

Beiträge zum Stuttgarter Maschinenbau

Martin Lukas

# Uncertainty PERMEATED - Explainable AI in a Condition Monitoring Framework for Industrial Assets



stuttgarter  
**maschinenbau**  
interdisziplinär und vielfältig



**University of Stuttgart**  
Institute for Control Engineering  
of Machine Tools and  
Manufacturing Units (ISW)





University of Stuttgart  
Germany



**Beiträge zum Stuttgarter Maschinenbau**

**Vol. 25**

Editors: Prof. Dr.-Ing. Oliver Riedel  
Prof. Dr.-Ing. Alexander Verl  
Jun.-Prof. Dr. rer. nat. Andreas Wortmann

Martin Lukas

**Uncertainty PERMEATED - Explainable AI  
in a Condition Monitoring Framework for  
Industrial Assets**

Fraunhofer Verlag

**Contact:**

Institute for Control Engineering of  
Machine Tools and Manufacturing Units ISW  
Seidenstr. 36  
70174 Stuttgart  
info@isw.uni-stuttgart.de  
<https://www.isw.uni-stuttgart.de>

Cover illustration: © TRUMPF SE + Co. KG

**Bibliographic information of the German National Library:**

The German National Library has listed this publication in its Deutsche Nationalbibliografie; detailed bibliographic data is available on the internet at [www.dnb.de](http://www.dnb.de).

ISSN: 2750-655X

ISBN: 978-3-8396-1990-2

D 93

Zugl.: Stuttgart, Univ., Diss., 2023

Print and finishing: Fraunhofer-Druckerei, Stuttgart

The book was printed with chlorine- and acid-free paper.

**© Fraunhofer Verlag, 2024**

Nobelstrasse 12  
70569 Stuttgart  
verlag@fraunhofer.de  
[www.verlag.fraunhofer.de](http://www.verlag.fraunhofer.de)

is a constituent entity of the Fraunhofer-Gesellschaft, and as such has no separate legal status.

Fraunhofer-Gesellschaft zur Förderung  
der angewandten Forschung e.V.  
Hansastraße 27 c  
80686 München  
[www.fraunhofer.de](http://www.fraunhofer.de)

All rights reserved; no part of this publication may be translated, reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. The quotation of those designations in whatever way does not imply the conclusion that the use of those designations is legal without the consent of the owner of the trademark.

## Preface

The German economy is well-known throughout the world for its plant and mechanical engineering. With its two mechanical engineering faculties housing 42 institutes, the University of Stuttgart is the largest university institution for mechanical engineering in Germany. Our scientific excellence in this field is based on our numerous doctoral students and their outstanding dissertations. Many of these dissertations arise out of local, national and international collaborations with renowned universities and non-university research institutions, such as the German Aerospace Center, the Fraunhofer-Gesellschaft and the Max Planck Society. The fields covered by the dissertations range from Bio-Engineering, Energy Engineering, Automotive Engineering, Cybernetics and System Engineering, Product Development and Design, and Production Engineering to Process Engineering, and are based on the six main research areas of Advanced Systems Engineering, Autonomous Production, Software-Defined Manufacturing, Resilient Supply, Biointelligence and Decarbonization of Industry. The research findings from the dissertations aim to develop customer-specific, product-, process- and employee-oriented technologies in a targeted and timely manner.

Many of the dissertations written within the framework of the research work at the institutes are published in this series »Beiträge zum Stuttgarter Maschinenbau«. Our wish for the doctoral candidates at the two faculties of Stuttgarter Maschinenbau is that their dissertations in the field of mechanical engineering will be recognized by the wider professional community as authoritative contributions and thus contribute to establishing a new standard of knowledge.

For Stuttgarter Maschinenbau



Stefan Weihe



Oliver Riedel

## Foreword by the editors

With the publication series »Beiträge zum Stuttgarter Maschinenbau«, the Institute for Control Engineering of Machine Tools and Manufacturing Units at the University of Stuttgart (ISW) reports on its research findings. The institute supports a wide variety of approaches to control engineering and industrial automation as well as the use of modern methods of information management. All of these approaches involve a constant exchange between basic research and application-oriented development, ensuring a continuous transfer of technology into practice.

Thus, the dissertations originating at the ISW are continued by the present directors of the institute, and meanwhile in the fourth generation, in the proven conception initiated by the founder of the ISW Prof. Stute and his successor Prof. Pritschow in 1972.

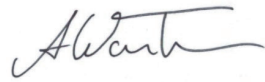
We would like to thank Martin Lukas M.Sc. for his work on the volume, the publisher for including this series of publications in its list, and the printers for the immaculate and speedy production of the volume. May the book meet with a positive reception within the wider professional community.



Alexander Verl



Oliver Riedel



Andreas Wortmann

# **Uncertainty PERMEATED - Explainable AI in a Condition Monitoring Framework for Industrial Assets**

Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik  
der Universität Stuttgart zur Erlangung  
der Würde eines Doktor-Ingenieurs (Dr.-Ing.)  
genehmigte Abhandlung

Vorgelegt von  
**Martin Lukas**  
aus Peine

Hauptberichter: Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl

Mitberichter: Univ.-Prof. Dr.-Ing. habil. Marco Huber

Tag der mündlichen Prüfung: 05. Dezember 2023

Institut für Steuerungstechnik der Werkzeugmaschinen und  
Fertigungseinrichtungen der Universität Stuttgart

2024



---

# Abstract

The first chapter introduces the topic of condition-based maintenance and contextualizes the importance of this technique, especially for critically important, complex and costly systems like machine tools.

Condition-based maintenance can be seen as a special case of diagnosis and data analysis. Consequently, the second chapter introduces terms and definitions, which serve as foundation for the following discussion.

The third chapter presents the state of research and a detailed review of publications in the context of data-driven diagnostics for condition-based maintenance. Different ideas behind and the purpose of model-based diagnostics, as well as signal-based diagnostics are outlined.

Chapter 4 focuses on uncertainty, which is the main challenge in condition-based maintenance. Recommending a maintenance action has potentially costly real-world impacts. It is therefore necessary be aware of the risks of decisions.

Uncertainty about the real state of the system seems to be inherent to the task of condition-monitoring. The lack of interpretability and auditability of decisions and the reasons for them are identified as main obstacles for a more widespread adoption of data-driven techniques. Subsequently chapter 5 introduces a diagnostics framework called PERMEATED, which embraces these results and is designed to deal with the existing uncertainties by incorporating them and emphasizing the importance of trust. The application of this framework to a real world application for machine tools is presented.

Chapter 6 discusses some existing machine learning approaches for condition-monitoring applications and applies them to a particular task regarding the dynamic behavior of a machine tool drive axis. Their performance is compared to an alternative, PERMEATED-compatible method, called SLIM. A different approach to satisfy the principles of the PERMEATED



diagnostics process is given in the last section of the chapter. Instead of using inherently interpretable machine learning models, this chapter uses so-called explainers to retrieve explanations from opaque machine learning models.

Chapter 7 summarizes the results of this thesis with regards to the task of condition monitoring for industrial assets and concludes with the identification of areas, where further research is necessary to make the application of techniques of machine learning more applicable for the task of condition-based maintenance.

---

# Kurzfassung

Das erste Kapitel führt in das Thema zustandsbasierte Wartung ein und kontextualisiert die Wichtigkeit der Methoden für produktionskritische, komplexe und kapitalintensive Systeme wie Werkzeugmaschinen.

Die zustandsbasierte Wartung kann als eine spezielle Form der Diagnose und Datenanalyse gesehen werden. Dieser Intention folgend, gibt das zweite Kapitel einen Abriss über die relevanten Definitionen und Konzepte, die als Diskussionsgrundlage der Arbeit dienen.

Im dritten Kapitel wird der Stand der Wissenschaft und Technik präsentiert und hierbei insbesondere auf Arbeiten im Kontext der datengetriebenen Diagnose zum Zweck der zustandsbasierten Wartung eingegangen. Dabei wird der Unterscheidung von modell- und signal-basierten Ansätzen zur Diagnose besonderes Augenmerk gewidmet.

Unsicherheit ist Thema des vierten Kapitels, welche die dominante Herausforderung für alle Ansätze der zustandsbasierten Wartung darstellt. Da eine fälschliche Wartungs- oder Reperaturempfehlung kostspielige Auswirkungen haben kann, müssen die Risiken integraler Bestandteil jeglicher Abwägung einer solchen Empfehlung sein.

Unsicherheit über den realen Zustand eines Systems scheint inhärenter Bestandteil des Problems der Zustandsrekonstruktion zu sein. Das Fehlen von Interpretationsmöglichkeiten für Entscheidungen wird als Hinderungsgrund für eine Verbreitung daten-getriebener Methoden identifiziert. Das fünfte Kapitel führt einen neuen Rahmen für Diagnosen namens PERMEATED ein, welcher auf dieser Erkenntnis gründet und dergestalt ist, dass Unsicherheiten in den

Entscheidungsprozess eingebunden werden können. Das „Vertrauen“ der Anwender in das System wird dabei durch die Verwendung interpretierbarer Methoden gewährleistet. Dieses Rahmenwerk wird für Werkzeugmaschinen angewandt.

In Kapitel sechs wird die Eignung einiger weitverbreiteter Machine Learning Methoden für die Diagnose des dynamischen Verhaltens einer Werkzeugmaschine untersucht und mit einem alternativen, interpretierbaren Ansatz, SLIM, verglichen. Anschließend werden nicht inhärent interpretierbare Machine Learning Methoden, sondern sogenannte Explainer verwendet, um interpretierbare Proxies aus opaken Modellen zu generieren.

Das siebte Kapitel fasst die Resultate der Thesis mit Hinblick auf die zustandsbasierte Wartung von industrieller Ausrüstung zusammen und identifiziert Bereiche für zukünftige Forschung, um den Mehrwert von Machine Learning für die zustandsbasierte Wartung zu steigern.

---

# Contents

<b>Abstract</b> . . . . .	<b>iii</b>
<b>Kurzfassung</b> . . . . .	<b>v</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Problem . . . . .	3
1.2 Requirements and Goals . . . . .	4
<b>2 Terms, Foundations and Systematization of Diagnosis</b> . . . . .	<b>9</b>
2.1 Definitions . . . . .	9
2.2 Transfer to machine tools with focus on drives . . . . .	11
2.3 Modes of Diagnosis . . . . .	12
2.3.1 Observation by the operator . . . . .	13
2.3.2 Measuring instrumentation/Inspection procedures . . . . .	13
2.3.3 Benchmark workpieces . . . . .	13
2.3.4 External Sensors . . . . .	14
2.3.5 Drive-based diagnostics . . . . .	14
2.4 Diagnostics approaches . . . . .	14
2.4.1 Model-Based Diagnostics . . . . .	16
2.4.2 Signal-based Diagnostics . . . . .	17
2.5 Maintenance . . . . .	18
2.5.1 Causes for degradation . . . . .	20
2.5.2 Types of damages . . . . .	22

- 2.5.3 Condition-based maintenance . . . . . 23
- 2.5.4 Prioritizing diagnostics needs from a maintenance perspective . . . . . 25
- 3 State of Research . . . . . 29**
  - 3.1 Model-based Diagnostics . . . . . 29
    - 3.1.1 Damage Processes . . . . . 30
    - 3.1.2 System Identification . . . . . 30
    - 3.1.3 Bayesian Filters . . . . . 32
  - 3.2 Signal-based Diagnostics . . . . . 34
  - 3.3 Feature Creation . . . . . 34
  - 3.4 Feature Selection . . . . . 37
  - 3.5 Regularization . . . . . 38
  - 3.6 Novelty Detection . . . . . 39
  - 3.7 Case-Based Reasoning . . . . . 42
  - 3.8 Classification . . . . . 44
  - 3.9 Regression . . . . . 46
  - 3.10 Method Selection . . . . . 50
  - 3.11 Summary . . . . . 50
- 4 Uncertainty . . . . . 53**
  - 4.1 Sources of Uncertainty . . . . . 54
    - 4.1.1 Simplified show-case model . . . . . 54
    - 4.1.2 Present uncertainty . . . . . 55
    - 4.1.3 Future uncertainty . . . . . 56
    - 4.1.4 Modeling uncertainty . . . . . 56
    - 4.1.5 Prediction method uncertainty . . . . . 57
  - 4.2 Uncertainty-related Activities . . . . . 57
    - 4.2.1 Uncertainty representation and interpretation . . . . . 57
    - 4.2.2 Uncertainty quantification . . . . . 58
    - 4.2.3 Uncertainty propagation . . . . . 58
    - 4.2.4 Uncertainty management . . . . . 59

---

4.3	Interpretation of Uncertainty as Probabilities . . . . .	60
4.3.1	Physical probabilities . . . . .	60
4.3.2	Subjective probabilities . . . . .	61
4.3.3	Logical probabilities . . . . .	62
4.4	Interpretation of Uncertainty via Fuzzy Set Theory . . . . .	63
4.5	Remaining Useful Life . . . . .	65
4.5.1	Generating an Estimate of the Remaining Useful Life . . . . .	67
4.5.2	State Estimation . . . . .	70
4.5.3	State Prediction . . . . .	71
4.5.4	Determining the End of Life . . . . .	71
4.6	Model Validation . . . . .	75
4.7	Model Calibration . . . . .	79
4.7.1	Least-squares methods . . . . .	80
4.7.2	Likelihood Method . . . . .	82
4.7.3	Regularizers for Model Calibration . . . . .	83
4.7.4	A Third Perspective . . . . .	83
4.8	Challenges . . . . .	85
<b>5</b>	<b>PERMEATED . . . . .</b>	<b>89</b>
5.1	Trust in Technical Systems . . . . .	90
5.1.1	Explainability . . . . .	92
5.1.2	Accuracy . . . . .	93
5.2	Interpretability of Diagnostic Systems . . . . .	94
5.2.1	What is Interpretability? . . . . .	95
5.2.2	Why Interpretability? . . . . .	95
5.2.3	How could Interpretability be Measured? . . . . .	99
5.3	The PERMEATED-Framework . . . . .	101
5.3.1	Guiding Ideas . . . . .	103
5.3.2	Data Acquisition . . . . .	105
5.3.3	Data Manipulation . . . . .	105
5.3.4	State Detection . . . . .	107
5.3.5	Health Assessment . . . . .	107

- 5.3.6 Prognostic Stage . . . . . 108
- 5.3.7 Advisory Generation . . . . . 109
- 5.3.8 Responsible Decision-Maker . . . . . 110
- 5.4 AxDiag: An Industrial Expert System . . . . . 111
  - 5.4.1 Use Case: Vibration of Cascade-controlled Axis . . . . . 117
  - 5.4.2 AxDiag: Assets . . . . . 118
  - 5.4.3 AxDiag: Data Acquisition . . . . . 121
  - 5.4.4 AxDiag: Data Manipulation . . . . . 122
  - 5.4.5 AxDiag: State Detection . . . . . 122
  - 5.4.6 AxDiag: Health Assessment . . . . . 122
  - 5.4.7 AxDiag: Prognostic Assessment . . . . . 123
  - 5.4.8 AxDiag: Advisory Generation . . . . . 124
  - 5.4.9 AxDiag: Responsible Decision-Maker . . . . . 124
- 5.5 Comparison to an Industrial Best Practice . . . . . 125
  - 5.5.1 Threshold setting procedure . . . . . 126
  - 5.5.2 Results . . . . . 127
- 5.6 Discussion . . . . . 129
  
- 6 Interpretable Machine Learning . . . . . 131**
  - 6.1 Non-interpretable Machine Learning Algorithms . . . . . 133
    - 6.1.1 Support Vector Machines for classification . . . . . 133
    - 6.1.2 Gaussian Process . . . . . 137
    - 6.1.3 Logistical Regression . . . . . 138
    - 6.1.4 Application within the PERMEATED-framework . . . . . 139
  - 6.2 SLIM . . . . . 142
    - 6.2.1 Methodology . . . . . 143
    - 6.2.2 SLIM Integer Program . . . . . 144
    - 6.2.3 Operational Constraints . . . . . 145
    - 6.2.4 Application within the PERMEATED-framework . . . . . 147
  - 6.3 Explainers . . . . . 149
    - 6.3.1 LIME . . . . . 151
    - 6.3.2 SHAP Values . . . . . 152

6.3.3 Application within the PERMEATED-framework . . .	155
6.4 Summary . . . . .	157
<b>7 Summary and Outlook . . . . .</b>	<b>159</b>
<b>Bibliography . . . . .</b>	<b>165</b>
<b>List of Figures . . . . .</b>	<b>207</b>
<b>List of Tables . . . . .</b>	<b>209</b>





---

# 1 Introduction

While the first industrial revolution was about augmenting brawn with steam power, the fourth revolution is about augmenting brains with computerized systems. This digital transformation of the manufacturing industry poses daunting challenges and exceptional opportunities simultaneously. The drivers of this transformation are multi-faceted: manufacturers' demand for increased productivity, shorter time-to-consumer and thus time-to-market and increased predictability of their manufacturing capacities, consumers' demand for price competitive individualized goods—preferably instantaneously—and the struggle of producers of manufacturing systems, such as machine tools, for unique selling points and alternative business models in increasingly global markets. A promising route towards such unique selling points and new business models is thought to be the creation of services that create continuous revenue streams after the sale of investment products such as manufacturing equipment. To be valuable a service has to solve a problem. And this is where the second law of thermodynamics enters the picture: everything decays and eventually stops working. It is an inescapable feature of existence. What is not inescapable, however, is what effect the failure of a particular production system has on the output of that production line. The worst case scenario is the unexpected failure of some critical asset that stops the complete production process and where it takes a long time to identify and repair or replace the broken component or subsystem. A collection of techniques that is commonly referred to as "Condition Monitoring", tries to mitigate that worst case scenario altogether or ease its impact at least somewhat. All of these techniques are driven by different amounts

of information. In the context of drive-based condition monitoring, the main topic of this thesis, the sources of information are restricted to the set of signals that are available within the control of the drive under consideration anyway, which is hoped to make it into a good candidate for an after sale service. There are three main functions a condition monitoring system can fulfill

- fault detection,
- fault identification,
- fault prognosis.

If within the set of monitored signals an anomaly can be detected, the Condition Monitoring System (CMS) is said to fulfill the function of fault detection. The information about the presence of a fault in the system can be useful for preventing the continued production of possibly substandard products or to trigger an immediate shutdown in case of safety critical faults. The usefulness of the fault detection function, however is not as high as that of fault identification. If instead of simply detecting the presence of a fault in a system, the affected component or subsystem or even their fault mode can be identified, the downtime of a system can be reduced significantly. Although this is highly useful, especially for maintenance and quality management personal, the most significant contribution of a CMS is thought to be the function of fault prognosis. Fault prognosis deals with predicting the occurrence of a fault and tries to eliminate unplanned downtimes by taking preventive maintenance actions on components identified to be critical—but still working—either during the next scheduled maintenance shutdown or, if the critical component is not expected to survive until that date, during an opportune time in the production schedule. Techniques that are concerned with this set of problems are often referred to as "Prognostic and Health Management (PHM)". All three of these techniques require knowledge about the system under consideration. Heuristically, it can be stated that the more useful the function, the more knowledge is required. A core component for providing these functions is widely thought to be machine learning, which is still conspicuously absent in manufacturing industries.

Among the obstacles for a more thorough adoption of machine learning methods in the industrial context are the deployment of models and their integration into existing MES, the robustness of deployed models against variations in the real plants, the explainability and interpretability of the models, connectivity to edge computing sites and of course the availability of statistically significant amounts of relevant data for the initial training of models. In the context of fault detection, fault identification and fault prediction, the generation of data of plants in a faulty state can be prohibitively expensive or even infeasible due to ethical and safety concerns. It is the goal of this thesis to investigate opportunities and challenges for the adoption of certain forms of machine learning algorithms in an industrial setting.

## **1.1 Problem**

Machine tools represent major investments for the majority of their typical customers and are expected to last for at least a decade. The realized useful life is dependent on the specific usage patterns, operating and environmental conditions as well as contaminants on functional surfaces and in lubricants, which can adversely affect the useful life and lower it significantly below initial expectations. Systems are therefore needed to identify failures as soon as possible to prevent potentially costly consequences of a fault. The subject of this work is the measurement and especially analysis of the condition of the electromechanical components of the drives of a machine tool as basis for a condition monitoring tool that exclusively uses already built-in resources of the drives. The lack of additional sensors qualifies this approach as drives-based or "sensorless". This approach is intended to keep the additional complexity and integration expenditures to a minimum as well as to maintain maximum backwards compatibility. Economical considerations of every condition monitoring systems will also take the impact of that solution on the productivity of the machine into account; the longer the solutions demands the machines to remain in a non-productive state, the more negative its impact on customer acceptance and its return on investment will be. Additionally, the diagnostic system's output

have to be actionable to the addressees, who have to assume responsibility and sometimes liability for the consequences of their decisions.

## 1.2 Requirements and Goals

This work is supposed to contribute to an increase in the availability of machine tools by developing a condition monitoring framework for industrial assets using some methods from the explainable artificial intelligence realm to improve the efficiency and effectiveness of maintenance actions. To achieve this goal, the condition of the drives of machine tools is to be automatically analyzed as part of an integrated methodology for downtime prevention. This necessitates the detection and identification of incipient faults due to long-term degradation processes. The system's complexity is supposed to be held roughly constant. This is achieved by focusing primarily on measurements of system variables that are already available in the machine tool at the level of the numerical control. Previous works have shown that faults caused by wear and tear can be detected using the drives' sensors. But in most cases, there are unfortunately only measurements of signals available, which are the sum of component specific fault signals.

To satisfy these general needs, a systematic way to analyze the machine tools is needed. A primary concern is the availability of data which satisfies certain quality standards. The available data has to be reviewed and classified by the domain's quality experts, who also define requirements for indicators. While reactive maintenance schemes can passively wait for the violation of at least one of the quality-related indicators, predictive maintenance schemes rely on the active forecast of the point of time at which a component or system enters a fault mode. Therefore, a major component of the value of an indicator for these purposes is its behavior over time. This changes the question from whether a signal is indicative of a certain fault to how good this fault is predictable by the measured signal.

Complex systems, like the controllers of machine tools' drives, also measure signals that are not necessarily used for the control of the machine tools. They of-

ten help to prevent acute damages in machines, like temperatures that are held in certain bounds to protect for example electronics from permanent damage.

The fundamental principle of a drive is the conversion of torque-producing current into a force that is utilized for the positioning of the respective axis of the machine tool. Modern machine tools are typically controlled by CNCs. In most machine tools, either the resulting velocity or the position itself is measured. Some drives are equipped with measurement systems for both physical entities. The derivative of speed with respect to time results in the acceleration of the movement, the derivative of the acceleration with respect to time results in the jerk of movement. Programming systems for modern machine tools often offer the possibility of generating jerk-limited trajectories for the drives. Information about the usage of any unit under consideration is therefore available at this level in the information system architecture; position, velocity, acceleration and jerk of each individual program can be mapped to a load integral as a measure of actually experienced stress of components of the drives of machine tools. [1]

The state of degradation of machines tools can also be measured by the difference of the current observation relative to either historical records or to simulations of the unit under consideration. For this purpose a quantitative-parametric model of the component or machine has to be available and specific enough to warrant the detection and identification of certain fault modes in the outputs that are measurable for the real plant. The precondition for this sort of comparisons is the availability of a deterministic model covering time frames of interest. These models can also help to investigate the effects of different operating conditions on the measured outputs. Models of the behavior of a system over time can be regarded as a form of highly compressed domain-specific knowledge about the system.

Most analysis methods of control theory assume linear-time invariant systems as the mathematical foundation for the modeling of machines. While this assumption is convenient with respect to the possibilities of the methods it opens up, the validity of it has to be checked for the specific unit under consideration. Backlash in a rack-pinion system for example introduces a non-trivial non-linearity that impacts for example frequency response measurements of

the drive. Depending on the type of drive, transient operating conditions like the behavior of a drive during reversal of the direction of movement can be especially useful for diagnostic purposes. For ball-screw drives the tracking error between set point and actual position of the drive during positioning can give valuable insights into the degradation state of the drive.

Of special interest in the context of condition monitoring systems is the effect of operating conditions on wear and tear and of course the reverse—the effect of wear and tear on the observable behavior of the system. A loss of stiffness in a drive for example will have an influence of certain eigenfrequencies and can therefore negatively impact the performance of the drives' control system. In order to extract the degradation signals from the sum of such signals, which are often the only available measurement, filters are commonly chosen in an attempt to amplify the signal components associated with degradation of certain components. Bearings are a particularly good example for this approach. The characteristic frequencies of bearings are mostly determined by the geometrical properties and can often be obtained from the manufactures of these components or the technical documentation. For certain events, like the ball passing the outside of the cage, characteristic frequencies can be given, i.e. the so-called Ball Passing Frequency Outside (BPFO). If the fundamental frequency is known or can be determined from the measured signals, these frequencies can be checked for signs of change, indicating either a fault or a change in the operating conditions. The sensitivity of these characteristic frequencies has to be investigated for the fault or degradation modes of interest. Their statistical and stochastic properties are of special importance for determining the suitability of these signals for condition monitoring purposes.

This work's focus on the economical realization of actionable and "sensorless" condition monitoring systems for drives in machine tools has certain implications: A frequently used method for machine diagnostic is the diagnosis of vibrations that result from machine operations. These measurements contain information about the mechanical properties of the machine tool as well as degradation conditions. But as this form of measurement necessitates the integration of additional sensors into the machine tool, this route is not pursued

here because of the associated increase in complexity and cost. The analyses will use only information contained in measurements of the available sensors or information from already existing databases. Another implication is that there are additional constraints on frequently used diagnostic methods that produce "inexplicable" results.

To summarize the desired characteristics for a condition monitoring systems with respect to the presented considerations:

- **Integration in a Condition Monitoring Framework:** The developed analyses shall be integrated into a Condition Monitoring Framework that is usable for managing the quality of machine tools and their output. In principle other sources of data like visual or acoustical impressions could be used for the analysis. Given that the automation of the analysis process relies on indicators that are easily processed by an information processing system, the focus will be on quantifiable variables.
- **Usage of available measurements devices:** Only signals of sensors that are already integrated into the machine tools as well as information in already existing databases are used. Additional sensors are disregarded for the impact on costs and complexity.
- **Modularity of the diagnostic system:** Diagnostic modules are software components that could be executed in principle on any machine that meets the minimum requirements with regard to data storage, connectivity and processing power. An integration of the software into the machine tool itself is desirable but introduces certain problems like updatability and possible performance issues. A cloud-native solution appears to offer significant advantages.
- **Robustness of the analysis:** This requirement is paramount for the applicability of any analysis method in an industrial setting. Variations in the operating conditions, the environment, noise in the measurements, huge and inconsistent time spans in between measurements and rather



small sample sizes for the configuration of the analysis pose a challenging setting for any data processing and analysis scheme.

- **Identification of the affected component:** The goal of the diagnosis is aimed at least at the identification of the component affected by a fault. The mere detection of some fault in the machine, while valuable for the start of an investigation, does probably not offer the tangible advantages a condition monitoring system has to have to justify its development and implementation.
- **Actionable analyses:** The results of the diagnostic system have to be actionable for its addressees. This implies that the results are presented in the form of recommendations and a rationale for the given recommendations.
- **No decrease in productivity:** A low sensitivity of a measurement necessitates frequent measurements. This impacts the economics of the solution adversely. Ideally, a measurement should not increase the downtime of machines.

For industrial assets like machine tools, the impact of maintenance in terms of costs and loss of productivity has become a major part in the assessment of the total cost of ownership. Future machine tools will probably have to be designed with serviceability and diagnosability in mind to remain competitive. This requirement will most probably impact the software and hardware architecture of future machine tools in a substantial way.

---

## 2 Terms, Foundations and Systematization of Diagnosis

### 2.1 Definitions

Process and condition monitoring, although closely related, are two different processes and should be treated as such. The focus of this thesis will not be put on process monitoring. The definitions of maintenance in [2] will be used as basis for further discussions. A **fault** in a machine is a condition of a machine when any of its components or their assembly is degraded and exhibits an abnormal behavior. Additionally to the definition of a fault, the concept of **wear limit** is useful, which allows a given component of systems to be considered worn down, without it having to exhibit unwanted or even unsafe behaviors. A definition of **failure** is critical for condition monitoring systems. In general failure can be defined as the loss of the ability to perform at least one required function. Depending on the needs of different applications, failure can also be further distinguished into

- a hard failure, e.g. broken parts in the system,
- a soft failure, the system fails to meet the requirement of reliability.

The failure time can be decided either explicitly by a mathematically-defined failure criteria, e.g. a threshold, or, implicitly based on the historical cases that are considered to have failed. In the first case, the failure can be defined as exact thresholds on individual variables or a function of them, or it can be defined

as a probabilistic threshold based on requirements of system reliability. The failure criteria can be given as design specification or it can be estimated from historical cases of failure. In the second case, the failure time of the monitored system is not decided based on a single or multiple measurements or health state predictions, instead it is estimated from the failure time of historically failed cases of other instantiations of the system. In some cases the failure time is decided by subjective judgement. For example, the recorded time of system overhaul, or the time of a certain major maintenance action can be logged as the failure time.

A failure can be said to have occurred, if a monitored state of a system passes the threshold for the first time or only after multiple detections in consecutive measurements. If a failure condition is reached irreversibly, the system is said to have reached its end of life.

**Wear** is a mostly gradual damaging mechanism that starts with the commissioning of a system and will ultimately lead to its failure. The main driver of wear is a movement that creates a **load** on the system. External factors like overloads or adverse environmental conditions contribute to the damaging of a system and can be much less gradual.

**Monitoring** of a system is a rather common practice that has been used since at least the introduction of steam driven equipment, to keep people and machine out of harm's way. Normally, a critical variable of the system is continuously measured and compared to certain preset limits to trigger predefined actions. In this context the limits are immutable and do not evolve dynamically. Checking against those limits results in a binary condition and can trigger actions, that span the spectrum from issuing a warning to the initiation of an emergency shutdown.

Certain **phenomena**, or detectable patterns of behavior, allow for the identification of faults of components or the effect of wear on the system.

The term **indicator** usually refers, but not necessarily, to a unitless relative number that is used to describe one or many attributes of the system under consideration. Indicators are the basis for the identification of the system's state.

**Diagnostics** is a more complex and farther reaching concept than monitoring. It uses checks on and classifications of indicators or sets of indicators with the goal of obtaining a general view of the system.

**Prognosis** is the scientifically informed prediction about the evolution of states or the occurrence of events that will happen at a future time instant. The basis for a prognosis is the state of the system resulting from the prior diagnostics step. Dependent on the estimate of the current state and certain assumptions about the future loading of the system, probabilities about certain future events can be derived.

**Performance** is the capability of a machine defined by one or more characteristic quantities such as power, flow, efficiency or speed.

The **baseline** of a system are parameters or derived quantities obtained under specific equipment configurations and specified operating conditions and can be used as reference values for the system.

## **2.2 Transfer to machine tools with focus on drives**

In the context of machine tools, the monitoring of readily available drive signals might not be sufficient for an accurate diagnosis of its condition. Especially the change of certain characteristics during the life-cycle of a machine is of high importance for an assessment of its condition. The definition of the already experienced damage is different for a complete system than for its components. While a permanent change in the material make-up of a component can indicate the end of life of a component, this event may be of no or only little relevance to the functionality of the complete system. Only a significant change in the transfer behavior of commands of a machine tool can be regarded as a severe fault or, when significant enough, failure. Wear and tear in machine tools can normally be regarded as rather slow process. Collisions can induce a sudden drop in the expected remaining useful life, as the damage for a system will generally be not linear in the magnitude of the shock: a collision at full speed with a sturdy measurement apparatus for example will tend to have a more immediate and severe impact on the system health than even a prolonged operation

under normal conditions. The total load a component has experienced is more promising as an indicator for the status of a component than the chronological age of a component [1]. Non-linearity in the damage induced by stress also need to be accounted for.

For a complex system, it might be infeasible to understand and model every detail of the system's behavior. What's modeled therefore has necessarily to be limited to the overall operating mechanisms of the system and some more detailed models for critical components constituting the system under consideration. The diagnostic and prognostic capability to cover more of the system's failure modes is limited to the amount of available information about the system or the respective subsystem. A complex system may exhibit highly non-linear, stochastic behavior due to various reasons, such as manufacturing variations of the vast number of components, their assembly, the simultaneous occurrence and evolution of numerous fault and failure mode and non-linear relations between those factors and the measured signals. These characteristics pose a great challenge to basically all modeling methods. Additionally, the system behavior from different time instances may behave inconsistently even under identical operational conditions, due to time-variance of the system, which adds to the hardness of the problem.

## 2.3 Modes of Diagnosis

Machine diagnosis can be categorized into the following groups

- Observation by the operator,
- Measuring instrumentation/Inspection procedures,
- Benchmark workpieces,
- External Sensors,
- Drive-based diagnostics.

There are also certain techniques that cannot be fitted easily into one of these categories.

### **2.3.1 Observation by the operator**

One of the most common ways of monitoring a manufacturing system is the monitoring by an operator of the machine. As the typical operator will spend multiple hours a day near a manufacturing system, the operator will have an intimate knowledge of the system and will thus be able to detect a variety of changes in the system's behavior ranging from sound emissions to the detection of vibrations or the detection of an unusual number of warnings of the monitoring system.

### **2.3.2 Measuring instrumentation/Inspection procedures**

Inspection procedures are mostly used to test single components and can be associated with the need for complex measurement equipment. Their main area of application is in highly safety-critical environments like aviation or power plants. During routine maintenance procedures in these sectors single components will be tested intensively to detect incipient faults like material fatigue or cracks. During the inspections the systems under consideration can be disassembled, checked, replaced if needed and reassembled. Measuring instrumentation in the context of machine tools are more commonly used during manufacturing or the commissioning procedure. The instrumentation is not a part of the machine but is only added for the duration of the measurements and removed afterwards.

### **2.3.3 Benchmark workpieces**

Workpieces that are only manufactured for the purpose of checking certain geometries for diagnostics are benchmark workpieces. Certain information about the state of the machine tool can be inferred from the workpiece. Additionally, not only information about the machine tool itself, but also about the operational processes can be gathered from a benchmark piece. It is a common practice, though, to minimize the influence of the process to maximize the information content about the state of the drives.

### **2.3.4 External Sensors**

External sensors for the monitoring of manufacturing equipment are already a readily available commodity. Ball screws are probably among the components with the most advanced monitoring solutions offered commercially. Their bearings are typically monitored with vibration sensors and the obtained signals are processed in proprietary software. The external sensors are typically not used for any additional task.

### **2.3.5 Drive-based diagnostics**

Drive-based diagnostics uses already available signal sources in a machine tool, like information about currents, accelerations, speeds and positions of the drives. Additionally, the drive-based diagnostics can be divided into two different conceptual ideas: model-based and signal-based approaches. The available knowledge about systems can be a determining factor for the feasibility of certain approaches. Signal-based approaches do not assume a level of knowledge about the system far beyond basic equations of movement that can be inferred from information about the geometry of the systems and the general ability to record data. The proportionality constants of bearing elements relative to the speed of rotation of the motor can be seen as an example of this basic knowledge.

## **2.4 Diagnostics approaches**

The diagnostic process is generally triggered by detection of an anomaly during routine monitoring, routine analysis, randomized analysis or human perception. This detection is carried out by making comparison between the present descriptor of the machine and baseline values that are either chosen by experience, from manufacturer's specifications, commissioning tests or are computed from statistical data, e.g. long term averages. Two main approaches can be used for the diagnosis of machine tools:

- **Data-driven** approaches (trending, neural networks, statistical etc.). These methods are generally automated and do not require deep knowledge of the mechanism or fault initiation and propagation but do require training of the algorithms using large sets of observed fault data.
- **Model-based** approaches, which rely on an explicit representation fault behavior or symptoms, e.g. through fault models or correct behavior models.

The utilization of a combination of both approaches is possible and even warranted for some use cases.

The described methods can also be divided into **online/offline** capable methods. Online methods work during the regular operation of the machine tool, while offline methods necessitate an interruption of the regular operation. Some measurements can also conceivably be integrated into the regular operation of the machine tool without any disturbance of its productivity.

Another differentiating category is the **domain** in which the assessment of the system's state is taking place in. Most sensors will sample data with a fixed sampling rate in the time domain. These signals can be post-processed to enable the analysis of system characteristics in the frequency, order or time-frequency domain.

[3] advises to consider the following questions to decide on a suitable diagnostic approach:

- application of the equipment,
- end user for the diagnostic approach,
- monitoring technique,
- complexity of the knowledge to be modeled,
- need to have an explanatory model,
- availability of existing data with known faults and normal operation.



### **2.4.1 Model-Based Diagnostics**

There are at least two different concepts offered by model-based diagnostics for the task at hand. One is to create a model of the machine tool, the other is to create a model of the failure.

If a model of the machine tool is created, there is again a differentiation to be made between models that are based on physical-mathematical fundamental equations and models that try to recreate the input-output behavior of the system without any additional layer of interpretation being inferable from the resulting model parameters. The physics-based approach will in general require a greater amount of work and system knowledge to arrive at the same approximation quality as a black-box model, but also allows for an easier interpretation and a far better parametrization. A problem with physics-based models is that their approximation quality can suffer during the life-span of a system, when effects start to occur in the real systems that cannot be captured by a simple parameter adjustments of the fixed model. While black-box models may suffer from the same performance degradation, a retraining using newly acquired data might suffice in fixing that problem.

A model-based approach can be seen as an indirect method for the problem of system monitoring. While it is not strictly necessary to have a model for the detection or even identification of an incipient fault in a machine tool, knowledge about the relationship between a system component and its effect on measurements can allow for the detection of gradual changes in specific components. A sufficiently good machine model or digital twin might also allow for the detection of not yet problematic changes in system components.

Machine-Learning algorithms, like neural networks, support vector machine or extreme learning machines can be and have been used to create a mapping from inputs of a system to outputs.

Another approach is to use an observer to infer the state of a simulation model. If the difference between the simulated output of a physics-based model and the real model deviates too much, a change in the real system will be the most likely culprit. Also, repeated system identification with fixed dimension

and model classes can be seen a type of model-based diagnostics, as changes in the inferred parameters can be attributed to changes in the system under consideration.

Creating a model of the failure itself is another interesting alternative, as it does not assume intimate knowledge of the complete system, but only about the effect of the failure on the system. These models do typically comprise a much smaller number of equations and states.

Failure modeling is usually applied at the material or component level and derived from wear or failure mechanism. A typical model for mechanical components is a fault propagation model, such as crack or spall propagation models. Fault propagation models usually have to incorporate the loading condition as an input to correctly estimate the cumulative damage over time. The more constant the usage over the life span of an engineering system is, the easier it seems to be to detect faults and estimate their progression. Many characteristic parameters for failure models have to be identified using experimental data. Backed up by the first principles regarding relations of the measured data and experimental setting, it is possible to use a fewer number of experiments to identify the characteristics of the system compared to what is needed for data-driven methods. However, due to modeling assumptions, modeling errors and unforeseeable uncertainty in the application, the model may not be as accurate for the real application as it is in the experimental setup. A mitigating strategy is to accommodate the model by online parameter updating methods based on the measured condition data at runtime [4]. One of their main drawbacks can be seen in the fact that they are a rather narrowly focused description of only one part of the system at hand; non-modeled failures will thus remain undetectable.

## **2.4.2 Signal-based Diagnostics**

As most damages to a system component will consist of changes in the functional surfaces, a change in the emitted signals of these components can be detected. Bearings, for example, will emit impulse-like excitations, every time

the damaged surface of a bearing is passed by a rolling element. These impulse-like signals are generated periodically, if the base-frequency of the drive is kept constant and the resulting frequency can be inferred from the geometry of the components. This characteristic frequency can be detected at measuring locations that are spatially separated from the damaged bearing. Signals of this type can be gathered either with external or drive-internal sensors. The signal-based approach can be seen as the more direct approach compared to the model-based techniques. The idea behind signal-based diagnostics is to use various processing methods to tease out information that is already present in the measured signals, but not necessarily obvious without some form of filtering. Signal-based diagnostics typically comprises the creation of indicators, the selection of relevant features and the construction of some form of either classification or regression on the selected features. The creation of indicators can involve the application of certain transformation like the short-time Fourier or wavelet transformation on the raw signal to infer information in another domain. It could also involve various steps, like denoising the signal or resampling the time signal with a measured or inferred fundamental frequency of a rotational drive, to obtain a signal that behaves as if it was measured at fixed angular increments instead of at fixed temporal increments. After applying suitable transformations it is common to extract characteristic features of the resulting signals and to consider these as potential indicators. The next step in signal-based diagnostics typically involves the selection of relevant features that are used in the resulting statistical model, which can either be some sort of detector of novelty for the fault detection case, some sort of classifier for the fault identification case or some form of regression for the fault prediction task.

## **2.5 Maintenance**

Highly integrated production processes and the tight schedules used in manufacturing today put a premium on the reliability and efficiency of the equipment. Maintenance is supposed to keep equipment operating at or near their peak capacity. More formally, maintenance can be defined as the

"combination of all technical, administrative and managerial actions during the life-cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function." [5]

Historically, maintenance consisted mostly of cleaning, lubricating and calibrating the equipment. Repairs were mostly triggered after a failure had occurred. The equipment was designed with larger margins of error than is common today, breakdowns were relatively rare and repairs were rather easy due to the simple design of the equipment. A single equipment failure had only limited effect on production, because of the lack of integrated production processes.

The second world war had a large impact on production processes. To meet the larger production requirements of war industry, the level of automation and integration was increased considerably. Production lines were created by connecting equipment sequentially. But this also affected the severity of equipment failure, since a single failure could bring the whole production line to a hold. Preventive maintenance process, consisting mainly of scheduled equipment overhauls, started to emerge as a countermeasure to reduce the number of breakdowns and a way of reducing costs for maintenance [6].

Modern communication technology and the global competition of markets have made the modern degree of integration both possible and necessary. The increase in usable technologies is also reflected in an increase in the availability of maintenance processes which try to improve upon corrective and schedules maintenance strategies. In [5] maintenance processes are divided into different categories, which are depicted in Fig. 2.1. The main categories are preventive maintenance, which is done before a specific fault has been detected and triggered a maintenance call, and corrective maintenance that is done after a specific fault has been detected. Preventive maintenance is further divided into condition-based maintenance and predetermined maintenance. Predetermined maintenance does not rely on any sort of investigation into the actual state of the unit under consideration.

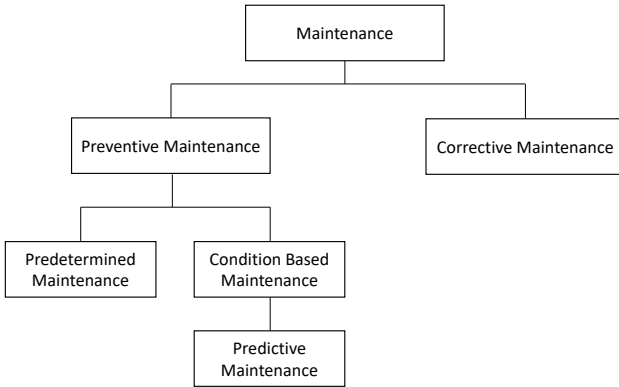


Figure 2.1: Taxonomy of maintenance actions. [5], adapted.

Preventive maintenance makes it possible to raise the reliability level of the production equipment. Effective preventive or even predictive maintenance enables the reduction of unplanned downtimes and the avoidance of penalties for delayed deliveries. The cost of preventive maintenance can rise dramatically, however, if the desired level of reliability is set too high. This necessitates the maintenance organization to decide on their preferred level of preventive maintenance as a function of the effects of equipment failure and the efficiency of preventive maintenance. For a more thorough introduction to maintenance and repair strategies, the reader is referred to the literature, for example [7].

### 2.5.1 Causes for degradation

There are two principal causes for the degradation of components with rolling contacts, like bearing and guide ways. One is the usage of the component according to their specifications during normal operations. Wear and tear is in this regard a special case of damage inflicted to the system. The second main cause for degradation is improper or unsuitable use or excessive workloads, which can lead to localized damages like plastic deformations of functional surfaces. The drives of a machine tool transform positioning signals into force or torque.

This causes a transmission of mechanical power via mechanical contacts and is thus the cause for the damage.

Degradation is also influenced by operating and environmental conditions. Aging is a further component of degradation that can influence the properties of lubricants and surfaces, i.e. through oxidation. Only effects of the normal operating regime can ever be used for predicting the condition of a machine tool or component in the future. The effects of other sources of damage, for example plastic deformations caused by improper usage, might be detectable by a condition monitoring system, but could only be predicted, if improper usage of a machine tool was assumed in a forecast. Damage modes that occur randomly, like pitting, can by their very nature only be considered probabilistically.

Tear is the loss of surface material primarily caused by mechanical interactions, i.e. by the application of forces and relative movement. If there exists friction in such a system, it is a tribological system and has to be analyzed with respect to the energy transformation processes by friction and to the loss of material that leads to changes in the geometry of the affected components and ultimately to a loss of functionality. [8, p.105]

Often, tribological systems are classified by their structure and the collective of their stresses. The structure consists of all bodies, solids and media of the system, the collective of stresses consists of load, relative velocities and temperature. Two frictional wear and tear mechanisms can be identified: body-to-body contact with or without lubricant and wear under the influence of abrasive particles. There are basically four damage types:

- abrasion by hard ridges on one of the bodies or by granules between them,
- fatigue of the microstructure by repeated application of mechanical stress in the border region,
- adhesion caused by molecular reactions of media in the zone of contact,
- ablation of material caused by excessive energy densities at the surface.

Most components of interest in machine tools have rolling contacts. For this type of contact abrasion and fatigue are the dominant types of damages the system will experience.

For a drive, this implies that the Hertzian contact pressure between rolling elements and guidance will introduce damages in the microstructure of the metallic building materials. Repeated application of stress to this area will facilitate the occurrence of pittings. Once these larger damages appear on the contact surfaces, the loss of functionality of this rolling contact becomes imminent. The very same damage phenomenon can be caused by improper usage and excessive stresses as well as by environmental conditions.

### **2.5.2 Types of damages**

To cluster certain phenomena of damages, a finer differentiation of types of damages is useful. There are two additional classes of degradation phenomena that are of special interest in the context of detecting damages: their periodicity.

#### **Periodic Damages**

Periodic types of damages necessitate the presence of a mechanic contact between several moved components. A rolling contact is the most classic and common example of this type of structure. A small, localized damage to the geometry, i.e. pitting, causes a jolt or localized change in the friction of the bodies while the damaged location on the surface is passed over. The impulse-like excitation is produced with every passing and is thus periodic with respect to the geometry of the components and, if the relative speed of the bodies is constant, also periodic with respect to time. In the latter case, the characteristic frequency of the train of impulses can be identified with tools of the frequency analysis. This frequency can be attributed directly to a mechanical component, if the geometrical characteristics of the components are taken into account. The frequency is a function of these characteristics and the fundamental rotational frequency or the translational speed, which is often known or inferable from measurements. It is not always possible to measure the effect of the impulse train directly, because vibrations in the surrounding material can be excited by it. The material will vibrate with its eigenfrequency, driven by the energy that is periodically extracted from the drive system [8].

## Aperiodic Damages

Aperiodic damages do exhibit phenomena that are not periodic, neither spatially nor temporally. They are therefore not associated with a characteristic, speed-proportional frequency. This type of damage can affect the drive at all positions. A typical example is a change in the friction of a bearing or the loss of stiffness of a component. Aperiodic types of damages exhibit a wide range of effects on the characteristics of the components. Aperiodic damages can be interpreted as ranging from slow to almost instantaneous changes of relevant linear or non-linear mechanical characteristics of the system. [8]

### 2.5.3 Condition-based maintenance

The intent of all types of preventive maintenance, such as condition based maintenance, is "to reduce the probability or the degradation of the functioning of an item" [5] and to prevent the irreversible damage to equipment. *Failure modes* are the effects that cause equipment failures such as adverse changes in the material of components or cracks. Some causes of such failure modes have been discussed in section 2.5.1.

As a general rule, the more complex a system is, the more complex its failure modes tend to be. In simple equipment, time-dependent failure modes dominate. In complex systems, the number of possible failure modes and their complex interactions make it unlikely that relying on regularly scheduled equipment overhauls suffices to keep such equipment reliable. The more knowledge about failure modes and their ensuing phenomena exists, the more likely it is to detect impeding failures with one of the measurement types discussed in Section 2.3.

The deterioration of equipment can be visualized conceptually with the Potential Failure curve, Fig. 2.2, with the two dominant points of "potential failure" (P) and "failure" (F).

The ability to generate an index of health for the equipment and of creating a large enough temporal gap between these two points determines the feasibility



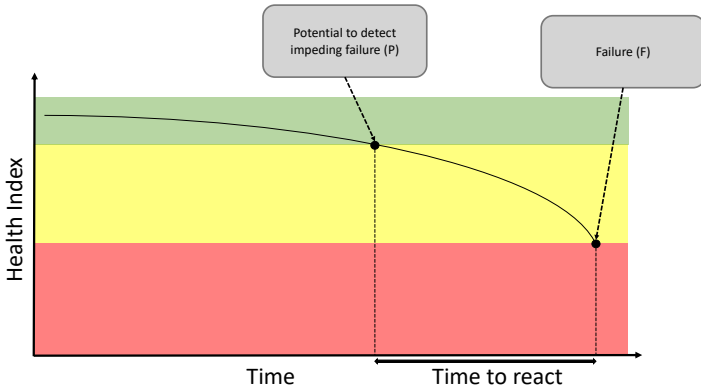


Figure 2.2: Potential Failure Curve. [5], adapted.

of condition-based maintenance strategies. If there is no health index or the timespan is too short to react, preventive maintenance cannot be used productively. Of course there is a multitude of attempts for generating such health indexes and for maximizing the timespan by the use of diverse mathematical methods that contribute to an increase in usability of condition based maintenance strategies. The state of research in this field and conceptual challenges to these attempts will be presented in Chapter 3 and 4, respectively.

Here, it is sufficient to note that these systems are always a function of the available information about the system: If there is no system expert that is able to correctly label the condition of a possibly very complex system, there are only statistical tools that can be used to detect anomalies, but there is no principled way to better categorize them. The relevance of these anomalies has to remain unclear until further investigation has been conducted. If there is no data available about whether or how a certain failure mode influences the observed signals of a system and no expert knowledge can provide a sufficiently accurate estimate of the effect, there is no way to create a trusted automated system for the identification of that specific failure. If there is no data available about the trajectory of the observed signals related to a specific failure mode, and no expert knowledge that could substitute it, there is no way to systemati-

cally build an accurate predictor of the fault. Different algorithms might utilize available data to different degrees of efficiency, none of them can work without any data.

An example of a system working with very little data is case-based reasoning. Case-based reasoning aims at providing actionable suggestions for situations or problems that are similar by some measure to an already experienced and solved problem. What worked for a historically solved case is suggested as solution for the "similar" case at hand. A root-cause analysis is typically conducted for each failure mode. On the other side of the spectrum are methods of deep learning, which have shown state-of-the-art performance in a wide range of tasks. Methods of deep learning, when deployed on systems with a lot of computational power, can handle - but also require - amounts of data that are widely considered to be "big" in volume, variety and velocity. These methods map the specifics of every training instance during training to some set of parameters with respect to some constructed measure of success, e.g. mean error rate. In most cases, there is no easy way of telling why a specific test gets mapped to a specific output.

#### **2.5.4 Prioritizing diagnostics needs from a maintenance perspective**

[3] specifies what components a monitoring system should be devised for. It could be thought of a guide for the respective owner of the engineering solution. The idea is straight forward: devise a measure for the severity of a fault, find those components for which faults have the highest negative impact and construct a condition monitoring solution to the extent that the available or obtainable information allows to be realized economically. Another important practical question is how to prioritize the development and implementation efforts for a condition monitoring solution. A seemingly natural approach is described in [3], the Failure Mode Symptoms Analysis, which advises to:

1. analyze the availability, maintainability and criticality of the system with respect to the whole production process;
2. list the major components and their functions;
3. analyze the failure modes and their causes as component faults;
4. determine the criticality, taking into account the gravity (safety, availability, maintenance costs, production quality) and the occurrence frequency;
5. decide accordingly which faults should be covered by diagnostics;
6. analyze under which operation conditions the different faults can be best observed and define reference conditions;
7. express the symptoms that can serve in assessing the condition of the machine and that will be used for diagnostics;
8. list the descriptors that will be used to evaluate the different symptoms,
9. identify the necessary measurements and transducers from which the descriptors will be derived or computed.

The process is essentially a modification of the FMEA process with a focus on the symptoms produced by each identified failure mode and the subsequent selection of the most appropriate detection and monitoring strategies. Correspondingly, this tool should be used in conjunction with an existing FMEA analysis that has already identified and ranked possible failure modes. During the Failure Mode Symptoms Analysis, the symptoms of each failure mode have been ranked by detection rate, severity of the fault, diagnosis and prognosis confidence. The Monitoring Priority Number (MPN) is the result of the multiplication of the four preceding rankings and results in an overall rating of each failure mode. A high MPN value indicates that the nominated technique is suitable for the detection, diagnosis and prognosis of the associated failure mode. A low MPN value does not imply that monitoring is not necessary but rather that a low confidence for the accuracy of detection, analysis and prognosis can be expected with the nominated monitoring technique and frequency. The least favorable case is a failure mode with high severity, low detectability,

and low diagnostic and prognostic confidence. The most favorable case is of course a failure mode with low severity, easily detectable, with known failure modes and associated patterns and therefore a high diagnosis and prognosis confidence level.

The more complex a system under consideration, the more difficult it becomes to apply the Failure Mode Symptoms Analysis thoroughly. It is highly unlikely to be able to predict all failure modes and to have access to all signals that would be needed to detect even the already identified ones. Where this information is available, however, this method seems like a reasonable approach to guide the development process of a diagnostics system.



---

## 3 State of Research

The topic of preventive maintenance, ranging from condition-based maintenance to predictive maintenance, has attracted a lot of research over the last decades. The applications reach from the detection of faults in ball bearings, gears, valves and ball screw drives to induction motors, batteries, and engines for Mars rovers. There exist a few overview articles about methods and developments in the realm of condition monitoring [9]–[13], some of those overview articles focus more on certain components of drives, such as the electric motors [14]–[18]. [19] gives an overview of even broader frameworks for factories in the context of Industry 4.0, in which data-based applications like condition monitoring are an integral part of the overall production system. This chapter will look at work done in the realm of condition monitoring and its overlap with methods of the realm of machine learning, that have been used to provide the functions of a condition monitoring system described in Chapter 1.

### 3.1 Model-based Diagnostics

One of the approaches mentioned in Section 2.4 is a model-based approach, where there are at least two different concepts, the modeling of the damage, which is often non-linear in nature, and the modeling of the system, which is often approximated with using linear state equations. While there are of course white-box models of systems, that are built up from basic physical equations to create a model of the system, in this section only work on data-driven methods is reviewed.

### **3.1.1 Damage Processes**

There has been some research that tries to offer general methods to model faults of a system. [20] uses a General Path Model, which assumes an underlying functional for the degradation path of a specific fault mode to estimate a time of failure distribution. A similar approach is used in [21], where the parameters of the system are tuned with a genetic algorithm. An alternative for the modeling of the damage process is described in [22], where the rate of degradation of a system's components are related to the irreversible entropy produced by the underlying dissipative physical processes. An approach that simplifies degradation phenomena into three characteristic categories, namely linear, concave and convex, is presented in [23]. Each such category can be modeled by one of the passive elements of a bond graph, namely by a resistor, a capacitor or an inductor, respectively. A degradation phenomenon of a system is assumed to be nothing but a continuous drift in one of the parameters of a system. The system fails, when the evolution of any set of parameters of interest violated at least one predefined threshold. [24] used a similar approach to decompose a system and assign generic damage evolution characteristics to the components of a system.

### **3.1.2 System Identification**

The proper identification of system parameters is a problem that has attracted a lot of research interest over the years. In machine tools, almost all of the time the drives will be operated in a closed-loop mode, due to either unstable behavior of the plant, or safety and efficiency concerns. This necessitates the use of techniques that are designed to create an estimate of the system from data that was acquired under closed-loop conditions. The main problem with data from closed-loop experiments is the lack of statistical independence between the disturbances entering the process and the input of the system, which puts a fundamental limitation to the use of standard open-loop identification methods [25], [26]. There exist several closed-loop parametric model identification

methods that make different assumptions about model structure or knowledge about the controller model. The model-based condition monitoring approach assumes the availability of a model of sufficiently high accuracy to compare either the predictions of this model to the actually measured values, or, alternatively, to detected changes in the coefficients of the models over time, which are interpreted to be at least correlated to degradation phenomena. The topic of closed-loop identification has attracted an increase in interest due to efforts creating and integrated identification and control framework. The key idea is to jointly identify the plant and design a control strategy with the objective of optimizing a control performance criterion. For such tasks, it would be advantageous, if the system identification could be done with routine operating data, as this would not decrease the performance of the system due to an increase in downtimes. [27], [28] showed what conditions need to apply to at least theoretically being able to identify a system using operational data only. For more formal proofs of these conditions, the reader is referred to [29]. Studying the Information Matrix in the prediction error identification setup, [30] also derived constraints on the excitation signal for the identifiability of a closed-loop system. The persistent excitation of a signal has to be larger than some thresholds that are functions of system lag and the order of the controller. The persistent excitation of a discrete signal  $u_k$  is equal to the largest  $n$ , for which  $R(n)$  is invertible, where

$$R(n) = \begin{bmatrix} \gamma(0) & \gamma(1) & \gamma(2) & \cdots & \gamma(n-1) \\ \gamma(1) & \gamma(0) & \gamma(1) & \cdots & \gamma(n-2) \\ \gamma(2) & \gamma(1) & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \gamma(1) \\ \gamma(n-1) & \gamma(n-2) & \cdots & \gamma(1) & \gamma(0) \end{bmatrix}$$

and  $\gamma(k)$  is the autocovariance function at lag  $k$ , defined as

$$\gamma(k) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=k}^N u_i - u_{i-k}.$$



The identification of a model based on input-output data is a common problem in control theory [31] and methods like Nonlinear autoregressive moving average with exogenous input (NARMAX) [32] have been developed for this task and adapted to better be able to deal with uncertainties in the acquired data [33]. While classical prediction error models have been used extensively for the task of system identification, other works try to approximate the system with neural nets of some form [34] or use subspace identification methods to directly estimate a linear-time invariant state space model [26], [35], [36].

### **3.1.3 Bayesian Filters**

If there exists a dynamical model of either the system or the fault, it is often necessary to update the current state of the (modeled) system given some observation. Following [37], the state is defined as

The state of a stochastic dynamic system is defined as the minimal amount of information about the effects of past inputs applied to the system that is sufficient to completely describe the future behavior of the system.

In practical application, the measurement of initial conditions and subsequent measurements are never precise, which necessitates the handling of uncertainties, a topic that is of such relevance for condition monitoring systems that it will be dealt with in Chapter 4. At this point it suffices to mention that updating the belief about the state of a system according to Bayes theorem is probably the most excessively used method.

While [38] shows conditions for the existence of exact finite dimensional nonlinear filters, which were applied in [39], there are basically two methods to approximate the ideal Bayesian filter, which are used for practical applications: some version of the Kalman filter or some form of Monte Carlo Sampling, mostly Particle Filtering.

Kalman Filters were originally designed for linear models that are subject to Gaussian noise. An estimate of the posterior distribution of the state is given

directly by this approach. Although it was developed of linear systems, the Kalman filter has been applied to linearized version of non-linear system nevertheless, which enabled its use in a variety of applications in the realm of condition monitoring. The Kalman filter can be seen as building block for any system that tries to describe either the system or the fault as a dynamic system [40]–[43]. An extension of the Kalman Filter, the unscented Kalman Filter was introduced in [44] and improved in [45]. This version of the Kalman filter does not approximate the non-linear state space model but instead uses a few carefully chosen sample points and transforms them using original dynamical equations of the system. This enables the accurate capturing of the posterior mean and covariance of any nonlinearity to the 3rd order. It has been used in a condition monitoring contexts [46]–[49]. [50] introduces another version of the Kalman filter, which exploits the so-called spherical-radial cubature rule to find a set of cubature points, which are sufficient to give approximations for certain integrals, which reduces the computational complexity of this method significantly. In [51] a constraint Kalman Filter is used, that truncates the probability density function in order to guarantee monotonicity in the models of degradation processes.

Particle Filtering tries to approximate the posterior distribution of the state of the system indirectly by the use of Monte Carlo simulation to make the Bayesian framework computationally tractable for nonlinear systems. Random samples of the present state are chosen with associated weights and transformed by the system equations to get an estimate of the posterior distribution. As the estimate of the posterior distribution becomes more accurate, when more samples are used in the simulation, computational cost have to be weighed against accuracy for most interesting problems. Particle filtering has been used extensively for condition monitoring applications [52]–[55]. Various combinations of the particle filter and other methods, like blind deconvolution for a planet gear boxes, or their application to dynamic Bayesian networks have been presented for the monitoring of systems [56]–[60]. A comparison of the performance of an unscented Kalman filter and a particle filter can be found in [61].

## 3.2 Signal-based Diagnostics

The other of the two approaches presented in Section 2.4 is called signal-based diagnostics. The idea is to create certain features, using transformations, filters, resampling or more advanced methods. Of course combinations of these methods can also yield viable options. After the features have been created, some of them are selected to be indicators for the statistical model that has to be created to enable the condition monitoring system to fulfill at least one of the functions of a condition monitoring system. It is not necessary to apply each of these stages, nor is it necessary to apply each of these stages only once. Some deep-learning algorithms for example work on raw data directly and apply sequential transformations by design.

## 3.3 Feature Creation

Feature creation is the name for the application of certain transformation, filters, etc. on a raw signal to create features with a high information content. This reduces the number of possible regressors for a statistical model. This task can be seen as key component of every diagnostic system, as the usage of ambiguous, noisy or uninformative features would undermine the overall performance of a diagnostics system. A lot of applications use signal processing techniques to extract relevant information from the raw signals. The Fourier transform is so ubiquitously used that an account for its usage in the literature is omitted. Other popular methods to transform a signal from the time domain to either the frequency or even the time-frequency domain are the wavelet transformation [62], [63] and the Hilbert-Huang-Transformation [64], [65].

The wavelet and wavelet package transformation have been used quite extensively for condition monitoring task. [66]–[74] are just a small selection of works that used this technique successfully for the creation of features from raw signals. These transformations are often combined with other techniques like in [75]–[78] to create features that have some beneficial properties for a diagnostics system, like the reduction of noise for example. The other major

transformation, that has been used repeatedly and productively for condition monitoring tasks is the Hilbert-Huang transformation [79]–[82].

Similar to the wavelet transformation, the Hilbert-Huang transformation is used frequently in combination with other methods like autoregressive models, which benefit from certain characteristics of the intrinsic mode functions, into which the Hilbert-Huang transformation decomposes the raw signal [83], [84]. Other examples for signal processing that try to decompose the measured signal into more easily analyzable components are the intrinsic timescale decomposition [85], the Taeger-Huang-Transformation [86] or the discrete Gabor expansion [87], some of which have also been used for the analysis of the condition of components.

Some researchers try to directly apply a filter to the measured signal in order to directly monitor the system in real-time. [88] for example utilizes sweeping-filters to implement a fast method for the health assessment of tools in machines, where programmable second-order filters are employed to amplify the fault signal of a cutting tool.

Another approach for designing filters is based on the spectral kurtosis of signals of rotating machines in possibly non-stationary operating conditions [89], [90]. The spectral kurtosis is a fourth-order spectral cumulant and signifies the peakedness of a probability function, which can be interpreted to signify the temporal dispersion of the time-frequency energy distribution or alternatively the peakedness of the squared envelope of the signal. The spectral kurtosis can be used to detect transient signals. Given the spectral kurtosis of a sum signal, a matched filter for the maximization of the kurtosis of the filtered signal can be devised. The derived filters have been applied to the problem of bearing monitoring [91].

There are also works that focus on the utilization of other higher order spectra of signals, like the bispectrum. The idea is that all cumulant spectra of order greater than 2 are identically zero for Gaussian random variables, which are assumed to be a good model of noise. If noise tends to be distributed normally and additive, it should therefore not influence the higher order spectra.

[92]–[95] are only a small selection of the application of these techniques to condition monitoring tasks.

Another signal processing technique that is used in the field of mechanical signature analysis is cyclostationarity [96]. A cyclostationary signal is one that exhibits some periodicity of its energy flow that is hidden. It is applied to rotating machines and specifically geared towards applications that produce signals that are not stationary according to the mathematical definition of that term. This concept has also been applied to the monitoring of components, like bearings and gears. [82] introduces a method based on characteristic distances for the monitoring of ball screw drives. Geometrical properties of the ball-screw drive are used to find characteristic distances associated with components. Measurements of a ball screw drive with induced degradation are taken with samples that are equidistant in space, not in time. Some advantages of this approach compared to vibrations-energy based features [97] have been demonstrated [98]. This can be seen as an application of the concept of the instantaneous frequency, which has been used to amplify fault signals of rotating machinery [99]–[103]. Typically, it is attempted to resample a signal that has been measured equidistantly in time to a signal that behaves similar to the type of signal, which [82] measured directly. All of these approaches attempt to make signals from machinery in non-stationary conditions usable. [104] presents a method of rescaling vibration-based features to also account for non-stationary operating conditions.

Fuzzy sets and systems were introduced in [105] and [106], respectively, which are a method for information granulation [107], which is likened to the way humans reason [108]. Fuzzy membership functions are often used to map an observation to a measure of similarity. These measures are then used as an input for diagnostic systems instead of the original measurements. Examples of the application of methods from the fuzzy set theory are given in the subsequent sections. Rough Set theory has been introduced in [109] and is, like fuzzy set theory, a method to granularize information by providing approximations of instances that are suited for similarity comparisons and providing a way to reason about uncertainties rigorously [110], [111]. As is the case with fuzzy

methods, rough set theory has been used as a preprocessing tool in the realm of condition monitoring, which yet again maps observations to a measure for similarity [112]–[114]. Despite sharing many characteristics, fuzzy and rough sets can be shown to not be equivalent [115]. Combinations of fuzzy and rough descriptions of observation have also been used successfully [116].

## 3.4 Feature Selection

Features extracted from raw signals are typically prone to containing large amounts of redundant information, which might impede the practical application of automated machine condition monitoring. For a lot of applications the demands for processing power and storage capacity grow far faster than linearly with the problem dimension, which can be taken as number of features that are used. For this reason, various works focus on selecting from among the available features a subset of relevant ones.

Artificial Neural Networks have been used to find optimized feature sets from many feature parameter types of vibration signals [117]. Another way of determining the relevance of certain features is to use a method called automatic relevance detection. It uses a probabilistic setup to find components of a feature vector that are irrelevant to a generative model  $\mathbf{y} = \Phi\mathbf{x} + \epsilon$ , where  $\Phi \in \mathcal{R}^{n \times m}$  is a dictionary of features and  $\mathbf{x} \in \mathcal{R}^m$  is a vector of unknown weights,  $\mathbf{y}$  is an observation vector and  $\epsilon$  is uncorrelated noise  $\epsilon \sim \mathcal{N}(0, \lambda I)$  [118]. Instead of identifying the "most important" features, unimportant features are removed. The technique has been used for a condition monitoring system for rotating machinery [119].

Originally developed for linear regression models, but later extended to more general models, the Lasso algorithm is another form for detecting relevant features. By introducing a constraint on the absolute value of the sum of coefficients, some coefficients are produced that are exactly zero [120] The resulting models can be viewed as belonging to the class of interpretable models. Chapter 6 will focus on interpretability more closely.

[121] introduces a method for the detection of relevant features that is based

on their predictive capabilities. An approach to select features based on their monotonicity and trendability is presented in [78].

STRASS (STrong Relevant Algorithm of Subset Selection) is an algorithm, which produces 3 categories of features: "strong relevant", "weak relevant" and "redundant". This algorithm has been used in [122] for the improvement of the performance of different classifiers on a variety of data sets.

A method that aims specifically at selecting the most important component state for maintenance decision is described in [123].

If certain assumptions about the mathematical geometrical properties of the data points themselves are made, data-points can be described as noisy samples of an underlying manifold, which in fact enables the reduction of the problem dimension and has also been applied to fault diagnosis tasks [124]. Generative Topographic Mapping is another approach utilizing similar assumptions and has been used for the health quantification and selection of relevant features [125]. [126] uses Dominant Feature Identification based on principal component analysis and clustering to determine dominant features.

## 3.5 Regularization

Regularization is a term describing the addition of information to an ill-posed problem in order to solve it, increase its robustness to noisy samples or to avoid overfitting issues [127]–[129]. In a lot of the algorithms that are to be presented in the following sections, regularizers are intrinsically employed to limit for example the complexity of the solution space or to smooth the input data in some form. It can be shown that training with noisy examples can be seen as a form of regularized learning [130]. Constraining the data to be regarded as samples from an underlying low-dimensional manifold has some interesting properties, like a linear time complexity to find the mapping onto the generating manifold [131], [132]. Mapping data to this manifold can reduce the dimensionality of the problem. It seems like there are some fundamental connections between manifold regularization and semi-supervised learning [133], [134]. Regularizers of this type have been applied to large data sets [135]. [136] intro-

duced a new form of regularization by randomly omitting feature detectors on each training case. An improved version of this sort of regularization has been introduced in [137] and is dubbed "DropConnect".

## 3.6 Novelty Detection

The simplest function to be accomplished in constructing a condition monitoring system is the capability of detecting novel system behavior. As the system should be in a healthy state upon delivery, this novel behavior can be seen to indicate a fault. It might even be impossible to proceed any further on the "hierarchy" of condition monitoring systems, because there is simply not enough data from "abnormal" cases in training sets to construct explicit models of exceptional cases. The main idea is to construct a model of normalcy from more readily available data under regular operation conditions. [138] gives an overview over different novelty detection schemes. Novelty detection can be seen as the task of recognizing that test data differs in some respect from data that was available during learning and which is assumed to represent normal behavior. The practical importance of this task has led to many proposed approaches. According to [138], there are at least five general categories:

- i) probabilistic,
- ii) distance-based,
- iii) reconstruction-based,
- iv) domain-based and
- v) information-theoretic

techniques.

Approach (i) often uses probabilistic methods that try to create a probability density estimation for the "normal" class. The underlying assumption is that low density areas in the training data set indicate that there is only a low probability of containing representations of "normal" data. Interesting methods in this realm include attempts to estimate the underlying generative probability



distribution function (pdf) of the data. The pdf estimate can then be used to define boundaries of normality by introducing thresholds. A new data sample can be tested against these boundaries to determine its membership status. [139] present a self-adaptive system for the automatic creation of such thresholds for a condition monitoring system. There exist parametric and non-parametric estimators for this task.

One example of the parametric approach is called extreme value theory [140], a branch of statistics that deals with extreme deviations of a probability distribution, i.e. extremely large or extremely tiny values in the tails of distributions assumed to generate the data. This method has been applied for condition monitoring purposes example in [141]. State-Space models are another example of parametric probabilistic methods used for novelty detection in time-series data. Their underlying assumption is that there exists a not necessarily directly observable state that evolves through time, possibly as a function of the inputs. Hidden Markov Models (HMM) fall into this category. Each state is associated with a probability distribution, the "emission probability", and each pair of states  $(i, j)$  has associated "transition probabilities" which represents the probability of being in state  $q$  given that the system was in state  $p$  on the previous time sample. While the features are observable, the system states are not and are called *unobservable*, *hidden* or *latent* states. The parameters of HMMs are trained using available data [142] and have been used extensively for diagnostics applications [143]–[148].

Non-parametric methods often make use of kernel-density estimators [149], where the pdf is typically modeled by Gaussian kernels centered on data points, for which the variance has to be trained. The sum of the contributions of nearby kernels gives an estimate of the probability for yet unobserved points in the data space. [150] gives an overview of certain types of density estimators, which have been used repeatedly for the purpose of novelty detection [151]–[155].

Approach (ii) includes concepts like nearest-neighbor and other clustering analyses that have also been used for the more general classification case. The idea here is that data representing a normal mode of operation should be near

to each other according to some measure. The Euclidean or Mahalanobis<sup>1</sup> distance are amongst the common choices for that underlying measure. Abnormal behavior should be represented by a large distance to other data points or clusters. The usage of a case-based reasoning process scheme is presented in [156]. The measure of similarity does not seem to be as well-defined as for other distance-based novelty detection schemes, though.

The third approach uses regression models of some form or another. The functional relationship of these models is established by using training data from a normal mode of operation. If data from an abnormal operational regime is mapped using these models, the reconstruction error is expected to be large, thus giving rise to a huge novelty score. Various versions of neural networks have been proposed and used for this task. [157] presents a novelty detection approach for multivariate datasets based multilayer perceptron, with the same number of input and output neurons, and three hidden layers. This layout is called replicator neural network (RNN). The RNN is supposed to reproduce the input points at the output layer with the minimum reconstruction error, after undergoing a compression through the hidden layers containing fewer neurons. A small set of input points with large reconstruction errors are considered as outliers. [158] used autoencoders to compute the bitwise difference between input and output to highlight novel components of the input.

Approach (iv) uses domain-based methods to characterize the training data. The goal of these methods in general is to create a boundary around the normal data points. This boundary is supposed to follow the outline of the underlying distribution of the data, but they do not explicitly create an estimate of that density. A good example of this method is Support Vector Data Description, which tries to model an "in-group" and an "out-group" by learning a model of normalcy [159]–[161]. This method has been used to detect faults in machines [162]. Deep Learning on a support vector data description has also been proposed [163].

<sup>1</sup> The Mahalanobis distance of an observation  $\mathbf{x}$  from a set of observations with mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$  is defined as  $d_M = \sqrt{(\mathbf{x} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})}$ . In case of standard normally distributed variables, the Mahalanobis reduces to the Euclidean distance.

Approach (v) uses information-theoretic measures to detect huge changes in the information content of a data set induced by the addition of "abnormal" data. In [164] a parameter-free method for anomaly detection based on compression theory is proposed. A sequence of continuous observations is divided via a sliding window into subsequences. The subsequences are then compared to the whole sequence by approximating their respective Kolmogorov complexity, which measures information content of a signal by trying to find the shortest computer program that produces the sequence as an output. In the general, this metric cannot be computed exactly and the size of the compressed file that contains the string representation of the sequences is used as a proxy of this measure. [165] proposes a local-search heuristic to identify outliers in categorical data based on entropy, which measures the information content of random variables. In this measure, outliers are those observations that reduce the entropy of the dataset significantly, when removed. The Kullback-Leibler divergence has also been used successfully as metric for the detection of outliers. [166], [167]

## 3.7 Case-Based Reasoning

Case-based reasoning (CBR) is a paradigm for solving problems that is in many respects fundamentally different from other statistical approaches. CBR is able to utilize the specific knowledge of previously experienced, concrete problem situations (cases). Instead of relying exclusively on general knowledge of a problem domain or making associations along generalized relationships between problem descriptors and conclusions, a new problem is solved by finding a similar past case and reusing it in for the new situation. A second important difference is that CBR also is an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems, which solves some problems of incremental learning common in other learning paradigms [168]. Case-based reasoning is an example of a system that can work with comparatively little data. It is aimed at recommending actionable suggestions for situations or problems

that are similar (by some measure) to an already experienced and solved problem. To be able to provide these suggestions, it is necessary that a root-cause analysis is conducted for each problem case. Case-based reasoning has been used in the Overall Management Architecture for Health Analysis (OMAHA) framework, which has been deployed in the civilian aviation industry. [169]. A central maintenance system is used to correlate failure messages from modules to observable effects like displayed messages. A fault item is generated for every failure message and the CMS tries to correlate new failure messages and effects to yet unexplained fault items. If they can not be correlated, a new failure item is created and added incrementally to the knowledge database. For each failure item a root-cause analysis is conducted. Unfortunately, not all failure items are mono-causal and in an application of this method for airplanes, up to ten root-causes for each fault item and up to three modules per root-cause were identified as possible culprits [170].

An interesting conceptual difference between CBR and other data-driven methods is the point in time, when the generalization from already experienced problem instances are drawn. CBR delays generalization of its cases until testing time and therefore realizes a "lazy" generalization strategy, while most other data-driven methods generalize during training time.

It has been argued that CBR is based entirely on anecdotal evidence, as there are in general no statistically significant amounts of data available. This leaves conclusions of a CBR-system vulnerable to random effects. Proponents counter with the observation that inductive reasoning, the primary way of natural human learning, almost always operates on data too scarce to have statistical relevance and still seems to work.

As it seems to be the case for most techniques from the realm of artificial intelligence, there is no universal CBR method suitable for every domain of application [168]. There are also developments trying to create a statistical framework for CBR and formalizing case-based inference as a specific type of probabilistic inference [171]. This enables the generation of case-based predictions equipped with a quantifiable level of confidence.

## 3.8 Classification

Classification is the task of mapping an input vector to a restricted number of outputs. This general ability is of course useful in the realm of condition monitoring to map a set of selected features from a machine to an output, which could indicate a certain state, for example a slightly worn tool or a tool that has to be replaced immediately.

There is a multitude of different classifiers and a comprehensive list of all of them is well beyond the scope of this section, which is intended to only give a general overview of the used data-driven classification tools and some of their applications in the realm of condition monitoring.

The derivation of a learning algorithm typically assume a set of training data  $\{\mathbf{x}_i, y_i\}_{i=1}^l$ , where  $\mathbf{x}_i$  is set of observed patterns,  $y_i$  are their corresponding labels. The task of a classification algorithm is to construct a function from this data that can map a yet unseen input  $\mathbf{x}_{new}$  as accurately as possible to a corresponding label  $y_{new}$ .

Support Vector Machines (SVM) are a method that is based on the statistical learning theory [172] and have been introduced in [173]. They can be viewed as specifically regularized networks [129], [174], [175]. Unlike most classifiers, SVMs do not try to minimize the empirical risk, but the structural risk. In their original formulation, Support Vector Machines are designed as binary classifiers, which can only decide to which of two classes a given instance belongs. This was too severe a constraint and various strategies to extend the capabilities of SVMs to multi-class classification problems have been proposed. The formulation of SVMs utilized inner products of feature vectors. This allows for the usage of the so-called "kernel trick" [176]–[178]. By carefully choosing certain kernel functions, this trick allows to do calculations in a feature space, "as if" the input data had undergone a possibly rather computationally expensive transformation [179], which often improves their performance capacities [180], [181]. The computational complexity of the SVM growth far faster than linearly with the number problem dimension, which has prompted several attempts to decrease the computational complexity of this method, to increase its training

speed [182] and its applicability to very large data sets [183]. This method has attracted considerable attention and several introductions are available for engineering practitioners [184]–[187] and a lot of research has applied this method in the context of condition monitoring [188]–[208].

Several variants of this technique exist, which try to improve the properties of these algorithms. SVMs construct a function, which maps feature vectors to outputs by using the combination of certain instance of the feature vectors that were available during training, the so-called support vectors. The formulation of the SVM imposes a certain sparsity constraint and leads to a convex quadratic optimization problem to be solved. There are formulations called Least Squares Support Vector Machines [209] and Weighted Least Squares Support Vector Machines [210], Bayesian Least Squares Support Vector Machines [211] or Recurrent Least Squares Support Vector Machines [212], which use ideas of the derivation of the SVM, but dispose of the sparsity constraint to create an algorithm, which can be formulated as solving a linear program. Laplacian Support Vector Machines exploit certain properties of manifolds, to which the training data is assumed to belong to, and can be used for semi-supervised learning [213], where not all learning instances are labeled. Some research utilizes certain aspects of the fuzzy set theory to construct so-called Fuzzy Support Vector Machines [214]–[217] that have given superior classification results in certain applications. The  $\nu$ -Support Vector Machines [218] is another interesting approach, which allows controlling certain properties via the parameter  $\nu$ , which is an upper bound on the fraction of training errors and a lower bound on the fraction of support vectors. Support Vector Machines have certain tunable parameters that control for example how sparse the resulting description of the function will be. Methods for finding a good parametrization are presented for example in [219], [220]. There is also some research based on particle swarm optimization for the tuning of parameters of SVMs for condition monitoring applications [221].

For some applications it is not sufficient to only have hard class labels as a result, but the confidence of the classification is also an output of interest. To satisfy that need a probabilistic extension to the Support Vector Machines, dubbed

the Relevance Vector Machines has been formulated [222]–[224]. Other frameworks to make the output of SVMs probabilistic also exist [225]–[229] and have been used in condition monitoring contexts successfully [230]. Although more frequently used for regression task, Gaussian Processes [231] can be used for classification tasks. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

The same is true for Extreme Learning machines, which are usable for multi-class classification tasks [232], [233], but are more frequently used in regression applications as well.

Deep Learning is the name of a collection of techniques that have shown state-of-the-art performance for various classification tasks. Given enough input data, multiple consecutive layers of artificial neural networks are trained efficiently to construct a classifier [234], [235]<sup>2</sup>. Some of these techniques have been adopted in the realm of condition monitoring [237], [238].

There are some theoretical results showing that an ensemble of different classifiers can be able to generate better classification results. Research utilizing the idea of ensembles for condition monitoring application has also been conducted [239]–[242], for example based on decision trees, which are also a classical example for classifiers [243]. Of course, artificial neural networks can also be utilized for classification tasks and have been used for condition monitoring applications [92], [189], [190].

## 3.9 Regression

Regression describes techniques to create a function approximation that maps some input values to some useful output, which is, unlike in the case of classification, not categorical. These techniques can be used for example to create an approximation model of the dynamic behavior of some machine or to predict the evolution of a fault. A variable of special interest in the realm of condition monitoring and especially for predictive maintenance is the remaining useful

---

<sup>2</sup> It seems like there are some interesting connections between the mathematical structure of physics and deep learning techniques. [236]

life (RUL) of a system. This quantity represents the expected time until a failure occurs. Various Methods have been adopted in order to create an estimate for the RUL.

There is a multitude of methods for the regression in the context of machine learning and while a complete description of all the proposed techniques is well beyond the scope of this thesis, this chapter is intended to give an overview of regression techniques in the context of condition monitoring. Hidden Markov Models, which we have visited upon in the previous chapter, are frequently used for regression tasks. They have been used in the context of system health diagnosis and prognosis [244]–[246]. An extension of HMMs, called Hidden Semi Markov Models (HSMM), does not explicitly make the Markov chain assumption [247], [248] and allows modeling the time duration of the hidden states, which is supposed to increase the prognostic performance of the models. These models have been applied in failure detection and prognosis tasks [249], [250], sometimes even in enhanced versions [251].

Along the same ideas that led to the development of the Support Vector Machine, an algorithm dubbed Support Vector Regression has been developed, which can be used for regression tasks [252]. There have been several extensions to the original method, for example the usage of fuzzy methods to improve the performance of the algorithm [253], [254], the introduction of incremental update capacities [255] or, similar to the  $\nu$ -Support Vector Machine, the  $\nu$ -Support Vector Regression, algorithm, where  $\nu$  is again an upper bound on the fraction of errors and a lower bound for the fraction of support vectors [256]. [257] gives a rather thorough introduction to the topic. [258], [259] applied this method for condition monitoring tasks.

Traditional artificial neural networks<sup>3</sup> have been used for the prognosis of tool-wear conditions [260], [261]. [262] demonstrated that probabilities can be estimated with certain artificial neural networks as well, which made it possible to forecast failure probabilities. To capture the time-dependence of processes more naturally, Recurrent Neural Networks have been developed [263] and

---

<sup>3</sup> See for example [37] for an introduction of the history of the development of artificial neural networks



are still an active research topic [264]. These networks have feedback paths for signals that they themselves generated. There have been successful usages in the condition monitoring context [265]–[268].

As briefly mentioned in the previous section, Gaussian processes are often used in regression applications [269]. They offer a lot of options to design application specific correlation functions, which can help to improve the prediction accuracy. A drawback of Gaussian Processes is that the computational burden grows far faster than linearly with the problem dimension. To mitigate this problem, fast implementations of Gaussian Processes have been developed [270]–[272], which are being applied to general big data problems [273], [274] as well as to condition monitoring applications [275], [276].

Dynamic Bayesian Nets are a special form of Probabilistic Graphical Models [277]. They have some interesting properties, like the ability to transparently model interactions in systems, account for inherent uncertainties due to natural variability, the capability of integrating many types of information and updatability of all distributions in a model when a new observation for at least one variable becomes available [59]. There have been successful usages of Dynamic Bayesian Networks for condition monitoring applications [59], [278]–[280].

Artificial Neural Fuzzy Inference systems (ANFIS) are reasoning systems that encode fuzzy IF-THEN rules which can be tuned or rather trained like artificial neural networks. ANFIS have been introduced in [281] and extended in [282]. Their universal approximation property [283], [284] has been shown [285], [286]. A thorough treatment of these methods can be found in [287]. There are some applications of this type of reasoning system to condition monitoring tasks [288]–[293]. To tackle some problems associated with the computational burden of too many inputs, [294] introduces a method to determine the structure of a fuzzy inference system. [295] presents a method for using fuzzy inference systems on large data sets. A comparison of the performance of fuzzy logic systems to artificial neural networks can be found in [296].

A particularly interesting regression algorithm is called Extreme Learning Machine (ELM). It has been introduced in [297] as a novel approach towards machine learning with universal approximation capability [298]. Extreme Learn-

ing Machines are dubbed "extreme" because of the speed of their training. They use randomly assigned biases and input-connection weights for their neurons and only the weights of the connection layers are found via tools of linear algebra, i.e. by utilizing the Moore-Penrose-Pseudo-Inverse of matrices [299]. It shows that hidden neurons do not need to be tuned iteratively and that all their parameters can be chosen independently of training samples from any continuous probability distribution and still achieve competitive classification and regression results. The Extreme Learning Theory also had some impact upon already well established tools like support vector machines [300], [301]. A lot of research interest has been spawned by the introduction of this theory and a lot of improvements to the original algorithm have been proposed, ranging from improvements in the numerical solution of the problem setup [302], [303] to Extreme Learning Machines for Big Data applications [304]. A lot of approaches focus on increasing the robustness of ELMs [305]–[307], introduce probabilistic setups to build sparse regressors [308], [309], utilize various kernels to increase the performance [310] or focus on specific types of regularizers to optimize the algorithm for certain specialized applications [311]–[314]. Other approaches focus on increasing the capabilities of this algorithm by extending their formulation to semi- or even unsupervised problem setups [315]–[317]. An interesting approach in this context actively asks for labels on instances that have produced a high degree of uncertainty about the corresponding output [318]. Some formulations of the algorithm have been proposed, which have enough stages to be considered "deep" [319]–[321]. There are specific formulations for highly correlated types of input data like images [322] and the combination of multiple regression algorithms has been used to create a more robust output [323], [324]. [325] uses Extreme Learning Machines as an optimization stage for the parameters of a generalized radial basis function neural network. As with classification tasks, there are also methods utilizing the fuzzy set theory to improve the performance of the algorithm [326]–[328]. There exist a couple of introductory articles to this sprawling topic [308], [329], [330]. Some of these techniques have been successfully employed in the context of condition monitoring [331]–[334].

As was the case with classification methods, there are usages of ensembles of regressors of either the same type [335], [336] or diverse types of regressors [337] to create a more robust predictor that does not suffer from the bias of any one regression function.

The relative performance of some methods mentioned in this section as well as some omitted algorithms in predicting the RUL can also be found [338]–[341].

## 3.10 Method Selection

Given that the field of artificial intelligence is vast and the particular constraints of a manufacturing setting, the choice of a suitable algorithm for the application at hand is not trivial. To ease the burden on practitioners, [342] presents empirical rules for the selection of the appropriate methods to use for a specific collection of data samples. An alternative guide for the selection of suitable methods is given in [343]. In a similar vein, [344] presents an overview specifically for condition monitoring applications of rotating machinery. The choice of an architecture of the condition monitoring system, constraints the applicability of the solution. For example, sites with a sufficiently good internet connectivity might be suitable candidates for cloud-based analysis solutions, like presented in [345], while for application in areas with limited connectivity embedded solutions with significantly lower computational capacities have to be employed. [346]

## 3.11 Summary

The toolbox for the creation and automation of condition monitoring tasks is well suited for a multitude of tasks along the whole data processing chain. Especially the addition of techniques of the field of machine learning for classification and regression tasks has opened new possibilities for the monitoring and even prediction of equipment conditions. The latter capability promises the highest additional value. The adoption of these techniques also introduces their respective weaknesses, like a relative lack of interpretability and explain-

ability, which are problematic for the condition monitoring context. Here, the costs of misdetection and misattribution can prove prohibitive, which makes an accurate characterization of the uncertainty of classification and prediction results paramount. This challenge will be discussed in the following chapter.



---

## 4 Uncertainty

The detection and identification of faults and especially the prediction of the remaining useful life or a specific failure mode has proved to be a challenge for industrial assets due to the broad range of usage profiles, load situations, operating temperature, environmental influences, lubrication and their effects upon the different wear and tear processes.

Each component and subsystem of a complex system is only similar up to manufacturing tolerances and the individual history of the component. Knowledge about individual components can therefore only be specified up to a certain level of certainty of the respective characteristics of the components. From a system theoretical perspective, interconnecting multiple systems with uncertainties in their parameters, even assuming that the structures of these uncertainties are known, poses challenges with respect to diagnostic capabilities and even mathematical tractability.

To accurately identify a fault in a system, it is mandatory to be able to distinguish normal from abnormal behavior, but this task becomes increasingly more challenging, when the complexity of the system and the number of influences on the systems grow. The greater the uncertainty about the state of the system and the larger the space of tolerable behavior, the harder the task. It will be shown in this chapter that for the task of predicting a system's future state the management of uncertainty is an even harder challenge. By looking into the following questions:

- "From where do uncertainties arise?",
- "What are uncertainties?",
- "How to cope with uncertainties?",
- "How to quantify uncertainties accurately?",

this chapter will give a more thorough understanding of uncertainties in condition monitoring systems.

There are different perspectives to these questions and the underlying differences in fundamental concepts and their relation to the problem at hand shall be reviewed in some detail. Given that the prediction of a fault mode is of the greatest value for such a system, the requirements on uncertainties for a prognostic setup will guide the discussion, which follows [347].

### 4.1 Sources of Uncertainty

Prognostics is the discipline of predicting the future state of system or component, identifying possible failure modes and thereby predicting the remaining useful life of a component or the system in general. There are several influences on the prediction of future behavior and in turn the remaining useful life. To facilitate prognostics-based decision-making, it is important to assess how these sources of uncertainty affect prognostics. It is therefore necessary to compute the overall uncertainty in the remaining useful life prediction. In practical applications, it can be challenging to even identify and quantify the individual sources of uncertainty that are capable of affecting the prognostics. Or it can be possible to identify sources of uncertainty, but it is then not straightforward or even possible to account for the effect due to modeling issues. [347], [348]

#### 4.1.1 Simplified show-case model

Let the health state of an engineering component at any time  $t$  be given by  $x(t)$ . Consider a simplified degradation model, where the rate of degradation of the health is proportional to the current state of the system,  $\dot{x} \propto x(t)$ . The

proportionality factor is a negative number. Put in a discrete context, the model can be expressed as

$$x(k+1) = ax(k) + b(k), \quad (4.1)$$

with  $k$  representing the discretized time-index.  $a < 0$  is the aforementioned negative proportionality factor and can be thought of as representing a load, while  $b$  represents action upon the system. Effective maintenance could be represented by  $b > 0$ , shocks to the system could be represented by  $b < 0$ . The system is globally asymptotically stable, which means that given enough time, and a lack of maintenance, any arbitrary initial state  $x_0$  will approach the origin of the system, which is considered as becoming dysfunctional.

In order to compute a remaining useful life, it is necessary to choose a threshold function that defines the occurrence of failure. Crossing that threshold function for the first time indicates the end of the useful life of the system. Predicting the time of this occurrence is the essence of remaining useful life estimation.

### 4.1.2 Present uncertainty

To estimate the evolution of the state of a system, it is critically to identify the current state of the system. Typically, the damage or fault is expressed in terms of the state of the system as a dynamic system of state variables. State estimation then becomes equivalent to the estimation of the extent of the damage. This state estimation is typically done by using at least one of a variety of filtering techniques. Input-Output data of the system, usually collected through internal or external sensors, is used to estimate the state and the remaining uncertainty of this state estimation. In the conceptual example, this uncertainty can be represented by  $x_0$  being a random variable. Longer observation periods and more sophisticated filtering techniques can reduce the remaining uncertainty in the state estimation, given the assumption that the generative system is time-invariant. There exists a true state of the system<sup>1</sup>, but we cannot assess

---

<sup>1</sup> At least in macro-scale systems. There are discussions about whether this holds true for quantum and meso-scale systems. This true state might not be representable in the chosen model class, though.



this true state directly. Our uncertainty therefore clearly describes a lack of knowledge. [347]

### 4.1.3 Future uncertainty

The most important source of uncertainty in the context of prognostics arises from a rather mundane observation: the future is not known to us. There is no information about the future evolution of loading, operation, environmental and usage conditions that are known precisely, and there have to be certain assumptions about these conditions, if any conclusions are to be drawn. This kind of uncertainty can be seen in the conceptual example in a variance in the value of  $a$ . If there were no uncertainty regarding the future, there would not be any contribution to the overall level of uncertainty regarding the true remaining useful life of the engineering system or component.

### 4.1.4 Modeling uncertainty

If the effects of different loading, environmental, operational and usage conditions, however uncertain they are, are to be incorporated into an estimation of the RUL, it is necessary to use a functional degradation model. The occurrence of an end-of-life condition is modeled as a Boolean threshold model. These two models are used to predict the RUL and they may either be physics-based or data-driven. It may be impossible to develop models that accurately predict the underlying reality in practical applications.<sup>2</sup> Modeling uncertainty represents the difference between the true response of the system, that is neither knowable nor measurable, and the predicted response of the model. It consists of different parts: model form, model parameters and process noise. While it is conceivable to know these characteristics of the system at the time of the prediction, it is in general not possible to know their values for future instances. In the conceptual model, the parameter  $b$  is representative of one form of modeling uncertainty,

---

<sup>2</sup> Even a totally known and correct system of ordinary differential equations can exhibit exponential divergence of state predictions, if there is uncertainty in the initial state

while the linear form of the model is another assumption that might be incorrect for a real degradation process.

### **4.1.5 Prediction method uncertainty**

Even in the unlikely case of being able to quantify all the aforementioned uncertainties accurately, it is necessary to quantify their combined effect on the RUL estimation to predict their influence. It might not be possible to do this. For example, in the absence of an analytical expression for the arising uncertainties, sampling-based approaches are used for predictions. The usage of any finite number of samples causes uncertainties regarding the final probability distribution of the state estimation.

## **4.2 Uncertainty-related Activities**

The problem of handling uncertainty in the domain of condition monitoring, prognostics and health management is often discussed from the points of view of representation, quantification and management. Even though these are distinct processes, there are often used interchangeably. Following [348], four uncertainty-related activities are specified, which are needed to accurately describe the uncertainty in the estimate of the Remaining Useful Life and to inform the decision-maker appropriately.

### **4.2.1 Uncertainty representation and interpretation**

To appropriately deal with uncertainty, it is important to first decide in which way the arising uncertainties are represented. For example, the model in Sec. 4.1.1 contains three uncertain terms, the initial condition  $x_0$ , as well as the degradation rate  $a$  and the offset  $b$  are uncertain, which have to be accounted for. There is a wide range of different tools for representing uncertainty on different levels of granularity and detail. These include, but are not limited to probability theory, fuzzy set theory, rough-set theory, imprecise probabilities, interval-form

analysis etc. [349] for example presents work on the propagation of uncertainties for mixed probabilistic-possibilistic inputs. Probability theoretic methods seem to be used most frequently in the relevant literature, which can be regarded as evidence of their particular usefulness or the relative obscurity of the other ideas. Also, the availability of fast and tested code for probabilistic problems could contribute to this phenomenon. In the probabilistic framework, the uncertainty in the aforementioned variables of the model would be represented as probability distributions.

### 4.2.2 Uncertainty quantification

The next step consists of identifying and characterizing the sources of uncertainty that do or may have and influence on the state estimation or the prognosis of the RUL. If the probabilistic representation is used, the quantities  $x_0$ ,  $a$  and  $b$  are represented by random variables, whose statistics need to be quantified and whose distribution types and parameters need to be estimated. The more accurate these estimates are, the more accurate the models using these quantities can become. At this stage, uncertainties are assessed individually for components or modules of the overall system. The two most commonly used types of Bayesian filters, the Kalman filter and particle filter can essentially be regarded as tools of uncertainty quantification. While for example a particle filter can deal efficiently with many probability distributions, Gaussian error distributions for the random variables and a linear model structure are often assumed in the problem setup for the estimation of the current state of the system given health monitoring data.

### 4.2.3 Uncertainty propagation

Uncertainty propagation is probably the most relevant stage for prognosis. It accounts for the effect of all the previously identified and quantified uncertainties on the future states, the RUL and the associated uncertainties that can be a result of mixing different sources of uncertainty. The future states of the system

are computed by propagating the various uncertainties through a degradation-prediction model shaping the uncertainty in the estimates of the future states. A boolean threshold function used to indicate the EOL and the predicted future states of the system are then used to get a distribution over the RUL. It is important to understand that future states and RUL predictions are dependent upon the decisions that were previously made with regard to the uncertainty characterization. This might result in significantly decreased prediction performance of the prognostic system, if for example simplifying assumptions about the type of uncertainty distribution of a random variable were not warranted. Similarly, the ex-post assignment of certain types of probability distribution to the RUL, like treating it as a Gaussian probability distribution, will probably not contribute to ensuring a satisfying performance of the RUL prediction. Quite the contrary is likely in fact, as the propagation of possibly non-Gaussian probability distributions through a generally non-linear state space model can give rise to highly non-gaussian probability distributions of the RUL.

#### **4.2.4 Uncertainty management**

Uncertainty management is a term "used to refer to different activities that aid in managing uncertainty in condition-based maintenance during real-time operation" [347]. One aspect of uncertainty management attempts to minimize uncertainties in the resulting relevant estimates by identifying the contribution of different sources of uncertainty and implementing measures to decrease the uncertainty about these contributors. For example, decreasing the uncertainty related to the future loading conditions of the machine by changing its production schedule, will most likely also decrease the uncertainty in the RUL. The term "uncertainty management" is also often used to refer to decision-making processes utilizing uncertainty-related information.

## 4.3 Interpretation of Uncertainty as Probabilities

The interpretation of probabilities has been a viciously discussed topic for approximately 200 years. In this section both major interpretations of the discussion as well as a third, more novel approach are presented. The interpretation of the nature of the probability has direct impacts on the applicability of certain methods of condition monitoring systems.

### 4.3.1 Physical probabilities

Physical probabilities, also referred to as objective or frequentist probabilities, assume the existence of a "true" relative frequency of occurrence of a specified event in an infinite series of repeated (physical) trials, like a two on a die roll. These probabilities are thought of as existing as an ontological property of nature. Furthermore, there are at least two major interpretations of these ontological features of nature, namely von Mises's frequentist [350] and Popper's propensity interpretation [351]. The main idea is that uncertainty arises *only* due to the existence and presence of physical probabilities. If a true value of any particular quantity is deterministic, it is not thought as meaningful to ascribe a physical probability to that quantity. This rules out the application of the mathematical tools of probability theory to problems where the uncertainty is epistemic, i.e. arising not from a fact of nature, but from a lack of knowledge about a possibly deterministic facts. For example, the mean of any given probability distribution will have a deterministic mean, if any at all. It is therefore thought to be nonsensical to talk about a probability distribution over that mean, if different samples give rise to estimates about the underlying probability distribution differing in their mean estimate.

The lack of knowledge displayed in the different estimations about the true probability distribution is neither to be thought of nor treated as randomness in the frame of physical probabilities. In stochastic models, each parameter is treated as deterministic but only estimates of them can be found. Uncertainty, which still exists in this framework, about parameters is represented through

confidence intervals. Uncertainty in parameters is conceptually not allowed to be used for further uncertainty quantifications [352].

### 4.3.2 Subjective probabilities

In the subjective interpretation probabilities can be assigned to a broader range of statements than in the physical interpretation. There is no need for any of these statements to be in regard to the possible outcomes of random experiments. Probabilities are thought of as degrees of belief given the available knowledge about or evidence for a statement. As there is in principle no reason for any of a person's beliefs to conform to all or even any of the axioms of probability, a coherence requirement arises. This consistency requirement is called Dutch book consistency and is the normative statement, that beliefs *ought* to be such that an external agent (commonly referred to as "Nature") cannot make a guaranteed profit off of the beliefs of the agent, given the possibility of nature to force the agent to gamble in a specific way according to the probability assignments of said agent.<sup>3</sup> In this subjective interpretation, even deterministic quantities can be represented by using probability distributions, reflecting the degree of belief about that value. As a result, probability distributions can be assigned to parameters of probability models that have to be estimated and can then be used to propagate uncertainties in a logically consistent way in this interpretation. The subjective interpretation of probability is often associated with the concept of likelihood in Bayes's theorem, which can be interpreted as a rule to update beliefs in the presence of new evidence.

---

3 For example, if an agent believes that event  $E$  will happen with probability 0.7, but simultaneously holds the belief that  $E$  will not happen with probability 0.5, "Nature" can show this agent the incoherence of his beliefs by the following gamble. Assume there are tickets that pay out a fixed amount, say 1Unit, if event  $E$  happens, and nothing, if  $E$  does not happen. "Nature" can now choose to buy OR sell these tickets and the agent has to accept the offers of "Nature". A fair price for the agent is the probability of occurrence times the payout, so "Nature" could sell the agent a ticket for  $E$  to happen for 0.7Units AND sell a 0.5Units ticket for  $E$  not to happen. If  $E$  does in fact occur, "Nature" will lose 0.3Units on the first ticket, but gain 0.5Units on the second ticket. If  $E$  does not occur, "Nature" will gain 0.7Units on the first ticket and lose 0.5Units on the second. In both cases, "Nature" is guaranteed a profit of 0.2Units. The only way to prevent this scheme is to hold consistent beliefs, i.e. beliefs that conform to Kolmogorov's axioms [353, p.77].

An analytical solution of the ideal Bayesian update is almost always intractable<sup>4</sup> and a whole range of techniques have been developed to give an approximate solution. Most prominent amongst these techniques are particle filtering and the well-known Kalman filter. But these techniques are known as Bayesian filtering not only because of their usage of the Bayesian belief update, but because they provide uncertainty estimates that need to be interpreted subjectively. In macro-systems, it is assumed that there is a true state at any given time, so the uncertainty that arises for example in the Kalman filtering process cannot be interpreted as a property of the system *per se*, but as an assessment about the incomplete knowledge about the state of the system. [352]

### 4.3.3 Logical probabilities

Another view on probabilities is given by the idea that probabilities are akin to formal logic in that it is a set of rules for the consistent manipulation of truth values between propositions [353].

In this view, any probability assignment is conditional on the evidence for it. If there has not been a technical error in manipulation of the numerical values of the relationships of these values towards one another, all probabilities are considered to be *local truths*. Like in the subjective interpretation of probabilities, probabilities are not seen as an ontic fact of reality, but as epistemic. In this view, however, the calculus of probability is a set of rules about how to manipulate the relationship between accepted propositions or evidence. There is no guiding principle about which evidence *ought* to be accepted or is relevant to the problem at hand.

This means of course that even the most sophisticated probabilistic argument can be entirely without merit, if the accepted evidence is faulty or incomplete. In this interpretation, logical consistency takes the role of a normative statement and not Dutch books, because there does not exist a *realizable* Dutch book

---

<sup>4</sup> This is due to a lack of analytical solutions to the integration in the denominator.

for all valid logical probability assignments.<sup>5</sup> In this view the question of relevance, which is outside the realm of probability theory, is assigned imminent importance.

## 4.4 Interpretation of Uncertainty via Fuzzy Set Theory

Fuzzy sets, introduced by [105], have sparked another discussion about the "nature" of uncertainties. While statistical uncertainty is about the occurrence of a well-defined event<sup>6</sup>, which can be the result of any combination of the sources of uncertainty already outlined in Sec. 4.1. The classical tools of statistics are well suited to treat this kind of uncertainty, regardless of which of the interpretations discussed in Sec. 4.3 is adopted. The utility, validity and even existence of the other kind of uncertainty handled by fuzzy sets other hand is not as clear and has been debated. This kind of uncertainty arises, when the occurrence of the event itself is ambiguous.<sup>7</sup> Even for domain experts it is not always possible to give crisp bounds on events and only qualitative assessments can be given, like "the torque induced by friction should not be too high.  $x$  Nm is still ok." This sort of linguistic and conceptual vagueness can hardly be handled properly with probabilities, debatably it cannot be handled at all. Instead, instruments from the conceptually adjacent fuzzy set theory can be used to deal with this kind of ambiguity and uncertainty and will be introduced here briefly.

---

5 An example of this is given in [353, p.78]:

"To amplify that last objection, let  $Q =$  "There are exactly 100 Martians and only one wears a hat and George is a Martian." The probability of  $P =$  "George wears a hat" given  $Q$  is 0.01. But a subjectivist can say, "Based on my utility, it's 83.7%!," or whatever. How can you prove him wrong? There are no experiments that can be run because there are no Martians. There are thus no bets that can be made, because there is no "event" to occur or not. Unless probability is treated as logic, you have nothing to say to the subjectivist and must accept his probability as being right, which is absurd."

6 An example of a well-defined event in the context could be "the motor current of drive  $x$  exceeds  $y$  mA at a feed rate of  $z$  using the normal dynamic parameter settings"

7 The membership of a machine to certain health class, for example "The status of the machine is ok".



Following [287], if  $X$  is a collection of objects denoted generically by  $x$ , then a **fuzzy set**  $A$  in  $X$  is defined by a set of ordered pairs:

$$A = \{(x, \mu_A(x)) | x \in X\}$$

where  $\mu_A$  is called the **membership function (MF)** for the fuzzy set  $A$ . The MF maps each element of  $X$  to a membership grade between 0 and 1. That means that elements can belong to a fuzzy set "to a degree", which is measured by the MF.

A few examples of one dimensional membership functions are given here, but these can easily be generalized for  $n$ -dimensional MFs. A **trapezoidal MF** is specified by

$$\text{trapezoid}(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases}$$

a **Gaussian MF** is specified by

$$\text{gaussian}(x; m, \sigma^2) = \exp\left(-\frac{1}{2} \frac{x-m}{\sigma^2}\right)$$

and a **sigmoidal MF** is specified by

$$\text{sig}(x; a, c) = \frac{1}{1 + \exp(-a(x-c))},$$

Binary membership functions or the definition of  $\alpha$ -cuts, where only elements  $x$  of the fuzzy set with a  $\mu_A(x) \geq \alpha$  are considered, can recreate the crisp sets that the probability measures are defined on.

These sets do not express the relative occurrence of a well-defined event in a sample, but express more of a degree of "subjective belief" that a certain event has even occurred.

## 4.5 Remaining Useful Life

The estimation of the Remaining Useful Life can be viewed as a main challenge for any condition monitoring system that tries to predict when a system will cease to function. To aid the discussion of uncertainty generally and prognostics in particular, some important terms are adopted from [354] for the remainder of this section:

- **Time Index:** The variable time in a prognostic application can be modeled as discrete or continuous. The time index  $k$  will be used instead of the actual time. This enables the framework to deal with non-uniformly sampled data. In [355] for example, a scheme based on Lebesgue Sampling is proposed, which does not take samples evenly spaced in the time but in the state dimension of the system. [356] also presents a method for aperiodic sampling. Additionally, time indexes are chosen to be invariant to time-scales.
- **Time of Detection of Fault:**  $k_D$  denotes the time index of the proper time ( $t_D$ ), at which the diagnostic or fault detection algorithm detected the fault. After the detections, the execution of prognostics algorithm should be triggered to start making RUL predictions as soon as enough data has been collected. For some applications there may not be an explicit declaration of fault detection, for example, when the systems are subject to a continuous decay process, as in the case of batteries. The time of fault detection for such systems can then be considered to be equal to the time of system creation.
- **Time to Start Prediction:** The actual time instance, at which the system actually starts to predict future instances, will be denoted with index  $k_P$ .

For certain algorithms, these time instances  $t_D$  and  $t_P$  will coincide, but in general there will be some time delay to collect the necessary data before predictions can be made.

- **Prognostic Features:** Let  $q_n^l(k)$  be a feature at time index  $k$ , where  $n = 1, \dots, N$  is the feature number and  $l = 1, \dots, L$  is the unit under test index. Irrespective of the analysis domain, i.e. signal-based diagnostic entities, like time statistics, spectral characteristics or wavelet coefficients, and model-based entities, like physically meaningful variables or system parameters, are considered as time-series in prognostics. In general, any quantity that can be computed from measurable variables of the system that aides the prognosis, is potentially a prognostic feature.
- **Operational Conditions:** Let  $u^l(k)$  be an operational condition at time index  $k$ , where  $l = 1, \dots, L$  is the unit under test index. The operational conditions describe how the system is being operated and are sometimes referred to as load.
- **Physical Health Indicator** is directly defined by a physical parameter of the system, such as crack length in a gear, or the vibration amplitude of a shaft. The threshold can be decided by the design specification.
- **Probability Health Indicator** is commonly defined by the probability of the current system being in a healthy condition. The value of this health indicator is usually between 0 and 1, where a threshold can be set by a statistical confidence level.
- **Mathematical Health Indicators** are defined by a variable with only mathematical meaning, such as a certain distance metric, e.g. Mahalanobis distance, L2 distance between two distributions, etc. A mathematical health indicator can virtually be any scalar value transformed from a multi-dimensional feature space. For example residuals in systems based on model identification can be used as a mathematical health indicator. The thresholds for this type of indicator have to be learned from training data sets or specified heuristically.

- **Health Index:** Let  $h^l(k)$  be a health index at time index  $k$  for unit under test  $l = 1, 2, \dots, L$ .  $h$  can be considered a normalized aggregate of relevant health indicators and operational conditions.
- **Historical data:** Historical data encapsulates all the information we know about a system a priori. Such information may be of the form of archived measurements or EOL distributions and can refer to variables in both the feature and health domains.
- **Point Prediction:** Let  $\pi^l(k|j)$  be a point prediction of a variable of interest at time index  $k$  given information up to time  $t_j$ , where  $t_j \leq t_k$ . For  $k = EOL$ ,  $\pi^l(k|j)$  represents the critical threshold for a given health indicator. Predictions can be made in the features or health domain.
- **Trajectory Predictions:** Let  $\Gamma^l(k)$  be the trajectory of predictions at time index  $k$  such that  $\Gamma^l(k) = \{\pi^l(k|k), \pi^l(k+1|k), \dots, \pi^l(EOL|k)\}$ .

Of particular interest for the discussion of uncertainty in health assessment and prognostics are state indicators and the Remaining Useful Life, which will be outlined next.

A system state is characterized by a continuous health indicator or discrete degradation stages. The health indicator is usually a continuous-value quantity defined by one of the following methods.

### 4.5.1 Generating an Estimate of the Remaining Useful Life

Unfortunately, analytical expressions for quantifying uncertainty in the RUL estimation are not ubiquitous available for general problem setups; even simple problems involving linear models with only Gaussian random variables, like in Sec. 4.1.1, are not always tractable. It is therefore necessary to use different techniques to quantify the emergent distribution of the RUL.

Being a rapidly developing subject, the researchers in the field of RUL prediction have employed a number of different techniques in multiple research areas, such as regression analysis, time-series forecasting, statistical

survival analysis, etc. that can roughly be clustered into model-based, data-driven and experience-based approaches.

Model-based and data-driven approaches rely on estimating the system's health state and predicting or extrapolating the system's state up to the time when the failure criterion is satisfied. The difference is that the model-based approach makes predictions through physics-based models or system models, while the data-driven approaches make predictions through models learned from the time-series of states through regression or trend analysis or stochastic process modeling. The experience-based approaches estimate RUL directly by modeling the relations between states, the current life and the recorded failure time without an explicit mathematical failure criteria.

In an experience-based prognostic setting, the RUL could for example be calculated by testing multiple specimens of the same component or system. Once a set of run to failure experiments has been conducted, ensuring the same usage and operating conditions, the realized RUL is measured and resulting differences in this quantity are interpreted to represent the presence of variability across the different specimens, or, equally as a realization of physical probabilities. There are various ways to assign a RUL to a system under consideration given information about such experiments. For example, [357] uses fuzzy methods to estimate the similarity of a measurement of the system under consideration to such historical records of runs to failure and estimates the RUL as a similarity-weighted mean of the recorded instances. Given that runs to failure for complete systems are costly for industrial assets, the focus of the further discussion will not be on these approaches.

The distinctive feature of model-based and data-driven approaches to condition-based monitoring is that each component or system is considered by itself, and it is therefore difficult to define variability across specimens, like in experience-based prognostics, as a meaningful concept. At any time instant  $k_p$  at which prognostics need to be performed, the existence of a specific state of the system is assumed. The state is thought to be purely deterministic<sup>8</sup>, i.e.

---

<sup>8</sup> If the idea of the Einstein-Gibbs Phase Space is to be taken seriously, however, the very idea of a well-defined state might be erroneous for mechanical systems

the "true" value is precise, but unknown. The arising uncertainty is therefore epistemic in nature and rules out the use of an easy frequentist interpretation. The goal of condition-based prognostics is to predict the RUL at the relevant level of aggregation in the system<sup>9</sup> at time instance  $k_p$  to be used for making decisions.

Model-based and data-driven approaches can be broken down into certain common tasks. At first the current state and the uncertainty in the state estimate have to be quantified. Then the degradation-prediction model is used to generate estimates of the state at future time instances.

Operational conditions have great impact on the system behavior and degradation processes. The general problem is that only past operational condition patterns can be known and thus used for prognostics. This necessitates assumptions about the future usage of the system. This adds another layer of uncertainty regarding future meaningful environmental conditions and usages. Typically, the future usage is assumed to be not too different from past usages, when additional knowledge, like production schedules, is unavailable. While it is true, as argued in [347] that the subjectivist interpretation of probabilities is consistent with the assignment of probabilities regarding the future use of the systems, it has to be emphasized that the resulting probabilities are only locally true, i.e. are *conditional* on these assumptions. If the assumptions are off by too much, the result will most likely not be useful. The propagation of uncertainties through the models is stopped, when failure or a sensible statistical definition thereof is reached.

For most engineering systems, and for mechanical systems in particular, degradation processes are irreversible endogenous effects that result from tribological phenomena; only effective maintenance, which has to be considered exogenous to the system, can improve the system's health state. A challenge with real-world systems in the field regarding the task of calculating a RUL is exactly that they are almost always maintained regularly with preventive maintenance practices. These actions will, if done effectively,

---

<sup>9</sup> While it is *necessary* to be able to ascertain the health state of each component, it is *not sufficient* to be able to predict the future state of the complete system

recover the system's conditions and change its behavior. Even if the time of preventive maintenance actions can be modeled, their impact is harder to grasp, which adds another challenge for deploying prognostics solutions.

RUL estimation are only meaningful for those engineering system that can exhibit evolving degradation behavior. For systems with stochastic failures, like certain electronic components, predictions can necessarily not be made for individual system instances. Stochasticity in fault modes implies that no particular outcome, but only the aggregate behavior of large enough collections of units or systems can be predicted.

To summarize, the derivation of the RUL can also be seen as an uncertainty propagation problem for model-based and data-driven approaches. The task of estimating it can be broken down into three steps,

- estimation of the current state,
- predicting future states,
- determining the EOL,

which will be discussed in the subsequent sections.

### 4.5.2 State Estimation

Consider the state space model

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (4.2)$$

where  $\mathbf{x}(t) \in \mathbb{R}^{n_x}$  is the state vector,  $\boldsymbol{\theta}(t) \in \mathbb{R}^{n_\theta}$  is the parameter vector,  $\mathbf{u}(t) \in \mathbb{R}^{n_u}$  is the vector of the inputs or loading to the system,  $\mathbf{v}(t) \in \mathbb{R}^{n_v}$  is the noise vector and  $\mathbf{f}(\cdot)$  represents the state function.

The state vector at time  $t_p$ , i.e.  $\mathbf{x}(t_p)$  and the system parameters  $\boldsymbol{\theta}(t_p)$  have to be estimated using information about the system collected until  $t_p$ . Let  $\mathbf{y}(t) \in \mathbb{R}^{n_y}$  be the output vector,  $\mathbf{n}(t) \in \mathbb{R}^{n_n}$  the measurement noise vector and  $\mathbf{g}(\cdot)$

the output equation of the model, then the observations of a model can be described by

$$\mathbf{y}(t) = \mathbf{g}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{n}(t)) \quad (4.3)$$

Bayesian Filtering approaches, like some version of the Kalman filter or particle filtering are used to infer the current state of the system. The resulting estimates of the states should be interpreted as subjective or *Bayesian* uncertainties. The state and the parameters are considered to be deterministic but not known precisely. The probabilities represent a lack of knowledge.

### 4.5.3 State Prediction

Having obtained an estimate of the state of the system at time  $t_p$  corresponding to time index  $k_p$ , the future states are predicted using the time-discretization of Eq. (4.2) and estimates about the future loading conditions at discrete time indices  $k$ . Because there is no way of obtaining data from future instances, there is no way to obtain corrective information for any error in the estimate of the initial state. Additionally, the process noise and loading conditions at future instances are necessarily uncertain, which will cause the uncertainty about future states of the system to become larger with an expanding prediction horizon. [358] introduces a method of uncertainty quantification, that does not make assumptions about the underlying distribution of the noise. The impact of input uncertainty on prognostic algorithms is examined in [359].

### 4.5.4 Determining the End of Life

The EOL of a system is reached, when it becomes unable to fulfill at least one requirement regarding the performance of the system. The health of the system is expressed through the Health Index  $h$ .  $h$  consists of  $n_h$  health indicators. Each health indicator  $d_i(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{y}(k))$  maps a given point in the joint state-parameter-observation space to the right half-open real interval  $\mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}_0^+$ . A value of 0 indicates a failure. These individual health indicators are aggregated in the Health Index,



$h(d_1(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{y}(k)), \dots, d_{n_h}(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{y}(k))) \in \mathbb{R}_0^+$ . This health index can be used in the boolean *threshold function*  $T_{EOL}$ , defined as

$$T_{EOL}(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{y}(k)) = \begin{cases} 1, & h(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{y}(k)) = 0 \\ 0 & \textit{otherwise} \end{cases} \quad (4.4)$$

$T_{EOL}$  is equal to 1, if the health index reaches zero. The EOL can then be defined as the earliest time instance at which the value of  $T_{EOL}$  becomes equal to 1. Let  $EOL(t_p)$  be the first time instance, for which  $T_{EOL} = 1$  holds. The Remaining Useful Life is the time duration between the present and  $EOL$ .

Practical problems in the health management and prognostic domain may consist of:

- non-Gaussian random variables effecting the RUL prediction,
- a possibly non-linear multidimensional state-space model,
- uncertain future loading conditions,
- a possibly complicated Health Index function over a multidimensional space.

The fact that the distribution of the RUL depends on the quantities indicated in Fig. 4.1, makes it clear that it is mistaken to assign artificial probability distribution types to it. The predictability of the RUL at time  $t_p$ , i.e.  $R(t_p)$  necessitates that the following conditions are met:

- Using the present state estimate ( $\mathbf{x}(k_p)$ ) and the state space equation, the future states ( $\mathbf{x}(k_p + 1), \mathbf{x}(k_p + 2), \dots, \mathbf{x}(k_E)$ ) can be calculated.
- Estimates about the future loading conditions ( $\mathbf{u}(k_p + 1), \mathbf{u}(k_p + 2), \dots, \mathbf{u}(k_E)$ ) are available to calculate the future states values using the state space equation.
- Parameter values form time-index  $k_p$  until time-index  $k_E$  ( $(\boldsymbol{\theta}(k_p), \boldsymbol{\theta}(k_p + 1), \dots, \boldsymbol{\theta}(k_E))$ ) are obtainable.
- Process noise ( $\mathbf{v}(k_p), \mathbf{v}(k_p + 1), \mathbf{v}(k_p + 2), \dots, \mathbf{v}(k_E)$ ) can be estimated.

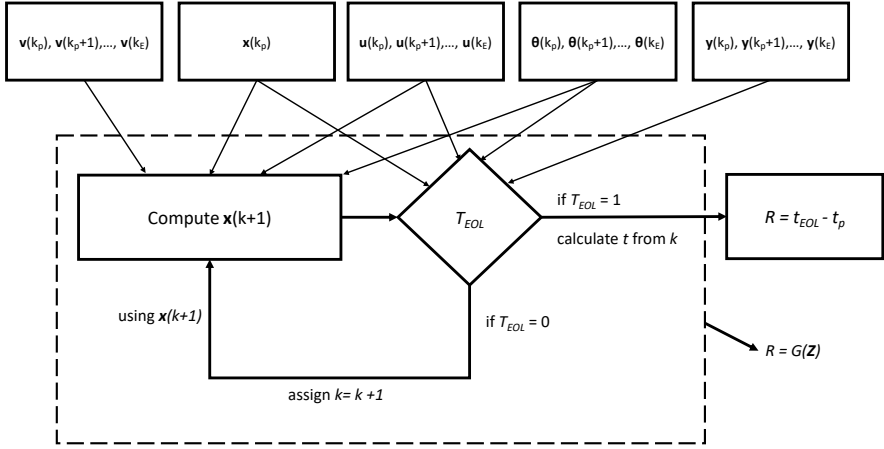


Figure 4.1: Graphical Representation of  $G(\mathbf{Z})$ . [347], adapted.

These quantities can be regarded as independent quantities with regard to the RUL prediction. The RUL thus becomes a dependent quantity. Let  $\mathbf{Z} = \{Z_1, Z_2, \dots, Z_i, \dots, Z_n\}$  signify the vector of the above quantities, where  $n$  is the length of the vector  $\mathbf{Z}$ , or, equivalently, the number of uncertain quantities influencing the RUL prediction. The calculation of the RUL  $R$  can be then be expressed as

$$R = G(\mathbf{Z}) \quad (4.5)$$

The elements of  $\mathbf{Z}$  are uncertain. They are used to compute their combined effect on the RUL prediction. Estimating the uncertainty of  $R$  is equivalent to propagating the uncertainty in  $\mathbf{Z}$  through  $G$ .

The problem of estimating the uncertainty in  $R$  using uncertainty propagation techniques is non-trivial even for rather simple problems. Only in rare cases is it possible to obtain an analytical solution. Some of these special cases are given by [347]:

- Every quantity contained in  $\mathbf{Z}$  is normally distributed and  $G$  can be expressed as a weighted linear combination of the quantities in  $\mathbf{Z}$ . In this

case,  $R$  will also be normally distributed and its statistics can be calculated analytically.

- Every quantity contained in  $\mathbf{Z}$  are log-normally distributed, and if the logarithm of  $G$  can be expressed as weighted combination of the quantities in  $\mathbf{Z}$  then  $\log(R)$  follows a normal distribution, for which the relevant statistics can be expressed analytically.

While Gaussian distributions and linear state space models may be a common occurrence in prognostics and health management, this does not imply in and of itself that  $G$  is linear. The use of a non-linear threshold function renders  $G$  non-linear. These results also hold true for the simplified model introduced in Sec. 4.1.1.

In order to compute the probability density function of  $R$  in more general cases, it is necessary to resort to rigorous computational methods. There are different types of sampling methods such as Monte Carlo sampling, Latin hypercube sampling, importance sampling and unscented transform sampling to tackle this task. There are also some analytical methods for the calculations of the probability distribution of  $R$ , that use reasonably many evaluations of  $G$  and approximate the probability distribution reasonably good. Amongst these methods are the first-order second moment method and the first-order reliability method. [347] provides references to research utilizing these methods. Additionally, there are also hybrid methods such as the efficient global reliability analysis method which involves sampling as well as analytical methods. All of these methods empirically calculate the probability distribution of the RUL; while some calculate the probability density functions, others calculate the cumulative distribution function, or generate samples from the target distribution. Due to some limitations in each of these methods, it might not be possible to accurately generate the probability distribution of  $R$  and only an infinite number of samples guarantees an accurate characterization of the true probability density function<sup>10</sup>. The choice of the propagation methods therefore introduces an

---

<sup>10</sup> An even this probability density function is only accurate in a mathematical sense, as its correctness is conditional on the accepted evidence and modeling assumptions, for which there is no rigorous mathematical criterion

additional amount of uncertainty. This uncertainty can possibly be reduced by more advanced sampling techniques or by an increase in computational power.

## 4.6 Model Validation

Model validation is the process of quantifying the extent to which a computational model is supported by available experimental data.

A particularly useful method is the comparison between the predicted outputs of the model for certain situations and the experimentally observed real outputs. A valuable tool in this regard is a computable measure for the comparison of the divergence of predictions and measurements.

An important aspect of model validation is of course the ability to treat multiple sources and types of uncertainty accurately. In the context of validation, both the model inputs and the outputs can, if at all, only be measured contaminated by measurement noise and are therefore uncertain in their own right. A rigorous model assessment should account for physical variability, measurement error, solution approximation error etc. and be able to ascertain the performance not only qualitatively, but quantitatively to aid in gauging the justified amount of confidence in the model's predictions.

Model validation mirrors some the preceding debate in Sec. 4.3 about the interpretation of probabilities. While the classical method of hypothesis testing, inspired by frequentist thinking, is based on the comparison of certain statistics of the model's prediction, mostly mean and variance, with the same statistics of the experimental data, the subjective interpretation of probability offers methods such as Bayesian hypothesis testing, which allow for a comparison of the distributions of model predictions and corresponding experimental data. They can also account for various types of uncertainty [352]. Along with developing the fundamental abilities of state estimation, uncertainty quantification and propagation in order to be able to create an estimate of the RUL, deploying such models to real world problems poses the rather intricate question of *performance evaluation*. Prognostic concepts still lack unambiguous standard definitions and consistent interpretations. It seems likely that this is at least par-

tially caused by the breadth of end-user requirements and domain-specificity regarding time scales, available information, sample sizes etc. [360]. Users of any prediction will naturally want to know "how good" prognostics estimates are before even begin to consider changing the maintenance plan according to the suggestions of the system. This makes it imperative that a fair amount of trust in the prognostic system is created before their predictions are incorporated into the decision-making process. See Sec. 5.1 for discussion of that aspect.

Without interpretable and rigorously derived confidence bounds, predictions will lose a lot of their possible benefits and could cause costly misunderstandings. While the confidence bounds are indicative of the ability of an algorithm to handle uncertainty, performance metrics are supposed to be a tool to judge the real world performance. According to [353, pp. 39 ff.], each and every probability statement is correct, bearing any mistakes in the calculus involved, but only locally so. The probability assignments are dependent on the accepted evidence. Only the comparison between the predictive probabilities and the actualized results will reveal whether the set of accepted evidence was a *useful*<sup>11</sup> choice. There are scientific, administrative and economic reasons that make the availability of meaningful metrics desirable: Metrics are necessary to evaluate the *performance* of an algorithm, for example to measure the influence of a change in the available data set on prediction accuracy. Metrics are also a necessary precondition for the formulation of *requirements* an algorithm has to meet before it can be deployed to production systems. Furthermore, they can help to ascertain the *risk* that is associated with decisions based on the suggestions of the algorithm [354]. Despite the obvious desirability of standardized performance metrics, there still seems to be lack of them in the literature.

Currently, there is no widely accepted performance evaluation method for prognostics applications, despite numerous performance metrics being available. The selection of them is usually a subjective, applications specific choice. Some metrics in use are derived from other prediction related areas such as the Mean Squared Error from regression analysis. There is some concern that these

---

<sup>11</sup> Usefulness is concept that is not well-defined in probability theory.

metrics are not well-suited to address the special needs of prognostics systems. [354] clusters the available *performance* metrics into three general groups:

- **Accuracy-based metrics:** These types of metrics evaluate the goodness of the fit of the prediction and an already recorded observation. Amongst this type of metric are for example Bias, Root Mean Square Error or Mean Absolute Error.
- **Precision-based metrics:** These metrics evaluate the spread of the prediction errors. These are invariant to bias. Amongst the more well known metrics are for example the standard deviation or them mean absolute deviation from the sample median.
- **Robustness-Based metrics:** Graphical tools like the Reliability Diagram or the Receiver Operating Characteristic can be used, if the prediction problem is transformed into a classification problem by repeated trials, to depict certain characteristics of the prognostic algorithm, like the trade-off between the false positive rate and the false negative rate. The Brier Score or, respectively, the Area under the ROC curve (AUC) can then be used as numerical measures derived from the diagrams.

Additional to these metrics, there are economic metrics, like the Return on Investment (ROI) or the Mean Time Between Failure (MTBF). From an implementations point of view, the Total Value of the system is of eminent importance, as it describes the monetary value of the implementation of the system. The total value is given by

$$V_{Total} = \sum_{i=1}^{n_{FM}} TV_i - A - O - (1 - P_c) \cdot \delta,$$

where  $TV$  is the Technical Value of a system,  $A$  is the development or acquisition and implementation cost,  $O$  is the operational and maintenance cost,  $P_c$  represents the computer resource requirements and  $\delta$  is the cost of the com-

puter system.  $n_{FM}$  is the number of monitored failure modes,  $i$  is their index. The Technical Value of a system is given by

$$TV = P_f(D \cdot \alpha + I \cdot \beta) - (1 - P_f)(P_D \cdot \Phi + P_I \cdot \theta),$$

where  $P_f$  is the probability of the occurrence of the failure mode,  $D$  is the overall detection confidence metric,  $\alpha$  represents the savings realized by detecting the fault in advance,  $I$  is the overall fault isolation confidence metric,  $\beta$  summarizes the savings realized by identifying the fault in advance,  $P_D$  is the probability of a false positive detection,  $\Phi$  is the average cost of a false positive detection,  $P_I$  is the probability of a misidentification and  $\theta$  is the average cost of that misidentification [354].

If estimates of the relevant terms can be given, the Total Value can be seen as rather comprehensive summary of the capabilities of a monitoring system with respect to its costs.

Predictions of the RUL are made based on the history of condition measurements. A challenge is that the consequences of errors in the prediction of the RUL are not symmetric with respect to the true value. If the predicted EOL is too early, still functioning components could be replaced or unnecessary verification procedures could be triggered. If the predicted EOL is too late, there might be too little time left to take corrective actions before the system fails. In most cases, it is therefore preferable to err on the early side. Predictions made early on have necessarily access to less information about the dynamics of the fault evaluation and are required to predict farther in time, which makes the prediction task more difficult as compared to predicting at a later stage. The traditional performance metrics given above seldom take this factor into consideration and are therefore only to a limited extent able to capture what would intuitively be considered the true performance of predictions. [360] attempts to address this problem by proposing a performance evaluation framework that consists of four performance metrics in a hierarchy. These are:

- **Prediction Horizon:** evaluates the actionable time window between the time instance when an algorithm under consideration can make predictions with a specified accuracy about the EOL.
- $\alpha - \lambda$  **Performance:** evaluates whether the predictions at a given time within the Prediction Horizon stays within the accuracy requirement. The accuracy requirement is given as a percentage of the actual RUL.
- **Relative Accuracy:** quantitatively evaluates the absolute percentage error of a prediction at a time within the prediction horizon, if the algorithm has met the requirements of the previous metrics.
- **Convergence:** evaluates how fast the prediction performance with respect to any chosen accuracy-based metric improves towards the EOL of the instance, if the algorithm has met the requirements of the previous metrics.

Other attempts to make predictions comparable, are for example, [361], which presents interpretable and verifiable thresholds on the performance of predictors in the form "The prognostic algorithm shall provide a minimum of  $\langle TTM \rangle$  hours time-to-maintenance such that between  $\langle Lower \rangle\%$  and  $\langle Upper \rangle\%$  of failures of component ABC will be avoided with  $\langle Confidence \rangle\%$  confidence." To qualify prognostic algorithms with regard to their applicability to real maintenance systems, [362] proposes a method based on the concept of technology readiness levels. [363] explores some existing scalar metrics like the Mean Absolute Error or the Confidence Convergence Horizon for evaluating prognostic models and claims to offer a more intuitive aggregate measure to represent the quality of the model.

## 4.7 Model Calibration

Model calibration refers to the adjustment of model parameters to increase the match between experimentally observed data and the model output. It belongs to a rather large group of mathematical problems, namely inverse problems. When a computational model is used to predict the outcome or effect of a par-



ticular, a priori known phenomenon<sup>12</sup>, this is referred to as forwards problem. The inverse problem makes use of measured effects and tries to infer some characteristic about the underlying generative system, which are considered to be the causes. Inverse problems could be defined as the class of problems that seek to determine unknown causes based on observation of their effects. For the remainder of this chapter inverse problems will be dealt with as synonymous with model parameter estimation or in the context of time-dependent outputs with system identification [352, pp. 118-119]. According to Hadamard, a well-posed problem should have the following properties: a solution exists, that solution is unique and the solution is stable [37, p. 315]. While forward problems are generally well-posed, inverse problems are not. As inverse problems are not well-posed, there can be no certainty about the actual cause of a certain effect and so the confidence associated with the multiple solutions becomes of eminent importance.

Parameter estimation, or inferring difficult to measure or even unobservable quantities through measurements of a dependent variable, has been a significant research topic in various fields. Consider the computational model  $y = G(\mathbf{x}, \boldsymbol{\theta})$ , with  $\mathbf{x}$  being the independent input and  $y$  the dependent output. Normally, the measurement is assumed to be unbiased, i.e.  $\epsilon_i = y_i - G(\mathbf{x}_i)$  is modeled as a zero mean normally distributed variable. The quantity  $\epsilon \sim \mathcal{N}(0, \sigma^2)$  is referred to as fitting error. The purpose of parameter estimation is to generate an estimate of  $\boldsymbol{\theta}$  that best tunes a model to explain the observed data.

### 4.7.1 Least-squares methods

The method of least squares is based on minimizing the squared difference between the model prediction and the actually observed data. An error measure  $L(\boldsymbol{\theta})$  is computed as

$$L(\boldsymbol{\theta}) = \sum_{i=1}^n (y_i - G(\mathbf{x}_i, \boldsymbol{\theta}))^2. \quad (4.6)$$

---

<sup>12</sup> Colloquially named *cause*, even though there is philosophical debate about the existence of this. To avoid some deep philosophical questions about the true nature of *cause*, the rather pragmatic approach of causal relativism will be adopted

The so-called least squares estimate of  $\theta$ , symbolized as  $\theta_{LS}$  is computed as

$$\theta_{LS} = \arg \min_{\theta \in \Theta} L(\theta). \quad (4.7)$$

Note the necessity for paired input-output values. If the model  $G(\mathbf{x}_i, \theta)$  is linear with respect to inputs  $\mathbf{x}$  and parameters  $\theta$ , the procedure becomes a linear regression problem, otherwise non-linear techniques have to be used and the procedure becomes known as non-linear regression. Another assumption is that the inputs are certain, i.e. there is no measurement error on them and that the output measurement error is totally contained in  $\epsilon$ . In the frequentist interpretation, the parameters  $\theta$  are assumed to be deterministic, and the estimate may not coincide with the true value of them. If the size of the data sequence becomes infinite, it can be proved that  $\theta_{LS}$  tends to the true value. Uncertainty in least squares estimates can be expressed by using confidence intervals on the least squares estimates. They are calculated at particular significance levels  $\alpha$ . The confidence bounds are given as intervals  $[\theta_{\alpha, min}, \theta_{\alpha, max}]$ . For a given  $\alpha$ , an error  $L_\alpha$  is defined as

$$L_\alpha = L(\theta^*) \left( 1 + \frac{p}{m-p} F_{p, m-p}^\alpha \right), \quad (4.8)$$

where  $F$  refers to the F-statistic evaluated at significance level  $\alpha$ ,  $p$  refers dimension of the parameters vector and  $n$  is the number of data available for calibration, as defined earlier. The confidence interval of  $\theta$  is the region, where  $L(\theta) \leq L_\alpha$  is satisfied. Obviously, the bounds on the parameters can be found by solving constrained minimization problems, but these will become increasingly computationally expensive with an increase in the dimension of  $\theta$ . Also, the confidence intervals should not be confused with a probability density function of  $\theta$ , since in the underlying interpretation the parameters are assumed to be deterministic, there can be no such entity as a probability density function for them. A propagation of uncertainties thus becomes an intellectually barren exercise [352, p. 123].

### 4.7.2 Likelihood Method

The least squares estimation procedure is in essence an optimization problem. But it can also be shown that the least squares approach maximizes the probability that the data can actually be observed under the conditions  $\epsilon \sim \mathcal{N}(0, \sigma^2)$ . This probability,  $P(D|\theta)$ , where  $D$  denotes all the available input-output data, is referred to as likelihood function of  $\theta$  and is denoted as  $L(\theta)$ . Note that, since data  $D$  has already been observed, "the probability of observing data" is meaningless, as there is no uncertainty about the data that has already been collected. Therefore, this quantity was renamed "likelihood". The likelihood function does not follow the laws of probability and must not be confounded with probability distributions or distribution function. In fact, the absolute values of the likelihood function are of no particular interest, as it is only meaningful up to a proportional constant. Only the relative values are important. The concept of likelihood is used in both interpretations of probability. From a frequentists point of view, the likelihood function can be maximized to obtain the maximum likelihood estimate of the parameters. It is also possible to construct likelihood-based confidence intervals for the inferred parameters. In the subjective probability interpretation the likelihood function can be viewed as a collection of weights and there they are also only meaningful up to a constant proportionality factor. Assuming that  $n$  pairs of data are independent, the likelihood function can be constructed as

$$L(\theta) \propto \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(y_i - G(\mathbf{x}_i, \theta))^2}{2\sigma_i^2}. \quad (4.9)$$

The maximum likelihood estimation tries to find those model parameters that maximize the likelihood function over the parameter space

$$\hat{\theta} = \arg \max_{\theta \in \Theta} L(\theta). \quad (4.10)$$

or, put alternatively, to find the model parameters with the highest likelihood of observing exactly that data that was observed.

### 4.7.3 Regularizers for Model Calibration

It was already stated that parameter estimation belongs to a class of inverse problems which are often ill-posed. To enforce unique solutions, regularizers are often added to the problem formulation, which can introduce additional constraints, i.e. penalties for the complexity or restrictions on the smoothness of the function or bounds on the parameters. Some popular regularizers can also be interpreted as priors on the parameters, while others do not have a direct correspondence in the probabilistic framework. In the least-squares framework this is realized by adding a regularizing term,

$$L(\boldsymbol{\theta}) = - \sum_n \log(p(\mathbf{y}_n | \mathbf{x}_n, \boldsymbol{\theta})) + \frac{1}{\lambda} \mathcal{R}(\boldsymbol{\theta}), \quad (4.11)$$

while in the probabilistic setup it is done by augmenting the likelihood function

$$\log(p(\mathbf{y}, \boldsymbol{\theta})) = - \sum_n \log(p(\mathbf{y}_n | \mathbf{x}_n, \boldsymbol{\theta})) + \log(p(\boldsymbol{\theta} | \lambda)). \quad (4.12)$$

For some popular regularizers there are exact correspondences, given in Table 4.1.

$\mathcal{R}(\boldsymbol{\theta})$	$p(\boldsymbol{\theta}, \lambda)$
$\ \boldsymbol{\theta}\ _2^2$	$\mathcal{N}(\boldsymbol{\theta}   0, \lambda)$
$\ \boldsymbol{\theta}\ _1$	$\text{Lap}(\boldsymbol{\theta}   0, \lambda)$
$\ \boldsymbol{\theta}\ _p, p > 0$	$\exp(-\lambda \ \boldsymbol{\theta}\ _p)$

Table 4.1: Correspondence between popular regularizers in the least-squares and the probabilistic setup

### 4.7.4 A Third Perspective

As was shown in the preceding sections, testing and estimation, whether frequentist or subjectivist, are the backbone of classical analysis.

Some research, such as [364], considers this to be problematic, because the

questions that tend to get answered by these tests are not necessarily what the typical user is asking. No matter how much certainty is present in the parameters of a model, the uncertainty in the observable itself must always be greater. The goal of all probability modeling is to obtain a probabilistic model for some set of measurable variables, like

$$\Pr(Y \in y|X, D, M), \tag{4.13}$$

where  $y$  are values of the observable  $Y$  which are of interest to some decision-maker,  $D$  is the collection of past observables,  $M$  is the collection of premises, which lead to the probability model.  $X$  is the assumed new observable. If a parametrized model is used, (4.13) corresponds to the posterior predictive distribution in the subjectivist interpretation of probability and  $M$  incorporates those premises or assumptions from which the priors are deduced. [364] argues that models should be rare, because according to him most probability is not quantifiable and caution should be exercised to not enforce the quantification of something genuinely unknown. In the "probability as logic" interpretation of Sec. 4.3.3, even in a case of purely made up numbers for certain probability, the resulting probabilities can be correctly induced, given the evidence. How meaningful, or even useful they are, has to be judged by the supplied premises for these creations. But this assessment is not a statistical concept by itself. There is no such thing as an unconditional probability of any proposition in this interpretation. If, at some later point, it is decided that the same variables in  $D$  and  $X$  are of no interest anymore or that some variables are missing, the model is updated accordingly, and the probabilities derived from the updated model will also be correct. At least conditional on the given evidence, that is.

One model may be superior, however, but this can *only* be judged with respect to the decisions made conditional on the models. The third way of probability is, conditional on  $D$  and  $M$ , to vary  $X$  in the range of expected by or important to some decision-maker and see how these change the probability of  $Y \in y$ .

If a particular  $X$ , as it ranges along the chosen values, do not change the probabilities in any *important* way, then these  $X$  are themselves not important.

The opposite is also true. Importance is a matter of decision, which varies by the decision-maker. Importance is not a statistical or probability concept and therefore cannot be ascertained within probability models.

Relevance on the other hand is a probabilistic concept. If the probability of  $Y \in y$  changes in any way, while  $X$  does, then  $X$  is relevant for understanding  $Y$ , else it is not.

There is no hypothesis testing as in the frequentist paradigm and no estimation of parameters as in the Bayesian paradigm. There are only plain, interpretable, and verifiable probability statements. These statements can, should and must be verified. This reduces over-certainty, but cannot eliminate it unless models are deduced from a priori statements. Importance and relevance, like probability itself, are always conditional on the accepted assumptions. Only information presumed probative to a model should be added, if there is a plausible belief that the information is related to the cause of the observable of interest, which enforces a certain sparseness of the models.

## 4.8 Challenges

As outlined in this chapter, there are several significant conceptual and practical challenges in using different uncertainty quantification and propagation methods for health management and especially prognostics. To facilitate decision-making in the presence of uncertainty, it is therefore important to understand these challenges and the requirements for a prognostics and health management system:

- **Feasibility:** The chosen uncertainty quantification and propagation methods need to be computationally feasible to be implemented in online health monitoring. This requires timely calculations, while uncertainty quantification methods can involve some rather time-consuming and computationally expensive calculations (i.e. inverses of large matrices). Also, there are rather demanding requirements on the availability of data. Without enough run-to-failure tests, domain knowledge, sufficiently many sen-

sors and high-enough sampling frequency, there simply is not enough data to implement a prognostics scheme.

- **Reproducibility:** Automated verification and certification protocols require algorithms to produce repeatable results. Several uncertainty quantification methods are non-deterministic, however, and will produce slightly different result for every run, due to the frequent use of random selection or initialization techniques.
- **Performance:** The chosen uncertainty quantification and propagation methods need to produce a good estimate of the probability distribution of the RUL: This implies that the entire probability distribution over  $\mathbf{Z}$  and the functional relationships represented by  $G$  in Eq. (4.5) need to be accounted for. A lot of methods try to ease the problem by only using a few statistics of the distribution of  $\mathbf{Z}$ , like the mean or standard deviation, or try to approximate  $G$  with simpler models, for example linear ones. The chosen methods also need to be able to handle possibly multimodal distribution types. The existence of a suitable performance metric for prognostic tasks is not guaranteed.
- **Granularity:** While the capability of accurately calculating the RUL is important, for certain decision bounds on RUL might be sufficient. The chosen methods have to be able to deliver these sufficiently quickly and accurately.

[365] compares fault management policies computed with and without future uncertainty to illustrate the limiting effects of model uncertainty on prognosis-informed fault management policies. Some authors have tried to implement such rigorous uncertainty propagation schemes for some select systems, like for example lithium-ion batteries in [366], [367]. For industrial systems, which tend to have non-linear interactions, to have only sensors necessary to fulfill their predefined functions, to be complex and to be used application specific, the acquisition of run-to-failure data is often, if at all feasible, either prohibitively expensive or too dangerous for the personnel to try for ethical reasons. The epistemic uncertainty thus induced about the current state of the system under

consideration, problems with modeling the system and the lack of knowledge about future loading conditions severely limits the applicability of prognostics-based decision-making schemes.





---

## **5 PERMEATED - Pragmatic Explainable Relativistic Modular Extensible Adaptive Trust-building Environment for Decisions**

Following the discussion about uncertainty in Chapter 4 in the health assessment and especially the prognostic part of condition monitoring systems, it has become clear, that it is necessary to use a systematic approach to capture the available information about a system and use it to the extent that is possible. There have been several attempts to standardize prognostics and health management systems. One among them is given in the appendix of [368] and details are given in [369]. This norm specifies the requirements for an information and processing model to which an open condition monitoring and diagnostics architecture needs to conform. [370]–[373] present work on the implementation of this OSA-CBM standard.

While the guidelines seem to be well suited to aid in the development of the technical side of the system, it simultaneously also seems to neglect a major component necessary for the successful implementation of a condition monitoring systems, namely the addressee of the diagnostic information. Humans are a major component of any such system. They will ultimately make the decisions and enabling better decision-making is what such systems are supposed to achieve. It is important to emphasize the differences in the concepts of prob-

ability, relevance and decision. Relevance is a choice. Different experts on the same topic can disagree about the relevance of certain factors and whether or how to include certain variables in the analysis of a problem. While probability theory offers the concept of conditional independence to test for any influence of a variable to a predicted result, it does not offer a rigorous way to test for the relevance of that variable. This choice is up to the creator and even more importantly the user of the model. A decision can be influenced by probabilities about certain events it is concerned with, but they are not determined by probabilities arising in their context. More often than not, a "hunch", "experience" or "intuition" is used as the basis for decisions. While this might not be desirable from the standpoint of researchers trying to build models of human behavior, this seems to be a constraint that condition monitoring systems have to deal with. If decisions about maintenance are not to be strictly separated from operators or even owners, human decision-makers are a key component of the overall system and should be handled accordingly. Their requirements ultimately shape how information needs to be presented to be useful, it is their trust or lack thereof that determines how a system is used. In Sec. 5.1 will take a closer look at the question of trust in technical systems. This chapter will make the case for biasing the selection of machine learning algorithms to the set of interpretable algorithms to facilitate a more rapid adoption of data-driven decision support systems.

## 5.1 Trust in Technical Systems

A system that is not used, can by definition not contribute to the success of any operation.

The dominant idea to increase the usage of a prognostics system has been to create automatic aids that are increasingly more accurate and thus, the rationale goes, trustworthy. This strategy assumes that this unilateral increase in performance will also increase the overall performance of the human-machine system. This assumption, however, does not appear to be necessarily correct.

[374] for example showed that the optimization of the automated systems' per-

formance, measured as a high detection rate and few false alarms, did not optimize the human-machine team's performance. While it is reasonable to expect synergies in human-machine teams [375], it is most likely only achievable by optimizing the joint system. If an operator's trust in a system is inappropriate, i.e. too much trust is put into an untrustworthy system, which is called *misuse*, or too little trust is put into a trustworthy one, which is called *disuse*, the results can be unsatisfying. [376] provide some anecdotal evidence for the disastrous results of disuse of automated monitoring systems, i.e. the ignoring of warning systems. Misuse of a system is known as contributing factor for complacency, i.e. a state of mind characterized by a low level of suspicion.

The level of trust put into an automated aid, like the trust put into a person, is thought to be formed dynamically, mirroring the trust dynamics identified in the interpersonal trust literature, i.e. a rapid decline of trust after a failure event and only a slow increase in trust after an extended period of satisfying performance [377].

It is likely that trust of AI systems will have to be gained over time, similar to personal relationships. If things behave repeatedly as they are expected to do, trust is earned. To facilitate the rapid increase of trust into AI systems, building them in a way conducive to the development of trust seems like a promising route.

A major concern with AI systems is *bias*, which could be an artifact of the data used for training or of the used algorithm. This concern seems to be especially pronounced for AI systems which are used to make decisions that pertain to humans and could disadvantage certain groups of people. The tasks of assigning credit scores or selection applicants are examples of such potentially problematic applications. Bias detection and management can be seen as an element of the larger problem of algorithmic accountability.

It is believed that AI systems must be able to explain how and more importantly why they arrive at particular conclusions so that human supervision of the rationale of the system remains possible. This intuition has led to the formulation of the "Principles for Accountable Algorithms" [378], which are responsibility, explainability, accuracy, auditability and fairness. While these principles are

formulated with the idea of widespread algorithmic decision on the societal level in mind, at least two of these principles seem to be well suited to increase the amount of trust that users put in the decisions of algorithm itself: explainability and accuracy. Responsibility and auditability are also important factors, but focus more on the deployment and operation of the condition monitoring system than on the algorithm itself. Trust is thus not simply confidence that a model will perform well, which any sufficiently accurate, albeit opaque, model could demonstrate.

Trust in algorithms also necessitates the ability to know which and why instances have been correctly or wrongly identified. [379]

If the model's mistake tend to be in instances, where humans are also frequently mistaken, and a model is accurate, where humans are typically accurate, it might be easier to give control to such a system. Otherwise, it might always be advantageous to maintain human supervision. The principles of explainability and accuracy will be outlined next.

### 5.1.1 Explainability

Explainability is the name for a criterion, which means to ensure that decisions made by algorithms as well as any data driving those decisions can be explained to end-users and stakeholders in non-technical terms. [378] suggests the following questions as guidelines for a system that tries to satisfy the demand for explainability:

- "Who are end-users and stakeholders?"
- "How much of your system can be explained to users and stakeholders?"
- "How much of the data sources can be disclosed?"

To be able to answer the posed questions, the system has to be designed to do so. This involves certain design decisions:

- Design a plan for how decisions are explained to users of those decisions.
- In case of usage of machine-learning models:

- consider using interpretable or explainable models,
- describe the training data, including how, when and why it was collected or sampled.
- Disclose the sources of any data used and as much as possible about the specific attributes of the data. Explain how the data was cleaned or otherwise transformed.

Especially the last design choice might face legal challenges, and it will not be possible to apply it in every situation. In those situations, it will be harder to generate trust in the system. The requirements for explainability are also a constraint on what tools a data scientist can choose from for creating diagnostic systems.

### 5.1.2 Accuracy

In line with the discussion in Chapter 4, the identification, logging and articulation of sources of error and uncertainty throughout the algorithm and its data sources is emphasized by [378] as a major task for the deployment of algorithmic decision support systems. The goal is to gain an understanding about the expected and worst case implications of this uncertainty and to use it to design mitigation procedures. The guiding question for accuracy in this rather specific sense are:

- "What sources of uncertainty are present and how to mitigate their effect?"
- "How confident are the decisions made by the algorithmic system?"
- "Have alternative data sources been tested?"

Suggested starting points towards building an accurate system are:

- Assess the potential for errors in the system and the resulting potential harm to users.
- Develop a process by which errors in input data, training data or decisions can be corrected.

- Perform a validity check by randomly sampling a portion of the data and manually checking for its correctness. Report on the error rate publicly.
- Determine how to communicate the uncertainty of each decision.

Knowledge about the uncertainty of information that is used in decisions is very important from a perspective of rational decision-making and to counter tendencies to misuse or disuse of prognostic systems. It should be noted, however, that the transfer of uncertainty present or produced in the data collection or analysis processes to a decision-maker is not always welcomed.

## 5.2 Interpretability of Diagnostic Systems

If the system can explain its reasoning, it can be verified whether that reasoning is sound with respect to auxiliary criteria. There is still little consensus on what interpretability is with respect to machine learning and how to evaluate or benchmark it.

Currently, evaluation typically falls into one of two categories. The first evaluates interpretability in the context of an application: if the system is useful in either a practical application or a simplified version of it, then it is assumed to be somehow interpretable.

The second evaluates interpretability via a quantifiable proxy. It is typically claimed that some model classes, e.g. sparse linear models, rule lists, gradient boosted trees, are interpretable and then algorithms to optimize within that class are presented. Essentially, both evaluation approaches follow Justice Stewart's famous "But I know it when I see it"-criterion.<sup>1</sup> [380] argue that this apparent lack of rigor simultaneous is and is not a cause for concern: They argue that the notions of interpretability above appear reasonable because that is what they are: they meet the first test of having face-validity on the correct test set of subjects: human-beings. Unfortunately, differences in familiarity with models in test subjects will probably have an influence on the complexity of a model that is still deemed interpretable.

---

<sup>1</sup> Although in a markedly different context.

However, the basic notion leaves many kinds of questions unanswerable like whether all models in the model classes defined to be interpretable are equally interpretable. Quantifiable proxies like sparsity seem to allow comparisons, but fail to answer questions like how to compare models sparse in features versus a model sparse in prototypes.

### 5.2.1 What is Interpretability?

In the context of machine learning systems, the rather mundane definition of interpretability will be adopted for the remainder of this discussion:

Interpretability in a machine learning system is the ability to explain or present in understandable terms to a human.

A formal definition of the concept remains somewhat elusive, which does not seem to be a limitation of the concept in this context in particular.

In the field of psychology explanations are defined as "[...] the currency in which we exchanged beliefs". Questions such as what constitutes an explanation, what makes some explanations better than others, how explanations are generated and when explanations are sought, are also just beginning to be addressed in that field as well [380]. [381] makes the argument that explanations are the very building blocks of scientific knowledge and that good explanation have to be hard to vary to be good.

### 5.2.2 Why Interpretability?

Practically, interpretability is used to confirm other important desiderata of machine learning systems: There may exist many auxiliary criteria that one may wish to optimize besides accuracy and generalization capacity. For example notions of reliability and robustness against variations in parameters, inputs and the environment, causality or the retrieval of additional information might be important for an application. Interpretability can assist in qualitatively ascertaining whether desiderata, such as those mentioned above, are met. In industrial contexts, being provided a feasible explanation that fails to correspond to



a known causal structure, provides grounds to be concerned about the validity of a diagnostic module. It does not seem that all machine learning systems require interpretability. Advertisement servers, postal code detection, etc. compute their output without human intervention. Explanation is not necessary, because either there are no significant consequences for wrong decisions or the problem is sufficiently well-studied and validated in real applications that the system has gained trust despite its imperfections [380]. In other realms however, where the stakes for mistakes are higher, this criterion seems to be beneficial in creating explanations for algorithmic decisions, which might ultimately create trust in algorithmic decisions. The question thus becomes when explanations become appropriate or even necessary. In the following subsections, criteria for the desirability of explanations are given.

### **Incompleteness**

Part of the answer for when explanations are needed seems to point to uncertainty or more generally incompleteness, which creates a fundamental barrier to optimization and evaluation<sup>2</sup>. Note that incompleteness is a special form of uncertainty. While the radar localization of an aircraft might be uncertain, there are rigorous ways to reason about and quantify the resulting variance. Incompleteness on the other hand introduces an unquantified, maybe even unquantifiable, bias. So while the effect of a small data set or of only a limited number of sensors results in a quantified bias, the effect of selecting a certain model based on domain knowledge is generally not quantifiable. For example, in complex systems, the complete system is only rarely testable; creating a complete list of all scenarios in which the system may fail is at best a daunting task. The enumeration of possible outputs given all possible inputs might be even computationally intractable. Furthermore, it is not certain that there is enough information available to identify all undesirable outputs. It might therefore not be possible to describe the problem completely. Multi-objective trade-offs are another example of an incomplete problem description: Two well-defined

---

<sup>2</sup> See 4 for a detailed discussion

desiderata in machine learning systems may compete with each other, such as causality and prediction accuracy. Even if each objective is fully specified, the exact dynamics of the trade-off may not be fully known, and the decision may have to be made on a case-by-case basis. In the presence of incompleteness, interpretability is one way to ensure that effects of gaps in the problem formulation itself become visible.

## Causality

Causality still is a hotly debated topic in statistics and mathematics. In general there are two ways to address it. One is to fully embrace the Bayesian paradigm of correlation and treating causation, in a sense, mathematically as an illusion. In that paradigm, causality is not hard-coded into any formula, most of the time not even as a constraint.

The other way, more in line with how humans tend to organize their knowledge, includes causation. If the (in)famous example of the Deep Neural Network, which classified huskies and wolves with a high degree of accuracy, is considered, it becomes clear that reliance on predictive accuracy without any principled method of ascertaining cause-effect-relationships can be deceptive [382]. In an industrial context, less benign consequences of reliance on a faulty, yet highly accurate predictor are easily imaginable.

While there are some methods of artificial intelligence that try to model causal effects, the majority of supervised learning algorithms are only optimized to work with correlations in data. Nevertheless, they are often used in the hope of finding properties or generating hypotheses about causal properties of the real world. The associations that are embedded by supervised learning algorithms are not guaranteed to reflect causal relationships. Confounding variables can still be "lurking" in the data<sup>3</sup> and have an influence on both associated values. It is hoped that by the interpretation of supervised models falsifiable hypotheses can be generated and tested experimentally. The task of inferring causal relationships from observational and interventionist data has been studied ex-

---

<sup>3</sup> See [383] for a treatment of confounding variables with extra-probabilistic, i.e. causal, methods.

tensively [383]. Generally speaking, an unfortunate characteristic of these methods is their strong reliance on prior knowledge and the dependence of their results on the assumptions about the causal structure of the process generating the data.

### **Transferability**

Typically, models are validated by training them on a randomly selected partition of the data and withholding some data of the same distribution for testing. The gap between the accuracy of a model on the training and on the test data is used to judge a model's capacity for generalization. Humans tend to exhibit a richer capacity for generalization and can often successfully transfer learned skills to different situations.

If a machine learning algorithm is supposed to work in a non-stationary environment, it also has to be able to exhibit this sort of behavior and be able to cope with the change of the environment that the deployment of the model itself causes.<sup>4</sup> In security contexts, the environment might be actively adversarial in trying to supply manipulated data to "fool" the learning algorithm. With small perturbations to original sensory input, like images, adversarial attacks can deceive a target model to produce completely wrong predictions. There have for example been attacks on semi-autonomous cars to take the wrong lane. [385] For credit scores, the variables measured to quantify the risk of default on a loan can be manipulated by the assessed individuals. Periodically requesting increases to credit lines while keeping spending patterns constant for example, will contribute positively on the credit score [379]. This gaming of the rating system may invalidate their predictive power. Interpretability of models and their decisions can thus help to assess the effects of a change of context to the deployed models.

---

<sup>4</sup> [384] details a case, where a model was trained to predict probabilities of death from pneumonia. For patients that also had asthma, the probabilities were lower owing to more focused treatment on such patients in the used data sets. Deployed in an environment where asthma was not considered a reason for more focused treatments, the model would underestimate the risk of death for such patients.

## **Informativeness**

A common way to use machine learning models is to make decisions based on the outputs of a model. In some cases, the supervised model is used to provide additional information to human decision-makers, instead. While the ostensible objective of the machine learning algorithms remains a reduction in predictive error, the real-purpose is to provide useful information. In most cases this is achieved via the model outputs, however, additional information might be conveyed to human decision-makers via some procedure, for example by providing similar cases in support of a diagnostic decision. This setup more closely resembles unsupervised learning. The real goal is the exploration of relevant data [379].

### **5.2.3 How could Interpretability be Measured?**

In a standard machine learning setting, there exist several evaluation metrics, that are used in different contexts, as discussed in Sec. 4.6. Applied work is generally evaluated by showing the performance of the model on the task at hand, while core methods are required to demonstrate their generalizability via evaluation on several synthetic and standard benchmarks. In this section the interpretation evaluation taxonomy according to [380] is explained.

#### **Application-grounded Metrics**

Application-grounded evaluation involves the conduction of human experiments within real application. The idea is that the best way to show that a model works is to evaluate it with respect to the task that a researcher wants to examine. The same rationale is commonly applied in fields such as human-computer interaction and data visualization. The quality of an explanation in this application-focused context is evaluated on the task at hand: How good can errors be identified? Are new insights to be gained?

Examples of experiments include domain experts testing the exact application or domain experts testing a simpler or only partial task to decrease the neces-

sary time for experiments and lower the barriers for participation. How well human-produced explanations assist other humans in completing tasks is an important baseline in both cases.

As with all experiments involving human subjects, care has to be taken to eliminate bias and confounding variables as much as possible.

### **Human-grounded Metrics**

Like application-grounded evaluation, human-grounded evaluation is about conducting human-subject experiments. The difference is that these tests are simplified, but designed to maintain the essence of the target application. The appeal of this form of evaluation is that it is not necessary to directly engage the target community, which might be difficult, but to let lay people complete these tasks. This increases the pool of potential test subjects and can also help in lowering the expenses, because no compensation for highly trained domain experts is necessary. This kind of evaluation is more appropriate, when less specific notions about the quality of an explanation are to be tested, i.e. what kind of explanation is better understandable under severe time constraints. A major challenge in this setup is to evaluate the quality of an explanation without a specific context and purpose of the application.

Ideally the evaluation of the explanations will only depend on the quality of the explanation regardless of the correctness of the associated prediction and the source, i.e. whether it is from the model itself or a post-hoc interpretation. But it is hard to isolate these factors from one another. Potential experiments include, *binary forced choices*, where humans must choose one of two presented explanations for their quality, *forward simulations*, where humans are presented with the task of correctly simulating a model's output, given an explanation and an input or *counterfactual simulations*, where, given an input, an output and an explanation, the user is asked to predict, what changes are necessary to change the method's prediction to a desired output.

## **Functionally-grounded Evaluation**

Functionally-grounded evaluations are supposed to not require any human experiments, instead some more formal definition of interpretability is used as a proxy for the quality of explanations, like the sparseness of inputs. The appeal for this type of experiments obviously stems from the fact that human experiments require considerable amounts of time and money. These tests are most appropriate, once a class of models or regularizers has already been validated by one of the other two evaluation setups or when human testing is not appropriate. A challenge is of course to find suitable proxies. Even once a proxy has been formalized, finding an optimal solution to the problem can also be challenging, as it is likely to be non-convex, non-differentiable and contain certain discrete variables. Given a suitable proxy for interpretability, these can be used to improve the system. It is for example possible to increase the prediction performance of already validated and interpretable model given constraints on the measure of the proxy, or inversely by fixing the performance, but improving the system with respect to a measure on the proxy, for example sparseness.

## **5.3 The PERMEATED-Framework**

To facilitate decision-making in the presence of substantial amounts of uncertainty, the PERMEATED-Framework is introduced. Like some of its predecessors, it is a systematic approach for capturing the available information about a system or subsystem under consideration. But it also takes into account some special needs of the industrial context, which also include small sample sizes, the infeasibility of conducting experiments on considerable numbers of components, subsystems or even a whole machine tool and the need to address human decision-makers with the result of the analyses to create the trust that is conducive to the continued gainful usage of any condition monitoring system.

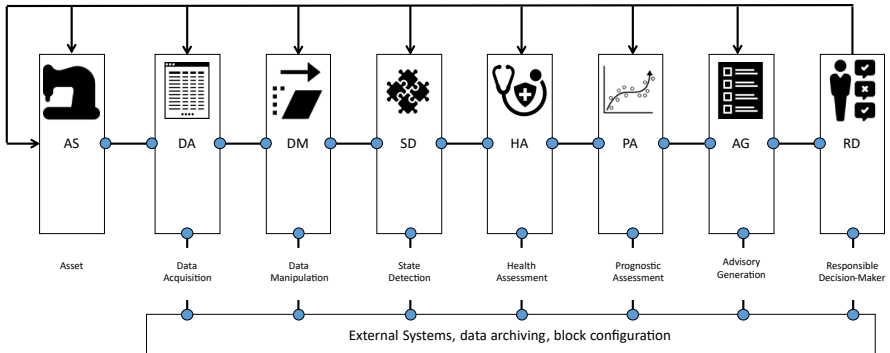


Figure 5.1: The PERMEATED-Framework. An approach for condition monitoring for industrial assets.

Its features are:

- Ability to handle information on different granularity levels.
- Explainability as an essential part of the framework.
- Focusing on feedback from the relevant decision-makers to continually improve the diagnostic capabilities.
- Stressing the generation of trust to aid implementation and continuous usage.
- Ability to facilitate every level of analytical capacity: Case-based reasoning (no generalization at all), fault detection, fault identification, fault prediction.
- Reliance on modules to support extensibility, adaptability and permanent improvements.
- Analyzes causes on the level of granularity they arise at.

The available information about a particular system or even unit under consideration is the determinant of what level of utility a diagnostic system can fulfill. The more information is available, the more useful functions can usually be served. On one end of the spectrum is a purely statistical model that identifies aberration from a specified norm, on the other end are highly complicated

deep neural networks that are used as regressors. For the latter applications, knowledge of the state of the system is not sufficient. Rather knowledge about the dynamics of faults of individual components within a complicated system is needed. Acquiring this knowledge is neither free nor easy. It is probably impossible to prevent every fault, it is very possible, however, to learn from failure. One of the main tasks for each maintenance system is to facilitate learning from errors and experimenting with different solution strategies. PERMEATED is designed to be a learning system. By design, there is no "done date" and no final release. It is adaptable and changes as new algorithms, monitored products, sensors or platforms become available. The PERMEATED-framework is built around empirical verification. It does not claim infallibility, to the contrary, making mistakes is seen as the most valuable opportunity for learning. It is imperative, though, to have the necessary feedback-loops in place, on a technical and organizational level, to learn from mistakes. The framework represents a shift from a "solutions" perspective to a "systems" perspective.

The following sections present the conceptual building blocks of the framework, depicted in Fig. 5.1.

### **5.3.1 Guiding Ideas**

The PERMEATED-Framework has been influenced by many ideas. The most important ones are presented here for reference.

#### **Data Presentation**

Humans are influenced by the way information is presented to them. The chosen representation can therefore not be neutral. Even if the same information content is presented, the presentation itself can skew what recipient of the information perceives as important. Care should be taken to present the information relevant to different user groups appropriately. While the presentation of more information than is appropriate for a certain user group technically sends more information, it is possible that less is actually received and understood. This can in fact reduce the amount of information that is effectively transmitted. If



the information is perceived to be confusing, operators will not use a system to its full capacity.

### **Pragmatism**

Deterministic theories represent the majority of present day scientific theories, with some notable exceptions such as quantum information theory. They relate certain measurements in the physical universe through rules of interpretation to mathematical terms. If the asserted relationships are violated, the model of reality is falsified.

Probabilistic theories cannot be falsified as easily. Only statements of extreme probabilities (i.e., 0 or 1) can act as potential falsifiers. Such potential falsifying events are called Borel criteria [386]. Furthermore, two probabilistic models  $P$  and  $Q$  cannot be distinguished, if they are mutually absolutely continuous, that is, when they assign probability 0 to the same events. These models will pass or fail any Borel falsifiability test together.

In this view, there is and cannot be a "true model" for the physical world. This is akin to Popperian standpoint that the best we can hope for with regard to any scientific theory is not to discover the ultimate truth, but that our currently unfalsified theories are useful for understanding and predicting the world during their limited lifespan [386]. The philosophy of taking usefulness as supreme and only considering real world differences of theories is called pragmatism<sup>5</sup> and adopted as governing principle for condition monitoring systems.

### **Causal Relativism**

The world can be described at various finer or coarser levels of detail. Causality is taken as probabilistic concept describing probabilistic invariances, which are stable over a shifting range of environments. In agreement with [386], causality in degradation processes in this sense can arise during analysis at any such level of detail. It need not be the case that causality at a higher level is an indication of causal relations at a deeper level.

---

<sup>5</sup> Although originally intended as a rationalization for various religious believes. [387]

This means that a reductionist program for condition monitoring, while doubtless of value in most cases, cannot be universally successful: certain aspects of a complex coarse-grained system may be emergent properties and can therefore only be dealt with on the level of analysis where they arise.

This conception of causality is called causal relativism and will be adopted as conception of causality for the PERMEATED framework.

### 5.3.2 Data Acquisition

Data acquisition represents the fundamental capability of the system to access information about the system or the components of the system under consideration. This stage implicitly or explicitly determines the capabilities of the overall system. If a system or fault is neither observable nor detectable, there is no information to be acquired at this stage, and it can thus not enter into a supervision system.<sup>6</sup> A graphical representation of such Data Acquisition stage and its environment can be seen in Fig. 5.2, which follows [369] closely, but closes the feedback loop from the Responsible Decision-Makers.

### 5.3.3 Data Manipulation

This stage takes the raw data of the previous stage and processes it according to the need of the application. There can be multiple data manipulation states in series or parallel for the same set of raw data inputs, there can also be not necessarily unique mixtures of raw data sets for different data manipulation blocks. While there are conceptually no limitations on possible data manipulations algorithms, which could range from Wavelets Package Decomposition to Hilbert-Huang transform, results have to be explainable from the point of view of a Responsible Decision-Maker. More classical data manipulation techniques like a Fourier transform of a raw current signals or various filtering techniques, like band-pass filters, might prove more valuable within this framework.

---

<sup>6</sup> given that the information cannot be reconstructed or extracted from a different sensor and be made accessible to the common data pool

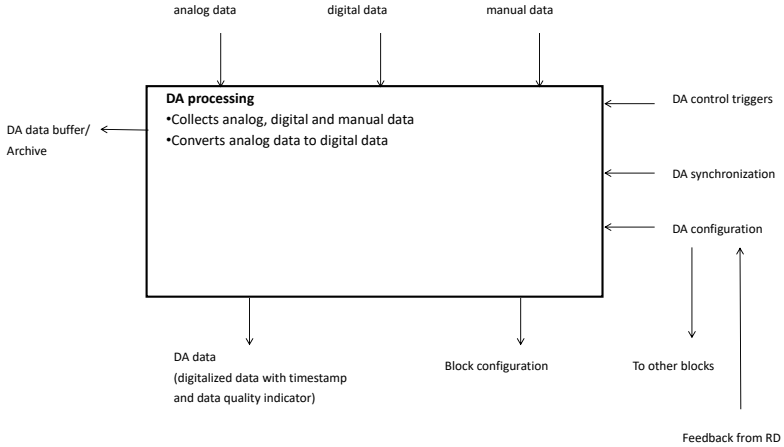


Figure 5.2: Data Acquisition Block. [369], adapted.

A graphical representation of such a Data Manipulation stage and its environment can be seen in Fig. 5.3.

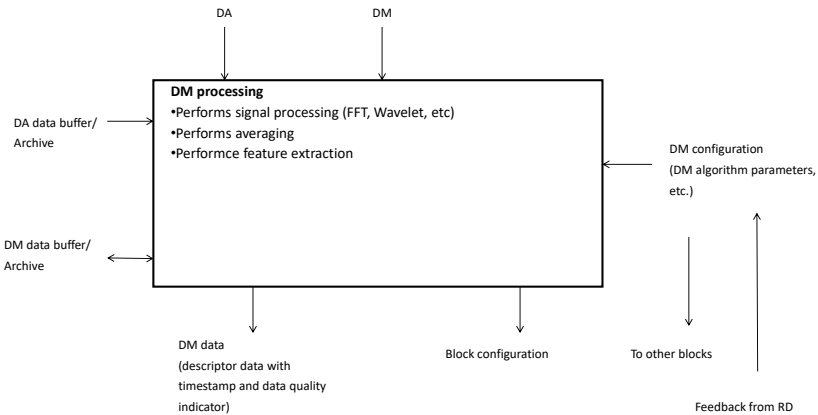


Figure 5.3: Data Manipulation Block. [369], adapted.

### 5.3.4 State Detection

The state of the system under consideration is the collection of at least one, but typically more, outputs of the data manipulation stage. The state is a mathematical representation of certain measured facts of the real system. It is the basis for subsequent processing steps. Components of the system can be represented by their own state. An example output of such a stage could be the eigenfrequency of a mechanical structure, which can be derived from the frequency response measurement of an axis in a machine tool or the friction within a drive train, which can be approximated by the average current need for a non-accelerated movement. A state vector can be composited from the individual states of the components and subsystems, which might have their own unique value ranges, engineering units, boundaries, etc. A graphical representation of a State Detection stage and its environment can be seen in Fig.5.4.

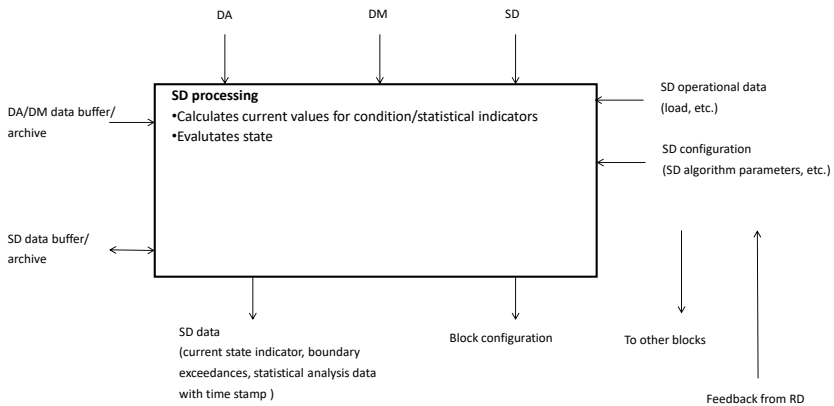


Figure 5.4: State Detection Block. [369], adapted.

### 5.3.5 Health Assessment

The Health Assessment state, represented in Fig. 5.5, is the first stage at which information about the context of the system's physical environment, ensembles and "peers" becomes important. The statistical relevance of detected anomalies

is only meaningful in comparison to a baseline or populations statistics.<sup>7</sup> Knowledge about the uncertainty of certain properties of the unit under consideration has to be injected at this stage. A difference to a defined reference group is only meaningful, if the variance of that property is small enough. This implies that the uncertainty about certain properties of the system and consequently the measured features is a deciding factor for the feasibility of a health assessment.

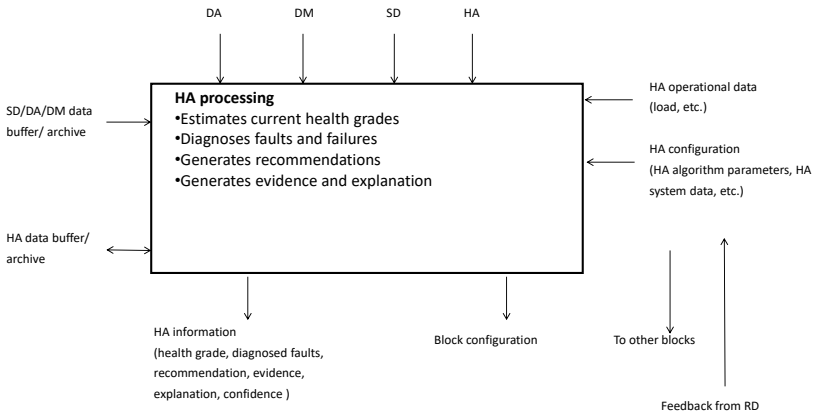


Figure 5.5: Health Assessment Block. [369], adapted.

### 5.3.6 Prognostic Stage

The Prognostic stage uses information about the health condition of the system under consideration, as well as knowledge about the trajectory of comparable systems, to generate projections of the trajectory of the system under consideration with respect to uncertainties in the usage profile and environmental variables. A graphical representation can be seen in Fig. 5.6. Typically, the result of these projections are compressed by using the RUL metric. To achieve this, this stage has to fulfill the tasks described in Sec. 4.5.1. Alternative usage strategies, that can have a huge impact on the RUL can be evaluated and the most useful strategy by some measure, e.g. revenue, can be selected. As was

<sup>7</sup> Except for cases in which certain measurable properties exceed some well understood and carefully implemented threshold.

discussed in the Chapter 4, different types of uncertainty are acting simultaneously on this estimate, which makes the creation of meaningful estimates challenging.

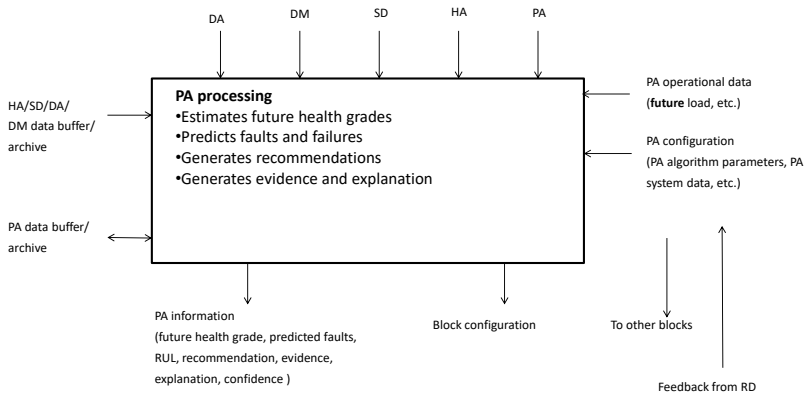


Figure 5.6: Prognostic Assessment Block. [369], adapted.

### 5.3.7 Advisory Generation

The Advisory Generation stage, depicted in Fig.5.7, is used to produce a suggestion for the course of action for the responsible decision-maker. The suggestions have to be specific, actionable, measurable and falsifiable. The rationale for the suggestion *should* be provided transparently. Obviously, the available information of the interaction of the components, the domain of the responsible decision-maker and the task at hand determine the type and usability of the generated advises, which can range from rather mundane ones, like "Don't do anything to machine Q435Z786" to strategies for the operation of a whole fleet of manufacturing sites.

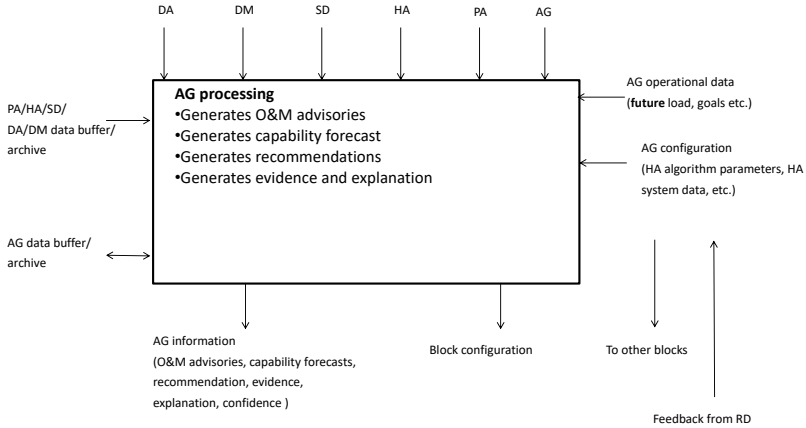


Figure 5.7: Advisory Generation Block. [369], adapted.

### 5.3.8 Responsible Decision-Maker

The Responsible Decision-Maker is the addressee of the analysis and generated advice. She has to make the final decision regarding any maintenance action, whether it is the shutdown of a system for safety reasons, the interpretation of the presented information of the Health Assessment or Prognostic Assessment stages and, naturally, the implementation of any of the generated advices. The Responsible Decision-Maker is the consumer and often customer of the advisory system. In accordance with the ideas regarding a third way of model calibration presented in Sec. 4.7.4, the Responsible Decision-Maker is relevant to the probabilities, because her judgement on the importance of variables and subsequent alterations to the models also determine the generated Health and Prognostic Assessments. She is the sole revealer of the true usability of the system. If she does not use or does not trust the system, the implementation effort has not succeeded. For a condition monitoring system, the deployment of a system is not the final stage of the development, the continued profitable usage is. A graphical representation of a Responsible Decision-Maker stage and its environment can be seen in Fig. 5.8.

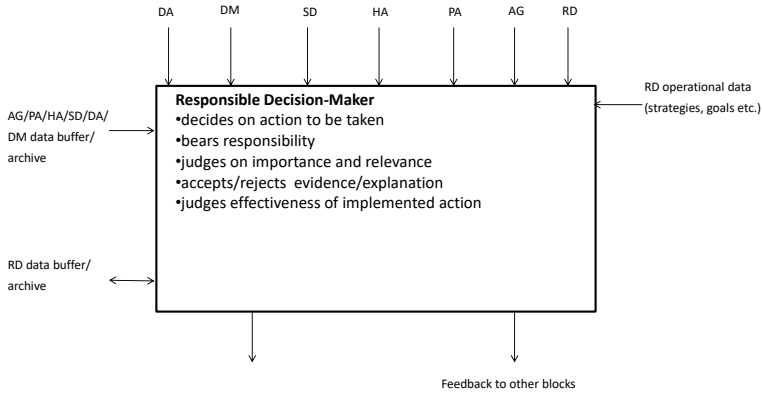


Figure 5.8: Responsible Decision-Maker Block

## 5.4 AxDiag: An Industrial Expert System

To facilitate the monitoring of the drives of machine tools, the PERMEATED-framework was used to create the AxDiag-Software-Tool. It consists of different software components. One component is executed by the control systems of the machine tools and communicates directly with the drives. This component triggers measurements, collects and persists the relevant signals. Another component is executable on normal PCs. This is used by service technicians and developers for manual analyses and for configuration purposes. A last component is hosted by a cloud service and facilitates the automated and scalable analysis of measurements.

The software component on the machine tool triggers measurement cycles that excite the system sufficiently in order to get usable representations of the dynamic behavior of the system<sup>8</sup>. The software is configurable and extensible. It allows for the inclusion of various measurement functions and ensures reproducible measurement conditions. The component makes sure that the parameters of the control cascades of the drives and their respective current set point filters are in a state that allows for the observability of interesting proper-

<sup>8</sup> See Sec. 3.1.2 for a discussion on the necessary levels of excitation



ties. For example, if the experimenter is interested in mechanical properties of the machine, it is often advisable to reduce the gain of the speed-control loop, but care has to be taken to keep the control cascade stable. The AxDiag utilizes mainly frequency response function measurements and measurements with consistent offset speed. The configuration allows the experimenter to position the axes relative to one another, which is especially interesting, if their cinematic chains are linked. The dynamic properties of several 5-axis machines for example are highly dependent upon their relative position, mainly due to an increase in lever length.

The manufacturers of machine tools do typically have a lot of specific domain knowledge about their assets. A lot of this knowledge with relevance for condition monitoring applications exits in the form of mechanical models. A very simplified, explanatory model of the mechanics of the machine tool under consideration is depicted in Fig. 5.9. It can be seen, that a 2D laser cutting machine can be represented by a mass-spring-damper model, where the major components of the machine, like its crossbeam, are flexibly coupled to one another.

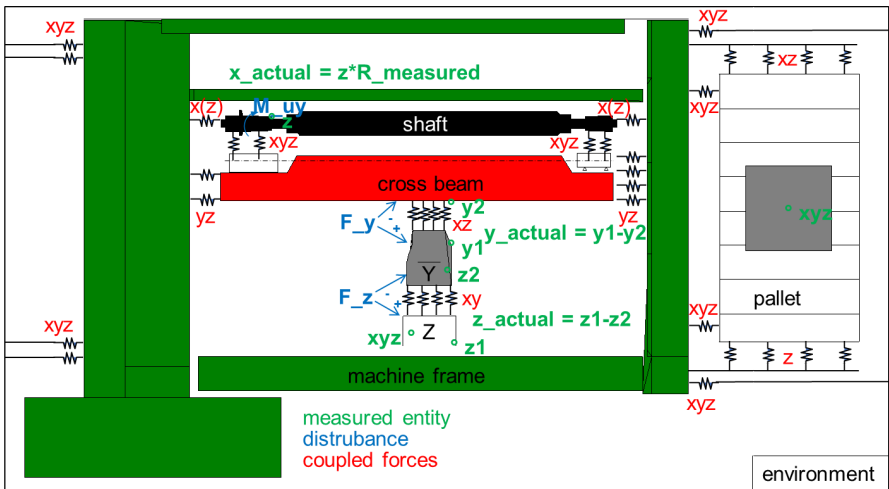


Figure 5.9: A simplified mechanical model of a machine tool.

A special simulation framework, which has been introduced in [388], has then been used to build mechanical models of various components. The framework allows for the parameterization of coupling conditions, the integration of controllers and is used to create dynamical models of the machine, which allow simulations in the time and frequency domain. These models can be used to investigate the effect of changes in mass caused by the installation of various machine options, for example of an advanced laser cutting unit, or they can be used to investigate the trade-off of using weight reduction techniques with associated loss of stiffness on the dynamic properties of the controlled drives.

The process of generating simulations models is depicted in Fig. 5.10. It starts from a CAD drawing of the relevant components, which are used for prototyping, production and eventually service purposes. These drawings are the basis for finite element models, which are primarily used during the development process of a machine to ensure that all stresses in the components are well within safety and functional bounds. In most cases, the granularity of these models is too fine to use them directly in simulations. For this reason, model reduction techniques, like the Krylov subspace method, are used to decrease the

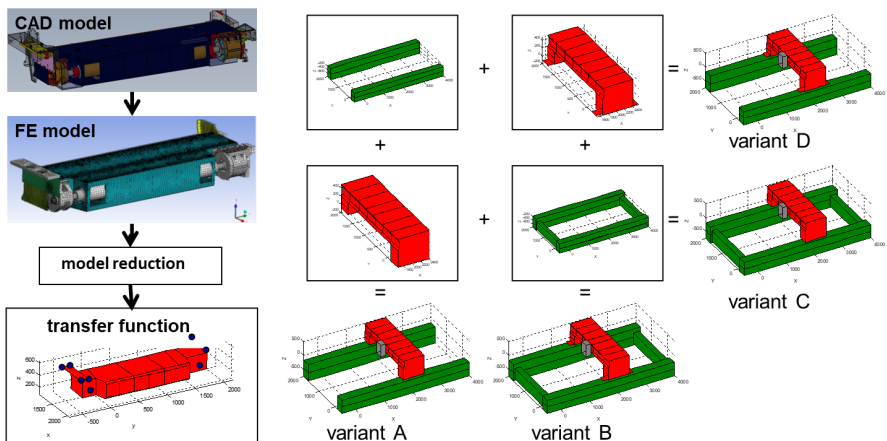


Figure 5.10: Depiction of the model generation process.

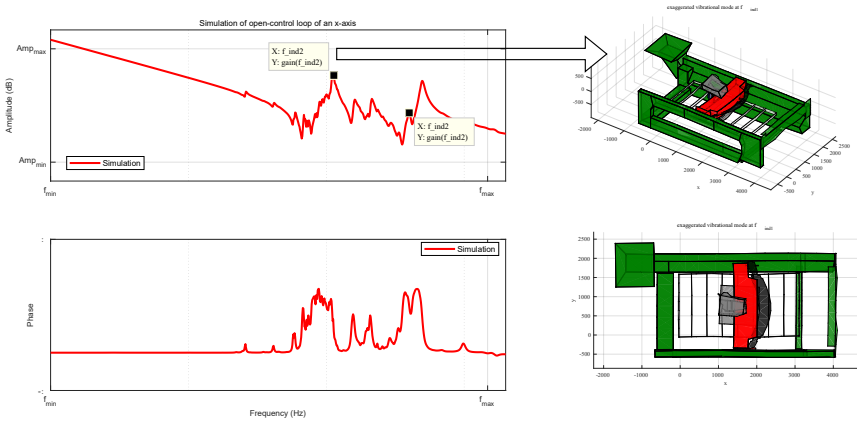


Figure 5.11: Visualization of the correspondence of a peak in the frequency response function to the first bending eigenmode of the driveshaft.

model dimension sufficiently. The reduced models can then be used to quickly build variants of machines.

Given these models, it is possible to investigate the behavior of the mechanical components of the machine at different frequencies. For example, Fig. 5.11 depicts a machine type, where a peak in the frequency response function is closely associated with the first bending eigenmode of the drive shaft, while Fig. 5.12 depicts the same machine type at a frequency that is closely associated with the second bending eigenmode.

With this simulation framework and sufficiently validated models, various kinds of faults can also be investigated. These models can for example be used to trace the effect of rack-pinion backlash by changing the coupling of the drive shaft to the machine frame. The effects of these change in parameters on the behavior in the time and frequency domain can provide valuable insights for the engineering of features for condition monitoring applications. For example, the vibrational energy of a drive, as defined in [97], could be confirmed to be as a measure of health for certain components. A common problem during the initial design and operation phase of the AxDiag condition monitoring system has been the derivation of limits and boundaries indicating the presence of a

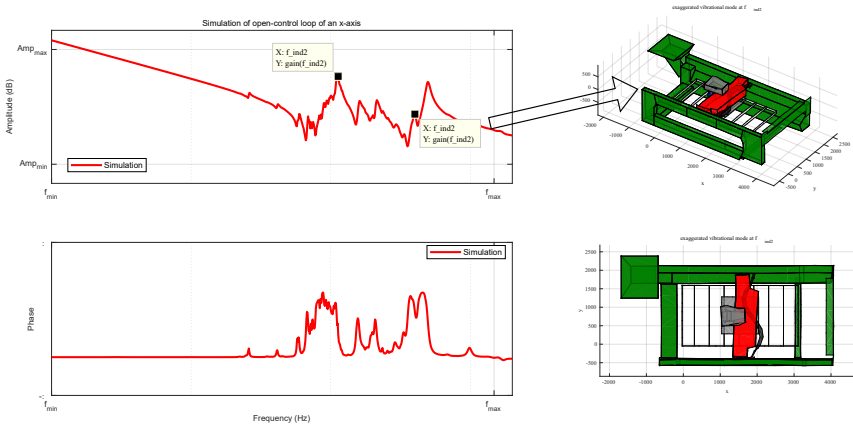


Figure 5.12: Visualization of the correspondence of a peak in the frequency response function to the second bending eigenmode of the drive-shaft.

fault mode, as knowledge about particular fault modes can be sparse and hard to model. For this reason, fuzzy sets, which have been described in Sec. 4.4, were used to express the relative degree of certainty of domain experts regarding the presence of a fault in a system under consideration. With the help of appropriately selected and parameterized fuzzy membership functions, knowledge about different kinds of various fault modes can be encoded and operationalized to the different types of measurements in the AxDiag-framework.

For example measurements with consistent offset speed make it possible to determine the effect of problems with specific components in the drive train, like the effect of a linear bearing on the spectrum of the torque-generating current, as shown in Fig. 5.13, where it can be seen that a defective component introduces an excitation at a frequency that is offset speed and component dependent. These excitations can lead to harmful vibrations, noise and a decrease in the quality of produced parts. Because not all statistically significant aberrations in the spectrum indicate a real fault in the machine which might influence its performance, fuzzy membership functions were used to encode and operationalize the tentative knowledge of the domain experts.

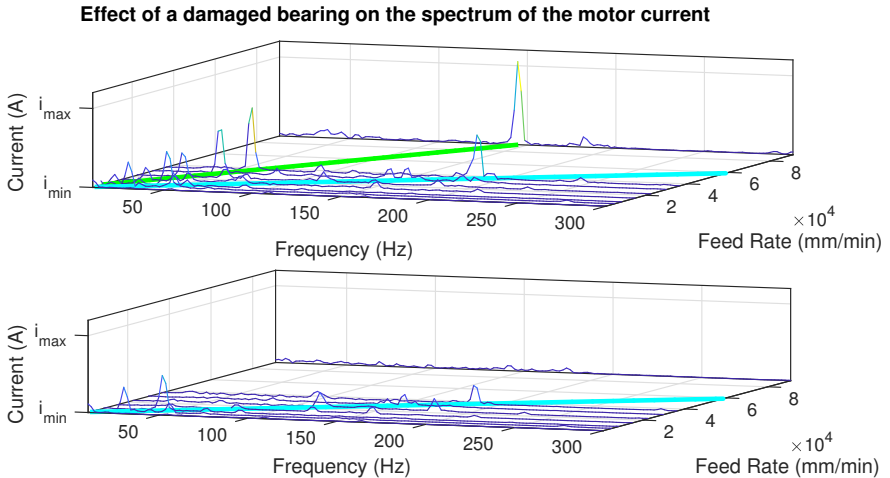


Figure 5.13: Effect of a damaged linear bearing on the spectrum of the motor current. The damaged component is depicted in the upper figure, while the lower one shows the same machine after a repair.

Frequency response function measurements can be used to gain a broad range of information about the system, for example the influence of a loss of stiffness of a motor bearing on the closed-loop frequency response function, as depicted in Fig. 5.14. During these measurements, the drives are fed a reference signal, which is created by adding a pseudorandom binary sequence to a static offset signal, and measuring the resulting controlled variable. This can in principle be done on the current, speed or position control loop, which are cascaded to create the high precision and positioning speed that is needed for modern machine tools. It can clearly be seen that the loss of stiffness changes the underlying plant of the cascade controller so significantly, that the preset controller parameters become far too aggressive. Excitation of the system in this frequency range by a disturbance or even regular inputs can lead to geometrical errors of produced parts during the cutting process. The effect of a collision protection system that has been used beyond its intended life-time stress threshold can be seen in Fig. 5.15. It can clearly be seen that the wear reduces the dampening effect of the coupling between the cutting head and

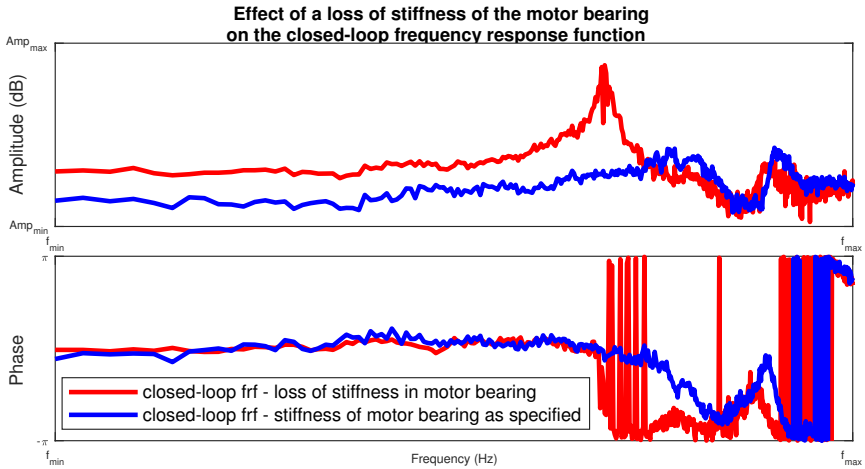


Figure 5.14: Effect of lost stiffness in the motor bearing on the closed-loop frequency response function.

its support structure. This weakening of the dampening coefficient introduces a gain in the associated frequency range of the closed control loop, which the controller is not necessarily designed for. Excitation of the system in this frequency range by a disturbance or ill-chosen input can lead to geometrical errors of produced parts during the cutting process. Again, in the previous two examples fuzzy membership functions proved essential for encoding the domain knowledge gained by simulations, experiments and service missions about the identified indicators into actionable threshold functions.

### 5.4.1 Use Case: Vibration of Cascade-controlled Axis

The software component that is installed on the machine tools has been deployed to more than 10000 machines so far. After the introduction of the PERMEATED-framework, machine tools undergo a mandatory quality control process at the end of the assembly line and are subjected to the measurement function described in Sec. 5.4. Some of these machines are configured to regularly collect and send measurements of the drives for an ongoing monitoring. The condition of the machine tools has been rated by quality

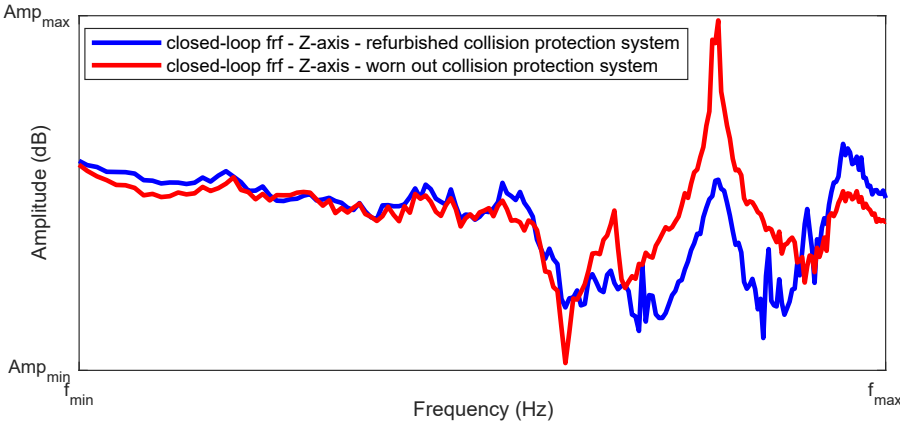


Figure 5.15: Effect of a worn out collision protection system on the closed-loop frequency response function.

control engineers and the results have been recorded. For a certain version of machine tools, an issue with the assembly of the pneumatic couplers between a support bracket of an axis and a specific type of laser cutting head has been identified. There are more than 1,250 individual machines of that type. This issue can prevent the cutting head from locking properly to its socket, which can lead to vibrations severe enough to be audible to even untrained operators. Initial, small disturbances of the measured velocity get amplified into a self-sustaining vibration of the drive. The vibration can also negatively impact the cutting quality. The root-cause analysis of this phenomenon had been significantly complicated by the presence of different types of uncertainties at different stages of the diagnosis process, but can now be reliably identified by using specifically designed fuzzy membership functions. The next subsections will present examples of the effect of uncertainties at the different stages along the process.

### 5.4.2 AxDiag: Assets

At the level of the asset, different environmental confounders can alter the measured characteristics of the system. In Fig. 5.16 the effect of temperature can be

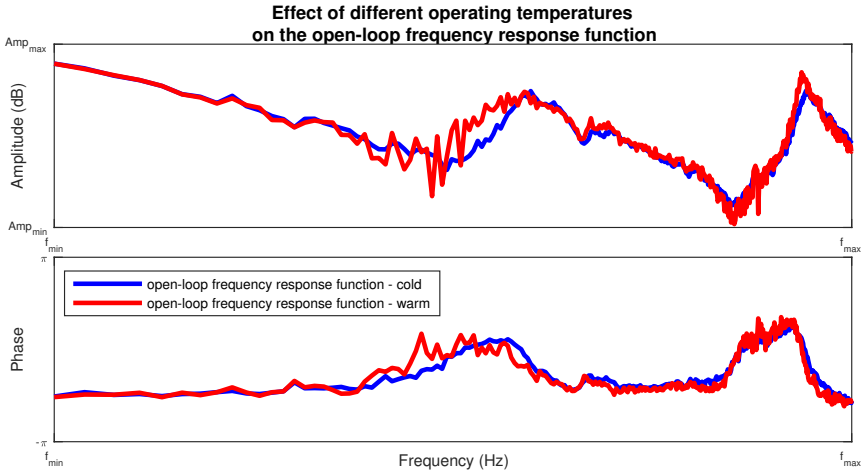


Figure 5.16: Influence of temperature on the open-loop frequency response function.

seen on the open-loop frequency response function of an axis with rack-pinion power-transmission system. If the temperature of the drive is not known, the measurements have to be regarded as uncertain. In Fig. 5.17 the effect of fastening the machine tool by applying less than the specified torque on the screws connecting the machine to the foundation on the open-loop frequency response function is depicted. The effect closely resembles the effect of a wider than specified rack-pinion backlash, which can be seen in Fig. 5.18. There are no sensors within the machine to check the apparent stiffness of the connection between machine and foundation. Without knowledge about the actual conditions at the installation site, it is therefore not easy to differentiate between these two competing hypotheses for explaining the observed phenomenon. A last example of uncertainties that arise about the very context of the measurement and the state of the asset at the physical level is depicted in Fig. 5.19. Here, the load of a linear-drive axis is subject to changes of control parameters of the control cascade in anticipation of a change in the moved mass. This change in mass is typically caused by the installation of an additional component to the machine tool to broaden the spectrum of material shapes that can be processed. For cer-



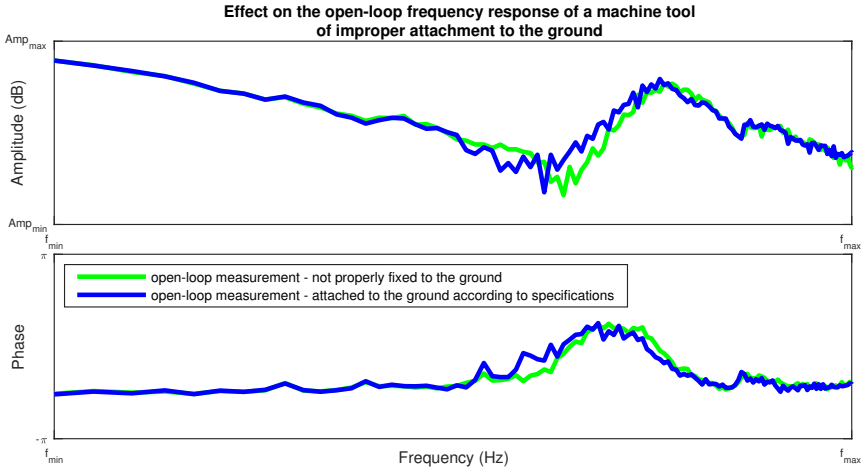


Figure 5.17: Effect of a worse than specified attachment to the machine tool to the ground on the open-loop frequency response function.

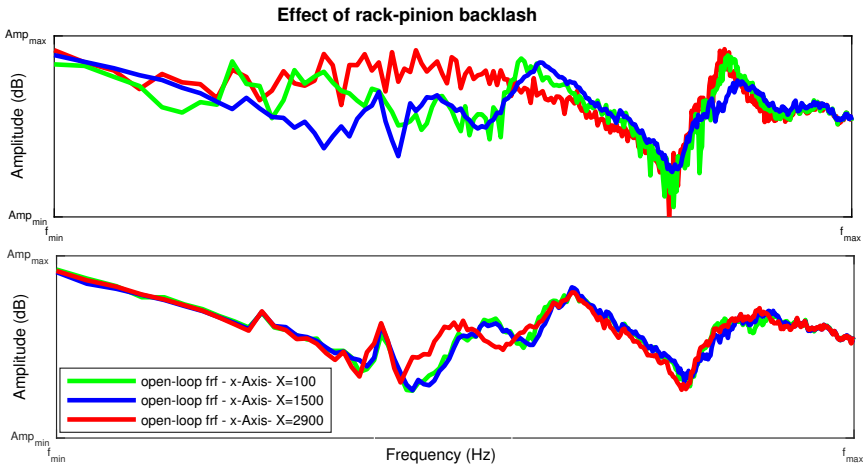


Figure 5.18: Comparison of the influence of greater than specified rack-pinion backlash (above) to a well-adjusted backlash (below) on the open-loop frequency response function.

tain versions of machine tools, this change in the configuration is not sent as a machine-readable message to the control of the machine. Instead, this feature has to be activated and deactivated manually. If this is done incorrectly, the subsequent analyses might mistake the effects of changing control parameters for a change in the electromechanical components.

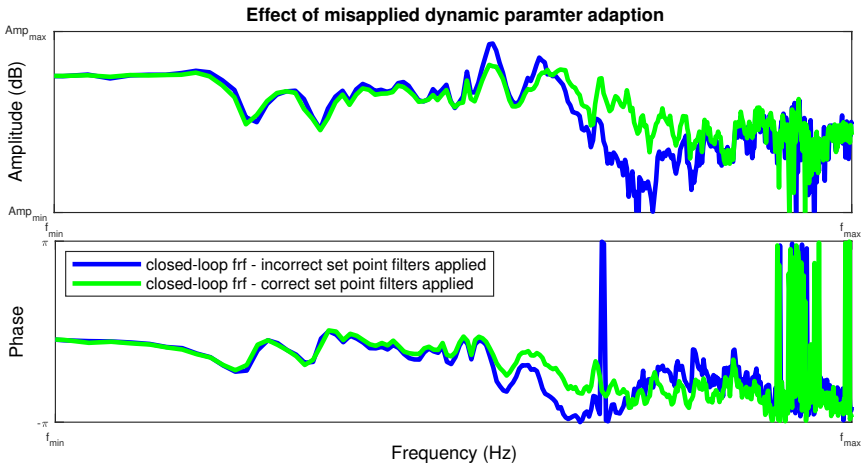


Figure 5.19: Effect of misapplied dynamic parameters on the closed-loop frequency response function.

### 5.4.3 AxDiag: Data Acquisition

During the data acquisition process, the automatic frequency response measurement function component of the AxDiag-software records the time samples for a specified number of cycles and averages the measured signal over the cycles. This signal is then exported for further analysis. Measurement noise is commonly present in measurement systems used in industrial applications and therefore a source of uncertainty. In this specific application, it is known that uncertainty is present in the data acquisition process, but information about the uncertainty is lost by the built-in data aggregation step performed by the numerical control. It is therefore not even available for propagation to downstream modules.

#### **5.4.4 AxDiag: Data Manipulation**

The Welch estimation method for frequency spectra uses overlapping time segments of the recorded signal. Typically, the longer a time series of a system under consideration with sustained and sufficient excitation, the more likely it is for measurement noise to be averaged out. [389, p. 211] This makes the application of overlapping time segments interesting for obtaining a faithful, unbiased estimate of the transfer function. Due to the specifics of the data acquisition process in this application, there are only averaged time-series available. This introduces an algorithmic uncertainty into the estimate of the transfer function estimate, which cannot be specified easily and is by itself unsuitable for uncertainty propagation. Additionally, window functions are used to convolve the time-series with. This is a standard procedure to mitigate undesirable effects in the spectral domain, i.e. leakages of spectral amplitudes to neighboring frequency bins. This introduces a trade-off between resolution in frequency and resolution in amplitude, which leads to additional bias errors. [389, pp. 212-217]

#### **5.4.5 AxDiag: State Detection**

Automatic extraction of indicators can also be complicated by the influence of the specifics of the Data Manipulation stage. If a spike that is an artifact of measurement noise and a subsequent Fourier transformations happens to occur close to a "real" resonance peak in a transfer function, it could be selected by an automatic indicator extraction system instead of the real, meaningful value. So the indicators themselves are uncertain to some degree. The likelihood of such a random spike appearing near a real resonance peak is hard to specify and therefore hard to propagate to subsequent analysis modules.

#### **5.4.6 AxDiag: Health Assessment**

In the health assessment stage, information about ensembles of systems are typically used in determining the health status of a system. The samples selected

to create estimates of the relevant statistics, for example mean values and the variance of the state indicators of the produced systems can be biased.

If it is not clear at the time of creation of the aggregates, whether all machines used in their calculation are indeed members of the "normal" group, the statistics will not reflect "normalcy" accurately. Labeling of the initial samples has to be done by domain experts, which themselves might have epistemic uncertainty. It seems like feedback about the usefulness of the assessments from a Responsible Decision-Maker is needed to gradually learn about the systems, i.e. to improve on the concept of what is normal behavior for a complex system, given the variation in their many constituting components, and what is indeed an aberration. Because of the difficulty to quantify the effect of a lack of knowledge and even problems of defining the crisp boundaries of an event of interest, it is hard to propagate the uncertainty of this stage to subsequent analysis modules.

#### **5.4.7 AxDiag: Prognostic Assessment**

Estimating trajectories of state indicators is highly uncertain, because machine tools are used to produce a variety of products in ways that are rather specific to individual owners and operators of these systems. It is hard to generate a representative load or usage profile that can be used in the estimation of the Remaining Useful Life of an individual system. As was outlined in Sec. 4.5, given stochasticity in failure modes or state indicators, only the estimates about the aggregate, given an assumed usage profile can be given. Individual realization might therefore vary greatly.

It is in general not possible to induce future usage patterns from past usages of systems. The uncertainty introduced by this is, if at all, hard to quantify and propagate.

### 5.4.8 AxDiag: Advisory Generation

During the Advisory Generation stage, given the information of the Health Assessment or Prognostic Assessment stages, an actionable recommendation has to be generated. These must represent a prioritization of what action is most likely to achieve a given goal, at least implicitly. These possible actions have to be selected from a previously defined set of actions. The implicit partitioning of the state space, which is the pre-image of the mapping of the analysis process from indicators to recommendations, introduces a compression problem. It will either be rather crisp around observed instances, thus leaving the analysis more prone to not attributing a state to a specific class, despite it belonging to that class. Or it could also be loose, thus making the analysis more prone to attribute a state to a specific class, although it does not belong to it. Yet unknown fault modes might generate measurable changes in state indicators that are close to the pattern of indicators of an already known fault mode. Uncertainty in the resulting advice is introduced by enforcing an assignment of an observed pattern to a given recommended action, i.e., by enforcing a lossy compression of the information available in the state vector on a predefined set of recommendations.

### 5.4.9 AxDiag: Responsible Decision-Maker

As already outlined in Sec. 4.7.4, the human component at this stage is relevant in the statistical sense to the generated probabilities, because the feedback of the Responsible Decision-Maker is used to modify the generators of subjective probabilities during the analysis stages. What is considered to be important by the respective Responsible Decision-Makers is an extra-statistical question. For example, if the Responsible Decision-Maker is demanding that the underlying system has to be described as linear, time-invariant system to be able to interpret indicators, this will constrain the set of possible indicators of the system and not allow for the handling of more than "mild" non-linearity. On the other hand this justifies the usage of certain tools of linear system analysis, that will be

familiar to more users and therefore probability that the system will be used, if the performance is in an "acceptable" range. The bias that is introduced by the incorporation of feedback from this module is hard to quantify.

## 5.5 Comparison to an Industrial Best Practice

The AxDiag instantiation of the PERMEATED-framework is an expert system and uses expert judgment to design thresholds on well understood and specifically crafted indicators. The problem of automatically setting thresholds has been investigated in [390], where the authors found that setting suitable threshold levels has been a dilemma for engineers in a number of science and industrial fields, because the derivation of such thresholds poses a significant, yet frequently underestimated problem. The number of necessary thresholds can grow very quickly, because there might be several dozen, sometimes even more than 100 individual indicators per drive of a machine series. Additionally, there are different combinations of options can change the characteristics of a machine tool. For example, the installation of a smart cutting unit with several sensors for better process control adds mass that has to be moved, or some machine tools, which are primarily used for cut metal sheets with lasers, can also be equipped to handle the cutting of tubes. This comes at the expense of an elongated z-axis and an altered machine frame. Any combination of these features might warrant the creation of a new baseline to compare the machines against. Additionally, it is customary to define warning and error thresholds, which might also be customized for different user groups: for example, a production line might have a lower tolerance for deviations on newly produced machines than a service organization, which is dealing with aging machines. This illustrates that the cost of manual threshold setting can easily become prohibitive.

But monitoring systems are easily capable of generating thousands of false alarms, when they are run on default threshold levels. Unfortunately, the cost of reacting to a false alarm can be high, given that such a reaction in a condition monitoring setting is most likely associated with a service mission and possibly

even with the replacement of fully functional components. But setting thresholds too low will decrease the value of a monitoring system significantly. The assessment of [390] is, that despite this problem's relevance, there is a relative scarcity of published research on this question.

The authors propose a method for any non-negative data, like vibration signal measures and for symmetrical process values. The approach takes the difference of probability distributions between these types of signals into consideration, but will derive thresholds automatically. Their approach is developed with the monitoring of wind turbines in mind, which are often working under non-stationary conditions. To circumvent the additional complexity of normalizing non-stationary data, they define a set of operational states of interest, identify timespans that correspond to these states and drop data from transitional periods. Similarly, the AxDiag measurements are designed to mitigate the handling of non-stationary measurements by specifying the measurement setup, like the positions of axes, offset speeds and excitation signals directly.

The next subsections will explain the automatic thresholding procedure and compare it against the PERMEATED-framework.

### 5.5.1 Threshold setting procedure

The most common approach is to assume a Gaussian distribution and to determine the mean  $\mu$  and the standard deviation  $\sigma$  of a data set. As is well known, the theoretical probability  $P$  that data from the data set will fall within the range  $\pm k\sigma$  is:

$$P(|X - \mu| < k\sigma) = \begin{cases} 0.6827, & k = 1 \\ 0.9545 & k = 2 . \\ 0.9973 & k = 3 \end{cases} \quad (5.1)$$

While this method is simple, the basic assumption does not always apply.

[390] specifies the following steps to set thresholds: definition of operational states, outlier removal, setting a minimum amplitude threshold, fitting of a dis-

tribution model, setting of a reference value, and automatic threshold derivation. The authors note that some randomness in the state-definition seems inevitable and that an explicit definition for outliers has been elusive despite numerous attempts, which renders this task subjective to some degree. The minimum amplitude threshold tries to remove data below a noise threshold. After these preparation steps, the method tries to fit data to the distributions, which requires relatively large data sets to converge plausibly. They recommend the generalized extreme value probability distribution as the most suitable function after testing different well-known distributions, like the Weibull or Poisson distribution. A random variable  $x$  is said to have a generalized extreme value probability distribution with location parameter  $a$ , scale parameter  $b$ , and shape parameter  $k \neq 0$ , if its density function is given by

$$f_{a,b,k}(x) = b^{-1} \exp \left( - \left( 1 + k \frac{x-a}{b} \right)^{-\frac{1}{k}} \right) \cdot \left( 1 + k \cdot \frac{x-a}{b} \right)^{-1-\frac{1}{k}}. \quad (5.2)$$

After fitting the distribution, the paper advises to generate a reference value as the 96-98 percentile of the associated cumulative distribution function and defines a warning threshold as 3dB and an error threshold as 6dB measured from that reference.

## 5.5.2 Results

Using this method, it is in deed possible to automatically create thresholds. The fitted generalized extreme value probability distribution can be seen in Fig. 5.20 The density functions are plotted so that they contain 99 percent of the probability mass. As can be seen, for an indicator with a suitable probability distribution, the proposed technique can be produce sensible error bounds.

But that only holds true, if the indicators are "well-behaved". For the machine tools under consideration this condition does not hold. The automatically generated thresholds do not show acceptable performance characteristics. As can be seen in Fig. 5.21, the automatic thresholds applied to the use case described in Sec. 5.4.1 generates far more false alarms compared to the fuzzy membership



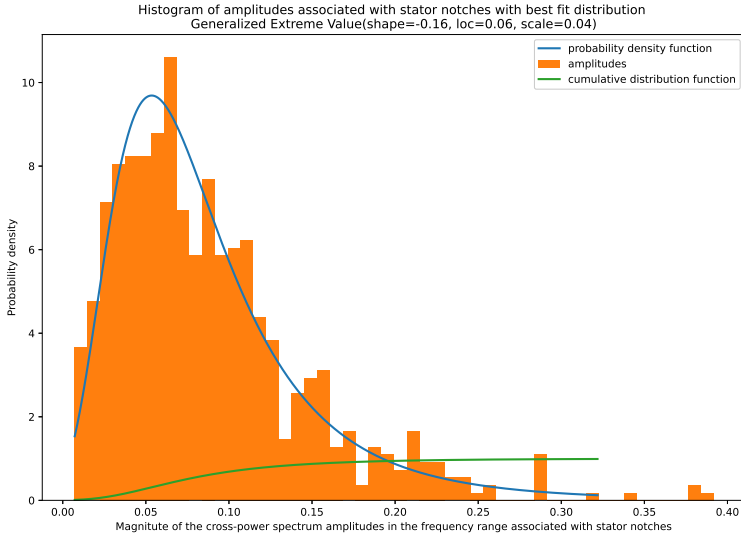


Figure 5.20: Fitted generalized extreme value distribution for amplitudes of the cross-power spectrum in a frequency range associated with stator notches.

function approach. By the judgment of the Responsible Decision-Makers, too many to be useful.

The application of the method demonstrates certain common problems:

- The removal of outliers still requires a highly technical degree of input from the domain experts.
- The selection of outliers introduces a degree of subjectivity
- The prescribed method cannot account for relevance in the scoring.

The PERMEATED framework, instantiated in the AxDiag expert system on the hand allows domain experts the description of relevance by using fuzzy functions. The explanatory power of a suitably parameterized sigmoidal fuzzy membership function is higher than a mere statistical argument and produces a higher degree of acceptance with Responsible Decision-Makers. Moreover, while both methods introduce a certain degree of subjectivity in the creation of the model, either by selecting the outliers or by selecting the type of the

### Comparison of confusion matrices of automatic thresholds and fuzzy recommendations

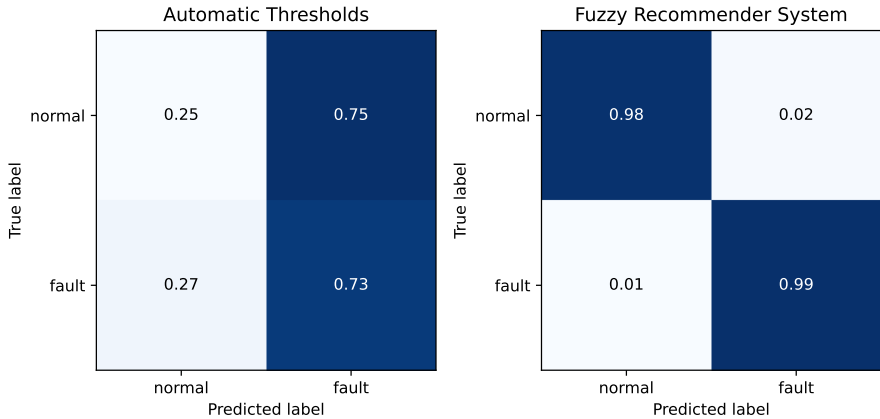


Figure 5.21: Confusion Matrix of an automatic threshold method described in [390] applied to a set of indicators of a y-axis of a machine tool series in comparison to the confusion matrix of the PERMEATED fuzzy recommender system.

membership function, the latter method allows for a rather intuitive way of expressing the "internal model" of domain expert, while the flexibility of the former method is highly constraint. These techniques are not mutually exclusive, though. On the contrary, the PERMEATED framework allows to flexibly choose different methods, as long as they allow for the generation of explanations. While automatic thresholding rules like the one described here can provide a starting point and default values, the usage of fuzzy membership functions allows for a highly customizable specifications of the relevance of indicators.

## 5.6 Discussion

The PERMEATED-framework was successfully instantiated as an industrial expert system that uses interpretable and purpose-built state indicators that are

transparently mapped via fuzzy membership functions to different classes of criticality. The system proved to be applicable to a rather broad spectrum of machine tool technologies: laser cutting machine tools for sheet metal processing, laser welding machine tools for 3D application, as well as for punching machines and metal tube cutting machines. This realization of the PERMEATED framework has already created significant saving through its application as quality control tool in the assembly line and as diagnostic tool for service missions.

While the maintenance of this system from crafting features to optimizing parameters of fuzzy membership functions and thresholds has proved to be quite demanding on the domain experts, the comparison against an automatic thresholding scheme in 5.5 has shown its superior performance. Trust in the generated advices has initially proved to be rather low, but has grown to high levels. This required the distribution of explanation to the Responsible Decision-Makers. To increase the utility of this instantiation of the PERMEATED-framework, while simultaneously easing the burden on the domain experts, the extensibility of the framework will be demonstrated in the subsequent chapter by integrating interpretable machine learning techniques and explanations obtained for opaque machine learning techniques.

---

## 6 Interpretable Machine Learning

Corresponding to the PERMEATED-framework, the simulation framework described in Sec. 5.4 was used to define more than 25 indicators for the linear-direct drive that are used to position the cutting head along the y-axis in a first Data Manipulation stage to tackle the use case described in Sec. 5.4.1. Among these are the eigenmodes of the motion unit's crossbeam, the excitation caused by different components, like magnets of the electric motor, or dynamic characteristics of the closed control loop, like the bandwidth of the controlled system. Subsequently, data sets were constructed and z-score normalized using the statistics of nominal systems in a second Data Manipulation stage. An excerpt of the resulting data for some indicators and machines can be seen in Tab. 6.1.

Additionally, the binary ratings of the quality engineers regarding the presence or absence of this particular fault were acquired by taking  $\alpha$ -cuts over the set of machines that have been deemed as "ok"<sup>1</sup> and added to the corresponding data sets. These data sets were then used to construct classifiers for the system, which are supposed to support the quality control experts in monitoring the products of the assembly line.

In the following sections, various classifiers are introduced that were applied to the constructed data set and assessed with respect to their ability to actually support the Responsible Decision-Maker. The same data set is then used for comparison with a special form of interpretable machine learning algorithm within the PERMEATED-framework.

---

<sup>1</sup> See the discussion in 4.4 for how fuzzy membership function help to concretize this ambiguous notion.

equipment_no	closedloopgain_p750	friction_p750	guide_carrage_1	measurementsystem_1	measurementsystem_2	measurementsystem_3	measurementsystemwinding_1	measurementsystemwinding_2	polepitchlinearmotor1	rollingmoment1	y_measurement1_p750	y_measurement2_p750	y_measurement4_p750	y_measurement6_p750	y_measurement13_p750	y_measurement15_p750	target
dcddf5a0bf93	-0.39	-0.57	-1.11	0.95	0.71	-0.37	-0.32	0.26	-0.73	-0.69	-1.13	0.18	1.98	-0.01	1.68	1.44	True
01d8f2625264d	-0.57	-1.22	-0.71	-0.22	-0.21	0.36	-1.39	-0.87	0.07	0.33	-0.85	-0.33	0.30	-0.75	-0.66	0.03	True
73063b2009442	-0.46	-0.41	-1.03	0.08	-0.20	0.07	-0.81	-0.76	-0.68	-1.16	-0.88	-0.45	0.78	0.47	-0.48	-0.67	True
301311ffa87fa	-0.33	-0.00	-1.28	0.14	-0.33	-0.57	-1.06	-0.62	-0.59	-1.17	1.31	1.07	-0.24	-0.73	-0.31	0.16	True
3ff6478a0e93f	-0.88	-0.95	-1.82	0.58	0.00	-0.54	0.45	-1.04	-0.86	-0.48	-0.85	-0.54	-0.70	-0.37	-0.38	-0.24	True
1a2e51a7838a	-0.51	-0.81	-0.23	-0.11	-0.44	-0.31	0.54	1.43	-0.10	-0.99	1.28	-0.20	0.07	0.66	-0.22	0.22	True
4d07b99dfddb	0.04	-1.06	-0.45	-0.21	-0.44	-0.51	-0.13	0.09	-0.12	0.12	0.91	-1.22	0.06	0.26	0.41	-0.22	True
9c7343e6420	0.23	-1.01	-0.91	1.12	-0.26	-0.24	0.99	0.21	0.16	-0.33	1.12	1.21	0.76	0.93	3.75	1.44	True
d947a2e58167	0.07	-0.34	-0.67	0.37	-0.15	0.29	1.07	1.39	0.75	0.79	-1.03	-0.31	-0.09	0.67	-1.05	-0.09	False
1e0914122dbd	-0.36	-0.14	-0.32	2.39	-0.02	0.28	0.48	-0.45	0.44	-0.54	-1.07	-1.01	-1.14	-1.08	-1.13	-0.77	True
d6b2f5e237f9	-0.39	-1.13	0.02	-0.02	-0.19	-0.23	-1.12	-1.17	-0.24	-0.78	1.06	-0.28	-0.52	0.44	0.33	-0.41	False
757cb088354	-0.36	-0.21	-0.98	0.45	0.21	-0.50	2.21	0.92	-0.74	-0.91	-1.01	-0.07	-0.41	0.71	-1.22	-0.34	False
b261253b5471	-0.53	-0.62	-0.62	-0.89	-0.33	-0.47	0.88	-0.76	0.01	-0.78	-1.09	0.43	0.14	0.63	0.02	-0.59	False
80f7eb7b0896	-0.11	-0.92	-1.04	-0.28	-0.35	0.33	-1.19	-0.67	-0.50	-0.74	0.94	-1.09	-0.46	0.90	-0.18	-0.16	False
ea14d663f616	-0.71	1.62	-0.87	0.89	-0.26	1.20	0.07	-0.95	-0.75	0.33	-0.94	-1.04	-0.64	0.28	0.58	-0.63	True
0e6f73f31af9	10.51	5.58	-1.38	1.84	-0.36	-0.59	4.74	-0.51	-0.90	0.77	1.14	0.40	-0.38	-1.11	0.50	-0.36	True
20ff407dd4a36	-0.72	-0.78	-0.94	1.11	-0.20	-0.21	-1.24	-0.58	0.16	-1.27	-1.22	-1.06	-0.54	-2.19	-0.16	-0.48	True
68d95d1c5a5c	-0.24	1.82	-0.46	0.50	0.33	-0.30	2.98	0.59	-0.10	-1.52	-1.06	0.72	0.91	-0.51	0.81	0.31	False
c2fe6070001	0.21	-0.86	-0.73	0.46	0.13	-0.57	-0.40	-0.10	-1.26	-0.14	-0.86	-0.17	-0.21	0.56	0.46	-0.63	False
3c9c9aa8445c	-0.91	-0.88	-0.61	0.66	0.26	-0.28	0.40	-0.10	-1.26	-0.14	-0.86	-0.17	-0.21	0.56	0.46	-0.63	False
cl0790871f3c	-0.64	-0.57	-0.03	1.11	0.25	-0.33	-1.65	-0.65	-0.43	-0.82	0.94	-1.77	0.08	-0.10	-0.49	-0.23	False
43617d493f61	0.08	-0.27	-0.79	1.07	-0.09	0.26	-0.13	-0.36	0.16	-0.68	-1.09	0.82	-0.87	-0.85	0.73	-0.22	True
5f3eae01603a	-0.45	-0.82	-1.22	0.63	-0.18	-0.63	-0.98	-0.61	-0.87	0.40	0.97	1.37	-0.41	-0.63	1.06	-0.44	True
2891f06afc5a	0.88	-0.38	0.94	-0.14	0.01	0.72	2.08	-0.93	-0.56	-0.19	0.93	-1.12	-0.67	0.70	-0.54	-0.22	True

Table 6.1: Excerpt of z-normalized indicators with pseudonymized equipment numbers, rounded to two decimal places

## 6.1 Non-interpretable Machine Learning Algorithms

In this section a selection of machine learning algorithms will be introduced and applied to the specified use case. These algorithms are not typically considered to be interpretable. The main concern of these algorithms is accuracy, not interpretability.

### 6.1.1 Support Vector Machines for classification

Support Vector Machines, first introduced by Vapnik<sup>2</sup>, are a tool deeply rooted in statistical learning theory.

Unlike most classifiers, SVMs do not try to minimize the empirical risk, but the structural risk. They have shown their capabilities for robust classification in a number of different applications. In [127] it has been shown that the formulation of the SVM is indeed equal to a robust optimization problem formulation, which provides an insight into the reason of the robust performance of the SVMs. The SVM is in its original formulation a binary classifier, i.e. only able to separate the members of two classes. Let  $\{\mathbf{x}_i, y_i\}_{i=1}^l$  be a set of observed patterns  $\mathbf{x}_i$  and their corresponding label  $y_i \in \{-1, 1\}$ . Such a set of observations is said to be separable by a hyperplane, if there exists a vector  $\mathbf{w}$  and a scalar  $b$ , such that

$$\begin{aligned}\mathbf{w}^T \mathbf{x}_i + b &\geq 1 & \forall i \in \{i | y_i = 1\}, \\ \mathbf{w}^T \mathbf{x}_i + b &\leq -1 & \forall i \in \{i | y_i = -1\}.\end{aligned}\tag{6.1}$$

The hyperplane  $\mathbf{w}_0^T \mathbf{x} + b_0 = 0$  is said to be the optimal hyperplane separating the patterns of the set of observations, if it separates the data with the maximal margin with respect to the direction  $\mathbf{w} / \|\mathbf{w}\|$ .

The "critical" points, i.e. those which are the closest to the decision boundary,

---

<sup>2</sup> See for example [172]

are those, that will satisfy one of the inequalities with equality. The optimal hyperplane is the unique maximizer of the distance

$$\rho(\mathbf{w}_0, b_0) = \frac{2}{\|\mathbf{w}_0\|} = \frac{2}{\sqrt{\mathbf{w}^T \mathbf{w}}}, \quad (6.2)$$

which is achieved by minimizing  $\mathbf{w} \cdot \mathbf{w}$ . Not all sets of observations are linearly separable, though. If the formulation of separability is modified by the introduction of positive slack variables  $\xi_i$ , the optimal hyperplane becomes that hyperplane, which manages the trade-off between separating the two classes with the greatest margin and allowing the fewest misclassifications the best.

$$\begin{aligned} \mathbf{w}^T \mathbf{x}_i + b &\geq 1 - \xi_i && \forall i \in \{i | y_i = 1\} \\ \mathbf{w}^T \mathbf{x}_i + b &\leq -1 + \xi_i && \forall i \in \{i | y_i = -1\} \\ \xi_i &\geq 0 && \forall i \in \{1, \dots, l\}, \end{aligned}$$

A misclassification of a pattern occurs, when the corresponding  $\xi_i$  has to exceed unity and  $\sum_i \xi_i$  can be seen as an upper bound on the number of errors. The penalty on the error is typically introduced as  $C(\sum_i \xi_i)^k$ , which leads to a convex optimization problem for all positive integers  $k$ , the choice of  $k \in \{1, 2\}$  renders the problem a quadratic optimization problem [185]. The parameter  $C$  has to be chosen by the user with a larger value corresponding to a higher penalty on errors.  $k$  is chosen as equal to one for the remaining derivation. The Lagrangian of the problem is given by

$$L(\mathbf{w}, b, \lambda, \xi, \mu) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \lambda_i [y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^l \mu_i \xi_i. \quad (6.3)$$

The Karush-Kuhn-Tucker conditions are given by

$$\frac{\partial L(\mathbf{w}, b, \lambda, \xi, \mu)}{\partial \mathbf{w}} = 0 = (\mathbf{w}_0 - \sum_{i=1}^l \lambda_i y_i \mathbf{x}_i), \quad (6.4)$$

$$\frac{\partial L(\mathbf{w}, b, \lambda, \xi, \mu)}{\partial b} = 0 = \sum_{i=1}^l \lambda_i y_i, \quad (6.5)$$

$$\frac{\partial L(\mathbf{w}, b, \lambda, \xi, \mu)}{\partial \xi_i} = 0 = C - \lambda_i - \mu_i \quad \forall i \in \{1, \dots, l\} \quad (6.6)$$

$$y_i(\mathbf{w}_0^T \mathbf{x}_i + b) - 1 + \xi_i \geq 0 \quad \forall i \in \{1, \dots, l\} \quad (6.7)$$

$$\xi_i \geq 0 \quad \forall i \in \{1, \dots, l\} \quad (6.8)$$

$$\mu_i \geq 0 \quad \forall i \in \{1, \dots, l\} \quad (6.9)$$

$$\lambda_i \geq 0 \quad \forall i \in \{1, \dots, l\} \quad (6.10)$$

$$\lambda_i [y_i(\mathbf{w}_0^T \mathbf{x}_i + b) - 1 + \xi_i] = 0 \quad \forall i \in \{1, \dots, l\} \quad (6.11)$$

$$\mu_i \xi_i = 0 \quad \forall i \in \{1, \dots, l\} \quad (6.12)$$

Eq. (6.6) shows  $\mu_i$  can be substituted by  $C - \lambda_i$ . Thus, by substituting Eq. (6.4) and Eq. (6.5) back into Eq. (6.3), the dual problem, only dependent on  $\lambda$ , can be formulated as

$$W(\lambda) = \lambda^T \mathbf{1} - \frac{1}{2} \lambda^T \mathbf{M} \lambda, \quad (6.13)$$

where  $\mathbf{1}$  is an  $l$ -dimensional unit vector and  $\mathbf{M}$  is a symmetric  $l$ -by- $l$ -matrix whose elements are  $M_{ij} = y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$ . The solution to the original problem is thus found by maximizing the dual problem Eq. (6.13) under the remaining Karush-Kuhn-Tucker conditions

$$\begin{aligned} \sum_{i=1}^l \lambda_i y_i &= 0, \\ 0 \leq \lambda_i &\leq C \quad \forall i \in \{1, \dots, l\}. \end{aligned}$$

It is noteworthy that the solution of the minimization problem Eq. 6.3 does neither depend upon the slack variables  $\xi_i$  nor their Lagrange multipliers  $\mu_i$ , which is an effect of setting  $k$  equal to one. In this case, only a few vectors



constitute support vectors, the solution to the problem is thus considered to be sparse. To estimate the class affiliation of a yet unseen pattern  $\mathbf{x}_{new}$ , it is sufficient to evaluate on which side of the hyperplane its projection will fall, i.e. the result of

$$f(\mathbf{x}_{new}) = \text{sign}(\mathbf{w}_0^T \mathbf{x}_{new} + b)$$

will give the estimate of the class label of the unseen pattern.

The method just derived only works for a case, where the examples are linearly separable, but the ideas can be generalized to the nonlinear case[185]. The solution to this is astonishingly easy and has been dubbed "kernel trick". First notice, that in the formulations for the optimization problems the observed patterns  $\mathbf{x}_i$  only enter via the inner product  $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ . If the data is first mapped into a possible infinitely dimensional *Hilbert space*, via a possibly nonlinear mapping  $\Phi(\cdot)$ , then the algorithm would of course only be dependent on the inner product of the mappings, i.e.  $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$ . Evaluating this inner product is equivalent to evaluating the kernel of the *reproducing kernel Hilbert space* (RKHS) induced by the mapping  $\Phi(\cdot)$ , i.e.  $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_i, \mathbf{x}_j)$ , where  $k(\cdot, \cdot)$  is the kernel. While computing the nonlinear transformation of  $\mathbf{x}_i$  via  $\Phi(\cdot)$  can be computational very expensive, the evaluation of the kernel can be rather efficient. The optimization problem for the separable and non-separable case can be formulated in the dual space as

$$W(\lambda) = \lambda^T \mathbf{1} - \frac{1}{2} \lambda^T \mathbf{M} \lambda, \quad (6.14)$$

where  $\mathbf{1}$  is an  $l$ -dimensional unit vector and  $\mathbf{M}$  is a symmetric  $l$ -by- $l$ -matrix whose elements are  $M_{ij} = y_i y_j \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$  and the same constraints as above. For estimating the class affiliation of a yet unseen pattern

$\mathbf{x}_{new}$ , it is sufficient to evaluate on which side of the hyperplane in the RHKS its projection will fall, i.e. the result of

$$\begin{aligned}
 f(\mathbf{x}_{new}) &= \text{sign}(\mathbf{w}_0^T \Phi(\mathbf{x}_{new}) + b) \\
 &= \text{sign}\left(\left(\sum_{i=1}^l \lambda_i y_i \Phi(\mathbf{x}_i)\right) \cdot \Phi(\mathbf{x}_{new}) + b\right) \\
 &= \text{sign}\left(\sum_{i=1}^l \lambda_i y_i \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_{new}) \rangle + b\right) \\
 &= \text{sign}\left(\sum_{i=1}^l \lambda_i y_i k(\mathbf{x}_i, \mathbf{x}_{new}) + b\right)
 \end{aligned}$$

will give the estimate of the class label of the unseen pattern. Notice, that it is again not necessary to evaluate the nonlinear mapping, but only the kernel.

## 6.1.2 Gaussian Process

**Definition:** A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution

A Gaussian process (GP) is completely defined by its mean function and covariance function. The mean function  $m(\mathbf{x})$  and the covariance function  $k(\mathbf{x}, \mathbf{x}')$  of a real process  $f(\mathbf{x})$  are defined as

$$\begin{aligned}
 m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\
 k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]
 \end{aligned} \tag{6.15}$$

and a GP will be denoted as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{6.16}$$

Random variables represent the values of the function  $f(\mathbf{x})$  at location  $\mathbf{x}$ . Gaussian Processes are often defined over time, i.e. the index set of the random variables is time. But more generally, the index set  $\mathcal{X}$  is the set of possible inputs, which could be more general, i.e.  $\mathbb{R}^D$ .

A Gaussian process is defined as a collection of random variables and thereby a *consistency* requirement also known as marginalization property is implied. This property simply states that if the GP specifies for example  $(y_1, y_2) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  then it must also specify  $y_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$ , where  $\boldsymbol{\Sigma}_{11}$  is the relevant submatrix of  $\boldsymbol{\Sigma}$ . The examination of larger sets of variables does not change the distribution of smaller sets.

The consistency requirement is automatically fulfilled, if the covariance function specifies entries of a proper covariance matrix. A simple example of a Gaussian process can be obtained from the Bayesian linear regression model  $f(\mathbf{x}) = \Phi(\mathbf{x})^T \mathbf{w}$  with prior  $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_p)$ .

We have for mean and covariance

$$\begin{aligned}\mathbb{E}[f(\mathbf{x})] &= \Phi(\mathbf{x})^T \mathbb{E}[\mathbf{w}] = \mathbf{0}, \\ \mathbb{E}[(f(\mathbf{x}))(f(\mathbf{x}')))] &= \Phi(\mathbf{x})^T \mathbb{E}[\mathbf{w}\mathbf{w}^T] \Phi(\mathbf{x}') = \Phi(\mathbf{x})^T \boldsymbol{\Sigma}_p \Phi(\mathbf{x}').\end{aligned}$$

A derivation of a classification strategy using Gaussian Processes, which involves several approximations, can be found in [269].

### 6.1.3 Logistical Regression

The main concept in a derivation of the logistical regression model are so-called *odds*, the fraction of the probability of a positive event  $P(Y = 1 | X = \mathbf{x}_i) = P(Y_i)$  to the probability of a negative event  $P(Y_i = 0)$ , which is by the law of total probability equal to  $1 - P(Y_i = 1)$ . The logarithm of the odds is the so called *logit* of the probability. In logistic regression, the logit is assumed to be a linear function of the observables  $\mathbf{x}$

$$\text{logit}P(Y_i = 1) = \ln \frac{P(Y_i = 1)}{1 - P(Y_i = 1)} = \theta_0 + \mathbf{x}_i^T \boldsymbol{\theta}. \quad (6.17)$$

This implies that the probability can be expressed as

$$P(Y_i) = \frac{\exp(\theta_0 + \mathbf{x}_i^T \boldsymbol{\theta})}{1 + \exp(\theta_0 + \mathbf{x}_i^T \boldsymbol{\theta})} = \frac{1}{1 + \exp(-(\theta_0 + \mathbf{x}_i^T \boldsymbol{\theta}))} = h_{\theta}(\mathbf{x}_i).$$

Although the observed variable is binary, logistic regression estimates the odds as a continuous function.

To estimate the parameters of the regressors, the existence of a set of  $N$  samples is assumed. If these samples are independently Bernoulli distributed, and an exponentiation trick is used, then their likelihood function is given by

$$L(\boldsymbol{\theta}|\mathbf{x}) = P(Y|X; \boldsymbol{\theta}) = \prod_i P(y_i|\mathbf{x}_i; \boldsymbol{\theta}) = \prod_i h_{\boldsymbol{\theta}}(\mathbf{x}_i)^{y_i} (1 - h_{\boldsymbol{\theta}}(\mathbf{x}_i))^{1-y_i}. \quad (6.18)$$

Typically, the logarithm of the likelihood function is normalized  $N^{-1} \log L(\boldsymbol{\theta}|\mathbf{x}) = N^{-1} \sum_i^N \log P(y_i|\mathbf{x}_i; \boldsymbol{\theta})$ .

It is often beneficial to have a probability associated with the output of a classifier. To get from binary outputs to probabilities, [225] proposed on mostly heuristic grounds to use the logistical regression on the transformed outputs of an SVM without applying the sign-function on the outputs first to create a probabilistic version of the algorithm.

#### 6.1.4 Application within the PERMEATED-framework

To test the selected set of algorithms, a hold-out set was constructed. The remaining data was used to train the different classifiers. To find suitable hyperparameters of the classifiers, a grid search with 10-fold cross-validation was used. The implementation of the Gaussian Processes Classifier automatically tunes its parameters and does not allow for specifically setting parameters for a grid search. To judge the general applicability of algorithms, the Receiver Operator Curves of the algorithms were constructed, which is a robustness-based metric as described in Sec. 4.6. To do this, "degrees of belief" or probabilities have to be constructed from the results of the SVM-type classifiers, for which Platt's algorithm is used. The results can be seen in Fig. 6.1.

An inspection of the plots indicates that in principle all algorithms in the set are suitable for the task, but the Gaussian Process Classifier exhibits the lowest performance for this application, as is indicated by the lowest AUC score and the shape of the curve. The best hyperparameters from the grid search were

Comparison of ROCs of various classifiers

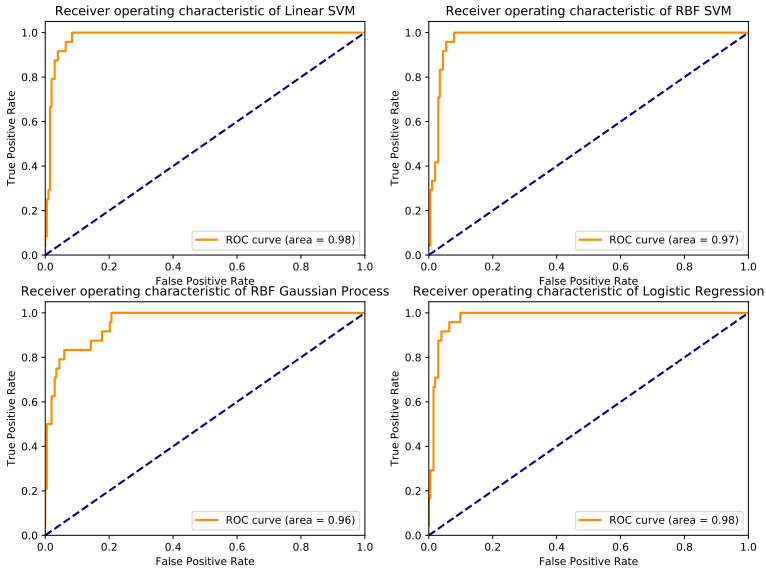


Figure 6.1: Receiver Operating Curves of selected classifiers. Where applicable, the classifiers were trained and tuned using a grid search and 10-fold cross validation. All classifiers are tested on a hold-out set.

selected and their classification results can be seen in the confusion matrices in Fig. 6.2. A natural question is, as described in Sec. 5.1.1, *why* these classifiers give the answers they do. It is not always easy to extract an answer to this question from these classifiers. As an example, the resulting linear SVM is inspectable. There are 114 instances that have been included as supporting vectors of the resulting classifiers and of course there is an equal number of associated non-zero weights. To get an answer to this natural question, even for this rather easy classification task, the trained classifier itself is approaching the limits of how many influencing variables can be handled simultaneously by humans. Projecting a test vector on support vectors and weighing these projections correctly to get closer to the answer is a rather challenging cognitive

task. An algorithm, which is designed to produce simple models, is presented in the next section.

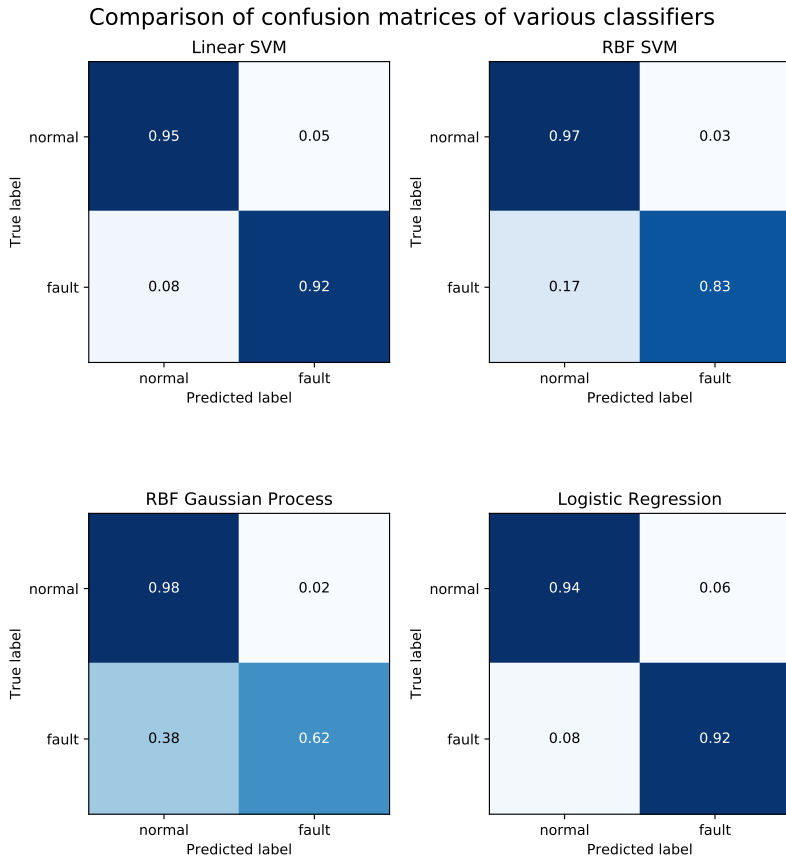


Figure 6.2: Confusion Matrices of certain classifiers. Where applicable, trained and tuned using a grid search and 10-fold cross validation. Tested on a hold-out set.

## 6.2 **Supersparse Linear Integer Models for Optimized Scoring Systems (SLIM)**

Scoring systems are linear classification models that require a user to only manipulate a few small numbers in order to make predictions. Models like that are regularly used in medical settings to assess the risk of numerous serious medical conditions since they allow physicians to make quick predictions, without extensive training and without the need for a computational device, which can be lifesaving. Medical scoring systems that are currently in use, are typically hand-crafted by practitioners, with a panel of experts agreeing on its validity. Scoring systems of the necessary sparsity have been difficult to create using traditional machine learning methods, because they need to be accurate, sparse and using small coprime integer coefficients. It has therefore been a common practice to use well established and tested data-driven techniques like regression analysis and to round-off the acquired regression coefficient in order to generate an easily usable system as an approximation.

The primary usage of such methods in medical contexts poses additional challenges regarding the need to explicitly satisfy constraints on operational quantities such as the false positive rate or the number of features before they can be deployed, which are, if at all, hard to specify in most machine learning methods. Some approaches, like sparse linear classification, such as "Lasso" or "Elastic Net", are able to control the accuracy and sparsity of models via convex surrogate functions, which typically also reduce the computational complexity, but also utilize rounding as an approximation to yield the final models with coprime integer coefficients. Such approximations have a negative influence on the predictive capabilities and make it hard to impose constraints deemed necessary.

In practice, machine learning methods are almost exclusively able to address operational constraints through parameter tuning, which necessitates a high-dimensional grid search process. "Supersparse Linear Integer Models for Optimized Scoring Systems" (SLIM) is a data-driven method for creating scoring

systems and thus interpretable, sparse models. The SLIM algorithm can be regarded as an integer programming problem that optimizes direct measures of accuracy and sparsity, while restricting coefficients to a small set of coprime integers and the possibility of explicitly modeling certain operational constraints. [391]

### 6.2.1 Methodology

Let there be a dataset of  $N$  independent and identically distributed examples  $D_N = \{(x_i, y_i)\}_{i=1}^N$ , where  $x_i \in X$  denotes a vector of features  $[1, x_{i,1}, \dots, x_{i,p}]^T$  and  $y_i \in \{-1, 1\}$  denotes a class label. In the context of SLIM, linear models of the form  $\hat{y} = \text{sign}\lambda^T \mathbf{x}$  are considered, where  $\lambda = [\lambda_0, \lambda_1, \dots, \lambda_p]^T$  represents a vector of coefficients and  $\lambda_0$  represents an intercept term. The coefficients are found by solving an optimization problem of the form:

$$\min_{\lambda} \text{Loss}(\lambda; D_N) + C \cdot \phi(\lambda) \quad \text{s.t.} \quad \lambda \in \mathcal{L}. \quad (6.19)$$

The *loss function* penalizes misclassifications, the *coefficient penalty*  $\phi(\lambda)$  induces soft qualities that are desirable but not mandatory and can be infringed upon for greater accuracy, while the *coefficient set*  $\mathcal{L}$  encodes hard constraints that must be satisfied. The *trade-off parameter*  $C$  controls the balance between accuracy and soft qualities. As a special case of this general optimization problem, SLIM can be formulated as

$$\min_{\lambda} \frac{1}{N} \sum_{i=1}^N \mathbf{1}[y_i \lambda^T \mathbf{x}_i \leq 0] + C_0 |\lambda|_0 + \epsilon |\lambda|_1 \quad \text{s.t.} \quad \lambda \in \mathcal{L}. \quad (6.20)$$

SLIM optimizes accuracy and sparsity by minimizing the 0 – 1 loss  $\frac{1}{N} \sum_{i=1}^N \mathbf{1}[y_i \lambda^T \mathbf{x}_i \leq 0]$  and the  $\ell_0$ -norm  $|\lambda|_0 = \sum_{j=1}^p \mathbf{1}[\lambda_j \neq 0]$ . SLIM also includes an  $\ell_1$ -penalty in the objective for the sole purpose of restricting coefficients to coprime values. This formulation is designed to produce scoring systems that achieve a pareto-optimal trade-off between accuracy and sparsity. By minimizing the 0 – 1 loss, the resulting models are robust to outliers and



attain the best learning theoretic guarantee of predictive accuracy. Similarly, controlling for sparsity by means of  $\ell_0$ -regularization prevents an additional loss in accuracy due to  $\ell_1$ -regularization. Furthermore, minimizing an approximation-free object function over a finite set of discrete coefficients implies that the free parameters in SLIM's object have special properties. The trade-off parameter  $C_0$  represents the maximum accuracy that a SLIM model will sacrifice to remove a feature from the scoring system.

### 6.2.2 SLIM Integer Program

Formulated as an Integer Program [391], SLIM can be stated as

$$\min_{\lambda, \psi, \Phi, \alpha, \beta} \frac{1}{N} \sum_{i=1}^N \psi_i + \sum_{j=1}^P \Phi_j \quad (6.21)$$

$$st. M_i \psi_i \geq \gamma - \sum_{j=0}^P y_i \lambda_j x_{i,j} \quad i = 1, \dots, N \quad (6.22)$$

$$\Phi_j = C_0 \alpha_j + \epsilon \beta_j \quad j = 1, \dots, P \quad (6.23)$$

$$-\Lambda_j \alpha_j \leq \lambda_j \leq \Lambda_j \alpha_j \quad j = 1, \dots, P \quad (6.24)$$

$$-\beta_j \leq \lambda_j \leq \beta_j \quad j = 1, \dots, P \quad (6.25)$$

$$\lambda_j \in \mathcal{L}_j \quad j = 0, \dots, P \quad (6.26)$$

$$\psi_i \in \{0, 1\} \quad i = 1, \dots, N \quad (6.27)$$

$$\Phi_j \in \mathbb{R}_+ \quad j = 1, \dots, P \quad (6.28)$$

$$\alpha_j \in \{0, 1\} \quad j = 1, \dots, P \quad (6.29)$$

$$\beta_j \in \mathbb{R}_+ \quad j = 1, \dots, P \quad (6.30)$$

The formulation can be modified to create certain specialized models. It is for example possible to create  $M$ -of- $N$ -tables.  $M$ -of- $N$  rule tables are simple rule-based models that, given a set of  $N$  rules, predict  $\hat{y} = +1$  if at least  $M$  of the rules evaluate to true. These models have the major benefit that they do not require the user to compute a mathematical expression.

### 6.2.3 Operational Constraints

The SLIM formulation provides a high degree of flexibility over the models allowing the encoding of a wide range of operational constraints into its IP formulation.

#### Loss Constraints for Imbalanced Data

The majority of classification problems in the real world, in a medical or technical environment, are imbalanced. Handling imbalanced data can be difficult for most classification methods since maximizing classification accuracy can result in trivial models<sup>3</sup>. SLIM has an advantage for such problems as it does not only avoid producing a trivial model, but can produce a model at any user-specified point on the ROC curve without parameter tuning. That is, the designer can encode hard constraints on the sensitivity as a loss constraint.

The solution of a single run of the IP results in the least specific or most sensitive model. If a maximum error rate of  $\gamma \in [0, 1]$  on negatively labeled examples is necessary, the IP can be formulated as

$$\begin{aligned} \min_{\lambda} \quad & \frac{W^+}{N} \sum_{i \in \mathcal{I}^+} \mathbf{1}[y_i \lambda^T \mathbf{x}_i \leq 0] + \frac{W^-}{N} \sum_{i \in \mathcal{I}^-} \mathbf{1}[y_i \lambda^T \mathbf{x}_i \leq 0] + C_0 |\lambda|_0 + \epsilon |\lambda|_1 \\ \text{s.t.} \quad & \frac{1}{N^-} \sum_{i \in \mathcal{I}^-} \mathbf{1}[y_i \lambda^T \mathbf{x}_i > 0] \leq \gamma \\ & \lambda \in \mathcal{L}, \end{aligned}$$

where  $W^+$  and  $W^-$  are user-defined weights that control the accuracy on the  $N^+$  positive examples from the set  $\mathcal{I}^+ = \{i : y_i = +1\}$  and  $N^-$  examples of the set  $\mathcal{I}^- = \{i : y_i = -1\}$ , respectively.

Assuming  $W^+ + W^- = 1$ , setting,  $W^+ > \frac{N^-}{1+N^-}$ , weighs the accuracy on each positive example as heavily as all negative examples combined. This parametrization of the formulation therefore returns a scoring system that

<sup>3</sup> i.e. if the probability of a certain event is 1%, a model that never predicts this event will be right 99% of the time

attains the highest sensitivity among models with a maximum error of  $\gamma$  on negative examples.

### Feature-Based Constraints for Input Variables

SLIM allows for a fine-grained control over the composition of input variables in a scoring system by including feature-based constraints in its formulation. Specifically, indicator variables can be used to encode the  $\ell_0$ -norm  $\alpha_j := \mathbf{1}[\lambda_j \neq 0]$  to formulate logical constraints between features such as "either-or" conditions and "if-then" conditions, which represents an alternative to creating classification models that obey structured sparsity constraints or hierarchical constraints. The indicator variables  $\alpha_j$  can be used to limit the number of input variables to at most  $\Theta$  by adding the constraint,  $\sum_{j=1}^P \alpha_j \leq \Theta$ . More complicated feature-based constraints include "if-then" constraints to ensure that a scoring system includes certain variables conditionally on the presence of another one, for example  $\alpha_{then} \leq \alpha_{if}$ . Hierarchical constraints, for example that a leaf feature can only be used if the nodes above it have been included, can be encoded as  $\alpha_{leaf} \leq \alpha_{nodes}$ .

### Feature-Based Preferences

Practitioners often have soft preferences for certain input variables. SLIM allows to encode such preferences by specifying a distinct trade-off parameters for each coefficient  $C_{0,j}$ . When a feature  $j$  is preferred over a feature  $k$ , it is possible to set  $C_{0,k} = C_{0,j} + \delta$ , where  $\delta > 0$  represents the maximum additional training accuracy that is allowed to be sacrificed to use feature  $j$  over  $k$ . Setting  $\delta = 0.05$  would ensure that feature  $k$  is only used, if an improvement of 5% can be attained compared to using feature  $j$ . This approach is also a possible avenue to handle missing data, as  $\delta$  can be set proportionally to the number of missing values, thus penalizing variables with a lot of missing values.

## 6.2.4 Application of SLIM within the PERMEATED-framework

Predict FAULT if SCORE  $\geq 3$

Indicator	Weight	Score
1. Y_MotionUnit3_p750	5	+ .....
<b>ADD POINTS FROM ROWS 1-1 SCORE</b>		= .....

Table 6.2: Textual form of the SLIM for the described test case.

Using an implementation in *python* of the SLIM algorithm and IBM's CPLEX optimizer, with settings tuned to high penalties for the inclusion of additional features, a rather trivial model was obtained, as can be seen from Tab. 6.2

It is apparent from inspecting the confusion matrix of the classifier in Fig. 6.3, that this classifier does not only perform well in comparison to the alternative algorithms outlined in Sec. 6.1.4, but is simple enough to allow for a rather straight forward interpretation: for the majority of cases, it is sufficient to look at the indicator *Y\_MotionUnit3\_p750* to be able to diagnose the status of the joint system. A root cause analysis using mechanical models of the machine could establish that an eigenmode of the combined system was severely influenced by changes in the modeled stiffness of joints. Changes in that eigenmode from "normal" were captured in this indicator. This lead to the conclusion, that the process of controlling the joining forces was not stable.

This example, admittedly selected for its clarity, shows conclusively that the usage of an interpretable model like this one can at least produce leads for further investigations into the root causes of a problem. The model structure is simple enough to lend itself to human inspection and can garner trust quickly by producing accurate results. A downside of this method is obvious: the class of models is restricted to a linear model. Non-linear effects cannot be handled appropriately. Another drawback is the need for an integer program solver. While there are excellent commercial products and free-for-research versions,

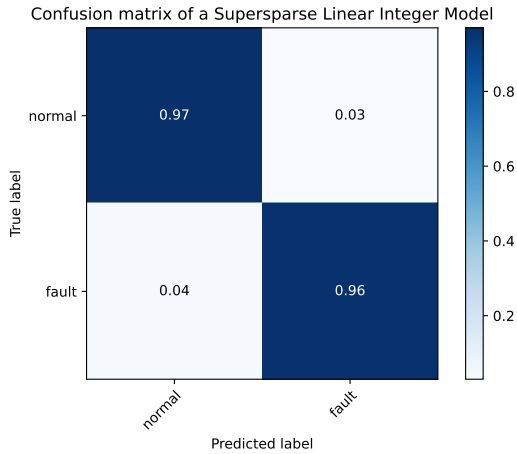


Figure 6.3: Confusion Matrix of the supersparse linear integer model. Tested on a hold-out set.

the SLIM algorithms was slow compared to other machine learning methods. A possible solution to this problem is presented in Sec. 6.3.

## 6.3 Explainers

As detailed in Chap. 5, the ability to generate explanations for the decisions of a system with relevant consequences, for example for maintenance decisions is of critical importance. The same needs arise from the application of machine learning systems to areas of even greater consequence, like for medical decision or aircraft collision avoidance systems.

Simultaneously, machine learning algorithms have faced some unexpected challenges when deployed in the real world, including an inability to distinguish causal effects from mere correlation, fairness and a lack of trust by end users. Interpretability is a promising approach to address these challenges, because the human user can be supported in diagnosing issues and verifying the correctness of machine learning models by providing insights into how the model arrived at its conclusions. There has been research on this field for some time, like [392], [393].

Prominent examples of this include applications, where models that are trained without knowledge of a feature that is considered to be unfit to be included in a model for ethical reasons, like gender or ethnicity, are still found to be prejudiced against these protected groups, since models seem to be able to reconstruct these omitted features from other features. It has been proposed that instead of omitting such features, models should include them and the effect of their inclusion should be controlled for. [394]

This necessitates the ability to understand the model's reasoning process, i.e. an explanation of how model predictions are affected by changing the prejudiced feature. Similarly, a user may want to determine whether a known causal connection is adequately represented in the model or understand the high-level structure of a model to gain confidence in its correctness. A method for retrieving such explanations from otherwise opaque models is discussed next. While there are models widely regarded as being interpretable, like linear models, decision trees, rule lists, etc., most of the algorithms that show the highest accuracies at benchmarks, are hard or even impossible to interpret, like various flavors of deep learning models.

For "simple" models, measured as the class of the model and the number of input parameters and weights, the model itself is the best explanation. The model generated by the SLIM algorithm in Sec. 6.2 is an example of that.

For more complex models, such as ensemble methods or deep networks, the models themselves are not suited as an explanation, because they are too complex. Instead, a simpler explanation model, which is some interpretable approximation of the original model has to be used. This is called an explainer.

*Explaining* a prediction in this context refers to the presentation of data suitable for gaining qualitative understanding of the relationship between the components of a particular instance and the model's prediction.

Humans typically have some prior knowledge about the application domain which is used to accept or to reject a model prediction, if they understand the reasoning behind the prediction. While an explanation by itself might not prove to be sufficient for garnering trust in the original model, it was discussed in Sec.5.1 and 5.2 why interpretable models can help to generate trust in such models and therefore foster their adoption, if the explanations are faithful and intelligible. Practitioners often overestimate the accuracy of their models and thus trust cannot solely rely on having constructed the model using cross-validation. The clearer the causality in a given model, the easier the interpretation becomes. Physics-based models are therefore typically more readily interpretable than neural networks.

The insights given by explanation are particularly helpful in identifying what must be done to convert an untrustworthy model into a trustworthy one. [382] presents the case, where a model with higher prediction accuracy is rejected by human users, as soon as explanations are presented. In the specific case, the task was to differentiate pictures of huskies from pictures of wolves. The classifier with higher accuracy was rejected as soon as the explanation of the classification revealed that the background of the picture had a huge influence on the classification result. The classifier was good at recognizing snow in the background and labeled pictures as "wolf". Desirable features of an explainer include

- interpretability,
- local-fidelity,
- model-agnosticity.

Interpretability means the provision of qualitative understanding between inputs and response. As already discussed, interpretability is a function of the user of a system. A machine learning practitioner will probably be able to interpret small Bayesian nets, but others might be more comfortable with a small number of features and their relative weight. It should also be noted, that features suitable for accurate predictions are not necessarily suitable for explanations, see also Sec. 5.2.

Although it will in general not be possible to achieve complete faithfulness unless it is a complete description of the model itself, an explanation must at least be locally faithful, i.e. it must correspond to how the model behaves in the vicinity of the instance being predicted. Local-fidelity of course does not imply global-fidelity. The converse is true, but it remains a challenge to obtain global-fidelity for complex systems.

An explainer should be able to explain any model, i.e. treat it as a black box. This is advantageous, given the fact that many state-of-the-art classifiers are not interpretable and also ensures applicability for future types of classifiers.

### 6.3.1 Local Interpretable Model-Agnostic Explanations (LIME)

Formally, we follow [382] in defining an explanation as a model  $g \in G$ , where  $G$  is a class of potentially *interpretable* models, such as linear models, decision trees or falling rule lists, i.e. a model  $g$  that can readily be presented to a user. The domain of  $g$  is  $\{0, 1\}^M$ , i.e.  $g$  acts over absence or presence of the interpretable components. Not every  $g \in G$  may be simple enough to be interpretable - thus let  $\Omega(g)$  be a measure of complexity of the explanation.  $\Omega(g)$  could be the depth of a decision tree or the number of non-zero weights in a linear model.



Let the model being explained be denoted  $f : \mathbb{R}^M \rightarrow \mathbb{R}$ , in classification,  $f(x)$  is the probability that  $x$  belongs to a certain class.

Additionally, a similarity measure  $\pi_x(z)$  is a proximity measure between an instance  $z$  to  $x$ , so as to define locality around  $x$ . Finally, let  $\mathcal{L}(f, g, \pi_x)$  be a measure of unfaithfulness of  $g$  in approximating  $f$  in the locality defined by  $\pi_x$ . To ensure interpretability and local-fidelity,  $\mathcal{L}(f, g, \pi_x)$  has to be minimized, while  $\Omega(g)$  remains sufficiently low.

The explanation  $\zeta(x)$  produced by LIME is obtained by solving the optimization problem

$$\zeta(x) = \arg \min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g). \quad (6.31)$$

This formulation can be used to test different explanation families  $G$ , fidelity functions  $\mathcal{L}$ , and complexity measures  $\Omega$ . While the problem setup of LIME is rather intuitive, it lacks some generality. In the next section a more general setup will be explained and applied to the LIME problem formulation.

### 6.3.2 SHapely Additive exPLANation (SHAP) Values

[395] introduces a method, called Shapely Additive Explanations (SHAP), which can be seen as a generalization of multiple other methods of generating explanations, one of which is LIME. This method is concerned with generating local methods to explain a prediction  $f(x)$  based on a single input.

Explanation models often use simplified inputs  $x'$  that map to the original inputs through a mapping function  $x = h_x(x')$ . It is noteworthy that the mapping function  $h_x$  is specific for a given input  $x$ . The simplified input can thus contain considerable less information than the original. A *local* method in this context, is a method that tries to ensure  $g(z') \approx f(h_x(z'))$  whenever  $z' \approx x'$ .

Additive feature attribution methods have an explanation model that is a *linear* function of *binary* variables:

$$g(z_0) = \phi_0 + \sum_{i=1}^M \phi_i z'_i, \quad (6.32)$$

where  $z' \in 0, 1^M$ ,  $M$  is the number of simplified input features, and  $\phi_i \in \mathbb{R}$  are real weights. Methods with explanation models matching this definition attribute an effect  $\phi_i$  to each feature, and summing over the effects of all feature attributions approximates the output of the original model. LIME is one of at least 6 different models of this form. [395] shows that there exist a single unique solution for this class of explanation models, if the problem is constraint to have three desirable properties.

The first property is *local accuracy*, which requires that the explanation model  $g(x')$ , matches the original model  $f(x)$ , when  $x = h_x(x')$ .

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^M \phi_i x'_i.$$

The second property, *missingness*, demands that features, for which  $x'_i = 0$ , do not have an attributed impact.

$$x'_i = 0 \implies \phi_i = 0$$

The last property, *consistency*, states that if a model changes in a manner so that some simplified input's contribution increases or stays the same regardless of other inputs, that input's attribution should not decrease.

More formally, let  $f_x(z') = f(h_x(z'))$  and  $z' \setminus i$  denote setting  $z'_i = 0$ . For any models  $f$  and  $f'$ , if

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i) \quad \forall z' \in \{0, 1\}^M,$$

then  $\phi_i(f', x) \geq \phi_i(f, x)$ .

The only possible explanation model  $g$  that follows the definition for additive feature attribution methods and simultaneously satisfies the three properties is

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)], \quad (6.33)$$

where  $|z'|$  is the number of non-zero entries in  $z'$  and  $z' \subseteq x'$  represents all  $z'$  vectors, where the non-zero entries are a subset of the non-zero entries in  $x'$ , [395]. SHAP values are proposed as a unified measure of feature importance. They are the solution to Eq. 6.33, where  $f_x(z') = f(h_x(z'))$  is the conditional expected value of observing  $\mathbb{E}[f(z)]$  given  $z_S$  and  $S$  is the set of non-zeros index in  $z'$ . Thus, they can be interpreted as attributing to each feature the change in the expected model prediction when conditioning on that feature, representing the increment from the base value  $\mathbb{E}[f(z)]$ , that would be predicted without knowledge of any simplified features to the current output  $f(x)$ .

Implicit in this formulation is a simplified input mapping,  $h_x(z') = z_S$ , where  $z_S$  has missing values for features that are not contained in  $S$ . Since most models cannot handle arbitrary patterns for missing input values,  $f(z_S)$  is approximated by  $\mathbb{E}[f(z)|z_S]$ .

From the definition of the SHAP values, it is clear that their exact computation is challenging. [395] provides approximation methods.

For a linear LIME model, the specific forms of  $\pi_{x'}$ ,  $\mathcal{L}$  and  $\Omega(g)$  that make the solution of the LIME optimization problem given in Eq. 6.31 consistent with local accuracy, missingness and consistency are:

$$\begin{aligned}\Omega(g) &= 0, \\ \pi_{x'}(z) &= \frac{(M-1)}{\binom{M}{|z'|} |z'| (M-|z'|)}, \\ \mathcal{L}(f, g, \pi_{x'}) &= \sum_{z' \in Z} [f(h_x(z')) - g(z')]^2 \pi_{z'},\end{aligned}$$

where  $|z'|$  is the number of non-zero elements in  $z'$ ,  $\pi_{x'} = \infty$ , when  $|z'| \in \{0, M\}$ . which enforces  $\phi_0 = f_x(\emptyset)$  and  $f(x) = \sum_{i=0}^M \phi_i$ .

### 6.3.3 Application of explainers within the PERMEATED-framework

Applying an implementation of the SHAP algorithm written in *python* [396] to the test use case specified in Chap. 6, it was possible to inspect the classifications made by the non-interpretable classifiers generated in Sec. 6.1 with the solution of the LIME problem derived in the preceding section. As can be seen from Fig. 6.4, the indicator with the dominant impact on the classification results was identified and is identical to the indicator that was identified by the SLIM algorithm of Sec. 6.2, without limiting the Responsible Decision-Maker to the set of algorithms, that might lack from accuracy due to their inability to cope with non-linear relationships in the training data. Explainers can thus be seen as a valuable addition to the PERMEATED-framework's Advisory Generation stage, although their construction can come with significant computational

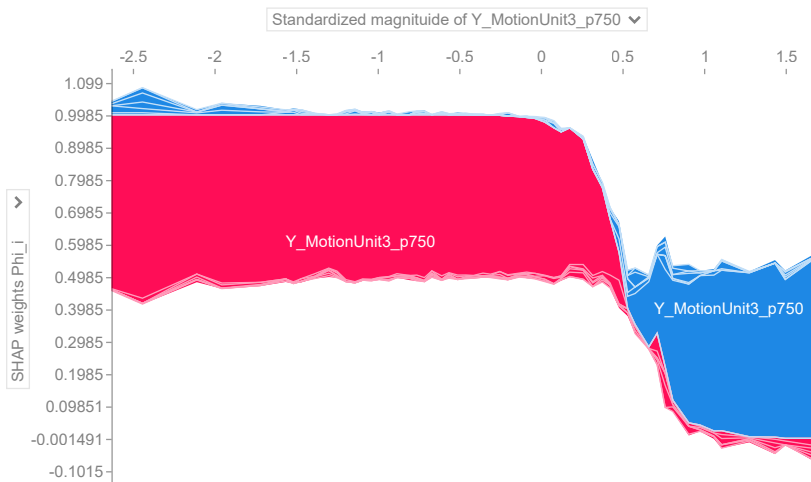


Figure 6.4: The Shapely Forces of individual indicators for an RBF SVM being applied to a hold-out set. The strength of the individual SHAP forces  $\phi_i$  is depicted on the y-axis, while the standardized magnitude of the indicator  $Y\_MotionUnit3\_p750$  is ordering the individual instances along the x-axis

burdens. The burden on the domain experts on the other hand can be lowered significantly, because they do not have to manually design fuzzy membership functions. In fact, their input is designed to be confined to labeling instances of measurements.

## 6.4 Summary

In this chapter, the extensibility of the PERMEATED-framework was demonstrated by integrating interpretable machine learning techniques and explanations obtained from opaque machine learning models. While the usage of opaque machine learning techniques, as demonstrated in Sec. 6.1, yields encouraging results with respect to classification accuracy, the lack of explanations for the classification results disqualifies algorithms of this class from an inclusion in the PERMEATED-framework. In Sec. 6.2, Supersparse Linear Integer Models have been investigated as an example of explainable machine learning models. For the use case under investigation, this model performed favorably in comparison to the opaque alternatives in terms of classification accuracy, while the resulting model was also simple enough for domain experts to probe it for its explanatory power. This clarity does not come without a price: the model is restricted to linear combinations of features, which precludes the appropriate handling of non-linear effects. Explainers were used in Sec. 6.3 to derive explainable models from opaque models. This approach enables domain experts to utilize in principle any class of machine learning models suitable for the problem domain, while retaining access to explanations. This approach is also not free of drawbacks: the choices of the class of possibly interpretable models and of the maximum allowed complexity shape in a quite literal way the depth and breadth of the generated explanations. The generated explanatory models are only locally faithful and a good explanation for the investigated instances does not guarantee a good explanation for uninvestigated ones, which might create an unjustified degree of trust in the explained model.

All of these techniques share the upside that the burden on the domain expert is eased significantly. Instead of having to explicitly specify the shape of fuzzy functions to map the relevance of specific indicators to error modes, using the tools of machine learning, their input is reduced to labeling a sufficiently large set of instances and judging the automatically produced explanations. But this comes at the cost of having to generate a sufficiently large training set, which might be prohibitively expensive, depending on the specific circumstances. Re-

ardless of the specific situation, though, the PERMEATED-framework has the capability to integrate the explicitly or implicitly modeled expert knowledge to generate useful decision for condition monitoring applications from it.

---

## 7 Summary and Outlook

Digitalization poses new challenges for manufactures of drives, machine tools and smart factories. Customers are increasingly interested in optimizing their production processes, for which the knowledge of the state of their assets is paramount.

Unplanned downtimes of individual machines can interrupt the complete production process and are therefore rather costly. But adding redundancy is not always an economical option, as spacial limitations on the factory floor and of course capital constraints may prohibit such a mitigation strategy. Condition Monitoring offers a path to a solution for this task. It tries to minimize unplanned downtimes by accelerating the diagnostic process by speeding up the process of finding a solution for a fault or by giving warning of looming faults early enough to reschedule certain production steps and converting the possible outage into a planned maintenance mission, albeit possibly on short notice.

In this thesis, it has been discussed that wear and tear of the machines is only partially responsible for downtimes and that the influence of environmental factors are also non-negligible contributors to unexpected failures. Certain faults are readily detectable from the signals of the drives, which are available in the numerical control of a machine tool. For some of these signals an unambiguous mapping to a fault is possible, which was demonstrated by the implementation of an interpretable diagnostic system at a production line and has globally service for diagnosing machine tools.

It has also been shown theoretically and empirically that uncertainties along



the diagnostic process pose the most severe challenge for health assessments in general and predictions in particular. The more uncertain diagnostic results are, the higher the risk of the decision predicated on them, which hinders the automation of decision processes.

The PERMEATED-framework was introduced, which stresses the importance of including the Responsible Decision-Maker, who is the addressee of health assessments, predictions, uncertainty quantifications and advices and ultimately decides on which course of action to take. The usability of a diagnostic system hinges critically on the trust that this responsible decision-maker has in its diagnostic capabilities.

To help the generation of trust in a diagnostic system, the PERMEATED-framework prescribes the usage of explainable models.

The application of the framework for a system based on fuzzy rules was demonstrated, which is used in the production line of a major machine tool manufacturer as a production-grade system. The limitations posed by the presence of uncertainty on this system were discussed. It proves the feasibility of a drives-based condition monitoring system for the detection of a broad range of phenomena and satisfies the requirement for explainability, but relies heavily on specifically tailored rule sets for the diagnosis of each machine series, which puts a high burden on the developer to provide concise and current rules for the types of assets that are subject to the diagnostic system. This increases the operational cost and decreases the total value of the diagnostics system. The specificity of the required skills for these kinds of tools introduce concerns about the scalability and maintainability of such solutions. To reduce these risks, the integration and adaption of the available tools for diagnostics to the particular demands of drives in machine tools in an industrial setting were explored.

The applicability of opaque machine learning algorithms and of a special interpretable form of machine learning algorithm from the medical context to a novel use case was demonstrated. The latter algorithm derives interpretable models that do not depend, at least theoretically, on any more input than is given by the set of labeled examples. While this is easing of the burden for the

	manual inspection	non-explainable AI	recommender system	eXplainable AI
automatability				
consistent quality				
detection spectrum				
cost of maintenance				
integration of feedback				
interpretability of results				
acceptance of advices				

Figure 7.1: Comparison of maintenance approaches using application-grounded metrics

programmer, it also comes with the price of constraining the available models to linear regressors.

Finally, explainers were explored. They enable the usage of highly non-linear and opaque models, that are for all practical purposes too complex to be interpreted by a human user, while still "opening the black box" of these models wide enough to extract explanations that can lead to insights about possible root causes and, more importantly, can serve as a route to falsify a diagnostic result, if an explainer reveals that an unimportant feature had a huge influence on an example for which the influences are known. This increases the range of tools available for the PERMEATED-framework considerably, but of course also adds the cost of creating explanations, possibly ad hoc, for instances of interest for the Responsible Decision-Maker. A comparison of the explored methods in the context of maintenance is given in Fig. 7.1.

Revisiting the goals laid out for this thesis in Sec. 1.2, it can be stated that all of them have been met, with the important exception that not all diagnostic measurement function are neutral with respect to the assets' productivity. The task of predicting failure modes remains somewhat elusive. Only a few of the so far identified failure modes lend themselves to making an estimate of the RUL.

In summary, this work presents a way to build a modular diagnostic system, which enables reasoning about uncertainty and risk. It also shows how to foster adoption of such a system by including on a fundamental conceptual level the need for and the needs of a Responsible Decision-Maker in the diagnosis

process, among which is the necessity for interpretable algorithms or at least explanations to enable the rejection of implausible models. The usability of this system was demonstrated for quality assurance tasks in an assembly line for machine tools and in the handling of many service-related incidents. With this, this thesis contributes to the solution of the task posed by the aforementioned growing demands for predictability of production assets without a need to retrofit existing machine tools in the field.

Possible direction for further research on this topic include:

- **Online Measurements:** The implementation of measurement functions with high enough sustained excitation to reliably and robustly identify the systems' and components' faults without negatively impacting productivity will increase the acceptance of data-driven maintenance methods. An additional requirement for such online measurements is for them to work with already installed sensors and controls to keep the effect on capital costs low.
- **Prognosis:** To build the capability of predicting future outages with enough certainty to base critical and costly maintenance decisions on the recommendations will require further research. The presentation and propagation of uncertainty is a computationally challenging task and the reduction of the possible pathways of a system's future to actionable recommendations is still unresolved.
- **Interpretable Algorithms:** A gainful avenue for further research is the development of methods to get to interpretable models that do not necessitate solving large integer programs, while maintaining a degree of control equivalent to the one offered by SLIM. Algorithms that would allow for importance ranked addition of indicators into the model could be of special interest, as the individual's limits of interpretability capacities could be fully utilized.

- **Explainers:** The inclusion of additional sensors into diagnostic systems, like acceleration sensors, microphones and even cameras becomes increasingly more feasible, given the growing availability of low-cost prosumer-grade devices. While the accuracy of analyses involving more specialized sensors should reduce uncertainty about many fault modes, the generation of interpretable models and explanations over a fused set of widely diverse types of information remains challenging.
- **Deployment:** Questions regarding where and how to physically host the components of a modular diagnostic process are largely unresolved. A deployment close to the edge offers advantages in regard to questions of latency as well as data ownership and trade secrets. The learning and updating process on the other hand become much more complicated. A cloud architecture based on containerized software or linearly scalable architecture like Hadoop offers virtually limitless computationally resources, but poses challenges in regard to trust in the providers with regard to intellectual property, bandwidth and customer acceptance. These are challenges that will necessitate further research on federated machine learning and privacy-preserving learning techniques.
- **Machine Reasoning:** Another gainful route for further research regards the application of techniques of machine reasoning. Knowledge about systems is captured in semantic graphs, which model the relationship between subjects and objects as well as between subjects and attributes. There are algorithms which can infer suggestions about yet unseen instances by contextualizing the observed instance and offering suggestions that worked for cases that were similar by some measure. Training the suggestions with methods of Reinforcement Learning seems to offer an alternative route to combat complexity and uncertainty.



---

## Bibliography

- [1] A. Huf, *Kumulative Lastermittlung aus Antriebsdaten zur Bewertung des Zustands von Werkzeugmaschinenkomponenten*. 2012.
- [2] “Condition monitoring and diagnostics of machines — general guidelines on using performance parameters,” International Organization for Standardization, Geneva, CH, Standard, Apr. 2002.
- [3] “Condition monitoring and diagnostics of machines – data interpretation and diagnostics techniques – part 1: General guidelines,” International Organization for Standardization, Geneva, CH, Standard, May 2012.
- [4] T. Wang, “Trajectory similarity based prediction for remaining useful life estimation,” Ph.D. dissertation, 2010.
- [5] “Instandhaltung - Begriffe der Instandhaltung; Dreisprachige Fassung,” DIN – Deutsches Institut für Normung e. V. (Hrsg.), Standard, Dec. 2010.
- [6] B. de Jonge, W. Klingenberg, R. Teunter, and T. Tinga, “Reducing costs by clustering maintenance activities for multiple critical units,” *Reliability engineering & system safety*, vol. 145, pp. 93–103, 2016.
- [7] H. Wang, “A survey of maintenance policies of deteriorating systems,” *European journal of operational research*, vol. 139, no. 3, pp. 469–489, 2002.
- [8] M. Walther, *Antriebsbasierte Zustandsdiagnose von Vorschubantrieben*. 2011.

- [9] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical systems and signal processing*, vol. 20, no. 7, pp. 1483–1510, 2006.
- [10] T. Van Tung and B.-S. Yang, "Machine fault diagnosis and prognosis: The state of the art," *International Journal of Fluid Machinery and Systems*, vol. 2, no. 1, pp. 61–71, 2009.
- [11] R. Teti, K. Jemielniak, G. O'Donnell, and D. Dornfeld, "Advanced monitoring of machining operations," *CIRP Annals-Manufacturing Technology*, vol. 59, no. 2, pp. 717–739, 2010.
- [12] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems—reviews, methodology and applications," *Mechanical systems and signal processing*, vol. 42, no. 1, pp. 314–334, 2014.
- [13] K. Javed, R. Gouriveau, and N. Zerhouni, "State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels," *Mechanical Systems and Signal Processing*, vol. 94, pp. 214–236, 2017.
- [14] G. Singh *et al.*, "Induction machine drive condition monitoring and diagnostic research—a survey," *Electric Power Systems Research*, vol. 64, no. 2, pp. 145–158, 2003.
- [15] M. Seera, C. P. Lim, S. Nahavandi, and C. K. Loo, "Condition monitoring of induction motors: A review and an application of an ensemble of hybrid intelligent models," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4891–4903, 2014.
- [16] M. Riera-Guasp, J. Pons-Llinares, V. Climente-Alarcon, *et al.*, "Diagnosis of induction machines under non-stationary conditions: Concepts and tools," in *Electrical Machines Design Control and Diagnosis (WEMDCD)*, 2013 IEEE Workshop on, IEEE, 2013, pp. 220–231.

- 
- [17] M. Riera-Guasp, J. A. Antonino-Daviu, and G.-A. Capolino, "Advances in electrical machine, power electronic, and drive condition monitoring and fault detection: State of the art," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1746–1759, 2015.
- [18] Y. Da, X. Shi, and M. Krishnamurthy, "Health monitoring, fault diagnosis and failure prognosis techniques for brushless permanent magnet machines," in *Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE*, IEEE, 2011, pp. 1–7.
- [19] M. Lukas, D. Stock, and A. Csiszar, "Fabos: Towards an open, distributed, real-time-capable, and secure operating system for production," *Procedia CIRP*, vol. 104, pp. 962–967, 2021.
- [20] J. Coble and J. W. Hines, "Applying the general path model to estimation of remaining useful life," *International Journal of Prognostics and Health Management*, vol. 2, no. 1, pp. 71–82, 2011.
- [21] F. Barbieri, W. Hines, M. Sharp, and M. Venturini, "Sensor-based degradation prediction and prognosis for remaining useful life estimation: Validation on experimental data of electric motors," *International Journal of Prognostics and Health Management*, 2015.
- [22] M. D. Bryant, "Modeling degradation using thermodynamic entropy," in *Proceedings of the 2014 IEEE Conference on Prognostics and Health Management, Cheney, WA, USA, 2014*, pp. 22–25.
- [23] F. Momeni, X. Jin, and J. Ni, "A microscopic approach for generic degradation modeling," *International journal of prognostics and health management*, vol. 7, 2016.
- [24] F. Peysson, M. Ouladsine, R. Outbib, J.-B. Leger, O. Myx, and C. Allemand, "A generic prognostic methodology using damage trajectory models," *IEEE transactions on reliability*, vol. 58, no. 2, pp. 277–285, 2009.
- [25] P. Van den Hof, "Closed-loop issues in system identification," *Annual reviews in control*, vol. 22, pp. 173–186, 1998.



- [26] B. Huang and R. Kadali, *Dynamic modeling, predictive control and performance monitoring: a data-driven subspace approach*. Springer, 2008.
- [27] Y. A. Shardt and B. Huang, "Closed-loop identification condition for armax models using routine operating data," *Automatica*, vol. 47, no. 7, pp. 1534–1537, 2011.
- [28] Y. A. Shardt, B. Huang, and S. X. Ding, "Minimal required excitation for closed-loop identification: Some implications for data-driven, system identification," *Journal of Process Control*, vol. 27, pp. 22–35, 2015.
- [29] Y. Shardt, "Data quality assessment for closed-loop system identification and forecasting with application to soft sensors," Ph.D. dissertation, 2012.
- [30] M. Gevers, A. S. Bazanella, X. Bombois, *et al.*, "Identification and the information matrix: How to get just sufficiently rich?" *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2828–2840, 2009.
- [31] S. Billings, "Identification of nonlinear systems—a survey," in *IEE Proceedings D (Control Theory and Applications)*, IET, vol. 127, 1980, pp. 272–285.
- [32] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.
- [33] R. H. Milocco and J. A. De Doná, "Robust deconvolution for armax models with gaussian uncertainties," *Signal Processing*, vol. 90, no. 12, pp. 3110–3121, 2010.
- [34] J. A. Suykens, J. Vandewalle, and B. De Moor, "Optimal control by least squares support vector machines," *Neural networks*, vol. 14, no. 1, pp. 23–35, 2001.
- [35] R. X. Gao, R. Yan, L. Zhang, and K. B. Lee, "Condition monitoring of operating spindle based on stochastic subspace identification," in *Proceedings of the ASME 2007 International Mechanical Engineering Congress and Exposition, United States, 2007*, pp. 1129–1135.

- 
- [36] S. J. Qin and L. Ljung, "Closed-loop subspace identification with innovation estimation," *IFAC Proceedings Volumes*, vol. 36, no. 16, pp. 861–866, 2003.
- [37] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 2009.
- [38] F. Daum, "Exact finite-dimensional nonlinear filters," *IEEE Transactions on Automatic Control*, vol. 31, no. 7, pp. 616–622, 1986.
- [39] F. Daum, "Nonlinear filters: Beyond the kalman filter," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57–69, 2005.
- [40] Z. Feng, S. Qin, and M. Liang, "Time–frequency analysis based on vold-kalman filter and higher order energy separation for fault diagnosis of wind turbine planetary gearbox under nonstationary conditions," *Renewable Energy*, vol. 85, pp. 45–56, 2016.
- [41] J. P. P. Gomes, B. P. Leão, W. O. Vianna, R. K. Galvão, and T. Yoneyama, "Failure prognostics of a hydraulic pump using kalman filter," in *Annual Conference of the Prognostics and Health Management Society, 2012*, 2012.
- [42] E. Bechhoefer, S. Clark, and D. He, "A state space model for vibration based prognostics," in *Annual Conference of the Prognostics and Health Management Society, 2010*, 2010, pp. 10–16.
- [43] M. J. Carr and W. Wang, "An approximate algorithm for prognostic modelling using condition monitoring information," *European Journal of Operational Research*, vol. 211, no. 1, pp. 90–96, 2011.
- [44] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, Ieee, 2000, pp. 153–158.
- [45] S. J. Julier, "The scaled unscented transformation," in *American Control Conference, 2002. Proceedings of the 2002*, IEEE, vol. 6, 2002, pp. 4555–4559.

- [46] X. Zhang and P. Pisu, "An unscented kalman filter based on-line diagnostic approach for pem fuel cell flooding," *International Journal of Prognostics and Health Management*, vol. 1, no. 5, 2014.
- [47] G. A. Terejanu, "Unscented kalman filter tutorial," *University at Buffalo, Buffalo*, 2011.
- [48] P. Lim, C. K. Goh, K. C. Tan, and P. Dutta, "Estimation of remaining useful life based on switching kalman filter neural network ensemble," *Rolls Royce Singapore Singapore Singapore*, Tech. Rep., 2014.
- [49] D. Chelidze and J. P. Cusumano, "A dynamical systems approach to failure prognosis," *TRANSACTIONS-AMERICAN SOCIETY OF MECHANICAL ENGINEERS JOURNAL OF VIBRATION AND ACOUSTICS*, vol. 126, no. 1, pp. 2–8, 2004.
- [50] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Transactions on automatic control*, vol. 54, no. 6, pp. 1254–1269, 2009.
- [51] J. Son, S. Zhou, C. Sankavaram, X. Du, and Y. Zhang, "Remaining useful life prediction based on noisy condition monitoring signals using constrained kalman filter," *Reliability Engineering & System Safety*, vol. 152, pp. 38–50, 2016.
- [52] M. Orchard, "A particle filtering-based framework for online fault diagnosis and failure prognosis," *PhD proposal, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA*, 2006.
- [53] M. Orchard, G. Kacprzynski, K. Goebel, B. Saha, and G. Vachtsevanos, "Advances in uncertainty representation and management for particle filtering applied to prognostics," in *Prognostics and health management, 2008. phm 2008. international conference on*, IEEE, 2008, pp. 1–6.
- [54] V. Verma, S. Thrun, and R. Simmons, "Variable resolution particle filter," in *IJCAI*, 2003, pp. 976–984.

- 
- [55] M. E. Orchard, P. Hevia-Koch, B. Zhang, and L. Tang, "Risk measures for particle-filtering-based state-of-charge prognosis in lithium-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, pp. 5260–5269, 2013.
- [56] B. Zhang, T. Khawaja, R. Patrick, G. Vachtsevanos, M. E. Orchard, and A. Saxena, "Application of blind deconvolution denoising in failure prognosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 2, pp. 303–310, 2009.
- [57] C. Chen, G. Vachtsevanos, and M. Orchard, "Machine remaining useful life prediction based on adaptive neuro-fuzzy and high-order particle filtering," in *Annual conference of the prognostics and health management society, 2010*, 2010.
- [58] C. Chen, B. Zhang, G. Vachtsevanos, and M. Orchard, "Machine condition prediction based on adaptive neuro-fuzzy and high-order particle filtering," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 4353–4364, 2011.
- [59] G. Bartram and S. Mahadevan, "Dynamic bayesian networks for prognosis," in *Proceedings of the Annual Conference of the Prognostics and Health Management Society, 2013*, 2013.
- [60] J. Yoon and D. He, "Development of an efficient prognostic estimator," *Journal of Failure Analysis and Prevention*, vol. 15, no. 1, pp. 129–138, 2015.
- [61] E. N. Chatzi and A. W. Smyth, "The unscented kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing," *Structural control and health monitoring*, vol. 16, no. 1, pp. 99–123, 2009.
- [62] J. Bergh, F. Ekstedt, and M. Lindberg, *Wavelets mit Anwendungen in Signal-und Bildverarbeitung*. Springer-Verlag, 2007.
- [63] A. Graps, "An introduction to wavelets," *IEEE computational science and engineering*, vol. 2, no. 2, pp. 50–61, 1995.

- [64] N. E. Huang, Z. Shen, S. R. Long, *et al.*, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," in *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences*, The Royal Society, vol. 454, 1998, pp. 903–995.
- [65] B. L. Barnhart, *The Hilbert-Huang transform: theory, applications, development*. The University of Iowa, 2011.
- [66] I. N. Tansel, C. Mekdeci, and C. Mclaughlin, "Detection of tool failure in end milling with wavelet transformations and neural networks (wt-nn)," *International Journal of Machine Tools and Manufacture*, vol. 35, no. 8, pp. 1137–1147, 1995.
- [67] P. Wang and G. Vachtsevanos, "Fault prognostics using dynamic wavelet neural networks," *AI EDAM*, vol. 15, no. 4, pp. 349–365, 2001.
- [68] K. Kunadumrongrath and A. Ngaopitakkul, "Discrete wavelet transform and support vector machines algorithm for classification of fault types on transmission line," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 2, 2012.
- [69] K. Zhu, Y. San Wong, and G. S. Hong, "Wavelet analysis of sensor signals for tool condition monitoring: A review and some new results," *International Journal of Machine Tools and Manufacture*, vol. 49, no. 7, pp. 537–553, 2009.
- [70] Z. Chen, R. Qi, and H. Lin, "Inter-turn short circuit fault diagnosis for pmsm based on complex gauss wavelet," in *Wavelet Analysis and Pattern Recognition, 2007. ICWAPR'07. International Conference on*, IEEE, vol. 4, 2007, pp. 1915–1920.
- [71] Z. Peng and F. Chu, "Application of the wavelet transform in machine condition monitoring and fault diagnostics: A review with bibliography," *Mechanical systems and signal processing*, vol. 18, no. 2, pp. 199–221, 2004.

- 
- [72] S. Zhang, J. Mathew, L. Ma, and Y. Sun, "Best basis-based intelligent machine fault diagnosis," *Mechanical systems and signal processing*, vol. 19, no. 2, pp. 357–370, 2005.
- [73] H. Yang, J. Mathew, and L. Ma, "Fault diagnosis of rolling element bearings using basis pursuit," *Mechanical Systems and Signal Processing*, vol. 19, no. 2, pp. 341–356, 2005.
- [74] V. Plapper and M. Weck, "Signale digitaler antriebe zeigen maschinenschäden," *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, vol. 97, no. 3, pp. 84–88, 2002.
- [75] L. Gelman, T. H. Patel, G. Persin, B. Murray, and A. Thomson, "Novel technology based on the spectral kurtosis and wavelet transform for rolling bearing diagnosis," *International Journal of Prognostics and Health Management*, ISSN, pp. 2153–2648, 2013.
- [76] X. Lou and K. A. Loparo, "Bearing fault diagnosis based on wavelet transform and fuzzy inference," *Mechanical systems and signal processing*, vol. 18, no. 5, pp. 1077–1095, 2004.
- [77] Q. Miao and V. Makis, "Condition monitoring and classification of rotating machinery using wavelets and hidden markov models," *Mechanical systems and signal processing*, vol. 21, no. 2, pp. 840–855, 2007.
- [78] K. Javed, R. Gouriveau, N. Zerhouni, and P. Nectoux, "A feature extraction procedure based on trigonometric functions and cumulative descriptors to enhance prognostics modeling," in *Prognostics and Health Management (PHM), 2013 IEEE Conference on*, IEEE, 2013, pp. 1–7.
- [79] Z. Peng, W. T. Peter, and F. Chu, "A comparison study of improved hilbert–huang transform and wavelet transform: Application to fault diagnosis for rolling bearing," *Mechanical systems and signal processing*, vol. 19, no. 5, pp. 974–988, 2005.
- [80] A. Bouchikhi and A.-O. Boudraa, "Multicomponent am–fm signals analysis based on emd–b-splines esa," *Signal Processing*, vol. 92, no. 9, pp. 2214–2228, 2012.

- [81] C. Ly, K. Ranney, K. Tom, H. Khatri, and H. Decker, "Effectiveness of empirical mode decomposition based features compared to kurtosis based features for diagnosis of pinion crack detection in a helicopter," ARMY RESEARCH LAB ADELPHI MD, Tech. Rep., 2010.
- [82] D. Maier, "Sensorlose online zustandserfassung von vorschubantrieb-skomponenten in werkzeugmaschinen," Ph.D. dissertation, 2015.
- [83] A. Soualhi, K. Medjaher, and N. Zerhouni, "Bearing health monitoring based on hilbert–huang transform, support vector machine, and regression," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 1, pp. 52–62, 2015.
- [84] C. Junsheng, Y. Dejie, and Y. Yu, "A fault diagnosis approach for roller bearings based on emd method and ar model," *Mechanical Systems and Signal Processing*, vol. 20, no. 2, pp. 350–362, 2006.
- [85] M. G. Frei and I. Osorio, "Intrinsic time-scale decomposition: Time–frequency–energy analysis and real-time filtering of non-stationary signals," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, The Royal Society, vol. 463, 2007, pp. 321–342.
- [86] A. Bouchikhi, "Am-fm signal analysis by teager huang transform: Application to underwater acoustics," Ph.D. dissertation, Université Rennes 1, 2010.
- [87] H. Shao, W. Jin, and S. Qian, "Order tracking by discrete gabor expansion," *IEEE Transactions on Instrumentation and measurement*, vol. 52, no. 3, pp. 754–761, 2003.
- [88] W. Amer, R. Grosvenor, and P. Prickett, "Machine tool condition monitoring using sweeping filter techniques," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 221, no. 1, pp. 103–117, 2007.
- [89] N. Sawalhi and R. B. Randall, "The application of spectral kurtosis to bearing diagnostics," in *Proceedings of ACOUSTICS*, 2004, pp. 3–5.

- 
- [90] J. Antoni, "The spectral kurtosis: A useful tool for characterising non-stationary signals," *Mechanical Systems and Signal Processing*, vol. 20, no. 2, pp. 282–307, 2006.
- [91] J. Antoni and R. Randall, "The spectral kurtosis: Application to the vibratory surveillance and diagnostics of rotating machines," *Mechanical Systems and Signal Processing*, vol. 20, no. 2, pp. 308–331, 2006.
- [92] C.-C. Wang and G.-P. J. Too, "Rotating machine fault detection based on hos and artificial neural networks," *Journal of intelligent manufacturing*, vol. 13, no. 4, pp. 283–293, 2002.
- [93] L. Gelman, T. H. Patel, and B. M. A. Thomson, "Rolling bearing diagnosis based on the higher order spectra," *International Journal of Prognostics and Health Management*, ISSN, pp. 2153–2648, 2013.
- [94] X. Yang, D. Lu, C. Ma, J. Zhang, and W. Zhao, "Analysis on the multi-dimensional spectrum of the thrust force for the linear motor feed drive system in machine tools," *Mechanical Systems and Signal Processing*, vol. 82, pp. 68–79, 2017.
- [95] X. Tian, J. X. Gu, I. Rehab, G. M. Abdalla, F. Gu, and A. Ball, "A robust detector for rolling element bearing condition monitoring based on the modulation signal bispectrum and its performance evaluation against the kurtogram," *Mechanical Systems and Signal Processing*, vol. 100, pp. 167–187, 2018.
- [96] J. Antoni, "Cyclostationarity by examples," *Mechanical Systems and Signal Processing*, vol. 23, no. 4, pp. 987–1036, 2009.
- [97] A. Verl, U. Heisel, M. Walther, and D. Maier, "Sensorless automated condition monitoring for the control of the predictive maintenance of machine tools," *CIRP Annals-Manufacturing Technology*, vol. 58, no. 1, pp. 375–378, 2009.



- [98] D. Maier and U. Heisel, "A comparison of model and signal based condition monitoring and mode separation for predictive maintenance of feed drives," *Journal of Machine Engineering*, vol. 11, no. 4, pp. 138–151, 2011.
- [99] F. Dalvand, A. Kalantar, and M. S. Safizadeh, "A novel bearing condition monitoring method in induction motors based on instantaneous frequency of motor voltage," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 364–376, 2016.
- [100] C. Capdessus, E. Sekko, and J. Antoni, "Speed transform, a new time-varying frequency analysis technique," in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, Springer, 2014, pp. 23–35.
- [101] E. Bechhoefer, B. Van Hecke, and D. He, "Processing for improved spectral analysis," in *Annual Conference of the Prognostics and Health Management Society, New Orleans, LA, Oct, 2013*, pp. 14–17.
- [102] P. Boskoski and D. Juricic, "Modeling localized bearing faults using inverse gaussian mixtures," in *V: Annual Conference on Prognostics and Health Management Society, PHM Society, New Orleans, USA, 2013*.
- [103] P. McFadden, "Detecting fatigue cracks in gears by amplitude and phase demodulation of the meshing vibration," *Journal of vibration, acoustics, stress, and reliability in design*, vol. 108, no. 2, pp. 165–170, 1986.
- [104] C. Cempel and M. Tabaszewski, "Multidimensional condition monitoring of machines in non-stationary operation," *Mechanical Systems and Signal Processing*, vol. 21, no. 3, pp. 1233–1241, 2007.
- [105] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.
- [106] L. Zadeh, "Toward a theory of fuzzy systems. erl report n 69-2. electronic research laboratories," *Univ of California. Berkeley*, 1969.

- 
- [107] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy sets and systems*, vol. 90, no. 2, pp. 111–127, 1997.
- [108] L. A. Zadeh, "Fuzzy logic= computing with words," *IEEE transactions on fuzzy systems*, vol. 4, no. 2, pp. 103–111, 1996.
- [109] Z. Pawlak, "Rough sets," *International Journal of Parallel Programming*, vol. 11, no. 5, pp. 341–356, 1982.
- [110] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Information sciences*, vol. 177, no. 1, pp. 3–27, 2007.
- [111] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron, "Rough sets: A tutorial," *Rough fuzzy hybridization: A new trend in decision-making*, pp. 3–98, 1999.
- [112] Y. Kaya and M. Uyar, "A hybrid decision support system based on rough set and extreme learning machine for diagnosis of hepatitis disease," *Applied Soft Computing*, vol. 13, no. 8, pp. 3429–3438, 2013.
- [113] X. Xu, G. Wang, S. Ding, X. Jiang, and Z. Zhao, "A new method for constructing granular neural networks based on rule extraction and extreme learning machine," *Pattern Recognition Letters*, vol. 67, pp. 138–144, 2015.
- [114] W. Pedrycz and G. Vukovich, "Granular neural networks," *Neurocomputing*, vol. 36, no. 1, pp. 205–224, 2001.
- [115] Z. Pawlak, "Rough sets and fuzzy sets," *Fuzzy sets and Systems*, vol. 17, no. 1, pp. 99–102, 1985.
- [116] A. Ganivada, S. Dutta, and S. K. Pal, "Fuzzy rough granular neural networks, fuzzy granules, and classification," *Theoretical Computer Science*, vol. 412, no. 42, pp. 5834–5853, 2011.
- [117] S. Shao, K. Nezu, K. Chen, and X. Pu, "Feature extraction of machinery diagnosis using neural network," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, IEEE, vol. 1, 1995, pp. 459–464.

- [118] D. P. Wipf and S. S. Nagarajan, "A new view of automatic relevance determination," in *Advances in neural information processing systems*, 2008, pp. 1625–1632.
- [119] D. Kateris, D. Moshou, X.-E. Pantazi, I. Gravalos, N. Sawalhi, and S. Loutridis, "A machine learning approach for the condition monitoring of rotating machinery," *Journal of Mechanical Science and Technology*, vol. 28, no. 1, pp. 61–71, 2014.
- [120] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
- [121] K. Javed, R. Gouriveau, R. Zemouri, and N. Zerhouni, "Features selection procedure for prognostics: An approach based on predictability," *IFAC Proceedings Volumes*, vol. 45, no. 20, pp. 25–30, 2012.
- [122] H. Senoussi, B. Chebel-Morello, M. Denai, and N. Zerhouni, "Feature selection and categorization to design reliable fault detection systems," in *Annual Conference of the Prognostics and Health Management Society, 2011*, 2011.
- [123] S. Si, H. Dui, Z. Cai, and S. Sun, "The integrated importance measure of multi-state coherent systems for maintenance processes," *IEEE Transactions on Reliability*, vol. 61, no. 2, pp. 266–273, 2012.
- [124] Y. Huang, X. F. Zha, J. Lee, and C. Liu, "Discriminant diffusion maps analysis: A robust manifold learner for dimensionality reduction and its applications in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 34, no. 1, pp. 277–297, 2013.
- [125] J. Yu, "A nonlinear probabilistic method and contribution analysis for machine condition monitoring," *Mechanical Systems and Signal Processing*, vol. 37, no. 1, pp. 293–314, 2013.
- [126] J.-H. Zhou, C. K. Pang, F. L. Lewis, and Z.-W. Zhong, "Intelligent diagnosis and prognosis of tool wear using dominant feature identification," *IEEE transactions on Industrial Informatics*, vol. 5, no. 4, pp. 454–464, 2009.

- 
- [127] H. Xu, C. Caramanis, and S. Mannor, "Robustness and regularization of support vector machines," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1485–1510, 2009.
- [128] T. Poggio and F. Girosi, "A theory of networks for approximation and learning," MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB, Tech. Rep., 1989.
- [129] A. Caponnetto, L. Rosasco, F. Odone, and A. Verri, "Support vector algorithms as regularization networks.," in *ESANN*, 2005, pp. 595–600.
- [130] C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural computation*, vol. 7, no. 1, pp. 108–116, 1995.
- [131] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of machine learning research*, vol. 7, no. Nov, pp. 2399–2434, 2006.
- [132] D. V. Chigirev and W. Bialek, "Optimal manifold representation of data: An information theoretic approach," in *Advances in Neural Information Processing Systems*, 2004, pp. 161–168.
- [133] P. Niyogi, "Manifold regularization and semi-supervised learning: Some theoretical analyses," *Computer Science Dept., University of Chicago, Tech. Rep. TR-2008-01*, 2008.
- [134] A. J. Smola, S. Mika, B. Schölkopf, and R. C. Williamson, "Regularized principal manifolds," *Journal of Machine Learning Research*, vol. 1, no. Jun, pp. 179–209, 2001.
- [135] I. W. Tsang and J. T. Kwok, "Large-scale sparsified manifold regularization," in *Advances in Neural Information Processing Systems*, 2007, pp. 1401–1408.
- [136] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

- [137] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th international conference on machine learning (ICML-13)*, 2013, pp. 1058–1066.
- [138] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.
- [139] M. Putz, U. Frieß, M. Wabner, A. Friedrich, A. Zander, and H. Schlegel, "State-based and self-adapting algorithm for condition monitoring," *Procedia CIRP*, vol. 62, pp. 311–316, 2017.
- [140] D. A. Clifton, S. Hugueny, and L. Tarassenko, "Novelty detection with multivariate extreme value statistics," *Journal of signal processing systems*, vol. 65, no. 3, pp. 371–389, 2011.
- [141] A. Hazan, J. Lacaille, and K. Madani, "Extreme value statistics for vibration spectra outlier detection," in *International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, 2012, p–1.
- [142] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [143] M. G. Mehrabi and E. Kannatey-Asibu Jr, "Hidden markov model-based tool wear monitoring in turning," *Journal of Manufacturing Science and Engineering*, vol. 124, no. 3, pp. 651–658, 2002.
- [144] D. A. Tobon-Mejia, K. Medjaher, N. Zerhouni, and G. Tripot, "A mixture of gaussians hidden markov model for failure diagnostic and prognostic," in *Automation Science and Engineering (CASE), 2010 IEEE Conference on*, IEEE, 2010, pp. 338–343.
- [145] I. Strachan and D. Clifton, "A hidden markov model for condition monitoring of a manufacturing drilling process," *IET Condition Monitoring*, pp. 803–814, 2009.

- 
- [146] Z. Chen, Y. Yang, Z. Hu, and Z. Ge, "A new method of bearing fault diagnostics in complex rotating machines using multi-sensor mixture hidden markov models," in *Proceedings of annual conference of the prognostics and health management society*, 2011, pp. 1–6.
- [147] C. Bunks, D. McCarthy, and T. Al-Ani, "Condition-based maintenance of machines using hidden markov models," *Mechanical Systems and Signal Processing*, vol. 14, no. 4, pp. 597–612, 2000.
- [148] L. Atlas, M. Ostendorf, and G. D. Bernard, "Hidden markov models for monitoring machining tool-wear," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, IEEE, vol. 6, 2000, pp. 3887–3890.
- [149] V. A. Epanechnikov, "Non-parametric estimation of a multivariate probability density," *Theory of Probability & Its Applications*, vol. 14, no. 1, pp. 153–158, 1969.
- [150] M. Markou and S. Singh, "Novelty detection: A review—part 1: Statistical approaches," *Signal processing*, vol. 83, no. 12, pp. 2481–2497, 2003.
- [151] P. Ram and A. G. Gray, "Density estimation trees," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2011, pp. 627–635.
- [152] C. He and M. Girolami, "Novelty detection employing an  $l_2$  optimal non-parametric density estimator," *Pattern Recognition Letters*, vol. 25, no. 12, pp. 1389–1397, 2004.
- [153] M. Girolami and C. He, "Probability density estimation from optimally condensed data samples," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 10, pp. 1253–1264, 2003.
- [154] Z. Deng, F.-L. Chung, and S. Wang, "Frsde: Fast reduced set density estimator using minimal enclosing ball approximation," *Pattern Recognition*, vol. 41, no. 4, pp. 1363–1372, 2008.

- [155] B. Silverman, "Algorithm as 176: Kernel density estimation using the fast fourier transform," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 31, no. 1, pp. 93–99, 1982.
- [156] P. Perner, "Concepts for novelty detection and handling based on a case-based reasoning process scheme," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 1, pp. 86–91, 2009.
- [157] G. Williams, R. Baxter, H. He, S. Hawkins, and L. Gu, "A comparative study of rnn for outlier detection in data mining," in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, IEEE, 2002, pp. 709–712.
- [158] B. B. Thompson, R. J. Marks, J. J. Choi, M. A. El-Sharkawi, M.-Y. Huang, and C. Bunje, "Implicit learning in autoencoder novelty assessment," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, IEEE, vol. 3, 2002, pp. 2878–2883.
- [159] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.
- [160] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak, "Fault prognostic of bearings by using support vector data description," in *Prognostics and Health Management (PHM), 2012 IEEE Conference on*, IEEE, 2012, pp. 1–7.
- [161] M.-Q. Pan, S.-X. Qian, L.-Y. Lei, and X.-J. Zhou, "Support vector data description with model selection for condition monitoring," in *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, IEEE, vol. 7, 2005, pp. 4315–4318.
- [162] S. Wang, J. Yu, E. Lapira, and J. Lee, "A modified support vector data description based novelty detection approach for machinery components," *Applied Soft Computing*, vol. 13, no. 2, pp. 1193–1205, 2013.
- [163] S. Kim, Z. Yu, R. M. Kil, and M. Lee, "Deep learning of support vector machines with class probability output networks," *Neural Networks*, vol. 64, pp. 19–28, 2015.

- 
- [164] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Towards parameter-free data mining," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 206–215.
- [165] Z. He, S. Deng, X. Xu, and J. Z. Huang, "A fast greedy algorithm for outlier mining," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2006, pp. 567–576.
- [166] M. Gamon, "Graph-based text representation for novelty detection," in *Proceedings of TextGraphs: The first workshop on graph based methods for natural language processing*, 2006, pp. 17–24.
- [167] M. Filippone and G. Sanguinetti, "Information theoretic novelty detection," *Pattern Recognition*, vol. 43, no. 3, pp. 805–814, 2010.
- [168] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI communications*, vol. 7, no. 1, pp. 39–59, 1994.
- [169] P. Reuss, A. Hundt, K.-D. Althoff, W. Henkel, and M. Pfeiffer, "Case-based agents within the omaha project," in *Case-based Agents. ICCBR Workshop on Case-based Agents (ICCB-CBA-14), located at International Conference on Case-Based Reasoning, September 29-October 1, Cork, Ireland*, S. Vattam and D. W. Aha, Eds., ICCBR, 2014.
- [170] P. Reuss, K.-D. Althoff, and W. Henkel, "Case-based decision support on diagnosis and maintenance in the aircraft domain," in *LWDA 2016 - Lernen, Wissen, Daten, Analysen - Workshop Proceedings. GI-Workshop-Tage "Lernen, Wissen, Daten, Analysen" (LWDA-2016), September 12-14, Potsdam, Germany*, R. Krestel, D. Mottin, and E. Müller, Eds., CEUR, 2016, pp. 249–256.
- [171] J. Lam, S. Sankararaman, and B. Stewart, "Enhanced trajectory based similarity prediction with uncertainty quantification," *PHM 2014*, 2014.
- [172] V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.
- [173] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.



- [174] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Advances in computational mathematics*, vol. 13, no. 1, pp. 1–50, 2000.
- [175] P. Andras, "The equivalence of support vector machine and regularization neural networks," *Neural Processing Letters*, vol. 15, no. 2, pp. 97–104, 2002.
- [176] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American mathematical society*, vol. 68, no. 3, pp. 337–404, 1950.
- [177] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- [178] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The annals of statistics*, pp. 1171–1220, 2008.
- [179] Z. Yang, A. Wilson, A. Smola, and L. Song, "A la carte—learning fast kernels," in *Artificial Intelligence and Statistics*, 2015, pp. 1098–1106.
- [180] P.-S. Huang, H. Avron, T. N. Sainath, V. Sindhvani, and B. Ramabhadran, "Kernel methods match deep neural networks on timit," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, IEEE, 2014, pp. 205–209.
- [181] Y. Cho and L. K. Saul, "Kernel methods for deep learning," in *Advances in neural information processing systems*, 2009, pp. 342–350.
- [182] J. Li and S. K. Halgamuge, "Improving training speed of support vector machines by creating exploitable trends of lagrangian variables: An application to dna splice site detection," in *Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007*, IEEE, 2007, pp. 230–233.
- [183] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast svm training on very large data sets," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 363–392, 2005.
- [184] A. Ben-Hur and J. Weston, "A user's guide to support vector machines," *Data mining techniques for the life sciences*, pp. 223–239, 2010.

- 
- [185] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [186] M. Heinert, "Support vector machines—theoretical overview and first practical steps," *Application of Artificial Intelligence and Innovations in Engineering Geodesy*, p. 71, 2010.
- [187] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, 2009.
- [188] M. Bin Hasan, "Current based condition monitoring of electromechanical systems. model-free drive system current monitoring: Faults detection and diagnosis through statistical features extraction and support vector machines classification.," Ph.D. dissertation, University of Bradford, 2013.
- [189] B. Samanta, K. Al-Balushi, and S. Al-Araimi, "Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection," *Engineering Applications of Artificial Intelligence*, vol. 16, no. 7, pp. 657–665, 2003.
- [190] L. Jack and A. Nandi, "Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms," *Mechanical systems and signal processing*, vol. 16, no. 2-3, pp. 373–390, 2002.
- [191] B.-S. Yang and A. Widodo, "Support vector machine for machine fault diagnosis and prognosis," *journal of system design and dynamics*, vol. 2, no. 1, pp. 12–23, 2008.
- [192] V. Sotiris and M. Pecht, "Support vector prognostics analysis of electronic products and systems," in *AAAI Fall Symposium on Artificial Intelligence for Prognostics*, 2007, pp. 120–127.
- [193] Z. Liu, D. Chen, G. Tian, M.-L. Tang, M. Tan, and L. Sheng, "Efficient support vector machine method for survival prediction with seer data," in *Advances in Computational Biology*, Springer, 2010, pp. 11–18.

- [194] T. S. Khawaja, *A Bayesian least squares support vector machines based framework for fault diagnosis and failure prognosis*. Georgia Institute of Technology, 2010.
- [195] O. P. Yadav, D. Joshi, and G. Pahuja, "Support vector machine based bearing fault detection of induction motor," *Indian Journal of Advanced Electronics Engineering*, vol. 1, no. 1, pp. 34–39, 2013.
- [196] J. K. Kimotho, C. Sondermann-Woelke, T. Meyer, and W. Sextro, "Machinery prognostic method based on multi-class support vector machines and hybrid differential evolution-particle swarm optimization," *Chemical Engineering Transactions*, vol. 33, pp. 619–624, 2013.
- [197] A. Widodo and B.-S. Yang, "Application of nonlinear feature extraction and support vector machines for fault diagnosis of induction motors," *Expert Systems with Applications*, vol. 33, no. 1, pp. 241–250, 2007.
- [198] A. Widodo, B.-S. Yang, and T. Han, "Combination of independent component analysis and support vector machines for intelligent faults diagnosis of induction motors," *Expert systems with applications*, vol. 32, no. 2, pp. 299–312, 2007.
- [199] K. C. Gryllias and I. A. Antoniadis, "A support vector machine approach based on physical model training for rolling element bearing fault detection in industrial environments," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 2, pp. 326–344, 2012.
- [200] R. A. Patel and B. R. Bhalja, "Condition monitoring and fault diagnosis of induction motor using support vector machine," *Electric Power Components and Systems*, vol. 44, no. 6, pp. 683–692, 2016.
- [201] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical systems and signal processing*, vol. 21, no. 6, pp. 2560–2574, 2007.
- [202] J. Zhang, T. Sato, and S. Iai, "Support vector regression for on-line health monitoring of large-scale structures," *Structural safety*, vol. 28, no. 4, pp. 392–406, 2006.

- 
- [203] H.-S. Tang, S.-T. Xue, R. Chen, and T. Sato, "Online weighted ls-svm for hysteretic structural system identification," *Engineering Structures*, vol. 28, no. 12, pp. 1728–1735, 2006.
- [204] M. G. Armaki and R. Roshanfekr, "A new approach for fault detection of broken rotor bars in induction motor based on support vector machine," in *Electrical Engineering (ICEE), 2010 18th Iranian Conference on*, IEEE, 2010, pp. 732–738.
- [205] C. K. Pang, J.-H. Zhou, and X. Wang, "A mixed time-/condition-based precognitive maintenance framework for zero-breakdown industrial systems," *Control and Intelligent Systems*, vol. 41, no. 3, pp. 127–135, 2013.
- [206] A. Rojas and A. K. Nandi, "Practical scheme for fast detection and classification of rolling-element bearing faults using support vector machines," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1523–1536, 2006.
- [207] D. Shi and N. N. Gindy, "Tool wear predictive model based on least squares support vector machines," *Mechanical Systems and Signal Processing*, vol. 21, no. 4, pp. 1799–1814, 2007.
- [208] T. Marwala and C. B. Vilakazi, "Computational intelligence for condition monitoring," *arXiv preprint arXiv:0705.2604*, 2007.
- [209] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.
- [210] J. A. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: Robustness and sparse approximation," *Neurocomputing*, vol. 48, no. 1, pp. 85–105, 2002.
- [211] T. Khawaja and G. Vachtsevanos, "A novel bayesian least squares support vector machine based anomaly detector for fault diagnosis," in *Annual Conference of the Prognostics and Health Management Society, 2009*, 2009.

- [212] J. A. Suykens and J. Vandewalle, "Recurrent least squares support vector machines," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 7, pp. 1109–1114, 2000.
- [213] S. Melacci and M. Belkin, "Laplacian support vector machines trained in the primal," *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 1149–1184, 2011.
- [214] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Transactions on neural networks*, vol. 13, no. 2, pp. 464–471, 2002.
- [215] X. Jiang, Z. Yi, and J. C. Lv, "Fuzzy svm with a new fuzzy membership function," *Neural Computing & Applications*, vol. 15, no. 3-4, pp. 268–276, 2006.
- [216] Y. Chen and J. Z. Wang, "Support vector learning for fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 716–728, 2003.
- [217] C. Thiel, S. Scherer, and F. Schwenker, "Fuzzy-input fuzzy-output one-against-all support vector machines," in *Knowledge-based intelligent information and engineering systems*, Springer, 2007, pp. 156–165.
- [218] P.-H. Chen, C.-J. Lin, and B. Schölkopf, "A tutorial on  $\nu$ -support vector machines," *Applied Stochastic Models in Business and Industry*, vol. 21, no. 2, pp. 111–136, 2005.
- [219] S. S. Keerthi, "Efficient tuning of svm hyperparameters using radius/margin bound and iterative algorithms," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1225–1229, 2002.
- [220] C. Gold, A. Holub, and P. Sollich, "Bayesian approach to feature selection and parameter tuning for support vector machine classifiers," *Neural Networks*, vol. 18, no. 5, pp. 693–701, 2005.
- [221] B. Samanta and C. Nataraj, "Use of particle swarm optimization for machinery fault detection," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 2, pp. 308–316, 2009.

- 
- [222] C. M. Bishop and M. E. Tipping, "Variational relevance vector machines," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., 2000, pp. 46–53.
- [223] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001.
- [224] M. E. Tipping, A. C. Faul, *et al.*, "Fast marginal likelihood maximisation for sparse bayesian models.," in *AISTATS*, 2003.
- [225] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.
- [226] T. Van Gestel, J. A. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor, and J. Vandewalle, "Bayesian framework for least-squares support vector machine classifiers, gaussian processes, and kernel fisher discriminant analysis," *Neural computation*, vol. 14, no. 5, pp. 1115–1147, 2002.
- [227] P. Sollich, "Bayesian methods for support vector machines: Evidence and predictive class probabilities," *Machine learning*, vol. 46, no. 1, pp. 21–52, 2002.
- [228] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 975–1005, 2004.
- [229] W. J. Park and R. M. Kil, "Pattern classification with class probability output network," *IEEE transactions on neural networks*, vol. 20, no. 10, pp. 1659–1673, 2009.
- [230] J. Liu, R. Seraoui, V. Vitelli, and E. Zio, "Nuclear power plant components condition monitoring by probabilistic support vector machine," *Annals of Nuclear Energy*, vol. 56, pp. 23–33, 2013.
- [231] C. K. Williams and D. Barber, "Bayesian classification with gaussian processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1342–1351, 1998.

- [232] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.
- [233] G.-B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1, pp. 155–163, 2010.
- [234] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [235] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.
- [236] H. W. Lin, M. Tegmark, and D. Rolnick, "Why does deep and cheap learning work so well?" *Journal of Statistical Physics*, pp. 1–25, 2016.
- [237] P. Tamilselvan and P. Wang, "Failure diagnosis using deep belief learning based health state classification," *Reliability Engineering & System Safety*, vol. 115, pp. 124–135, 2013.
- [238] H. Shao, H. Jiang, H. Zhao, and F. Wang, "A novel deep autoencoder feature learning method for rotating machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 95, pp. 187–204, 2017.
- [239] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1, pp. 1–39, 2010.
- [240] G. Wang, Y. Yang, and Z. Li, "Force sensor based tool condition monitoring using a heterogeneous ensemble learning model," *Sensors*, vol. 14, no. 11, pp. 21 588–21 602, 2014.
- [241] S. Binsaeid, S. Asfour, S. Cho, and A. Onar, "Machine ensemble approach for simultaneous detection of transient and gradual abnormalities in end milling using multisensor fusion," *Journal of Materials Processing Technology*, vol. 209, no. 10, pp. 4728–4738, 2009.

- [242] Q. Nie, L. Jin, and S. Fei, "Probability estimation for multi-class classification using adaboost," *Pattern Recognition*, vol. 47, no. 12, pp. 3931–3940, 2014.
- [243] J. K. Kimotho, C. Sondermann-Woelke, T. Meyer, and W. Sextro, "Application of event based decision tree and ensemble of data driven methods for maintenance action recommendation," *International Journal of Prognostics and Health Management*, 2013, vol. 4, 2013.
- [244] M. Dong and D. He, "Hidden semi-markov model-based methodology for multi-sensor equipment health diagnosis and prognosis," *European Journal of Operational Research*, vol. 178, no. 3, pp. 858–878, 2007.
- [245] N. Wang, S.-d. Sun, Z.-q. Cai, S. Zhang, and C. Saygin, "A hidden semi-markov model with duration-dependent state transition probabilities for prognostics," *Mathematical Problems in Engineering*, vol. 2014, 2014.
- [246] A. Lorton, M. Fouladirad, and A. Grall, "A methodology for probabilistic model-based prognosis," *European Journal of Operational Research*, vol. 225, no. 3, pp. 443–454, 2013.
- [247] S.-Z. Yu, "Hidden semi-markov models," *Artificial intelligence*, vol. 174, no. 2, pp. 215–243, 2010.
- [248] T. Liu and J. Lemeire, "Effective and efficient identification of persistent-state hidden (semi-) markov models.," in *STAIRS*, 2014, pp. 171–180.
- [249] M. Dong and D. He, "A segmental hidden semi-markov model (hsmm)-based diagnostics and prognostics framework and methodology," *Mechanical systems and signal processing*, vol. 21, no. 5, pp. 2248–2266, 2007.
- [250] F. Salfner and M. Malek, "Using hidden semi-markov models for effective online failure prediction," in *Reliable Distributed Systems, 2007. SRDS 2007. 26th IEEE International Symposium on*, IEEE, 2007, pp. 161–174.
- [251] M. Dong, "A novel approach to equipment health management based on auto-regressive hidden semi-markov model (ar-hsmm)," *Science in china series F: information sciences*, vol. 51, no. 9, pp. 1291–1304, 2008.



- [252] K.-R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Using support vector machines for time series prediction," *Advances in kernel methods—support vector learning*, pp. 243–254, 1999.
- [253] P.-Y. Hao, "Interval regression analysis using support vector networks," *Fuzzy sets and systems*, vol. 160, no. 17, pp. 2466–2485, 2009.
- [254] P.-Y. Hao and J.-H. Chiang, "Fuzzy regression analysis by support vector learning approach," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 2, pp. 428–441, 2008.
- [255] J. Ma, J. Theiler, and S. Perkins, "Accurate on-line support vector regression," *Neural computation*, vol. 15, no. 11, pp. 2683–2703, 2003.
- [256] B. Schölkopf, P. L. Bartlett, A. J. Smola, and R. C. Williamson, "Shrinking the tube: A new support vector regression algorithm," in *Advances in neural information processing systems*, 1999, pp. 330–336.
- [257] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [258] O. A. Omitaomu, M. K. Jeong, A. B. Badiru, and J. W. Hines, "On-line prediction of motor shaft misalignment using fast fourier transform generated spectra data and support vector regression," *Journal of Manufacturing Science and Engineering*, vol. 128, no. 4, pp. 1019–1024, 2006.
- [259] Z. Liu, M. J. Zuo, and Y. Qin, "Remaining useful life prediction of rolling element bearings based on health state assessment," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 230, no. 2, pp. 314–330, 2016.
- [260] B. Sick, "On-line and indirect tool wear monitoring in turning with artificial neural networks: A review of more than a decade of research," *Mechanical Systems and Signal Processing*, vol. 16, no. 4, pp. 487–546, 2002.
- [261] J. Dong, K. Subrahmanyam, Y. San Wong, G. S. Hong, and A. Mohanty, "Bayesian-inference-based neural networks for tool wear estimation," *The International Journal of Advanced Manufacturing Technology*, vol. 30, no. 9-10, pp. 797–807, 2006.

- 
- [262] D. F. Specht, "Probabilistic neural networks," *Neural networks*, vol. 3, no. 1, pp. 109–118, 1990.
- [263] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [264] J. Bayer and C. Osendorfer, "Learning stochastic recurrent networks," *arXiv preprint arXiv:1411.7610*, 2014.
- [265] D. Atherton, "Prediction of machine deterioration using vibration based fault trends and recurrent neural networks," *Journal of vibration and acoustics*, vol. 121, pp. 355–362, 1999.
- [266] Y. Shao and K. Nezu, "Prognosis of remaining bearing life using neural networks," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 214, no. 3, pp. 217–230, 2000.
- [267] J. Liu, A. Saxena, K. Goebel, B. Saha, and W. Wang, "An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries," NATIONAL AERONAUTICS and SPACE ADMINISTRATION MOFFETT FIELD CA AMES RESEARCH CENTER, Tech. Rep., 2010.
- [268] R. Zemouri and N. Zerhouni, "Autonomous and adaptive procedure for cumulative failure prediction," *Neural Computing and Applications*, vol. 21, no. 2, pp. 319–331, 2012.
- [269] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, 2006, vol. 1.
- [270] A. Wilson and H. Nickisch, "Kernel interpolation for scalable structured gaussian processes (kiss-gp)," in *International Conference on Machine Learning*, 2015, pp. 1775–1784.
- [271] J. P. Cunningham, K. V. Shenoy, and M. Sahani, "Fast gaussian process methods for point process intensity estimation," in *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, pp. 192–199.

- [272] R. Herbrich, N. D. Lawrence, and M. Seeger, "Fast sparse gaussian process methods: The informative vector machine," in *Advances in neural information processing systems*, 2003, pp. 625–632.
- [273] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," *arXiv preprint arXiv:1309.6835*, 2013.
- [274] Y. Saatçi, "Scalable inference for structured gaussian process models," Ph.D. dissertation, Citeseer, 2012.
- [275] C. Anger, R. Schrader, and U. Klingauf, "Unscented kalman filter with gaussian process degradation model for bearing fault prognosis," in *Proceedings of the european conference of the prognostics and health management society*, 2012.
- [276] D. An, N. H. Kim, and J.-H. Choi, "Options for prognostics methods: A review of data-driven and physics-based prognostics," in *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, 2013, 2013.
- [277] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [278] M. Dong and Z.-b. Yang, "Dynamic bayesian network based prognosis in machining processes," *Journal of Shanghai Jiaotong University (Science)*, vol. 13, no. 3, pp. 318–322, 2008.
- [279] S. Wahl and J. W. Sheppard, "Extracting decision trees from diagnostic bayesian networks to guide test selection," in *Annual Conference of the Prognostics and Health Management Society*, 2010, 2010.
- [280] G. Bartram and S. Mahadevan, "Probabilistic prognosis with dynamic bayesian networks," *Int. J. Progn. Health Manag.*, vol. 6, pp. 1–23, 2015.
- [281] J.-S. Jang, "Anfis: Adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [282] M.-H. Wang and C. Hung, "Extension neural network and its applications," *Neural Networks*, vol. 16, no. 5, pp. 779–784, 2003.

- [283] D. F. Specht, "A general regression neural network," *IEEE transactions on neural networks*, vol. 2, no. 6, pp. 568–576, 1991.
- [284] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural computation*, vol. 3, no. 2, pp. 246–257, 1991.
- [285] J.-S. Jang and C.-T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE transactions on Neural Networks*, vol. 4, no. 1, pp. 156–159, 1993.
- [286] B. Kosko, "Fuzzy systems as universal approximators," *IEEE transactions on computers*, vol. 43, no. 11, pp. 1329–1333, 1994.
- [287] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence," 1997.
- [288] W. Q. Wang, M. F. Golnaraghi, and F. Ismail, "Prognosis of machine health condition using neuro-fuzzy systems," *Mechanical Systems and Signal Processing*, vol. 18, no. 4, pp. 813–831, 2004.
- [289] Y. Lei, Z. He, Y. Zi, and Q. Hu, "Fault diagnosis of rotating machinery based on multiple anfis combination with gas," *Mechanical systems and signal processing*, vol. 21, no. 5, pp. 2280–2294, 2007.
- [290] V. T. Tran and B.-S. Yang, "Machine fault diagnosis and condition prognosis using classification and regression trees and neuro-fuzzy inference systems," *Control and Cybernetics*, vol. 39, no. 1, pp. 25–54, 2010.
- [291] B. Samanta and C. Nataraj, "Prognostics of machine condition using soft computing," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 6, pp. 816–823, 2008.
- [292] W. Wang, "An adaptive predictor for dynamic system forecasting," *Mechanical Systems and Signal Processing*, vol. 21, no. 2, pp. 809–823, 2007.
- [293] J. Liu, W. Wang, and F. Golnaraghi, "A multi-step predictor with a variable input pattern for system state forecasting," *Mechanical Systems and Signal Processing*, vol. 23, no. 5, pp. 1586–1599, 2009.

- [294] J.-S. Jang, "Structure determination in fuzzy modeling: A fuzzy cart approach," in *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on*, IEEE, 1994, pp. 480–485.
- [295] F.-L. Chung, Z. Deng, and S. Wang, "From minimum enclosing ball to fast fuzzy inference system training on large datasets," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 1, pp. 173–184, 2009.
- [296] A. Salimiasl and A. özdemir, "Analyzing the performance of artificial neural network (ann)-, fuzzy logic (fl)-, and least square (ls)-based models for online tool condition monitoring," *The International Journal of Advanced Manufacturing Technology*, vol. 87, no. 1-4, pp. 1145–1158, 2016.
- [297] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.
- [298] G.-B. Huang, L. Chen, C. K. Siew, *et al.*, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.
- [299] G.-B. Huang, "An insight into extreme learning machines: Random neurons, random features and kernels," *Cognitive Computation*, vol. 6, no. 3, pp. 376–390, 2014.
- [300] B. Frénay, M. Verleysen, *et al.*, "Using svms with randomised feature spaces: An extreme learning approach.," in *ESANN*, 2010.
- [301] Q. He, C. Du, Q. Wang, F. Zhuang, and Z. Shi, "A parallel incremental extreme svm classifier," *Neurocomputing*, vol. 74, no. 16, pp. 2532–2540, 2011.
- [302] X. Tang and M. Han, "Partial lanczos extreme learning machine for single-output regression problems," *Neurocomputing*, vol. 72, no. 13, pp. 3066–3076, 2009.

- 
- [303] M. Fernández-Delgado, E. Cernadas, S. Barro, J. Ribeiro, and J. Neves, "Direct kernel perceptron (dkp): Ultra-fast kernel elm-based classification with non-iterative closed-form weight calculation," *Neural Networks*, vol. 50, pp. 60–71, 2014.
- [304] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational learning with extreme learning machine for big data," *IEEE intelligent systems*, vol. 28, no. 6, pp. 31–34, 2013.
- [305] P. Horata, S. Chiewchanwattana, and K. Sunat, "Robust extreme learning machine," *Neurocomputing*, vol. 102, pp. 31–44, 2013.
- [306] G. A. Barreto and A. L. B. Barros, "A robust extreme learning machine for pattern classification with outliers," *Neurocomputing*, vol. 176, pp. 3–13, 2016.
- [307] K. Cao, G. Wang, D. Han, M. Bai, and S. Li, "An algorithm for classification over uncertain data based on extreme learning machine," *Neurocomputing*, vol. 174, pp. 194–202, 2016.
- [308] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: A survey," *International journal of machine learning and cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.
- [309] E. Soria-Olivas, J. Gomez-Sanchis, J. D. Martin, *et al.*, "Belm: Bayesian extreme learning machine," *IEEE transactions on neural networks*, vol. 22, no. 3, pp. 505–509, 2011.
- [310] X. Liu, L. Wang, G.-B. Huang, J. Zhang, and J. Yin, "Multiple kernel extreme learning machine," *Neurocomputing*, vol. 149, pp. 253–264, 2015.
- [311] Q. Yu, Y. Miche, E. Eirola, M. Van Heeswijk, E. SéVerin, and A. Lendasse, "Regularized extreme learning machine for regression with missing data," *Neurocomputing*, vol. 102, pp. 45–51, 2013.
- [312] Y. Miche, M. Van Heeswijk, P. Bas, O. Simula, and A. Lendasse, "Trop-elm: A double-regularized elm using lars and tikhonov regularization," *Neurocomputing*, vol. 74, no. 16, pp. 2413–2421, 2011.

- [313] A. Iosifidis, A. Tefas, and I. Pitas, "Dropelm: Fast neural network regularization with dropout and dropconnect," *Neurocomputing*, vol. 162, pp. 57–66, 2015.
- [314] A. Iosifidis, A. Tefas, and I. Pitas, "Graph embedded extreme learning machine," *IEEE transactions on cybernetics*, vol. 46, no. 1, pp. 311–324, 2016.
- [315] G. Huang, S. Song, J. N. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2405–2417, 2014.
- [316] C. K. L. Lekamalage, T. Liu, Y. Yang, Z. Lin, and G.-B. Huang, "Extreme learning machine for clustering," in *Proceedings of ELM-2014 Volume 1*, Springer, 2015, pp. 435–444.
- [317] J. Liu, Y. Chen, M. Liu, and Z. Zhao, "Selm: Semi-supervised elm with application in sparse calibrated location estimation," *Neurocomputing*, vol. 74, no. 16, pp. 2566–2572, 2011.
- [318] H. Yu, C. Sun, W. Yang, X. Yang, and X. Zuo, "Al-elm: One uncertainty-based active learning algorithm using extreme learning machine," *Neurocomputing*, vol. 166, pp. 140–150, 2015.
- [319] W.-b. Huang, F.-c. Sun, *et al.*, "A deep and stable extreme learning approach for classification and regression," in *Proceedings of ELM-2014 Volume 1*, Springer, 2015, pp. 141–150.
- [320] W. Yu, F. Zhuang, Q. He, and Z. Shi, "Learning deep representations via extreme learning machines," *Neurocomputing*, vol. 149, pp. 308–315, 2015.
- [321] S. Ding, N. Zhang, X. Xu, L. Guo, and J. Zhang, "Deep extreme learning machine and its application in eeg classification," *Mathematical Problems in Engineering*, vol. 2015, 2015.
- [322] E. Cambria, G.-B. Huang, L. L. C. Kasun, *et al.*, "Extreme learning machines [trends & controversies]," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 30–59, 2013.

- 
- [323] P. Zhang and Z. Yang, "Ensemble extreme learning machine based on a new self-adaptive adaboost. rt," in *Proceedings of ELM-2014 Volume 1*, Springer, 2015, pp. 237–244.
- [324] B. Han, B. He, R. Nian, *et al.*, "Larsen-elm: Selective ensemble of extreme learning machines using lars for blended data," *Neurocomputing*, vol. 149, pp. 285–294, 2015.
- [325] F. Fernández-Navarro, C. Hervás-Martínez, J. Sanchez-Monedero, and P. A. Gutiérrez, "Melm-grbf: A modified version of the extreme learning machine for generalized radial basis function neural networks," *Neurocomputing*, vol. 74, no. 16, pp. 2502–2510, 2011.
- [326] W. Zhang and H. Ji, "Fuzzy extreme learning machine for classification," *Electronics Letters*, vol. 49, no. 7, pp. 448–450, 2013.
- [327] Z.-L. Sun, K.-F. Au, and T.-M. Choi, "A neuro-fuzzy inference system through integration of fuzzy logic and extreme learning machines," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1321–1331, 2007.
- [328] Y. Qu, "Local coupled extreme learning machine," *Neural Computing and Applications*, vol. 27, no. 1, pp. 27–33, 2016.
- [329] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Networks*, vol. 61, pp. 32–48, 2015.
- [330] M. van Heeswijk *et al.*, "Advances in extreme learning machines," 2015.
- [331] Y. Lan, X. Xiong, X. Han, and J. Huang, "Multifault diagnosis for rolling element bearings based on extreme learning machine," in *Proceedings of ELM-2014 Volume 2*, Springer, 2015, pp. 209–222.
- [332] Q. Ye, H. Pan, and C. Liu, "A framework for final drive simultaneous failure diagnosis based on fuzzy entropy and sparse bayesian extreme learning machine," *Computational intelligence and neuroscience*, vol. 2015, p. 4, 2015.



- [333] P. K. Wong, Z. Yang, C. M. Vong, and J. Zhong, "Real-time fault diagnosis for gas turbine generator systems using extreme learning machine," *Neurocomputing*, vol. 128, pp. 249–257, 2014.
- [334] Y. Tian, J. Ma, C. Lu, and Z. Wang, "Rolling bearing fault diagnosis under variable conditions using lmd-svd and extreme learning machine," *Mechanism and Machine Theory*, vol. 90, pp. 175–186, 2015.
- [335] C. Hu, B. D. Youn, and P. Wang, "Ensemble of data-driven prognostic algorithms with weight optimization and k-fold cross validation," in *Annual Conference of the Prognostics and Health Management (PHM) Society*, 2010, pp. 10–16.
- [336] P. K. Wong, J. Zhong, Z. Yang, and C. M. Vong, "Sparse bayesian extreme learning committee machine for engine simultaneous fault diagnosis," *Neurocomputing*, vol. 174, pp. 331–343, 2016.
- [337] K. Goebel, N. Eklund, and P. Bonanni, "Fusing competing prediction algorithms for prognostics," in *Aerospace Conference, 2006 IEEE*, IEEE, 2006, 10–pp.
- [338] M. Schwabacher and K. Goebel, "A survey of artificial intelligence for prognostics," in *Aaai fall symposium*, 2007, pp. 107–114.
- [339] B. Saha, K. Goebel, and J. Christophersen, "Comparison of prognostic algorithms for estimating remaining useful life of batteries," *Transactions of the Institute of Measurement and Control*, vol. 31, no. 3-4, pp. 293–308, 2009.
- [340] S. Butler, "Prognostic algorithms for condition monitoring and remaining useful life estimation," *National University of Ireland, Maynooth*, 2012.
- [341] C. Okoh, R. Roy, J. Mehnen, and L. Redding, "Overview of remaining useful life prediction techniques in through-life engineering services," *Procedia CIRP*, vol. 16, pp. 158–163, 2014.
- [342] J. I. Aizpurua and V. M. Catterson, "Towards a methodology for design of prognostic systems," in *Annual Conference of the Prognostics and Health Management Society 2015*, 2015, pp. 504–517.

- 
- [343] D. An, N. H. Kim, and J.-H. Choi, "Practical options for selecting data-driven or physics-based prognostics algorithms with reviews," *Reliability Engineering & System Safety*, vol. 133, pp. 223–236, 2015.
- [344] A. Heng, S. Zhang, A. C. Tan, and J. Mathew, "Rotating machinery prognostics: State of the art, challenges and opportunities," *Mechanical systems and signal processing*, vol. 23, no. 3, pp. 724–739, 2009.
- [345] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, 2018.
- [346] R. I. Grosvenor and P. W. Prickett, "A discussion of the prognostics and health management aspects of embedded condition monitoring systems," in *Annual Conference of Prognostics and Health Management Society*, 2011, pp. 1–8.
- [347] S. Sankararaman, "Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction," *Mechanical Systems and Signal Processing*, vol. 52, pp. 228–247, 2015.
- [348] S. Sankararaman and K. Goebel, "Why is the remaining useful life prediction uncertain," in *Annual conference of the prognostics and health management society*, vol. 2013, 2013.
- [349] T. Whalen, "Propagation of uncertainty in systems with both probabilistic and possibilistic inputs," in *Norbert Wiener in the 21st Century (21CW)*, 2014 *IEEE Conference on*, IEEE, 2014, pp. 1–5.
- [350] E. N. and R. von Mises, "Probability, statistics and truth," *Journal of Philosophy*, vol. 36, no. 25, p. 696, 1939.
- [351] K. R. Popper, "The propensity interpretation of the calculus of probability, and the quantum theory," in *Observation and Interpretation*, S. Körner, Ed., Butterworths, 1957, pp. 65–70.
- [352] S. Sankararaman, "Uncertainty quantification and integration in engineering systems," Ph.D. dissertation, 2012.

- [353] W. Briggs, *Uncertainty: the soul of modeling, probability & statistics*. Springer, 2016.
- [354] A. Saxena, J. Celaya, E. Balaban, *et al.*, "Metrics for evaluating performance of prognostic techniques," in *Prognostics and health management, 2008. phm 2008. international conference on*, IEEE, 2008, pp. 1–17.
- [355] B. Zhang and X. Wang, "Fault diagnosis and prognosis based on lebesgue sampling," University of South Carolina Columbia United States, Tech. Rep., 2014.
- [356] H.-C. Yan, J.-H. Zhou, and C. K. Pang, "Gamma process with recursive mle for wear pdf prediction in precognitive maintenance under aperiodic monitoring," *Mechatronics*, vol. 31, pp. 68–77, 2015.
- [357] F. Di Maio and E. Zio, "Failure prognostics by a data-driven similarity-based approach," *International Journal of Reliability, Quality and Safety Engineering*, vol. 20, no. 01, p. 1 350 001, 2013.
- [358] H. Chen, "Distribution free prediction interval for uncertainty quantification in remaining useful life prediction," in *Annual Conference of the Prognostics and Health Management Society, 2013*, 2013.
- [359] D. Edwards, M. E. Orchard, L. Tang, K. Goebel, and G. Vachtsevanos, "Impact of input uncertainty on failure prognostic algorithms: Extending the remaining useful life of nonlinear systems," GEORGIA INST OF TECH ATLANTA SCHOOL OF ELECTRICAL and COMPUTER ENGINEERING, Tech. Rep., 2010.
- [360] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel, "Metrics for offline evaluation of prognostic performance," *International Journal of Prognostics and Health Management*, vol. 1, no. 1, pp. 4–23, 2010.
- [361] N. S. Clements and D. S. Bodden, "Prognostic algorithm verification," in *Annual Conference of the Prognostics and Health Management Society*, 2013.
- [362] I. Roychoudhury, A. Saxena, J. R. Celaya, and K. Goebel, "Distilling the verification process for prognostics algorithms," 2013.

- 
- [363] M. E. Sharp, "Simple metrics for evaluating and conveying prognostic model performance to users with varied backgrounds," in *Annual Conference of the Prognostics and Health Management Society 2013*, 2013.
- [364] W. M. Briggs, "The third way of probability & statistics: Beyond testing and estimation to importance, relevance, and skill," *arXiv preprint arXiv:1508.02384*, 2015.
- [365] B. Bole, K. Goebel, and G. Vachtsevanos, "Controlling tracking performance for system health management—a markov decision process formulation," *International Journal of Prognostics and Health Management*, vol. 6, 2 2015.
- [366] M. Daigle, A. Saxena, and K. Goebel, "An efficient deterministic approach to model-based prediction uncertainty estimation," NATIONAL AERONAUTICS and SPACE ADMINISTRATION MOFFETT FIELD CA AMES RESEARCH CENTER, Tech. Rep., 2012.
- [367] L. Tang, G. Rizzoni, and M. Lukas, "Comparison of dynamic programming-based energy management strategies including battery life optimization," in *2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC)*, IEEE, 2016, pp. 1–6.
- [368] "Condition monitoring and diagnostics of machines – data processing, communication and presentation – part 1: General guidelines," International Organization for Standardization, Geneva, CH, Standard, Mar. 2003.
- [369] "Condition monitoring and diagnostics of machines – data processing, communication and presentation – part 2: Data processing," International Organization for Standardization, Geneva, CH, Standard, Jul. 2007.

- [370] K. Swearingen, W. Majkowski, B. Bruggeman, D. Gilbertson, J. Dunsdon, and B. Sykes, "An open system architecture for condition based maintenance overview," in *Aerospace Conference, 2007 IEEE*, IEEE, 2007, pp. 1–8.
- [371] A. Löhr, C. Haines, and M. Buderath, "Data management backbone for embedded and pc-based systems using osa-cbm and osa-eai," in *AAIA Infotech@ Aerospace Conference*, 2012.
- [372] T. Sreenuch, A. Tsourdos, and I. K. Jennions, "Distributed embedded condition monitoring systems based on osa-cbm standard," *Computer Standards & Interfaces*, vol. 35, no. 2, pp. 238–246, 2013.
- [373] A. Löhr and M. Buderath, "Evolving the data management backbone: Binary osa-cbm and code generation for osa-eai," *European PHM Conference Nantes, France*, 2014.
- [374] R. D. Sorkin and D. D. Woods, "Systems with human monitors: A signal detection analysis," *Human-computer interaction*, vol. 1, no. 1, pp. 49–75, 1985.
- [375] N. P. Dalal and G. M. Kasper, "The design of joint cognitive systems: The effect of cognitive coupling on performance," *International Journal of Human-Computer Studies*, vol. 40, no. 4, pp. 677–702, 1994.
- [376] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," *Human factors*, vol. 39, no. 2, pp. 230–253, 1997.
- [377] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck, "The role of trust in automation reliance," *International Journal of Human-Computer Studies*, vol. 58, no. 6, pp. 697–718, 2003.
- [378] N. Diakopoulos and e. a. Friedler S. "Principles for accountable algorithms and a social impact statement for algorithms principles for accountable algorithms." (2016).
- [379] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.," *Queue*, vol. 16, no. 3, pp. 31–57, Jun. 2018.

- 
- [380] F. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning," *ArXiv e-prints*, Feb. 2017. arXiv: 1702.08608 [stat.ML].
- [381] D. Deutsch, *The beginning of infinity: Explanations that transform the world*. Penguin UK, 2011.
- [382] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 1135–1144.
- [383] J. Pearl, *Causality*. Cambridge university press, 2009.
- [384] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 1721–1730.
- [385] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, and M. Kim, "An analysis of adversarial attacks and defenses on autonomous driving models," in *2020 IEEE international conference on pervasive computing and communications (PerCom)*, IEEE, 2020, pp. 1–10.
- [386] A. P. Dawid *et al.*, "Probability, causality and the empirical world: A bayes–de finetti–popper–borel synthesis," *Statistical Science*, vol. 19, no. 1, pp. 44–57, 2004.
- [387] W. James, "Pragmatism," 1907.
- [388] P. Nieding, "Aufwandsoptimierte mechatronische simulation von werkzeugmaschinen auf basis mitwachsender modelle," Ph.D. dissertation, 2012.
- [389] A. Brandt, *Noise and vibration analysis: signal analysis and experimental procedures*. John Wiley & Sons, 2011.

- [390] A. Jablonski, T. Barszcz, M. Bielecka, and P. Breuhaus, "Modeling of probability distribution functions for automatic threshold calculation in condition monitoring systems," *Measurement*, vol. 46, no. 1, pp. 727–738, 2013.
- [391] B. Ustun and C. Rudin, "Supersparse linear integer models for optimized medical scoring systems," *Machine Learning*, vol. 102, no. 3, pp. 349–391, 2016.
- [392] R. Setiono, W. K. Leow, and J. Y. Thong, "Opening the neural network black box: An algorithm for extracting rules from function approximating artificial neural networks," in *Proceedings of the twenty first international conference on Information systems*, Association for Information Systems, 2000, pp. 176–186.
- [393] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. MÅžller, "How to explain individual classification decisions," *Journal of Machine Learning Research*, vol. 11, no. Jun, pp. 1803–1831, 2010.
- [394] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.
- [395] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., Curran Associates, Inc., 2017, pp. 4765–4774.
- [396] S. M. Lundberg, B. Nair, M. S. Vavilala, *et al.*, "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery," *Nature Biomedical Engineering*, vol. 2, no. 10, p. 749, 2018.

---

# List of Figures

2.1	Taxonomy of maintenance action . . . . .	20
2.2	Potential Failure Curve . . . . .	24
4.1	Graphical Representation of $G(\mathbf{Z})$ . . . . .	73
5.1	The PERMEATED-Framework . . . . .	102
5.2	Data Acquisition Block . . . . .	106
5.3	Data Manipulation Block . . . . .	106
5.4	State Detection Block . . . . .	107
5.5	Health Assessment Block . . . . .	108
5.6	Prognostic Assessment Block . . . . .	109
5.7	Advisory Generation Block . . . . .	110
5.8	Responsible Decision-Maker Block . . . . .	111
5.9	A simplified mechanical model of a machine tool. . . . .	112
5.10	Depiction of the model generation process. . . . .	113
5.11	Visualization of the correspondence of a peak in the frequency response function to the first bending eigenmode of the drive-shaft. . . . .	114
5.12	Visualization of the correspondence of a peak in the frequency response function to the second bending eigenmode of the drive-shaft. . . . .	115



5.13	Effect of a damaged linear bearing on the spectrum of the motor current. The damaged component is depicted in the upper figure, while the lower one shows the same machine after a repair.	116
5.14	Effect of lost stiffness in the motor bearing . . . . .	117
5.15	Effect of a worn out collision protection system . . . . .	118
5.16	Influence of temperature on the open-loop frequency response function. . . . .	119
5.17	Effect of a worse than specified attachment to the machine tool to the ground on the open-loop frequency response function. .	120
5.18	Comparison of the influence of greater than specified rack-pinion backlash to a well-adjusted backlash on the open-loop frequency response function . . . . .	120
5.19	Effect of misapplied dynamic parameters on the closed-loop frequency response function. . . . .	121
5.20	Fitted generalized extreme value distribution . . . . .	128
5.21	Confusion Matrices of an automatic threshold method and the PERMEATED Fuzzy Recommender System . . . . .	129
6.1	Receiver Operating Curves of selected classifiers . . . . .	140
6.2	Confusion Matrices of certain classifiers . . . . .	141
6.3	Confusion Matrix of the supersparse linear integer model . . .	148
6.4	The Shapely Forces of individual indicators . . . . .	155
7.1	Comparison of maintenance approaches using application-grounded metrics . . . . .	161

---

# List of Tables

4.1	Correspondence between popular regularizers in the least-squares and the probabilistic setup . . . . .	83
6.1	Excerpt of z-normalized indicators with pseudonymized equipment numbers, rounded to two decimal places . . . . .	132
6.2	Textual form of the SLIM for the described test case. . . . .	147



**University of Stuttgart**  
Germany

This publication introduces the PERMEATED framework for the diagnosis and condition monitoring of industrial assets. PERMEATED recognizes that the usability of a diagnostic system hinges critically on the trust that a responsible decision-maker, the addressee of health assessments, predictions, uncertainty quantifications and recommendations, has in its capabilities. To foster the generation of trust, PERMEATED prescribes the usage of explainable recommendations. Its usability is demonstrated by implementations as fuzzy recommender system, inherently interpretable machine-learning models and as opaque machine-learning models aided by explainers. PERMEATED's performance is validated on real-world data of various types and series of machine tools as part of a quality control process in the production line, and as support tool for service missions in the field.

 **Fraunhofer**  
VERLAG

