parameters optimization [74] [142] [143] [144]. They combine two or more algorithms to solve the same problem optimally.

The motivation regarding blind fasteners installation resides in the lack of intelligent strategies for an automatic on-line evaluation of installed blind bolts. When blind fasteners are used to join closed structures their evaluation after installation is not feasible unless by using time and cost intensive equipment (i.e. boroscopes). Even sometimes no evaluation at all is possible. Quite often, these issues are solved by overcalculating the number of fasteners to meet safety requirements thus leading to a weight increase. In any case, much of the benefits of using blind fasteners are not currently being fully exploited. Additional to weight aspects, the increase of the production costs due to overcalculation is also very significant. Safety coefficient being applied will depend on the aircraft area but can reach a value of 2. That means that the number of installed fasteners will be twice that estimated by the design. Considering that a small-medium sized aircraft contains around 85,000 fasteners (all types, not just blind ones) and that the price of each fastener is about 30€, plus the installation and other consumables cost (drilling, sealant application, fastening) the large economic benefits of an automatic inspection can be foreseen. An on-line evaluation system for blind bolts installation is therefore required. In this work a kernel density-based pattern classification approach for the automatic identification of behavioral patterns from monitoring data related to blind fasteners installation is presented. Patterns found can highly support the online classification of new fasteners.

Proposed approach

In order to automatically classify the installation of blind fasteners a data-driven approach is proposed. It is based on the multivariate density analysis of the head diameter (J) and the head height (K) of the formed heads in a set of installations, and on behavioral patterns identification

from high density regions found. Then, a distance-based classification of new monitoring torque-rotation diagrams can be applied.

KDE for behavioral patterns identification

Multivariate KDE is applied as it was introduced in Equation 4.14 and 4.15. In this case, and given the fastening torque (J) and the fastener rotation (K) as features, $X = \{J, K\}$, and a set of n instances $\mathbf{x}_i = (j_i, k_i)$, the KDE formula becomes:

$$\hat{f}_h(X) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_J, h_K} G\left(\frac{J - j_i}{h_J}, \frac{K - k_i}{h_K}\right) \quad (4.30)$$

Regions in feature space that show a high density imply a behavior of interest, whereas instances that are isolated, far from any behavior, can be considered as outliers and therefore they are filtered out. At each step of the process, the instance with the minimum density is removed aiming to eliminate faulty torque-rotation diagrams and noise from patterns to be defined and that will be used to classify new monitoring torque-rotation diagrams. Densities are then recalculated iteratively, using the Scott's factor, $bw = n^{(-1/(m+4))}$, computed in the first iteration for bandwidth selection, until the minimum density obtained at the i -th iteration is over the minimum density at iteration $i + 1$. Instances are grouped together to establish a behavioral pattern on the basis of their density and their proximity to each other, using the Euclidean distance $D(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(j_j - j_i)^2 + (k_j - k_i)^2}$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two instances $\in X = \{J, K\}$. Resulting distances are normalized between 0 and 1.

Behavioral patterns computation

Behavioral patterns are based on torque-rotation diagrams, since they describe the evolution along the time of the fastener installation and it can be monitored in real-time. Having the set of n blind fastener installations, they are first aligned in rotation axis (equivalent to time dimension) to the highest point by cross-correlation. Diagrams are then normalized in both dimensions, the rotation (R) and the torque (T), between 0 and 1 in order to filter out the effect of conditions variation. As a result, a normalized torque-rotation diagram is obtained, $\mathbf{d}_i = (d_1, \dots, d_z)$.

For a set S of diagrams a behavioral pattern, $\mathbf{p}_S = (p_1, \dots, p_z)$, is then simply defined as the average values in fastening torque, J , for every fastener rotation, K , as it is shown in Equation 4.31.

$$\mathbf{p}_S = \frac{1}{|S|} \sum_{i=1}^{|S|} \mathbf{d}_i \quad (4.31)$$

Distance-based classification

In order to calculate the distance between a pattern, $\mathbf{p}_S$, and a torque-rotation diagram, $\mathbf{d}_i$, Euclidean metric is computed similarly to Equation 4.3, as it can be seen in Equation 4.32.

$$D(\mathbf{p}_S, \mathbf{d}_i) = \|\mathbf{p}_S - \mathbf{d}_i\| = \sqrt{\sum_{l=1}^z (p_{Sl} - d_{il})^2} \quad (4.32)$$

Then a torque-rotation diagram can be easily classified by its proximity to a behavioral pattern. A maximum distance, D_{max} , to each pattern is established during the training phase. Having a set of k patterns, $\{\mathbf{p}_1, \dots, \mathbf{p}_k\}$, and the corresponding set of maximum distances, $\{D_{max_1}, \dots, D_{max_k}\}$, a new diagram $\mathbf{d}_{new} \in p_S$ if:

$$D_S = \underset{\{S=1, \dots, k\}}{\operatorname{argmin}} D(\mathbf{p}_S, \mathbf{d}_{new}) \quad (4.33)$$

Therefore, new patterns can be found when $D(\mathbf{p}_S, \mathbf{d}_{new}) > D_{max_S}, \forall S = 1, \dots, k$. The underlying idea is based on proximity and clustering algorithms [145].

The overall algorithm can be seen in Algorithm 4.5.

Algorithm 4.5 Health status assessment and pattern classification in blind fasteners installation. KDE for behavioral patterns identification

Input: a set of n instances of (J, K) features, $\mathbf{x}_i = \{j_i, k_i\}$ and a set of n corresponding torque-rotation diagrams $\mathbf{d}_i = (d_1, \dots, d_z)$, $i = 1, \dots, n$

```

1: Compute  $\hat{f}_h(X)$  using Equation 4.30
2: Set  $min' = 0$ 
3: Set  $density = \hat{f}_h(X_{|n|})$ 
4: Set  $bw = n^{(-1/(m+4))}$ 
5:  $outliers = \{\}$ 
6: while  $min(density) > min'$  do
7:    $min' = min(density)$ 
8:    $\mathbf{x}_{min'} \rightarrow outliers$ 
9:   Remove  $\mathbf{x}_{min'}$ 
10:  Compute  $density = \hat{f}_h(X_{|<n|})$  using Equation 4.30
11: end while
12:  $patterns = \{\}$ 
13: for all  $\mathbf{x}_i, i = (1, \dots, |<n|)$  do
14:   for all  $\mathbf{x}_j, j = (1, \dots, |<n|), j \neq i$  do
15:     if  $D(\mathbf{x}_i, \mathbf{x}_j) < bw$  then
16:        $\mathbf{x}_j \rightarrow S$ 
17:     end if
18:   end for
19:   Compute  $\mathbf{P}_S$  using Equation 4.31
20:    $\mathbf{P}_S \rightarrow patterns$ 
21: end for
```

Experimental results

The material used in experimental tests consists of a set of 35 torque-rotation diagrams. They all correspond to fasteners with the same reference and dash (MBF 2313, dash 5), though some variations in their installation conditions exist: grip length, stack thickness, preload and spindle revolutions per minute (rpm).

Despite the small amount of diagrams available, they are representative examples of the main behavioral patterns that can be found in the blind fastener installation process. Experimental data contained a total of 9 diagrams showing a correct installation. Other 26 diagrams are more or less deviated from admissible limits.

The proposed approach allows identifying 3 differentiated high density regions, as it can be seen in Figure 4.31. One region contains the 10 faulty torque-rotation diagrams whereas the other regions contains the other 23 diagrams, 9 showing a typical normal behavior and 14 corresponding to a faulty behavior but closer to admissible limits.

Figure 4.31: Health status assessment and pattern classification in blind fasteners installation. High density regions found in data.

From computed densities, two outliers are also detected, which correspond to (J, K) pairs isolated in feature space, with low pdf values. They are presented in Figure 4.32.

Figure 4.32: Health status assessment and pattern classification in blind fasteners installation. KDE-based pattern classification approach. Outliers found.

After removing such outliers, three different patterns are obtained: pattern A, representing the faulty diagrams, pattern B, representing a faulty behavior close to admissible limits, and pattern C, containing the normal diagrams. K-means method is also executed with $K=3$ in order to also find three different groups of similar diagrams in data. The distance metric used is the same Euclidean distance presented in Equation 4.32. Resulting patterns slightly differ from those obtained by KDE-based approach. The corresponding (normalized) torque-rotation diagrams obtained by both methods are shown in Figure 4.33.

Figure 4.33: Health status assessment and pattern classification in blind fasteners installation. Patterns found in data by (a) kernel density-based pattern classification approach and (b) K-means ($k=3$).

Torque-rotation diagrams taken from the experimental setup were tested on a 5-fold cross-validation basis, by segmenting the total data set into 5 equal parts. Therefore, the accuracy and generalization of the proposed approach when classifying new diagrams can be tested. The confusion matrix presented in Table 4.16 is obtained, containing the global results of the 5 folds for both methods.

Table 4.16: Health status assessment and pattern classification in blind fasteners installation. Confusion matrix.

	Pattern	Predicted A	Predicted B	Predicted C
KDE-based pattern classification	A	8	1	1
	B	1	12	1
	C	0	3	6
K-means ($K=3$)	A	4	4	2
	B	2	8	4
	C	1	7	1

It can be appreciated that the KDE-based pattern classification outperformed the classification results obtained by K-means for all patterns. Moreover, most of the correctly installed blind fasteners, represented by pattern C, are classified as pattern B diagrams in the case of K-means approach. This may lead to important false positive rates.

In order to evaluate the results, the $\text{precision} = TP / (TP + FP)$, $\text{recall} = TP / (TP + FN)$ and $\text{accuracy} = (TP + TN) / (TP + TN + FP + FN)$ of the classification are calculated, being TP the true positives, FP the false positives, FN the false negatives and TN the true negatives. They are three widely used quality measures in this kind of processes.

As it is shown in Table 4.17, precision, recall and accuracy are globally above 77%, so the approach accurately establishes patterns from data. Interestingly, the correct classification of diagrams rises above the false positive rate, $FPR = FP / (FP + TN)$, in all patterns, being $FPR = 0.03$ for pattern A, $FPR = 0.2$ for pattern B and $FPR = 0.07$ for pattern C, respectively. In contrast, K-means obtained a global precision, recall and accuracy below 40%. These results highlights the

difficulty and challenging of the given test scenario, containing only 35 representative diagrams, being two of them potential outliers that can highly influence the patterns to be drawn.

Table 4.17: Health status assessment and pattern classification in blind fasteners installation. Precision, recall and accuracy.

		Pattern A	Pattern B	Pattern C	Global Results
KDE-based pattern classification	Precision	88.89%	75%	75%	79.63%
	Recall	80%	85.71%	66.67%	77.46%
	Accuracy				78.79%
K-means ($K=3$)	Precision	57.13%	42.14%	14.28%	37.83%
	Recall	40%	57.13%	11.1%	36.07%
	Accuracy				39.38%

4.3 ML methods for quality estimation and production optimization

4.3.1 A case study on animal farming

Motivation

Broiler meat chickens are the most abundant farmed animal in the European Union (EU), and a key component of EU food supply. Much data is already collected by the member states (MS), predominately at slaughter plant level ¹. Data acquisition in itself is, however, of little value unless collected data are standardized among MS and further processed to produce useful information to improve broiler health and welfare, and hence offering the potential to reduce antimicrobial use across the EU, a real and current concern for producers and legislators.

The broiler meat chicken industry is the second largest meat industry in the world. Yearly, 70 billion birds are produced around the world under similar, well established management practices with similar genetic stocks produced majorly by two international companies; Cobb-Vantress (Cobb) and Aviagen (Ross). Environmental models put in practice nowadays are based on the theoretical curves proposed by this two companies. However, the large production volumes and the great complexity of the production chain, implies that the possibility to control system parameters to optimal values is probably fictitious. Additionally, deviation of values from optimality may go undetected due to different reasons including the lack/misuse of information at various stages, or late problem detection. Such circumstances may lead to a loss of efficiency of the system. Lack of an adequate control during each production process step may also lead to impaired animal health and welfare, and to the occurrence of meat quality and safety problems. Therefore, system adjustments to maximize, for instance, energy efficiency, may easily derive in unforeseen animal health or welfare issues. In addition, the economic perspective must also be considered as potential changes in management practices may be impossible in practice when all inputs and outputs of the system, many of them of different nature and coming from diverse sources, are simultaneously considered.

In order to provide the broiler meat chicken industry with an intelligent system able to support an efficient and sustainable broiler production, most recent technological approaches must be put together with traditional broiler production. The model of sustainable production also needs to be based on considerations for animal welfare and environmental responsibility. Advanced knowledge-based, empirical models to optimally manage broiler processes are needed. They are able to provide the players of the production chain such as farmers, veterinarians and technical personnel with key recommendations to optimize production and to design better production strategies. ML algorithms are usually applied to automatically generate such data-driven knowledge models. To do so, features of interest must be identified according to best management practices and most recent scientific knowledge in broiler production, health and welfare. Resulting models can be then deployed in a unified DSS that should be able to provide the necessary tools to assure efficient and sustainable production according to a social responsibility-based production model.

This work aims at developing a DSS that uses environmental parameters automatically collected for each corresponding flock, together with additional live production health and welfare data. This work will allow to generate intelligent practical tools to improve flock performance by better managing the health and welfare of broiler flocks. Our main contribution is to provide such tools on the basis of data fusion and a quantile regression forests-based approach. Resulting growth, welfare

¹Under the Broiler Directive 2007/43/EC

and production models are robust and comprehensive, yet accurate decision support tools in animal farming.

Proposed approach

The aim of this research work is to provide the meat chicken industry with an intelligent system able to support an efficient and sustainable broiler production. To do so, the key parts of the proposed DSS are the following:

- Automatic data acquisition and cloud-based storage capabilities.
- Environmental indicators, which are defined as deviations from optimal environmental conditions, previously learned, over time.
- Quantile regression forests approach used to model data and make predictions, dealing with uncertainty.
- Data fusion of environmental indicators and weights, leg problems and mortality ranges for the generation of growth, welfare and production models able to provide optimal decision support in animal farming.

The system architecture consists of a set of sensors installed to automatically measure and collect different environmental conditions in farm, a cloud-based storage of such information and a set of ML-based models learned from historical data to manage the production smartly and in an online fashion.

Data acquisition and cloud-based storage

Data acquisition devices measure environmental parameters every 15 minutes for each flock of birds from farm arrival to the end of production. For each parameter, there are three sensors located around the farm in order to minimize the variations in environmental conditions given in different locations. Collected data is automatically transferred to a cloud-based service, where it is stored for further analysis. Cloud service used is ownCloud ².

Growth, welfare and production parameters (e.g. weights, leg problems and mortality rate) are periodically acquired by experts by transect walk methodology [146] and other manual techniques, due to the complexity and expertise needed:

- Growth parameter: weights are collected at arrival and during week 1, 3, 5 and 6 of age on a representative sample of 50 birds/flock.
- Welfare parameter: lame and immobile birds are considered; they are collected using the transect method as described in [146] as the frequency (%) of each problem per transect.
- Production parameter: mortality rate is considered; it is calculated using farmer records from day 1 to slaughter, and including birds found dead and culled every day.

All this information is also integrated in the cloud-based data system.

²<https://owncloud.org/>

Environmental indicators

Information given by environmental conditions can support predictions made since they have serious effects in growth and animal welfare. Since the acquisition frequency regarding the environmental parameters is much higher than in the case of growth, welfare and production parameters, environmental data, $\mathbf{env} = \{env_1, \dots, env_m\}$, are re-sampled in order to have an average value per hour, $\langle \mathbf{env} \rangle = \{\langle env \rangle_1, \dots, \langle env \rangle_{n \leq m}\}$, where $\langle \cdot \rangle$ denotes the average over the sampled data.

Figure 4.34: Quality estimation and production optimization in animal farming. Example of relative humidity model learned from a set of farms.

Observations that are far from the mean are filtered out, based on 3 standard deviations from the mean, $\mu_{\langle \mathbf{env} \rangle}$. They are bad readings caused by hardware issues and can strongly affect the resulting environmental indicators. An example of resulting relative humidity model from a set of farms is shown in Figure 4.34.

Cumulative deviations from environmental models over time, $\mathbf{d} = (d_1, \dots, d_n)$ are then computed in order to provide useful indicators. They are calculated for values above, $\mathbf{d}^a$, and below, $\mathbf{d}^b$, the model, as it can be seen in Equation 4.34.

$$d_i = \begin{cases} d_i^a = \sum_{j=1}^i (\langle env \rangle_j - \langle env \rangle'_j) & \forall \langle env \rangle_j \geq \langle env \rangle'_j \\ d_i^b = \sum_{j=1}^i (\langle env \rangle'_j - \langle env \rangle_j) & \forall \langle env \rangle_j \leq \langle env \rangle'_j \end{cases} \quad (4.34)$$

where $\langle \mathbf{env} \rangle' = (\langle env \rangle'_1, \dots, \langle env \rangle'_i)$ are the environmental model values and $\langle \mathbf{env} \rangle = (\langle env \rangle_1, \dots, \langle env \rangle_i)$ are the current environmental values until day $i = 1, \dots, n$, respectively.

It is assumed that large cumulative deviations from normal environmental conditions over time have a big impact in growth, in the welfare of the birds and in production.

Quantile Regression Forests

Random forests is an ensemble method that grows an ensemble of trees [51]. It employs averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement (bootstrap). A large number of trees is therefore grown. Random forests for regression analysis, or regression forests, are an ensemble of different regression trees in which each leaf draws a distribution for the continuous target feature, $\mathbf{y} = (y_1, \dots, y_n)$. More precisely, given a set of m features, $X = \{X_1, \dots, X_m\}$ where each feature X_i can take a value from its own set of possible values χ_i , and n feature vectors or instances, $\mathbf{x}_i = (x_1, \dots, x_m) \in \chi = (\chi_1, \dots, \chi_m)$, with $i = 1, \dots, n$, a random forest is a collection of K tree predictors $T(\theta_k)$, with $k = 1, \dots, K$, being θ_k the random parameter vector that determines how the k -th tree is grown, or which variables are considered to split on at each node when approximating $\mathbf{y}$. For each tree and each node, randomness is employed when selecting a variable to split on and for each tree, a bagged version of X is used. In addition, only a random subset of predictor variables is considered for splitpoint selection at each node.

It is assumed that X and θ_k are independent and identically distributed, and tuples $(\mathbf{x}_i, y_i)$ are independently drawn from the joint distribution.

Every leaf of the tree, $l = 1, \dots, L$, corresponds to a rectangular subspace of χ denoted as $R_l \subseteq \chi$. Then for every $\mathbf{x}_i$ there is one and only one leaf $l(\mathbf{x}_i, \theta)$ for tree $T(\theta)$ such that $\mathbf{x}_i \in R_l$.

For a new feature vector, $\mathbf{x}_{new}$, the prediction of a single tree $T(\theta)$ is the average of the observed values in leaf $l(\mathbf{x}_{new}, \theta)$. Let the weighted vector $w_i(\mathbf{x}_{new}, \theta)$ be defined as:

$$w_i(\mathbf{x}_{new}, \theta) = \begin{cases} 1/\#\{j : \mathbf{x}_j \in \chi_{l(\mathbf{x}_{new}, \theta)}\} & \text{if } \mathbf{x}_i \in \chi_{l(\mathbf{x}_{new}, \theta)} \\ 0 & \text{otherwise} \end{cases} \quad (4.35)$$

The prediction of a single tree can be then computed as the weighted average of the target feature values, $\mathbf{y}$, as it is shown in Equation 4.36.

$$\hat{\mu}(\mathbf{x}_{new}) = \sum_{i=1}^n w_i(\mathbf{x}_{new}, \theta) y_i \quad (4.36)$$

In the case of random forests, Equation 4.36 is generalized as the average prediction of K single trees, as it can be seen in Equation 4.37.

$$\hat{\mu}(\mathbf{x}_{new}) = \sum_{i=1}^n K^{-1} \sum_{k=1}^K w_i(\mathbf{x}_{new}, \theta_k) y_i \quad (4.37)$$

By means of the Quantile Regression Forests approach, confidence intervals can be obtained from predictions made [147]. A Confidence Interval (CI) provides valuable information about the dispersion of observations around the predicted value, reinforcing the reliability of predictions made. Instead of recording the mean value of response variables in each tree leaf in the forest, all observed responses in the leaf are recorded. The prediction thus becomes the full conditional distribution $P(\mathbf{y} \leq y_i | X = \mathbf{x}_i)$, with $i = 1, \dots, n$, given by the probability that, for $X = \mathbf{x}_i$, $\mathbf{y} < y_i \in \mathbb{R}$.

The corresponding conditional distribution function $F(\mathbf{y} | X = \mathbf{x}_i)$ can be also expressed as $E(1_{\{\mathbf{y} \leq y_i\}} | X = \mathbf{x}_i)$, which is approximated by the weighted mean over the observations of $1_{\{\mathbf{y} \leq y_i\}}$, as it can be seen in Equation 4.38.

$$\hat{F}(\mathbf{y}|X = \mathbf{x}_i) = \sum_{i=1}^n w_i(\mathbf{x}_i) 1_{\{\mathbf{y} \leq y_i\}} \quad (4.38)$$

where $w_i(\mathbf{x}_i) = K^{-1} \sum_{k=1}^K w_i(\mathbf{x}_i, \theta_k)$ is the weighted vector.

The α -quantile, $Q_\alpha(\mathbf{x}_i)$ is defined such that the probability of $\mathbf{y} < Q_\alpha(\mathbf{x}_i) = \alpha$. The quantiles give more complete information about the distribution of $\mathbf{y}$ as a function of the predictor features, X , than the conditional mean $E(\mathbf{y}|X = \mathbf{x}_i) = \underset{z}{\operatorname{argmin}} E\{(\mathbf{y} - z)^2|X = \mathbf{x}_i\}$.

Given a new feature vector, $\mathbf{x}_{new}$, the estimate of the distribution function in 4.38 is computed and prediction intervals are created by simply applying the appropriate percentiles of the distribution. A 95% prediction interval for the value of $\mathbf{y}$, for instance, will be given by Equation 4.39.

$$I(\mathbf{x}_{new}) = [Q_{.025}(\mathbf{x}_{new}), Q_{.975}(\mathbf{x}_{new})] \quad (4.39)$$

Quantile regression forests approach can therefore be used to create prediction intervals that contain very useful information about the dispersion of observations around the predicted value. Besides, quantile regression estimates are more robust in presence of outliers and uncertainty in data.

Data fusion models

Environmental indicators are combined with weights, leg problems and mortality ranges to generate useful growth, welfare and production models able to provide optimal decision support in animal farming.

The growth indicator is given by the average of the weights at weeks 3, 5 and 6, $\langle \mathbf{weight}_s \rangle$, being $s = 3, 5, 6$. The idea is to anticipate heavy deviations during the growth process on the basis of deviations from optimal environmental values, modifying the farm conditions accordingly in advance.

Then, the growth model, $\langle \mathbf{weight}_s \rangle \approx G(d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b)$, is defined as it can be seen in Equation 4.40, being $(d_1, \dots, d_m)$ the deviations from the m environmental parameters under study.

$$\begin{aligned} \hat{G}(\langle \mathbf{weight}_s \rangle | X = \{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}) = \\ \sum_{i=1}^n w_i(\{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}) 1_{\{\langle \mathbf{weight}_s \rangle \leq \langle \mathbf{weight}_{si} \rangle\}} \end{aligned} \quad (4.40)$$

with $w_i(\{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}) = K^{-1} \sum_{k=1}^K w_i(\{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}, \theta_k)$ being the weighted vector.

Furthermore, the proportion of occurrence of leg problems (lame and immobile birds) over the total population (%), lp , at weeks 3, 5 and 6 of the entire growth process, is established as a welfare indicator.

Similarly as in the case of G , the welfare model, $lp_s \approx W(d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b)$, becomes (see Equation 4.41).

$$\hat{W}(lp_s | X = \{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}) =$$

$$\sum_{i=1}^n w_i(\{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}) 1_{\{lp_s \leq lp_{si}\}} \quad (4.41)$$

with $w_i(\{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}) = K^{-1} \sum_{k=1}^K w_i(\{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}, \theta_k)$ being the weighted vector.

Finally, the production indicator is given by the cumulative, proportion of mortality over the total population (%), $mort$, at weeks 3, 5 and 6.

In this case, $mort_s \approx M(d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b)$ defines the production model, as it can be seen in Equation 4.42.

$$\begin{aligned} \hat{M}(mort_s | X = \{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}) = \\ \sum_{i=1}^n w_i(\{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}) 1_{\{mort_s \leq mort_{si}\}} \end{aligned} \quad (4.42)$$

with $w_i(\{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}) = K^{-1} \sum_{k=1}^K w_i(\{d_{1s}^a, d_{1s}^b, \dots, d_{ms}^a, d_{ms}^b\}, \theta_k)$ being the weighted vector.

From resulting models, prediction intervals are generated. On the basis of such predictions, useful recommendations to adjust environmental conditions when they are outside the optimal limits can be provided.

Experimental results

The experimental data comprised a set of 20 flocks of meat chickens from different farms in various locations around Spain. Given the temperature and the relative humidity parameters of the 20 flocks under study, the environmental deviations from optimal conditions are computed on a LOOCV (Leave-One-Out Cross-Validation) basis for each sample, by segmenting the total set of samples into 20 parts. Environmental model is thus calculated on $n - 1$ samples and cumulative deviations are obtained for the sample left out, as it was presented in Equation 4.34. The resulting indicators will be used as input features of the quantile regression forests-based growth, welfare and production models.

Figure 4.35: Quality estimation and production optimization in animal farming. Cumulative deviations in (a) temperature and (b) relative humidity.

The average of cumulative deviations from temperature and relative humidity models used in this study can be seen in Figure 4.35. It can be appreciated how the cumulative deviations values increase as the growth period gets close to its end in week 6, especially in relation to deviations in temperature below the model. Heavy variations were also observed depending on the control system applied in each farm.

Available weights, leg problems and mortality rates information is then combined with environmental indicators to obtain the growth, welfare and production models. In order to estimate the generalization performance of the quantile regression forests-based modeling approach, the same LOOCV strategy was applied. Therefore, all models were trained on $n - 1$ samples, including data related to weeks 3, 5 and 6, and tested for each sample by comparing estimated target parameters in weeks 3, 5 and 6 to unseen, real values.

In the case of the growth model, the prediction intervals in weeks 3, 5 and 6 for all farms can be seen in Figure 4.36. Prediction intervals in week 3 for samples with id 13, 16, 18 and 20 are quite precise, showing very small sizes. Some interesting trends regarding prediction intervals can be appreciated in several samples, e.g. samples with id 3, 4 or 8. Additionally, in many cases a low final weight could have been anticipated at an early stage of the process, at week 3, e.g. in samples with id 1, 5, 6, 11, 12, 16 or 20, since the upper bound of predicted intervals are very low.

Figure 4.36: Quality estimation and production optimization in animal farming. Random forests-based growth model. Illustration of real values vs. predicted values and corresponding quantile intervals in weeks 3, 5 and 6.

The results presented in Table 4.18 are obtained, containing the average outputs of the 20 folds. From samples detected outside the prediction intervals, there are more samples above than below them, which means that other factors may also influence the growth curve.

Table 4.18: Quality estimation and production optimization in animal farming. Results obtained by the growth model.

	Week 3	Week 5	Week 6
Predicted in the 95% CI	18	19	16
Above the upper 95% CI limit	0	1	4
Below the lower 95% CI limit	2	0	0

The prediction intervals obtained by the welfare model in weeks 3, 5 and 6 for all samples is presented in Figure 4.37. Note that in the case of the flock 17th there were no leg problems registered in week 6, since the growth period ended before in this case. In some cases, e.g. in samples with id 8, 9, 10 or 11, big differences could be found regarding the evolution of prediction intervals from weeks 3 to 6, highly increasing the predicted values and the size of the intervals. This negative welfare effect must be dealt with in advance by, for instance, readjusting the environmental parameters in accordance with the optimal conditions, in order to minimise its impact in the production at the end of the growth period.

Figure 4.37: Quality estimation and production optimization in animal farming. Random forests-based welfare model. Illustration of real values vs. predicted values and corresponding quantile intervals in weeks 3, 5 and 6.

The LOOCV process with relation to the welfare model produced the following results (see Table 4.19). Samples outside the prediction intervals are equally distributed below and above them in weeks 3 and 6, respectively. This means a sharp change in the global trend that could have been anticipated in week 5, when it turns out that a couple of samples are above the prediction intervals.

Table 4.19: Quality estimation and production optimization in animal farming. Results obtained by the welfare model.

	Week 3	Week 5	Week 6
Predicted in the 95% CI	16	18	16
Above the upper 95% CI limit	0	2	4
Below the lower 95% CI limit	4	0	0

Regarding the production model, Figure 4.38 presents computed prediction intervals for all samples in weeks 3, 5 and 6. A high rate of mortality can be appreciated in the 10th and 15th samples, which could have been anticipated in week 5, when a value extremely high, above the prediction interval, was detected.

Figure 4.38: Quality estimation and production optimization in animal farming. Random forests-based production model. Illustration of real values vs. predicted values and corresponding quantile intervals in weeks 3, 5 and 6.

Following are the production model results (see Table 4.20), in which the average outputs of the 20 folds for samples predicted within the 95% CI, and above and below it, are presented. Interestingly, in general terms, there are more samples below the prediction intervals than above them (6 and 4, respectively), especially in week 3, which means low mortality rates despite the environmental conditions.

Table 4.20: Quality estimation and production optimization in animal farming. Results obtained by the production model.

	Week 3	Week 5	Week 6
Predicted in the 95% CI	16	17	17
Above the upper 95% CI limit	0	2	2
Below the lower 95% CI limit	4	1	1

As a whole, it can be observed that predictions that are more accurate have smaller prediction intervals. This is due to the fact that they are related to more common, expected input environmental parameters, closer to the optimal conditions, and with slight variations in the considered flocks. Therefore, the corresponding output parameters are easier to predict.

In Table 4.21, the global prediction accuracy and interval size obtained by growth, welfare and production models at weeks 3, 5 and 6 are shown. They are computed as the average of the 20 folds. In general, prediction accuracy is very high during weeks 3 and 5, and it decreases in week 6. More precisely, the best estimates are achieved in week 5, being 0.95, 0.9 and 0.85 for growth, welfare and production models, respectively. Only the production model obtained slightly better results in week 6 than in week 3, rising from 0.8 to 0.85. This is probably due to the increase in environmental conditions variance during the last days of the growth period, and to some extent due to external factors in the farm conditions, e.g. the way a particular farmer works. By defining a 95% CI, larger intervals are obtained and few samples are classified outside the limits. They can be considered as outliers.

Table 4.21: Quality estimation and production optimization in animal farming. LOOCV average score results of growth, welfare and production models.

	Week	Interval size	% over the total range	Prediction accuracy	
Growth model	3	920.074	46.181%	0.9	88.332%
	5	1558.894	78.246%	0.95	
	6	1635.609	82.098%	0.8	
Welfare model	3	0.734	52.957%	0.8	81.667%
	5	0.961	69.335%	0.9	
	6	0.946	68.254%	0.75	
Production model	3	2.547	45.95%	0.8	83.332%
	5	3.422	61.735%	0.85	
	6	3.993	72.037%	0.85	

Global prediction accuracy is over 81% in every proposed model, which means that given predictions can highly support the in-farm decision making by providing useful recommendations to adjust environmental conditions when they are outside the optimal limits. Although the obtained global prediction interval size denotes a high degree of uncertainty, it is still encouraging that the correctly predicted values rises above the interval size. These results demonstrate the validity of the proposed DSS for a representative set of farms, involving 20 flocks of meat chickens from different locations in Spain.

5.1 Conclusions

This work has been devoted to make a step forward in the ML for data-driven prognostics field. In particular, the MDI 4.0 challenge has been pursued in order to create common foundations for different application fields and industrial problems. The experience in different industrial projects and their scientific analysis have been on the basis of the ideas developed in this dissertation. Within this context, besides the scientific methodology put in practice, other aspects such as potential benefit on industry, technical feasibility and experimental validation and demonstration have been the key drivers of this work.

Following the specific constraints and requirements raised by the real projects presented, it becomes obvious that the learning and application of data-driven models is needed. The MDI 4.0 concept tries to gather all Industry 4.0 requirements from a data science perspective, by providing data-driven prognostics, health status assessment, pattern classification, quality estimation and optimal production models on the basis of ML methods.

One of the most critical parts of the MDI 4.0 is to apply a general framework to automatize the modeling process, optimally and regardless the monitoring asset under study. As it has been discussed, the industrial sector and the problem to be addressed strongly determines the analysis strategy to be executed. The data itself is crucial in this regard. The experience acquired in the development of the different approaches presented demonstrates that despite of this fact, a common methodology or analytical work-flow can be applied. Greater stress must be put on the model selection and on the free parameters to be specified. When a deeper knowledge on the application domain is available or an exhaustive data preparation task is performed, the modeling process is usually easier and more effective, and therefore the resulting models turns to be more accurate. But in many occasions this information is very limited or even nonexistent, as it was commented in the case of the marine diesel engines and propulsion systems, the Sydney Harbour Bridge and the blind fasteners installation. This is one of the main objectives of this work: to support the learning and behavior modeling phase in an automatic manner.

Finally, all this work and the corresponding proposed approaches have not been only theoretic-

cally developed, but also implemented and deployed in real scenarios to provide intelligent solutions and to support the smartization of the industry. The final goal can be therefore achieved depending of the application under consideration, e.g. CBM, predictive maintenance, SHM or sustainable production for an optimal decision making on business. The MDI 4.0 must be further developed, focusing on other technological aspects that support the data science paradigm and that contribute to the Industry 4.0 revolution as a whole.

5.1.1 Lessons learned

During the realization of this PhD dissertation several lessons have been learned, which must be mentioned. They are summarized as follows:

- **Dealing with real field data is not an easy task.** Besides the difficulties derived from the nature of the variables to be analysed (e.g. latency, nonlinear relationships and dependencies in a high-dimensional feature space), data must be smartly preprocessed (e.g. identifying outliers, processing missing values and applying normalization and standardization) and properly combined to successfully infer and model behaviors from them. This is quite apart from handling dummy or synthetic data, generated in controlled or simulated environments.
- **When doing research, the ways to achieve a goal are *infinite*.** Solving complex problems is challenging and requires a lot of efforts and a deep knowledge in problem solving strategies and tools. When data is involved, this task becomes easier, and even if the optimal solution is not achieved some interesting insights can be obtained by applying the right ML method.
- **There is always space to improve.** A research work is never finished fully and satisfactorily. New promising methods and techniques that may highly improve results achieved by a proposed solution are constantly arising. Sometimes this can be very frustrating, but also exciting and it encourages you to continue to learn.
- **The simplest solution is almost always the best.** Not only in terms of conciseness and intelligibility, but also regarding time and resources needed. Most of the times high levels of accuracy can be obtained with no need of applying the most complex algorithm ever.
- **It is very important to understand your data.** A good data scientist may design and approximate a scientific approach to solve a problem by *simply* taking a first look at the data. This is not about a test-error approach, wasting time and resources. By all means, a fine tuning of the proposed method will be needed, but it is advisable to go for the right solution, if any, straightforward.

5.2 Future work

Although this PhD process finishes with the contributions presented in this dissertation, new methods and applications are proposed in the same research line, as new challenging problems and computational algorithms arise and the complexity of monitoring assets increases. DS explosion is a fact nowadays. Nevertheless, there are some elements that have not been fully considered in this work, e.g. the analysis of massive data or the conceptualization and generalization of a common ML-based framework.

State-of-the-art ML methods, such as deep learning for spatial-temporal modeling, and novel BD and cloud-based strategies to efficiently monitor fleets of assets are creating a very promising framework to do applied research on Industry 4.0. According to this futuristic view, other research works in the context of the MDI 4.0 paradigm must be carried out, following the research line presented in this PhD.

5.2.1 Deep learning for spatial-temporal modeling

Deep learning is one of the key driving forces for many technology revelations. It opens another door for us to rethink how spatial and temporal events can be modelled, helping us to accurately predict future events, such as behaviors of interest, asset and infrastructure failures, production demand and product quality.

Further work will aim to study the basic deep learning concept, develop novel deep learning algorithms for spatial and temporal modelling, and apply them to solving real-world industrial problems within the scope of the MDI 4.0 concept.

- [1] I. . W. Group, et al., Recommendations for implementing the strategic initiative industrie 4.0, Final report, April.
- [2] M. Blanchet, T. Rinn, G. v. Thaden, G. Thieulloy, Industry 4.0: The new industrial revolution-how europe will succeed, Hg. v. Roland Berger Strategy Consultants GmbH. München. Abgerufen am 11.05. 2014, unter http://www.rolandberger.com/media/pdf/Roland_Berger_TAB_Industry_4_0_2014_0403.pdf.
- [3] T. Devezas, J. Leitão, A. Sarygulov, Industry 4.0: Entrepreneurship and Structural Change in the New Digital Landscape, Springer, 2017.
- [4] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, M. Harnisch, Industry 4.0: The future of productivity and growth in manufacturing industries, Boston Consulting Group (2015) 14.
- [5] S. Nathan, Understanding industry 4.0: Factories go digital, Engineer (Online Edition) 2.
- [6] G. Vasco, Pcti euskadi 2020, Una estrategia de especialización inteligente. Bilbao.
- [7] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer networks 54 (15) (2010) 2787–2805.
- [8] M. A. Beyer, D. Laney, The importance of big data: a definition, Stamford, CT: Gartner (2012) 2014–2018.
- [9] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, 2016.
- [10] C. M. Bishop, et al., Pattern recognition and machine learning, Vol. 1, springer New York, 2006.
- [11] S. Russell, P. Norvig, A. Intelligence, A modern approach, Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs 25 (1995) 27.
- [12] V. Dhar, Data science and prediction, Communications of the ACM 56 (12) (2013) 64–73.

- [13] G. Siemens, D. Gasevic, C. Haythornthwaite, S. Dawson, S. B. Shum, R. Ferguson, E. Duval, K. Verbert, R. Baker, Open learning analytics: an integrated & modularized platform, Proposal to design, implement and evaluate an open platform to integrate heterogeneous learning analytics techniques.
- [14] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, *ACM Computing Surveys (CSUR)* 41 (3) (2009) 15.
- [15] I. H. Witten, E. Frank, *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann, 2005.
- [16] J. R. Quinlan, *C4. 5: programs for machine learning*, Elsevier, 2014.
- [17] J. Hipp, U. Güntzer, G. Nakhaeizadeh, Algorithms for association rule mining a general survey and comparison, *ACM sigkdd explorations newsletter* 2 (1) (2000) 58–64.
- [18] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, in: *Acm sigmod record*, Vol. 22, ACM, 1993, pp. 207–216.
- [19] R. Agrawal, R. Srikant, Mining sequential patterns, in: *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, IEEE, 1995, pp. 3–14.
- [20] J. Rabatel, S. Bringay, P. Poncelet, Contextual sequential pattern mining, in: *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, IEEE, 2010, pp. 981–988.
- [21] L. A. Zadeh, Fuzzy sets, *Information and control* 8 (3) (1965) 338–353.
- [22] C. R. Turner, A. Fuggetta, L. Lavazza, A. L. Wolf, A conceptual basis for feature engineering, *Journal of Systems and Software* 49 (1) (1999) 3–15.
- [23] S. B. Kotsiantis, I. Zaharakis, P. Pintelas, Supervised machine learning: A review of classification techniques (2007).
- [24] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*, Vol. 1, MIT press, 1992.
- [25] R. Rojas, *Neural networks: a systematic introduction*, Springer Science & Business Media, 2013.
- [26] V. J. Hodge, J. Austin, A survey of outlier detection methodologies, *Artificial Intelligence Review* 22 (2) (2004) 85–126.
- [27] A. Widodo, B.-S. Yang, Support vector machine in machine condition monitoring and fault diagnosis, *Mechanical Systems and Signal Processing* 21 (6) (2007) 2560–2574.
- [28] O. Chapelle, B. Scholkopf, A. Zien, Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews], *IEEE Transactions on Neural Networks* 20 (3) (2009) 542–542.
- [29] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, *Neural computation* 13 (7) (2001) 1443–1471.

- [30] T. Hastie, R. Tibshirani, J. Friedman, Unsupervised learning, in: The elements of statistical learning, Springer, 2009, pp. 485–585.
- [31] W. Wu, H. Xiong, S. Shekhar, Clustering and information retrieval, Vol. 11, Springer, 2004.
- [32] S.-H. Cha, Comprehensive survey on distance/similarity measures between probability density functions, City 1 (2) (2007) 1.
- [33] A. K. Jain, Data clustering: 50 years beyond k-means, Pattern recognition letters 31 (8) (2010) 651–666.
- [34] T. Cover, P. Hart, Nearest neighbor pattern classification, Information Theory, IEEE Transactions on 13 (1) (1967) 21–27.
- [35] T. Kohonen, The self-organizing map, Neurocomputing 21 (1) (1998) 1–6.
- [36] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise., in: Kdd, Vol. 96, 1996, pp. 226–231.
- [37] T. Hofmann, B. Schölkopf, A. J. Smola, Kernel methods in machine learning, The annals of statistics (2008) 1171–1220.
- [38] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, IEEE Intelligent Systems and their Applications 13 (4) (1998) 18–28.
- [39] I. Steinwart, A. Christmann, Support vector machines, Springer Science & Business Media, 2008.
- [40] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
- [41] M. C. Munteanu, A. Caliman, C. Zaharia, Convolutional neural network, uS Patent 9,665,799 (May 30 2017).
- [42] G. E. Hinton, Deep belief networks, Scholarpedia 4 (5) (2009) 5947.
- [43] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, S. Khudanpur, Recurrent neural network based language model., in: Interspeech, Vol. 2, 2010, p. 3.
- [44] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.
- [45] C. Howson, P. Urbach, Scientific reasoning: the Bayesian approach, Open Court Publishing, 2006.
- [46] L. E. Sucar, Probabilistic Graphical Models, Springer, 2015.
- [47] K. P. Murphy, Dynamic bayesian networks: representation, inference and learning, Ph.D. thesis, University of California, Berkeley (2002).
- [48] V. Saligrama, M. Zhao, Local anomaly detection., in: AISTATS, 2012, pp. 969–983.
- [49] R. Polikar, Ensemble learning, in: Ensemble machine learning, Springer, 2012, pp. 1–34.

- [50] T. G. Dietterich, Ensemble methods in machine learning, in: International workshop on multiple classifier systems, Springer, 2000, pp. 1–15.
- [51] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.
- [52] L. Breiman, Bagging predictors, Machine learning 24 (2) (1996) 123–140.
- [53] Y. Freund, R. Schapire, N. Abe, A short introduction to boosting, Journal-Japanese Society For Artificial Intelligence 14 (771-780) (1999) 1612.
- [54] Y. Freund, R. E. Schapire, et al., Experiments with a new boosting algorithm, in: icml, Vol. 96, 1996, pp. 148–156.
- [55] D. H. Wolpert, Stacked generalization, Neural networks 5 (2) (1992) 241–259.
- [56] S. Arlot, A. Celisse, et al., A survey of cross-validation procedures for model selection, Statistics surveys 4 (2010) 40–79.
- [57] R. Kohavi, et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Ijcai, Vol. 14, Stanford, CA, 1995, pp. 1137–1145.
- [58] D. J. Hand, R. J. Till, A simple generalisation of the area under the roc curve for multiple class classification problems, Machine learning 45 (2) (2001) 171–186.
- [59] P. Langley, Elements of machine learning, Morgan Kaufmann, 1996.
- [60] M. M. Meerschaert, Mathematical modeling, Academic press, 2013.
- [61] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo, A geometric framework for unsupervised anomaly detection, in: Applications of data mining in computer security, Springer, 2002, pp. 77–101.
- [62] L. H. Chiang, E. L. Russell, R. D. Braatz, Fault detection and diagnosis in industrial systems, Springer Science & Business Media, 2000.
- [63] F. Salfner, M. Lenk, M. Malek, A survey of online failure prediction methods, ACM Computing Surveys (CSUR) 42 (3) (2010) 10.
- [64] M. Schwabacher, A survey of data-driven prognostics, in: Infotech@ Aerospace, 2005, p. 7002.
- [65] X.-S. Si, W. Wang, C.-H. Hu, D.-H. Zhou, Remaining useful life estimation—a review on the statistical data driven approaches, European Journal of Operational Research 213 (1) (2011) 1–14.
- [66] C. E. Rasmussen, Gaussian processes for machine learning.
- [67] B. Yegnanarayana, Artificial neural networks, PHI Learning Pvt. Ltd., 2009.
- [68] U. Yolcu, E. Egrioglu, C. H. Aladag, A new linear & nonlinear artificial neural network model for time series forecasting, Decision support systems 54 (3) (2013) 1340–1347.
- [69] F. Amato, A. López, E. M. Peña-Méndez, P. Vañhara, A. Hampl, J. Havel, Artificial neural networks in medical diagnosis, Journal of applied biomedicine 11 (2) (2013) 47–58.

- [70] M. H. Esfe, M. Afrand, W.-M. Yan, M. Akbari, Applicability of artificial neural network and nonlinear regression to predict thermal conductivity modeling of al 2 o 3–water nanofluids using experimental data, *International Communications in Heat and Mass Transfer* 66 (2015) 246–249.
- [71] A. Carrascal, A. Díez, A. Azpeitia, [Unsupervised methods for anomalies detection through intelligent monitoring systems](#), in: *International Conference on Hybrid Artificial Intelligence Systems*, Springer, Berlin, Heidelberg, 2009, pp. 137–144. doi:10.1007/978-3-642-02319-4_17.
URL http://dx.doi.org/10.1007/978-3-642-02319-4_17
- [72] A. Carrascal, A. Díez, J. Font, D. Manrique, [Evolutionary generation of fuzzy knowledge bases for diagnosing monitored railway systems](#), in: *Condition Monitoring and Diagnostic Engineering Management, COMADEM*, 2009, Vol. 12, COMADEM, 2009.
URL <http://www.gbv.de/dms/tib-ub-hannover/634600168.pdf>
- [73] U. ETSII, [Industriales Research Meeting 2016](#), ETSI Industriales, Madrid, 2016.
URL <http://oa.upm.es/40073/>
- [74] A. Diez-Olivan, J. A. Pagan, R. Sanz, B. Sierra, [Data-driven prognostics using a combination of constrained k-means clustering, fuzzy modeling and lof-based score](#), *Neurocomputing* 241 (2017) 97–107. doi:10.1016/j.neucom.2017.02.024.
URL <http://www.sciencedirect.com/science/article/pii/S0925231217302941>
- [75] A. Diez, A. Carrascal, [A multiclassifier approach for drill wear prediction](#), in: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, Springer, Berlin, Heidelberg, 2012, pp. 617–630. doi:10.1007/978-3-642-31537-4_48.
URL http://dx.doi.org/10.1007/978-3-642-31537-4_48
- [76] A. Diez, N. L. D. Khoa, M. M. Alamdari, Y. Wang, F. Chen, P. Runcie, [A clustering approach for structural health monitoring on bridges](#), *Journal of Civil Structural Health Monitoring* 6 (3) (2016) 429–445. doi:10.1007/s13349-016-0160-0.
URL <http://dx.doi.org/10.1007/s13349-016-0160-0>
- [77] N. Galarza, B. Rubio, A. Diez, F. Boto, D. Gil, J. Rubio, E. Moreno, et al., [Implementation of signal processing methods in a structural health monitoring \(shm\) system based on ultrasonic guided waves for defect detection in different materials and structures](#), *The e-Journal of Nondestructive Testing & Ultrasonics*.
URL <http://www.ndt.net/search/docs.php3?showForm=off&id=20159>
- [78] D. E. Barber, Shipboard condition based maintenance and integrated power system initiatives, Ph.D. thesis, Massachusetts Institute of Technology (2011).
- [79] W. C. Greene, Evaluation of non-intrusive monitoring for condition based maintenance applications on us navy propulsion plants, Ph.D. thesis (2005).
- [80] J. He, A.-H. Tan, C.-L. Tan, S.-Y. Sung, [On quantitative evaluation of clustering systems](#), in: *Clustering and information retrieval*, Springer, Berlin, Heidelberg, 2004, pp. 105–133. doi:10.1007/978-1-4613-0227-8_4.
URL http://dx.doi.org/10.1007/978-1-4613-0227-8_4

- [81] C. C. Aggarwal, Proximity-based outlier detection, in: Outlier Analysis, Springer, 2013, pp. 101–133.
- [82] C. Legány, S. Juhász, A. Babos, [Cluster validity measurement techniques](#), in: Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, AIKED’06, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2006, pp. 388–393.
URL <http://dl.acm.org/citation.cfm?id=1364262.1364328>
- [83] S. Basu, I. Davidson, K. Wagstaff, [Constrained clustering: Advances in algorithms, theory, and applications](#), CRC Press, 2008.
URL <http://dl.acm.org/citation.cfm?id=1404506>
- [84] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al., [Constrained k-means clustering with background knowledge](#), in: ICML, Vol. 1, 2001, pp. 577–584.
URL <http://dl.acm.org/citation.cfm?id=645530.655669>
- [85] P. Bradley, K. Bennett, A. Demiriz, [Constrained k-means clustering](#), Microsoft Research, Redmond (2000) 1–8.
URL <https://www.microsoft.com/en-us/research/publication/constrained-k-means-clustering/>
- [86] P. Cingolani, J. Alcalá-Fdez, jfuzzylogic: a java library to design fuzzy logic controllers according to the standard for fuzzy control programming, International Journal of Computational Intelligence Systems 6 (sup1) (2013) 61–75.
- [87] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, [Lof: identifying density-based local outliers](#), in: ACM sigmod record, Vol. 29, ACM, 2000, pp. 93–104. doi:10.1145/335191.335388.
URL <http://doi.acm.org/10.1145/342009.335388>
- [88] J. L. Fleiss, B. Levin, M. C. Paik, [Statistical methods for rates and proportions](#), John Wiley & Sons, 2013. doi:10.1002/0471445428.
URL <http://dx.doi.org/10.1002/0471445428>
- [89] F. Tian, M. Voskuijl, Automated generation of multiphysics simulation models to support multidisciplinary design optimization, Advanced Engineering Informatics 29 (4) (2015) 1110–1125.
- [90] S. Yin, G. Wang, H. Gao, [Data-driven process monitoring based on modified orthogonal projections to latent structures](#), IEEE Transactions on Control Systems Technology 24 (4) (2015) 1480–1487. doi:10.1109/TCST.2015.2481318.
URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7297846&isnumber=7488303>
- [91] S. C. Chapra, R. P. Canale, Numerical methods for engineers, Vol. 2, McGraw-Hill New York, 2012.
- [92] A. T. Azar, S. Vaidyanathan, Computational intelligence applications in modeling and control, Springer, 2015.

- [93] Q. Zhu, A. T. Azar, Complex system modelling and control through intelligent soft computations, Springer, 2015.
- [94] F. Rousseaux, Big data and data-driven intelligent predictive algorithms to support creativity in industrial engineering, Computers & Industrial Engineering.
- [95] H. E. P. Espinosa, J. R. Ayala-Solares, The power of natural inspiration in control systems, in: Nature-Inspired Computing for Control Systems, Springer, 2016, pp. 1–10.
- [96] A. Brabazon, M. O'Neill, S. McGarraghy, Natural Computing Algorithms, Springer, 2015.
- [97] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, Time series analysis: forecasting and control, John Wiley & Sons, 2015.
- [98] P. Hayton, S. Utete, D. King, S. King, P. Anuzis, L. Tarassenko, Static and dynamic novelty detection methods for jet engine health monitoring, Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 365 (1851) (2007) 493–514.
- [99] L. Guo, N. Li, F. Jia, Y. Lei, J. Lin, A recurrent neural network based health indicator for remaining useful life prediction of bearings, Neurocomputing 240 (2017) 98–109.
- [100] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, Long short term memory networks for anomaly detection in time series, in: Proceedings, Presses universitaires de Louvain, 2015, p. 89.
- [101] D. T. Shipmon, J. M. Gurevitch, P. M. Piselli, S. T. Edwards, Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data, arXiv preprint arXiv:1708.03665.
- [102] P. A. Whigham, et al., Grammatically-based genetic programming, in: Proceedings of the workshop on genetic programming: from theory to real-world applications, Vol. 16, Citeseer, 1995, pp. 33–41.
- [103] A. Carrascal, J. Font, D. Pelta, Evolutionary physical model design, in: Research and Development in Intelligent Systems XXVI, Springer, 2010, pp. 487–492.
- [104] P. Bentley, Evolutionary design by computers, Morgan Kaufmann, 1999.
- [105] V. Çelik, E. Arcaklioğlu, Performance maps of a diesel engine, Applied Energy 81 (3) (2005) 247–259.
- [106] A. K. Choudhary, J. A. Harding, M. K. Tiwari, Data mining in manufacturing: a review based on the kind of knowledge, Journal of Intelligent Manufacturing 20 (5) (2009) 501–521.
- [107] G. W. Vogl, B. A. Weiss, M. Helu, [A review of diagnostic and prognostic capabilities and best practices for manufacturing](#), Journal of Intelligent Manufacturing None (2016) 1–17. doi:10.1007/s10845-016-1228-8. URL <http://dx.doi.org/10.1007/s10845-016-1228-8>
- [108] R. Kothamasu, S. H. Huang, W. H. VerDuin, System health monitoring and prognostics—a review of current paradigms and practices, in: Handbook of maintenance management and engineering, Springer, 2009, pp. 337–362.

- [109] A. Das, J. Maiti, R. Banerjee, Process monitoring and fault detection strategies: a review, *International Journal of Quality & Reliability Management* 29 (7) (2012) 720–752.
- [110] S. Lee, Y. Ng, Hybrid case-based reasoning for on-line product fault diagnosis, *The International Journal of Advanced Manufacturing Technology* 27 (7-8) (2006) 833–840.
- [111] A. Krishnakumari, A. Elayaperumal, M. Saravanan, C. Arvindan, [Fault diagnostics of spur gear using decision tree and fuzzy classifier](#), *The International Journal of Advanced Manufacturing Technology* 89 (9) (2017) 3487–3494. doi:[10.1007/s00170-016-9307-8](#). URL <http://dx.doi.org/10.1007/s00170-016-9307-8>
- [112] J. Banks, J. Hines, M. Lebold, R. Campbell, C. Begg, Failure modes and predictive diagnostics considerations for diesel engines, Tech. rep., DTIC Document (2001).
- [113] T. Mitchell, *Machine Learning*, McGraw Hill, 1997.
- [114] N. Jones, Y.-H. Li, A review of condition monitoring and fault diagnosis for diesel engines, *Tribotest* 6 (3) (2000) 267–291.
- [115] K. Tidriri, N. Chatti, S. Verron, T. Tiplica, Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges, *Annual Reviews in Control* 42 (2016) 63–81.
- [116] I. Irigoien, B. Sierra, C. Arenas, Towards application of one-class classification methods to medical data, *The Scientific World Journal* 2014.
- [117] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, in: *ACM SIGMOD Record*, Vol. 29, ACM, 2000, pp. 427–438.
- [118] B. W. Silverman, *Density estimation for statistics and data analysis*, Vol. 26, CRC press, 1986.
- [119] G. R. Terrell, D. W. Scott, Variable kernel density estimation, *The Annals of Statistics* (1992) 1236–1265.
- [120] D. Scott, *Multivariate density estimation: Theory, practice and visualisation*. John Wiley and sons, New York.
- [121] D. M. Bashtannyk, R. J. Hyndman, Bandwidth selection for kernel conditional density estimation, *Computational Statistics & Data Analysis* 36 (3) (2001) 279–298.
- [122] J. Babaud, A. P. Witkin, M. Baudin, R. O. Duda, Uniqueness of the gaussian kernel for scale-space filtering, *IEEE Transactions on pattern analysis and machine intelligence* (1) (1986) 26–33.
- [123] R. Unnthorsson, T. P. Runarsson, M. T. Jonsson, Model selection in one-class ν -svm using rbf kernels, in: *Proc. 16th Int. Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management*, 2003.
- [124] A. Munoz, J. M. Moguerza, One-class support vector machines and density estimation: the precise relation, in: *Progress in Pattern Recognition, Image Analysis and Applications*, Springer, 2004, pp. 216–223.

- [125] B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the fifth annual workshop on Computational learning theory, ACM, 1992, pp. 144–152.
- [126] N. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, *The American Statistician* 46 (3) (1992) 175–185.
- [127] P. Bhattacharya, Y. Mack, Weak convergence of k-nn density and regression estimators with varying k and applications, *The Annals of Statistics* (1987) 976–994.
- [128] C. R. Farrar, K. Worden, An introduction to structural health monitoring, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365 (1851) (2007) 303–315.
- [129] H. Sohn, C. R. Farrar, F. M. Hemez, D. D. Shunk, D. W. Stinemates, B. R. Nadler, J. J. Czarnecki, A review of structural health monitoring literature: 1996-2001, Los Alamos National Laboratory Los Alamos, NM, 2004.
- [130] D. Collings, Lessons from historical bridge failures, in: Proceedings of the ICE-Civil Engineering, Vol. 161, Thomas Telford, 2008, pp. 20–27.
- [131] D. M. Frangopol, M. Liu, Maintenance and management of civil infrastructure based on condition, safety, optimization, and life-cycle cost, *Structure and infrastructure engineering* 3 (1) (2007) 29–41.
- [132] K. Worden, G. Manson, The application of machine learning to structural health monitoring, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365 (1851) (2007) 515–537.
- [133] R. Jafarkhani, S. F. Masri, Finite element model updating using evolutionary strategy for damage detection, *Computer-Aided Civil and Infrastructure Engineering* 26 (3) (2011) 207–224.
- [134] H. Wenzel, Health monitoring of bridges, John Wiley & Sons, 2008.
- [135] A. W. Moore, An introductory tutorial on kd-trees.
- [136] J. Fourier, Mémoire sur le refroidissement séculaire du globe terrestre, *Ann. Chim. Phys.*,(2) 13 (1820) 418–437.
- [137] J. W. Cooley, J. W. Tukey, An algorithm for the machine calculation of complex fourier series, *Mathematics of computation* 19 (90) (1965) 297–301.
- [138] E. S. Madsen, A. Bilberg, D. G. Hansen, Industry 4.0 and digitalization call for vocational skills, applied industrial engineering, and less for pure academics, in: 5th World Conference on Production and Operations Management P&OM, 2016.
- [139] S. Yin, S. X. Ding, D. Zhou, Diagnosis and prognosis for complicated industrial systems part i, *IEEE Transactions on Industrial Electronics* 63 (4) (2016) 2501–2505.
- [140] S. Yin, S. X. Ding, D. Zhou, Diagnosis and prognosis for complicated industrial systems part ii, *IEEE Transactions on Industrial Electronics* 63 (5) (2016) 3201–3204.

- [141] K. Severson, P. Chaiwatanodom, R. D. Braatz, Perspectives on process monitoring of industrial systems, *Annual Reviews in Control* 42 (2016) 190–200.
- [142] F. Serdio, E. Lughofer, A.-C. Zavoianu, K. Pichler, M. Pichler, T. Buchegger, H. Efendic, Improved fault detection employing hybrid memetic fuzzy modeling and adaptive filters, *Applied Soft Computing* 51 (2017) 60–82.
- [143] K. Wang, H. L. Gelgele, Y. Wang, Q. Yuan, M. Fang, A hybrid intelligent method for modelling the edm process, *International Journal of Machine Tools and Manufacture* 43 (10) (2003) 995–999.
- [144] V. Vera, J. Sedano, E. Corchado, R. Redondo, B. Hernando, M. Camara, A. Laham, A. E. Garcia, A hybrid system for dental milling parameters optimisation, in: *International Conference on Hybrid Artificial Intelligence Systems*, Springer, 2011, pp. 437–446.
- [145] S. Y. Kung, *Kernel methods and machine learning*, Cambridge University Press, 2014.
- [146] J. Marchewka, T. Watanabe, V. Ferrante, I. Estevez, Welfare assessment in broiler farms: Transect walks versus individual scoring, *Poultry science* 92 (10) (2013) 2588–2599.
- [147] N. Meinshausen, Quantile regression forests, *Journal of Machine Learning Research* 7 (Jun) (2006) 983–999.
- [148] A. Diez-Olivan, M. Penalva, F. Veiga, L. Deitert, R. Sanz, B. Sierra, [Kernel Density-Based Pattern Classification in Blind Fasteners Installation](#), Springer International Publishing, Cham, 2017, pp. 195–206. doi:10.1007/978-3-319-59650-1_17.
URL http://dx.doi.org/10.1007/978-3-319-59650-1_17