Quantitative Methods for Data Driven Reliability Optimization of Engineered Systems

Dissertation
zur Erlangung des Grades eines Doktors der Naturwissenschaften
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt/eingereicht von

Lukas Felsberger

29.10.2020

Erstgutachter:	Prof. Dr. Dieter Kranzlmüller
Zweitgutachter:	Prof. Dr. Rüdiger Schmidt
Tag der mündlichen Prüfung:	25.01.2021

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. .5.)

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Genf, am 29.10.2020 ....

Lukas Felsberger

Acknowledgements

In retrospect, the recent years taught me the ability to see my work, myself, and others from new and valuable perspectives. I am convinced that this will have a lasting, positive effect on my future career and personal development. This has only been possible due to people to whom I would like to express my sincere gratitude.

Prof. Dr. Dieter Kranzlmüller, for his supervision of my thesis at the Ludwig-Maximilians-Universität München and for providing the right questions and answers at the right time from the very first moment on.

Prof. Dr. Rüdiger Schmidt, for his immediate availability to review my thesis, his detailed and thought-through comments, and inspiring discussions.

Dr. Benjamin Todd, for giving me freedom to pursue my goals, the necessary support whenever needed, and defending my interests when required. Yours was the greatest contribution in making these three years a rewarding and interesting journey.

Prof. Andreas Müller, for inspiring discussions and determined support even beyond office hours.

Jan Uythoven and Andrea Apollonio, for fruitful collaborations in the Reliability and Availability Studies Working Group.

My colleagues, David Nisbet, Yves Thurel, Slawosz Uznanski, Thomas Cartier-Michaud, Volker Schramm, Arto Niemi, Jochen Schwenk, Christophe Martin, Raul Murillo Garcia, Konstantinos Papastigerou, and the whole CCE section, for sharing their expertise and opinion, which helped me developing my ideas and approaches further, in a productive yet friendly atmosphere.

The German Doctoral Student Programme, the Future Circular Collider Study at CERN, for offering and funding this interesting research project and the TE-EPC group led by Jean Paul Burnet for hosting my research in a compelling environment.

Finally, I want to express my gratitude to my parents and sister for giving me confidence in myself even if I had utterly failed in this ambitious project. In short, thank you for remembering me of the many important things outside a PhD cosmos. Thanks to my awesome friends for making the outside PhD cosmos as fun and as exciting as it can be.

Abstract

Particle accelerators, such as the Large Hadron Collider at CERN, are among the largest and most complex engineered systems to date. Future generations of particle accelerators are expected to increase in size, complexity, and cost. Among the many obstacles, this introduces unprecedented reliability challenges and requires new reliability optimization approaches.

With the increasing level of digitalization of technical infrastructures, the rate and granularity of operational data collection is rapidly growing. These data contain valuable information for system reliability optimization, which can be extracted and processed with data-science methods and algorithms. However, many existing data-driven reliability optimization methods fail to exploit these data, because they make too simplistic assumptions of the system behavior, do not consider organizational contexts for cost-effectiveness, and build on specific monitoring data, which are too expensive to record.

To address these limitations in realistic scenarios, a tailored methodology based on CRISP-DM (CRoss-Industry Standard Process for Data Mining) is proposed to develop data-driven reliability optimization methods. For three realistic scenarios, the developed methods use the available operational data to learn interpretable or explainable failure models that allow to derive permanent and generally applicable reliability improvements: Firstly, novel explainable deep learning methods predict future alarms accurately from few logged alarm examples and support root-cause identification. Secondly, novel parametric reliability models allow to include expert knowledge for an improved quantification of failure behavior for a fleet of systems with heterogeneous operating conditions and derive optimal operational strategies for novel usage scenarios. Thirdly, Bayesian models trained on data from a range of comparable systems predict field reliability accurately and reveal non-technical factors' influence on reliability.

An evaluation of the methods applied to the three scenarios confirms that the tailored CRISP-DM methodology advances the state-of-the-art in data-driven reliability optimization to overcome many existing limitations. However, the quality of the collected operational data remains crucial for the success of such approaches. Hence, adaptations of routine data collection procedures are suggested to enhance data quality and to increase the success rate of reliability optimization projects. With the developed methods and findings, future generations of particle accelerators can be constructed and operated cost-effectively, ensuring high levels of reliability despite growing system complexity.

Zusammenfassung

Teilchenbeschleuniger, wie z.B. der Large Hadron Collider am CERN, gehören zu den bisher größten und komplexesten technischen Systemen. Es wird erwartet, dass zukünftige Generationen von Teilchenbeschleunigern an Größe, Komplexität und Kosten zunehmen werden. Unter den vielen Hindernissen führt dies zu noch nie dagewesenen Herausforderungen an die Zuverlässigkeit und erfordert neue Ansätze zur Zuverlässigkeitsoptimierung.

Mit der zunehmenden Digitalisierung nimmt die Geschwindigkeit und Granularität der Betriebsdatenerfassung rasch zu. Diese Daten enthalten wertvolle Informationen für die Optimierung der Systemzuverlässigkeit, die mit datenwissenschaftlichen Methoden extrahiert werden können. Viele existierende datenbasierte Zuverlässigkeitsoptimierungsmethoden nutzen diese Möglichkeit nur unzureichend aus, weil sie zu vereinfachende Annahmen über das Systemverhalten treffen, organisatorische Kontexte für die Kosteneffizienz nicht berücksichtigen und auf spezifischen Betriebsdaten aufbauen, deren Erfassung zu teuer ist.

Um diese Einschränkungen in realistischen Szenarien zu überwinden, wird eine maßgeschneiderte Methodik vorgeschlagen, die auf CRISP-DM (CRoss-Industry Standard Process for Data Mining) basiert, um datenbasierte Zuverlässigkeitsoptimierungsmethoden zu entwickeln. Für drei realistische Szenarien nutzen die entwickelten Methoden die verfügbaren Betriebsdaten, um interpretierbare Versagensmodelle zu erlernen, die es erlauben, dauerhafte und allgemein anwendbare Zuverlässigkeitsverbesserungen abzuleiten: (1) Neuartige erklärbare Deep-Learning Methoden sagen zukünftige Alarme aus wenigen protokollierten Alarmbeispielen genau voraus und unterstützen die Ursachenermittlung. (2) Neuartige parametrische Zuverlässigkeitsmodelle erlauben die Einbeziehung von Expertenwissen für eine verbesserte Quantifizierung des Ausfallverhaltens für eine Flotte von Systemen mit heterogenen Betriebsbedingungen und die Ableitung optimaler Betriebsstrategien für neuartige Nutzungsszenarien. (3) Bayes'sche Modelle, die mit Daten aus einer Reihe vergleichbarer Systeme trainiert wurden, sagen die Zuverlässigkeit genau voraus und zeigen den Einfluss nicht-technischer Faktoren auf die Zuverlässigkeit auf.

Eine Bewertung der angewandten Methoden bestätigt, dass die maßgeschneiderte CRISP-DM-Methodik viele bestehende Einschränkungen überwindet. Die Qualität der gesammelten Betriebsdaten bleibt jedoch entscheidend für den Erfolg solcher Ansätze. Daher werden Anpassungen der routinemäßigen Datenerfassungsverfahren vorgeschlagen, um die Datenqualität zu verbessern und die Erfolgsrate von Zuverlässigkeitsoptimierungsprojekten zu erhöhen. Mit den entwickelten Methoden und Erkenntnissen kann kostengünstig ein hohes Maß an Zuverlässigkeit für zukünftige Teilchenbeschleuniger gewährleistet werden.

Contents

Chapter 1

Introduction

1.1 Motivation

CERN, High Energy Particle Accelerators and Future Challenges Particle accelerators study the fundamental constituents of matter as well as the laws which govern their interaction. The Large Hadron Collider (LHC) at the European Organization for Nuclear Research (CERN) has significantly advanced human understanding of matter by probing it at unprecedented particle collision energies of up to 14 TeV. Most notably, the Higgs Boson was discovered in 2012. It was the last missing particle of the Standard Model of particle physics, which describes most of the known universe. Yet, open questions about the behavior of the unknown universe remain. These include dark matter, the asymmetry of matter and antimatter, and neutrino masses.[1]

Particle accelerators with even higher collision energies promise to shine a light on these unresolved mysteries. Few organizations have the capabilities and experience to build such accelerators. The Future Circular Collider (FCC) study has been initiated to study options for future accelerators at the high-energy frontier at CERN. A new 80-100 km tunnel is proposed to house particle accelerators with collision energies of up to 100 TeV.[2, 3]

The operation of the LHC poses many challenges due to its complexity, its highly specialized sub-systems which are produced in low volumes, as well as the geographical expand of its infrastructure around the 27 km accelerator tunnel. The stored beam energy as well as the magnetic energy in the superconducting magnetic circuits posed novel risks for particle accelerators. New accelerators with circumferences of 100 kms and energies eight times higher than LHC are expected to set unprecedented requirements for their safe and reliable operation.

This thesis aims to develop quantitative, data-driven methods to optimize the reliability of particle accelerators and their sub-systems. A particular focus for the practical validation of the methods is put on power converters. They process and control the flow of electrical energy by supplying voltages and currents in a form that is optimally suited for electrical loads [4]. Particle accelerators consume electrical energy for their operation and contain various types of electrical loads. Among them, the powering of magnetic circuits and radiofrequency systems can represent up to 70-90% of the energy consumption [5]. As power converters are numerous and essential for operation, they impact the overall reliability of a particle accelerator significantly. Hence, they are chosen as representative sub-system to validate the developed methods, which can be applied to other systems as well.

The developed methods should help to improve the reliability of particle accelerator sub-systems despite their complexity growth at moderate additional investments to meet performance, reliability, and cost targets that make future particle accelerators projects feasible.

Reliability People, organizations and society have adopted engineered systems that function reliably. Systems that fail to function as expected have either not been adopted or abandoned quickly. Systems we use every day, such as doors, elevators, or cars, often only receive our attention when they fail to work. We have become used to them because they carry out their function or purpose as we expect it. **Reliability** is the ability of a product or system to perform as intended for a specified time, in its life cycle conditions [6].

In the widest sense, a system is a group of interacting entities that form a whole. It is enclosed by a boundary and surrounded by an environment through which it interacts through inputs and outputs. In engineering settings, a system is an aggregation of elements organized to fulfill one or several stated purposes [7]. System specifications should include its intended purposes and the conditions in which it needs to achieve them. A failure occurs when a system stops achieving its purpose despite being operated according to its specifications. E.g., a system failing during an earthquake is denoted "reliable" when its specifications exclude earthquakes as tolerable operating conditions. However, it is considered "unreliable" when it is supposed to withstand earthquakes.

Unreliability of an engineered system can principally be traced back to human activity. It may be the designer specifying wrong tolerances, the producer not manufacturing within correct tolerances, the user not operating in specified environments, or the management not providing the means and strategies to achieve the targeted reliability. This suggests that systems that always work as specified (i.e. 100% reliable) can be created in principle. However, even if all project stakeholders pay care to reliability aspects, systems can fail due to the practically unforeseeable complexity of failure mechanisms and variation inherent in natural and human processes. Hence, being able to trace back a failure to a human activity should not be confused with blaming system unreliability to human error. Instead, systems should be designed robustly to function reliably despite the possibility of human errors and unexpected environmental impacts. Then, systems can approach close to 100% reliability in practice [8].

Failures in systems, such as transportation, can compromise the safety of people. Failures in production systems, e.g. in the semiconductor industry, can cause multi-million losses due to the interruption of a global supply chain. These are direct financial consequences of failure. However, failures also cause indirect financial consequences, such as loss of consumer trust, which leads to reduced revenue in the future. Therefore, many successful organizations have identified reliability as a key enabler of long-term success.

Reliability Engineering is the technical discipline which aims to increase the reliability of systems. Its goals in order of priority are

1. to prevent or to reduce the likelihood or frequency of failures,
2. to identify and correct the causes of failures, and
3. to determine ways of coping with failures that do occur.

The order reflects the expected effectiveness in terms of minimizing cost and increasing reliability [8].

Highly reliable systems are not achieved by reliability engineers but by a concerted effort of designers, test engineers, manufacturers, suppliers, maintainers, and users. The role of reliability engineers is to support such efforts by providing effective tools, specialized training, and data-driven insights.

Economics of Reliability and the System Life Cycle The system life cycle concept includes all phases of the existence of a system from its first idea to disposal or reuse. In this thesis, the life cycle is separated in concept, design, production, field use and maintenance, and end-of-life phases.

It is generally observed that the cost to fix a problem rapidly increases until the production phase of the life cycle of a system as more and more project costs are committed. This is illustrated in Figure 1.1. It shows the cost to fix a problem (y axis) as a function of the life cycle phase (x axis). The rate of increase of the cost to fix a problem is particularly high during the design and production phases.

Therefore, it is important to ensure a high system reliability as early as possible in the life cycle. Failure to do so will result in change requests later in the life cycle with unnecessarily high costs and project delays. To ensure reliability early in the life cycle, system designers need to be able to make correct decisions. Therefore, reliability engineering is most effective when it makes the necessary knowledge, tools, and data for decision making available to system stakeholders as early as possible.

A common and established approach to achieve correct decision making in early life cycle phases is to rely upon the knowledge of experienced project stakeholders. They gather their expertise in a structured way and used it to improve new systems early in their life cycle. Such methods include Failure Mode and Effect Analysis (FMEA) [8], Fault Tree Analysis (FTA) [6], and design review [9]. These methods are based on a manual analysis of the investigated systems. This manual approach is increasingly difficult for modern complex, interconnected, and adaptive systems.

Instead of relying on manual analysis, rapid advances in information technology, computer science, and electronics allow to observe the behavior of a system directly. Data-science algorithms can automatically learn models of system behavior and derive reliability optimizations. Such improvements can help system stakeholders to follow up on the reliability of complex systems cost-effectively and improve their expertise.

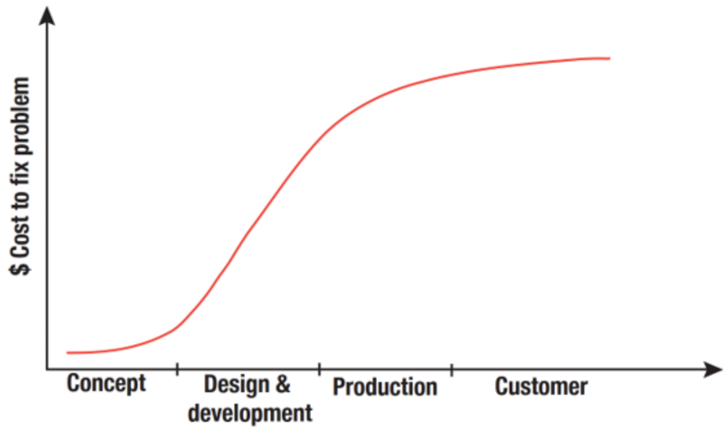


Figure 1.1: Cost of change in an engineering project throughout a system life cycle. [6]

This thesis presents a range of such methods to model system behavior, derive reliability improvements, and provide insights to system experts to help them build better systems in the future. In the following, the technological enablers of such data-driven approaches and their current limitations with respect to system reliability are discussed.

Digital Transformation Figure 1.2 shows the evolution of computing devices in terms of volume (red line and axis), price (blue line and axis), and number of installed devices (green line and axis) over the last decades (x axis). Their volume and price have decreased by more than ten (!) orders of magnitude since the 1950s and as a result, the number of installed computing devices has exploded. This has had a massive impact on the way people, organizations and societies function. The recent hardware, software, and methodological developments are discussed with respect to their impact on system reliability:

- In terms of hardware, electronics and control systems have become ubiquitous in modern machinery. More recently, an increased use of sensors makes systems a valuable source of data. Computing power has evolved with the growth of the data size it needs to manipulate. In a few decades, systems have changed from simple mechanical apparatus to internet connected, communicating, adaptive and partially autonomous entities. This drastic change provides both opportunities and challenges for system reliability, which are discussed below.

  With respect to the opportunities, system behavior and degradation can be measured at unprecedented granularity through increased sensing capabilities at reduced cost. Remote diagnostics of machinery is possible due to instant worldwide data transfer and communication. Advances in robotics promise autonomous or remotely controlled interventions in hazardous environments.

  Despite such improvements, there is a list of potential problems. Increased system complexity bears more potential failure mechanisms. Equipping systems with sensors

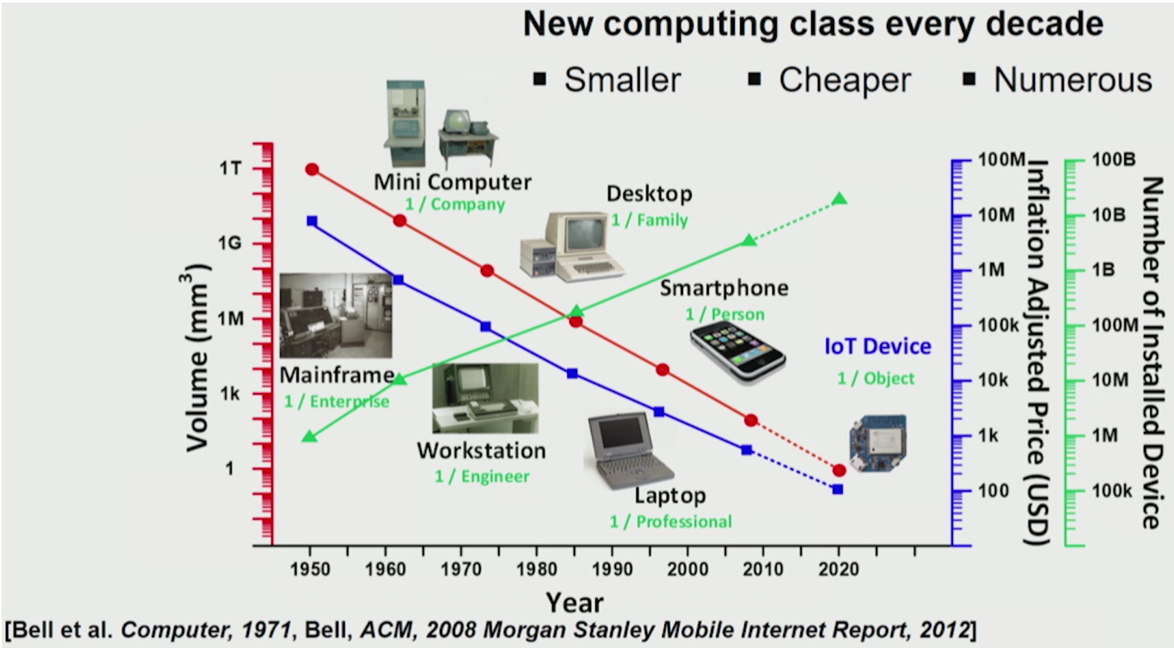


Figure 1.2: Evolution of computing devices over recent decades in terms of volume, price, and number of installed devices.

and computing capabilities increases their cost and energy consumption. Development and handling of machines that include mechanics, electronics, and software, demand an extended skill set from project stakeholders. Rapid technological change requires continuous adaption and investment.

- In terms of software, so-called Industrial Internet of Things (IIoT) platforms provide integrated frameworks of data collection, storage, visualization and analysis. They are offered by both commercial providers and as open source implementations [10]. Often, existing organizational platforms can be extended to cater for increased data collection and storage requirements. For data analysis and visualization, numerous software packages have been introduced.

  Modern data analytics platforms allow monitoring, diagnosing and predicting system behavior remotely and in real time. As for the introduction of electronics and sensing hardware, added software systems bear the risk of new system failure modes, increase system cost, and require project stakeholders to master additional skills.

- In terms of methodological advances, Artificial Intelligence (AI) and Machine Learning (ML) are increasingly popular. Methods based on deep neural networks have achieved or surpassed human performance in tasks such as image recognition or language translation [11]. Such methods benefit from a complex but versatile internal model structure and large data sets from which to detect patterns. In reliability engineering, problems are usually characterized by small data sets due to the scarcity of failures or anomalous behavior during operation and challenges in data collection. Furthermore, information is not just available in the form of quantified data but as expert knowledge, which cannot easily be 'used' by a neural network. Therefore, the benefits of neural networks are not always fully exploitable in reliability problems.

Classical ML models include Random Forests [12], Naïve Bayes [13] and Support Vector Machines (SVMs) [14]. They achieve comparable performance to deep neural networks for simple modeling tasks with small data sets at a fraction of the computational demand. Often they allow making use of expert knowledge as well as quantified data. However, deep neural networks generally outperform their modeling capabilities on large complex data sets.

Bayesian probabilistic methods provide a systematic reasoning framework for limited data scenarios and quantify confidence in the parameters and predictions they produce. However, their computational demand can exceed those of non-probabilistic methods for the same tasks.

1.2 Research Questions, Objectives and Contributions

1.2.1 Current Limitations

With the introduction of these novel technologies, maintenance optimization has become a research trend within reliability studies. Manyika et al [15] estimated that such an optimization has a global economic potential of $1.2 - 3.7 \cdot 10^9$ USD (as of 2015). Research on monitoring, diagnosing and predicting system health has arguably received a lot of attention in recent decades. The goal of such research is often to predict failures of systems precisely in time. In comparison to the traditional approach of running machinery until breakdown, this promises to avoid unexpected system downtime. In comparison to rigid time-scheduled preventive maintenance, a reduction of unnecessary interventions is expected.

Such optimization methods can lead to increased system reliability and decreased cost despite the required capital investment for sensors, data handling infrastructure and implementation. Therefore, research has focused to monitor and predict system failure using latest technologies in sensing, data analysis, and ML. However, there are three frequent limitations of such approaches.[1]

Firstly, many studies are carried out under laboratory conditions and are validated on unrepresentative data sets, if at all [16]. Moreover, the algorithmic and mathematical aspects are overemphasized, whereas, the required data collection and data quality, as well as the embedding in organizational processes receive less attention [17, 18]. Therefore, success stories of data-driven reliability improvements in industrial settings remain rare [19].

[1] A more detailed treatment of limitations of existing approaches is provided in Section 4.2.

Secondly, existing approaches often rely on detailed condition monitoring of machinery. The required sensing equipment may be expensive and can introduce additional failure modes. Given that the expected benefit is uncertain, many organizations do not want to invest in additional condition monitoring systems [20, 21]. However, most systems routinely collect operational data through their control systems, which may indirectly provide reliability information [22, 23, 24]. Methods suited for such (sub-optimal) data scenarios are underrepresented in the literature.

Thirdly, studies aim to determine the correct timing of maintenance. However, such studies can also reveal findings that can be used to improve future systems at early life cycle stages [21]. Referring to Figure 1.1, the potential cost benefit can be increased by orders of magnitude when insights lead to improved future system instead of optimized maintenance of existing systems. In other words, knowing when and how a system will fail is not very valuable in comparison to knowing how to modify an existing system or design a future system so that such failures will not happen in the first place. Yet, the focus of many current methods remains solely on determining when systems are going to fail.

1.2.2 Research Questions

The previously mentioned limitations prevent a wider adoption of data-driven reliability optimization methods despite their potential to improve the reliability of complex systems at low cost and workforce investment.

Hence, the umbrella research question (Umbrella RQ) of this thesis asks how the aforementioned limitations can be resolved to develop robust data-driven system reliability optimization methods from operational data and infer generally applicable strategies for data-driven reliability optimization. It is addressed by proposing a general methodology for the development and implementation of reliability optimization methods, which has the potential to address the mentioned limitations of existing methods.

The proposed methodology is tested by executing it for the three representative reliability optimization use cases at CERN and evaluating whether it facilitates the development of data-driven methods that reach the state of the art performance or go beyond. Each scenario is characterized by different optimization objectives, availability of data, and a priori expert knowledge. Hence, they provide strong evidence towards the Umbrella RQ but also give rise to three scenario-specific research questions, RQ1-3, which are independently addressed in the respective scenario Chapters 6-8. In this regard, the following research questions are addressed:

Umbrella RQ: How to develop robust data-driven system reliability optimization methods from operational data and infer generally applicable strategies for data-driven reliability optimization?

- RQ1: How to detect and analyze predictive failure patterns from system alarm and operational environment logging data of technical infrastructures?

- RQ2: How to optimize the life cycle cost of existing and future systems by combining expert knowledge on failure mechanisms and fault logs of a fleet of systems?
- RQ3: How to assess the most relevant factors influencing field reliability of systems based on field data and engineering documentation for groups of comparable systems?

In the following, research gap, objective and contribution are outlined for each of the research questions.

RQ1

Research Gap and Objective Systems and infrastructures are becoming more complex and connected. System experts are faced with the challenge of analyzing and understanding interdependent failure mechanisms. Explainable ML [25, 26] might provide a solution as it scales to problems of high complexity. Existing research has studied many situations using different approaches, algorithms, problem complexities, application fields. A framework for complex infrastructures, applicable to heterogeneous raw time series data, with good predictive performance and providing predictions with explanations is still missing in the particle accelerator domain.

The objective is to develop and test a fault prediction and analysis framework for particle accelerator infrastructures applicable to high-dimensional raw time series data.

Research Contribution Explainable deep learning based on raw sensor data is used to detect predictive failure patterns in systems of systems with human interaction. A proof of concept application to a particle accelerator infrastructure is provided.

Certain failures can be predicted in advance from as few as 5 training examples embedded in complex data. Non-trivial failure mechanisms (e.g. boolean logic between precursor events) can be reconstructed using the explanation mechanisms.

RQ2

Research Gap and Objective Information on system degradation behavior is often available both in collected data and expert knowledge. This requires a flexible modeling approach that can include both forms of information. Such approaches have been suggested for various application scenarios in the past. A solution for non-constant hazard rates, handling the effect of load histories, multiple failure modes, and propagation of parametric uncertainty has not been reported so far. However, such an approach is necessary to model the degradation of power converters realistically.

The objective is to present such a modeling framework and test it on a power converter use case. Life cycle costs should be evaluated and operational optimization, as well as future design improvements, derived.

Research Contribution A hierarchical model is proposed to quantify the dependency between load, stress and degradation. It can be applied to situations with and without condition, load, or environment monitoring and allows for the integration of expert knowledge. The models can be interpreted by experts and are applicable to new operating conditions and systems sharing similar components. For commonly encountered situations of limited data and knowledge, uncertainty can be quantified and propagated. In combination with a Monte Carlo simulation engine and a cost model, life cycle costs can be quantified.

Applying the framework to a power converter for which failure times, load, environment temperature, and expert knowledge on the failure mechanisms are known, a load and environment dependent failure behavior quantification is obtained. In combination with the simulation engine and a cost model, the load sharing strategy for redundant power converters with lowest life cycle costs can be determined. For repairable systems with wear-out characteristics and high downtime costs, imbalanced load sharing tends to have lower life cycle cost than balanced load sharing.

RQ3

Research Gap and Objective An accurate prediction of the field reliability of a system is desirable as it would allow to determine optimal design alternatives, required amount of spares, or expected warranty costs. The field reliability of a system is influenced by activities during all stages of a system life cycle. To predict it accurately, all processes and interactions would have to be known and quantified. Since this is impracticable, common reliability prediction methods focus on certain stages or aspects of a system life cycle. Their field reliability predictions can deviate by orders of magnitude despite requiring significant modeling efforts and often they do not quantify predictive accuracy. Accurate, uncertainty-quantifying methods that allow to integrate knowledge from different stages of a system life cycle are missing.

The objective is to develop and test a probabilistic framework to predict system field reliability more accurately based on system life cycle knowledge.

Research Contribution A new approach is taken in this work by posing field reliability prediction as inverse problem: Based on observed field reliability and life cycle descriptors of past and existing systems, statistical models of field reliability are learned. Thereby, information from all system life cycle stages can be included, uncertainty can be quantified, and it can be applied at any stage of a system life cycle. Applying the method to power converters, yields predictive models of field reliability with state-of-the art accuracy at a greatly reduced data collection and modeling effort. Using transparent Bayesian methods, the predictive uncertainty can be quantified and the importance of influencing factors can be inferred. For a use case of power converters, the most important factor was the produced number of power converters per type, which indicates that non-technical aspects may have a very strong impact on field reliability. Moreover, the descriptor is available very early in the life cycle, allowing accurate predictions early in the life cycle.

Umbrella RQ - Research Contribution

The studied scenarios cover a wide range of realistic situations in terms of data and knowledge availability, as well as choice of algorithms and methods. In Chapter 9 it is shown that for these scenarios, most practical limitations can be resolved using the tailored CRISP-DM methodology to develop data-driven reliability optimization methods that advance the state-of-the-art in the respective scenarios.

Across scenarios, it is noticed that considerable efforts have to be invested on data collection and preparation to reach acceptable levels of data quality for modeling and analysis. Time spent on data preparation could be reduced when a *Design for Reliability Data Collection* is implemented from the beginning of a system life cycle. Then, continuous data-driven reliability improvement can be used effectively to augment established reliability methods. To achieve this goal, following approaches can be recommended:

1. Design for Reliability Data Collection (Chapter 9): Systems and Processes need to be designed with data collection for reliability analysis and corresponding decision making in mind. A detailed listing of relevant data and recommendations to facilitate their collection are provided in Section 9.3. These data should be available to project stakeholders throughout the system life cycle.

2. Automatic data-driven failure pattern identification (Scenario 1 - Chapter 6): A data-driven approach to obtain predictive models of system failures and supporting information on failure mechanisms from logging data. The predictive models can help to prevent unforeseen failures and the failure mechanism information narrows the focus of in depth failure analysis for complex systems.

3. Degradation quantification and generalization (Scenario 2 - Chapter 7): A systematic approach to combine failure mechanism information, failure data, and expert knowledge into a transparent quantification of system degradation. It can be generalized to systems with different operating conditions and reused for future generations of systems.

4. Reliability prediction and effectiveness evaluation (Scenario 3 - Chapter 8): The recorded field reliability of a system can be correlated with major events during its system life cycle. For a set of comparable systems, multivariate statistical models can be obtained. They allow to estimate future systems' field reliability at early life cycle stages and provide insight on the most impacting factors on reliability during system life cycles for early strategic decision making.

These methods help to meet increasing reliability demands despite a complexity growth of modern systems in a cost-effective manner.

1.3 List of Peer-Reviewed Publications and Declaration of Authorship

Several peer-reviewed studies have been published during the preparation of this thesis. Their contribution to this thesis and the roles of the co-authors of the publications are clarified below.

Dr. Todd has been involved in all published studies except the very first one. He was the supervisor at CERN, the research institute where the experimental studies were carried out. He provided a compelling research environment for the author of this thesis by facilitating the interaction with stakeholders from CERN, pointing out valuable data sources as well as relevant ongoing reliability projects, helping out with organizational and technical matters, and providing feedback on the research projects and the written reports.

1. Felsberger, L., & Koutsourelakis, P. S. (2018). Physics-constrained, data-driven discovery of coarse-grained dynamics. Communications in Computational Physics, 25, 1259-1301.

   The methods of Chapter 7 and 8 partially reuse the Bayesian modeling approach for uncertainty quantification and propagation of the publication. Both authors have been involved in developing the presented method, discussing intermediate and final results, preparing illustrations for the thesis, and reviewing the publication draft. The numerical experiments were carried out by the author of this thesis.

2. Felsberger L., Kranzlmüller D., & Todd B. (2018) Field-Reliability Predictions Based on Statistical System Lifecycle Models. Lecture Notes in Computer Science, 11015, 98-117.

   Chapter 8 re-uses the structure, results and illustrations of the publication. The author of this thesis conceived the original research contributions, performed all implementations and evaluations, wrote the initial draft of the manuscript, and did most of the subsequent corrections.

3. Felsberger, L., Todd, B., & Kranzlmüller, D. (2019). Cost and Availability Improvements for Fault-Tolerant Systems Through Optimal Load-Sharing Policies. Procedia Computer Science, 151, 592-599.

   Chapter 7 re-uses the structure, results and illustrations of the publication. The author of this thesis conceived the original research contributions, performed all implementations and evaluations, wrote the initial draft of the manuscript, and did most of the subsequent corrections.

4. Felsberger, L., Todd, B. & Kranzlmüller, D., (2019), November. Power Converter Maintenance Optimization Using a Model-Based Digital Reliability Twin Paradigm. In 2019 4th International Conference on System Reliability and Safety (ICSRS) (pp. 213-217). IEEE.

Chapter 7 re-uses some results and illustrations of the publication. The author of this thesis conceived the original research contributions, performed all implementations and evaluations, wrote the initial draft of the manuscript, and did most of the subsequent corrections.

5. Felsberger L., Apollonio, A., Cartier-Michaud, T., Müller, A., Todd B., & Kranzlmüller, D. (2020) Explainable Deep Learning for Fault Prognostics in Complex Systems: A Particle Accelerator Use-Case. Lecture Notes in Computer Science, 12279, 139-158.

   Chapter 6 re-uses the structure, results and illustrations of the publication. The author of this thesis conceived the original research contributions and wrote the initial draft of the manuscript, and did most of the subsequent corrections. Implementations and evaluations were carried out by both T. Cartier-Michaud and the author of this thesis. A. Apollonio and A. Müller were involved in discussing results and reviewing drafts of the manuscript.

6. Felsberger, L., Todd, B., & Kranzlmüller, D. (2020). A Cost and Availability Comparison of Redundancy and Preventive Maintenance Strategies for Highly-Available Systems. To be submitted at ICSRS2021.

   Chapter 7 provides some results and illustrations of the publication. The author of this thesis conceived the original research contributions, performed all implementations and evaluations, wrote the initial draft of the manuscript, and did most of the subsequent corrections.

1.4 Structure of the Thesis

The remainder of this thesis is structured as follows. Chapter 2 gives an introduction to particle accelerators and their reliability challenges. Chapter 3 provides the necessary reliability engineering and ML backgrounds. Chapter 4 discusses existing research relevant to all research questions. Chapter 5 outlines the methodology used in this thesis. Scenario-specific research questions 1 to 3 are addressed in Chapters 6 to 8, respectively. An evaluation of the implemented methods and the general framework to address the Umbrella RQ is outlined in Chapter 9. Finally, conclusions and future research directions are presented in Chapter 10.

Chapter Learning Summary

Existing data driven methods often fail to address challenges arising in realistic reliability optimization settings and lead to few success stories. This thesis develops a methodology to resolve many limitations and executes it on three realistic use cases.

Chapter 2

Particle Accelerators

The goal of this chapter is to give the reader a better understanding of the domain in which the use cases of this thesis are situated. This should help to understand the value of particular findings of this thesis and to assess whether they could be valid for other domains that the reader is familiar with.

This chapter contains:

- An introduction to particle accelerators.
- An introduction to CERN and its Large Hadron Collider (LHC), Proton Synchrotron Booster (PSB), and Future Circular Collider (FCC) study.
- A discussion of existing and future reliability challenges for particle accelerators.
- A discussion of magnet power converters, which constitute a significant fraction of power converters in a particle accelerator. They are a representative example of power converters, which are the subject of experimental validations of this thesis.

2.1 Introduction to Particle Accelerators

Particle Accelerators use electromagnetic fields to drive charged particles to very high and very precise speeds and energies. The particles are confined in so-called particle beams with a controlled energy, phase, chromaticity, among other characteristics. These beams are required for various applications, e.g. particle physics research to study the fundamental constituents of matter, radiotherapy for cancer treatment, and ion implantation for semiconductor device fabrication.

Particle physics research aims to describe interactions of subatomic particles. It uses particle accelerators as experimental tools to provide empirical evidence for or against postulated models of particle interaction.

A common type of experiment is to collide particles with each other or against fixed targets. At sufficiently high energies, new particles are created during such collisions.

Analysis of the resulting particles allows scientists to understand the constituents of the subatomic world and the laws that govern them.

Over time, the postulated models required experimental evidence at increasingly higher interaction energies. Hence, in particle physics research there is a constant push for building particle accelerators that allow higher collision energies.

The main constituents of particle accelerators are the electrostatic or -dynamic accelerator system, the particle beam focusing and bending magnets, the beam measurement systems, and the particle collision targets or points as well as their detectors. Experiments in this thesis focus on power converters for particle accelerator operation, but can be generalized to other systems. [27]

2.2 CERN

CERN was founded in 1954 with the goal of establishing a leading fundamental physics research institution in Europe. It is devoted to pure science and aims to makes its findings accessible to everyone. Currently, 23 member states are funding CERN. It provides particle accelerators as well as supporting infrastructure for fundamental particle physics research. Its main particle physics achievements include the discovery of a range of constituents of the standard model of particle physics, culminating in the discovery of the Higgs boson in 2012 [1]. These achievements were made possible by operating a range of experiments and particle accelerators. As a side effect of building and operating such complex infrastructure, several technological innovations have been pioneered at CERN and made available to the public; most notably the world wide web.

2.2.1 LHC

The LHC is the worlds largest and most powerful particle accelerator. It is also considered to be among the most sophisticated and complex scientific instruments built to date. The project began 25 years before its operation started and it is expected to stay in service for 20 years. The word hadron describes composites of quarks, such as protons or neutrons. In its tunnel with 27 km circumference, the LHC houses approximately 1232 superconducting Nb-Ti magnetss. They are constantly cooled to 1.9 K by superfluid helium. The energy consumption of the LHC and its injectors, described below, during operation amounts to 200 MW. [28, 1]

The LHC has four particle collision points at which particle detectors are located to measure the resulting secondary particles created in collisions. The ATLAS (A Toroidal LHC ApparatuS) and CMS (Compact Muon Solenoid) experiment were built to study the Higgs Boson and supersymmetry. ALICE (A Large Ion Collider Experiment) studies collisions of heavy ions, which produce conditions similar to the first time instants of our universe. LHCb (LHC beauty) investigates the matter-antimatter imbalance. [28]

The LHC requires injection of particles at an energy of 450 GeV. To reach this energy, particles go through a chain of smaller particle accelerators (injectors) with increasing

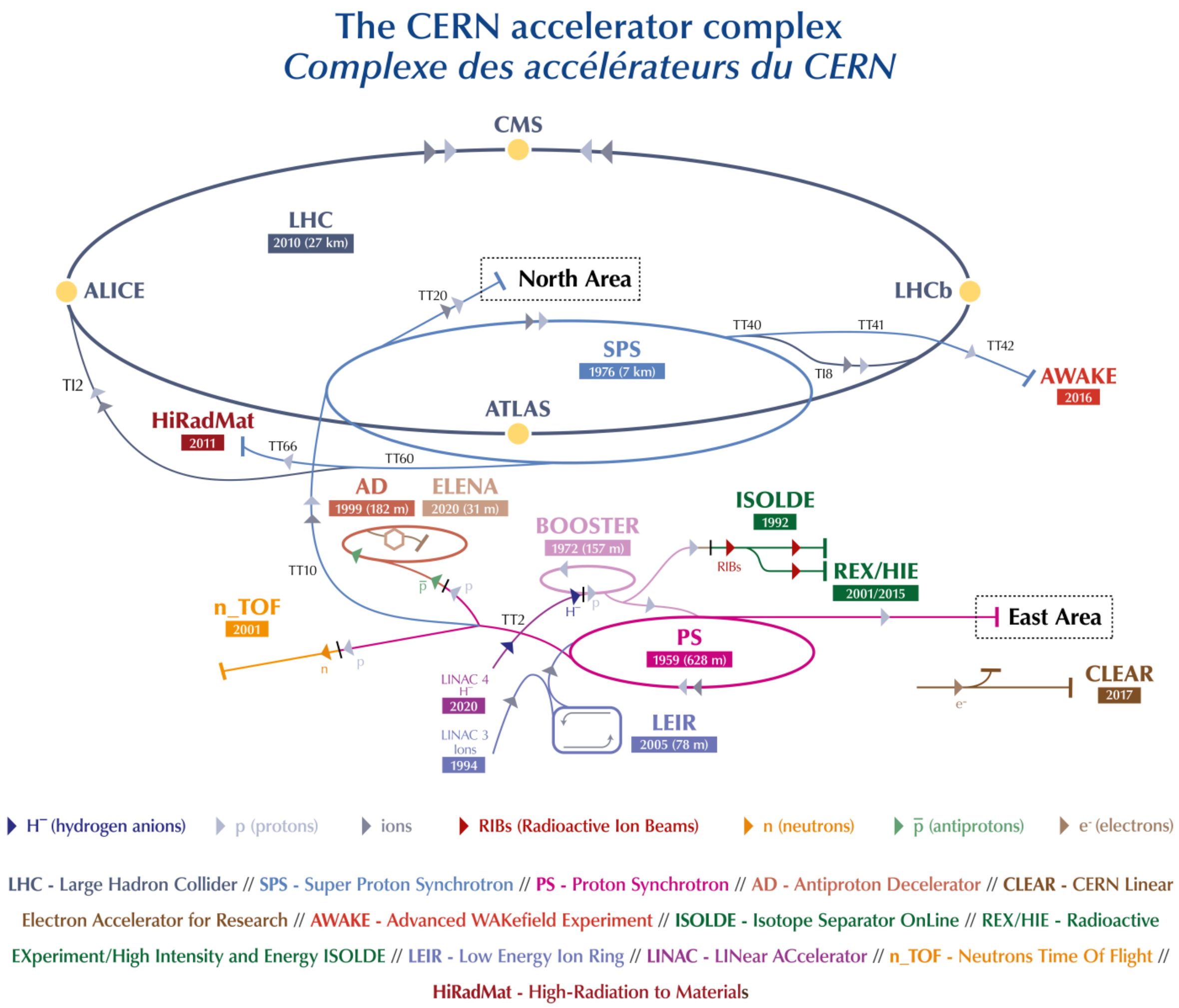


Figure 2.1: Schematic overview of the CERN accelerator complex. [29]

size and energy. Protons are accelerated by LINAC4, PS-BOOSTER (PSB), PS, and SPS before entering the LHC. Ions pass through LINAC3, LEIR, PS and SPS. This is illustrated in Figure 2.1. The solid lines in different colors depict the different particle accelerators and the arrows the direction in which the particles move. The yellow dots mark the four particle collision points of the LHC, as previously described. One can see that a vast network of particle accelerators is operational at CERN. The oldest accelerator (the Proton Synchrotron - PS) dates back to 1959 and has been continuously maintained and upgraded. For the operation of the LHC, all injectors need to be working at the same time in a coordinated manner, which poses significant reliability challenges.

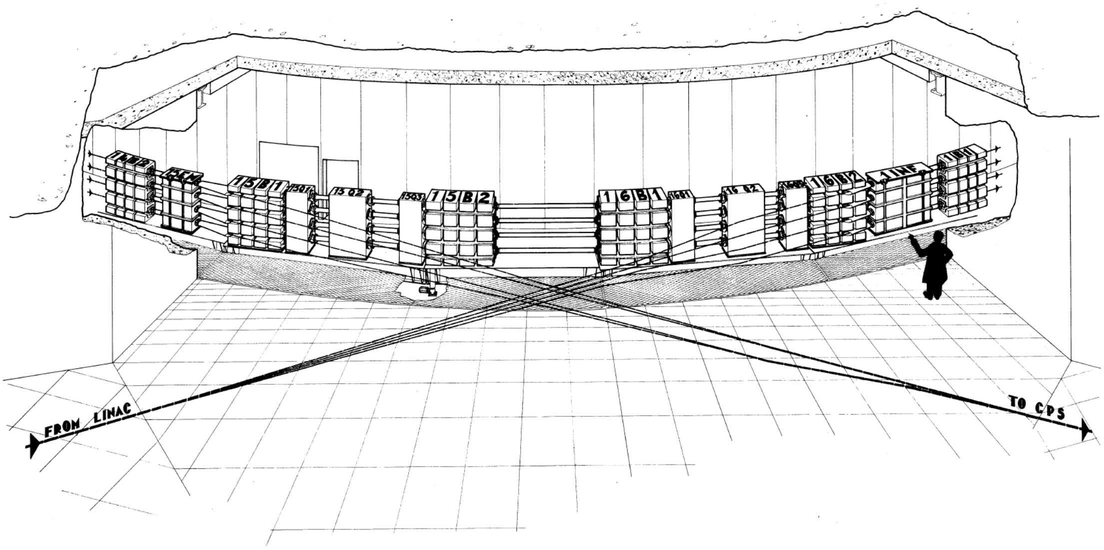


Figure 2.2: Simplified illustration of the four superimposed rings of the PSB and its beam transfer lines. [32]

2.2.2 PSB

The PSB is discussed in more detail as an example of a particle accelerator because it serves as use case in Chapter 6. The PSB accelerates protons which it receives from the LINAC4 at 160 MeV to an energy of 1.4 GeV. It is composed of four superimposed rings with a radius of 25 meters each [30]. These rings and the beam transfer lines are schematically illustrated in Figure 2.2. It shows a fraction of the full circumference of the four superimposed rings. The incoming beam from the LINAC (entering from lower left corner in Figure 2.2) is split by a series of pulsed magnets into separate beams for each of the four rings. After acceleration in the PSB, the four beams are merged again before being ejected to the PS (leaving towards lower right corner in Figure 2.2). [31]

The layout of the PSB is shown in Figure 2.3. It shows the circular particle accelerator with its 16 sections. Each section is equipped with two dipole magnets to bend the beam and a triplet composed of three quadrupole magnets to focus the beam [33].

The PSB produces different kinds of beams for a variety of experiments carried out at CERN. A beam is 'produced' within a cycle 1.2 seconds. A change of beam parameters and destinations can be executed between any two beam cycles. This makes the PSB a versatile particle accelerator. [31]

2.2.3 FCC

In 2012, the LHC has helped the discovery of the Higgs Boson [34]: the last missing particle of the standard model that describes the behavior of most of the matter of the known universe. Nevertheless, open questions on dark matter, the imbalance of matter

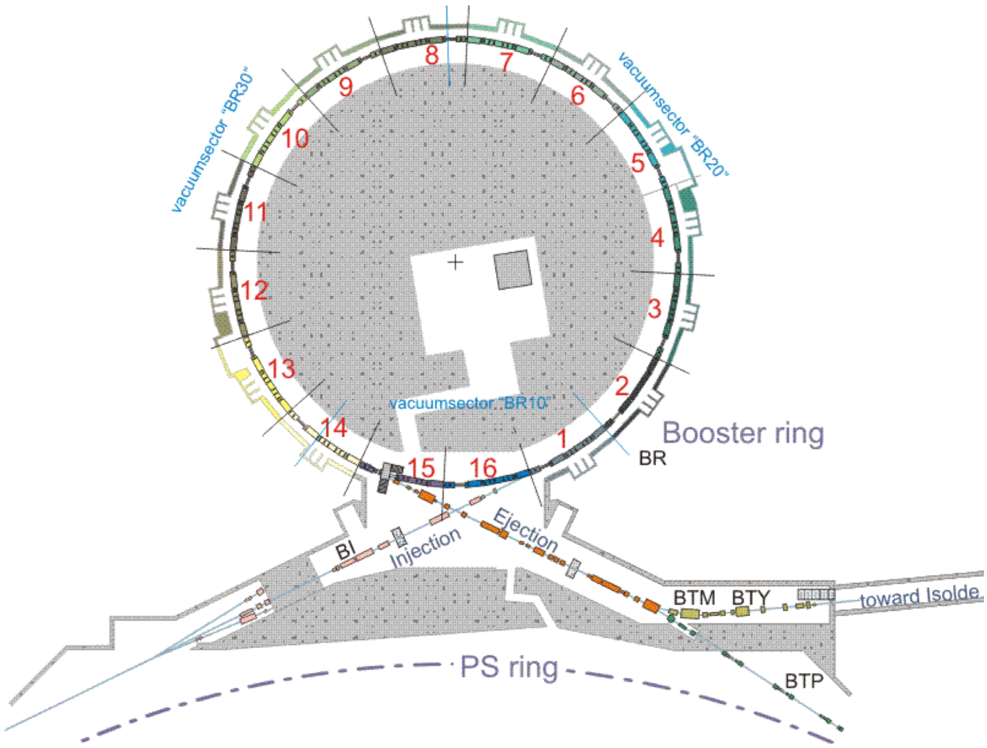


Figure 2.3: Layout of the PSB and its beam transfer lines. [33]

and antimatter, and neutrino masses remain. Pushing the energy and precision frontier by building larger and more powerful accelerators is expected to shine light on these phenomena. Following the 2013 update of the European Strategy for Particle Physics [2], the Future Circular Collider (FCC) study was launched at CERN to study options of proton and electron colliders at unprecedented energy levels as well as the required accelerator technology advancements.

Among the study's main results is the proposition of a 100km tunnel to house an electron collider (FCC-ee) which is later replaced by a hadron machine (FCC-hh) reaching collision energies of 100 TeV. The recent 2020 update of the European Strategy for Particle Physics [3] supports further in-depth financial and technical feasibility studies for an FCC.

2.3 Existing and Future Reliability Challenges for Particle Accelerators

To carry out scientific measurements effectively, it is important that particle accelerators deliver the desired beam collisions whenever required. Thus, the particle accelerator and its many sub-systems must be reliable. Considering that particle accelerators often use many complex sub-systems, which can be at technological frontiers and are produced in small quantities, achieving high reliability can be challenging.

Standard components of suppliers are often not qualified for use in particle accelerators as suppliers have optimized their products for use in the main markets, such as consumer electronics with much shorter life time requirements. Another limitation is that systems in particle accelerators are often exposed to radiation. Specialized equipment for such environments is costly and standard systems require specialized qualification campaigns for usage in radiation environments [35].

The life cycle of a particle accelerator can span several decades. Within such long periods, technologies evolve and specific expertise needs to be passed along generations of engineers. The sheer size of machines, such as the LHC, cause maintenance challenge due to the long distances to intervention sites. These factors pose further challenges in achieving high reliability.

However, there are also factors that facilitate reliability projects for particle accelerators. Many of their sub-systems are developed, built, operated, and maintained in-house. This renders reliability efforts which aim at the whole system life cycle easier in comparison to industries where systems are handed over to costumers after production and follow up on reliability is more difficult. Additionally, particle accelerator systems are often operated in environments with controlled temperature and humidity.

With the push to higher collision energies, future particle accelerators are expected to increase in size, energy, and complexity by an order of magnitude in comparison to the LHC. This generally translates to much stricter reliability requirements for each of the sub-systems of a future particle accelerator to maintain LHC availability levels. At the same time, the existing reliability challenges will be exacerbated by the increased size, complexity, levels of radiation, and further specialization of employed technologies.

If existing strategies to achieve reliable systems are maintained, the reliability goals of future particle accelerators will not be met. Hence, new methods for reliability improvement have to be investigated.

In a range of potential strategies to overcome these challenges, data-driven methods promise to improve the reliability of systems despite increases of complexity at moderate investment costs. This thesis develops quantitative reliability optimization methods to improve the reliability of particle accelerator systems by deriving reliability improvements from the operational history of existing systems. The main subject of study are magnet power converters, which are introduced in the following. The developed methods can be generalized to other kinds of systems.

2.4 Magnet Power Converters

Magnet power converters supply a specific voltage and current waveform to magnets. The controlled current in the magnets produce magnetic fields that precisely bend particle beams through the Lorentz force. A schematic overview of a power converter is given in Figure 2.4. It consists of a power part, a measurement part and a control part.

The power part receives electric power from an electric supply network and provides the power to the magnetic circuit. The input can be alternating or direct current. The output

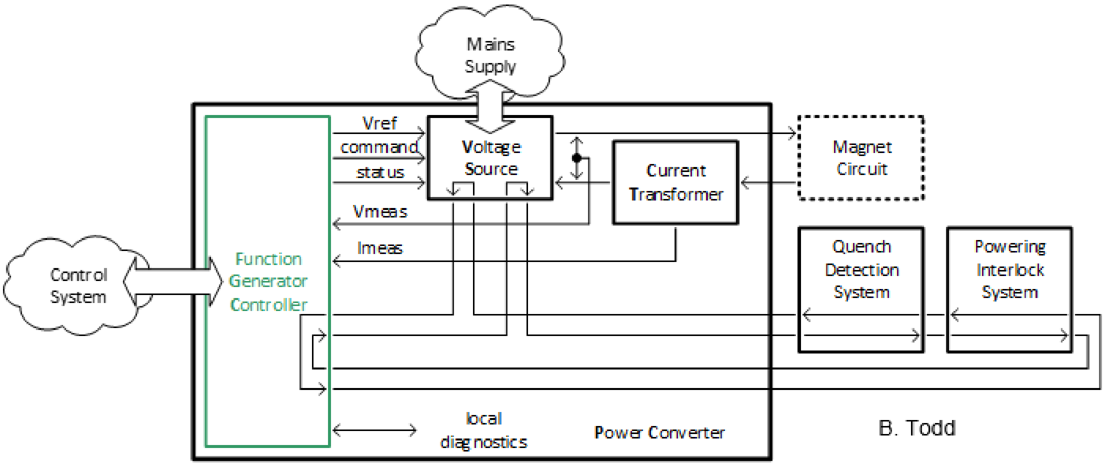


Figure 2.4: Schematic overview of a power converter.

can take any desired waveform depending on the type of power converter. The power is transformed in power electronics drive stages. The key types include power diodes, Bipolar Junction Transistor (BJT), Metal–Oxide–Semiconductor Field-Effect Transistors (MOSFET), Insulated-Gate Bipolar Transistors (IGBT), and thyristors. Heat dissipation from power electronics can require active air or water cooling systems.

The measurement part senses the actual voltage and current supplied to the magnet circuit. The control part (Function Generation Controller in Figure 2.4) receives the desired output waveform from a centralized control system and regulates the power part to obtain it at the output. It requires electronics hardware and software to translate the input signals from the control system into the desired output waveform of the power converter.

External interlock signals for machine protection and safety purposes allow shutting down the power converter operation in a safe manner. Additional magnet protection systems ensure that the energy stored in the magnetic circuit cannot cause damage. The power converter parts can be combined in dedicated racks. Configurations with line replaceable units (LRU) allow to carry out repairs by replacing faulty units. Thereby, repair times are reduced.

The control part collects monitoring and diagnostics data for the power converter. The following data are commonly collected:

- Desired and actual voltage and current.
- Warnings, which indicate an error but do not lead to shut down of the converter.
- Faults, which indicate an error and lead to immediate shut down of the converter.
- Depending on the converter type additional monitoring signals, such as the temperature, radiation levels, or current to ground, are collected.

Such operational data is used for the data-driven reliability studies presented in this thesis.

Chapter Learning Summary

The next generation of particle accelerators is expected to be more complex than existing infrastructures. Reliability approaches used for existing accelerators will not lead to a satisfying operational reliability of such future infrastructures.

Chapter 3

Backgrounds

3.1 Reliability Engineering

Reliability engineering is an engineering discipline for applying scientific know-how to a system (component, product, plant, or process) in order to ensure that it performs its intended function, without failure, for the required time duration in a specified environment [36]. A system **S** is defined as an aggregation of entities with a defined purpose. It has a range of inputs **I** and outputs **O** through which it serves its intended purpose and it is separated from its environment **E** through a boundary.

A system has a failure **F** when it fails to serve its purpose despite its environment and inputs being within specifications. The failure can be characterized by a range of properties, which are presented in Table 3.1. The first column shows the different terms to describe a failure. The second column defines the terms of the first column. The second to fifth columns show four different failure examples with their corresponding failure description.

Generally, operating systems leads to loads, which depend on system input, output and environment. The loads cause internal stresses, which can lead to failures through specific mechanisms.

In the broken phone screen example in Table 3.1, a very high load leads to overstress and immediate failure. In this case, the load history of the screen is not relevant as the failure would both occur in a new or old glass under the applied force. Contrary, in the worn car tyre example, the history of loads reduces the tyre profile through a so-called degradation or wear-out process. It is important to point out that the degradation-type of failure can be forecasted easier due to its gradual development in time.

The first two failure examples are hardware faults with mechanical loads and stresses. Hardware failures can also be driven by electrical, thermal, chemical, and other physical stresses. The third and fourth example are software failures. The failure property concepts were developed for the physical nature of hardware faults. Hence, they apply less intuitively to software failures, which are of discrete and non-physical nature. In the third example, an overstress concept still applies as the server demand exceeds its capacity. This is comparable to the mechanical force exceeding the strength of the phone screen glass in

Table 3.1: Failure concepts and definitions with hypothetical examples.

Term	Definition	Example 1:	Example 2:	Example 3:	Example 4:
Failure/Fault	System fails to serve its purpose despite its environment and inputs being within specifications	Broken phone screen	Worn car tyres	Streaming platform malfunction	Shopping website not functional
Failure mode	The observable effect of a failure	Broken phone screen glass after phone dropped on tile from less than half a meter height.	Car tyres without profile after driving less than 10000km	Video streams keep interrupting and videos are in low resolution	Shopping website does not allow to put products into shopping cart
Failure site	The location of the failure	The area where the glass is broken	Tyre profile	not applicable	not applicable
Failure mechanism	The process that leads to a failure	Mechanical overstress	Wearout due to friction	Streaming server capacity does not meet demand	Hyperlink area is located away from hyperlink text
Failure stress	The driving force of the failure mechanism	Mechanical stress	Mechanical stress	Number of streaming requests from users	Web browser and operating system of user
Failure Load	The application or environmental condition which causes stress	Force acting on screen	Contact force between car tyre and road surface	Number of streaming requests from users	Request to put product into shopping cart
Root cause (technical)	The root cause is the most basic causal factor or factors that, if corrected or removed,	Glass mounted under too high stress	Wrong mix of tyre rubber	Computing capacity is too small to meet peek demand	Failure occurs because of unexpected GUI rendering on user's operating system and web browser.
Root cause (organizational)	The root cause is the most basic causal factor or factors that, if corrected or removed,	Under cost pressure, an old glass mounting machine was adopted improperly for production of a new phone	The graphical interface of the tyre rubber mixing machine is not intuitive. The operator was not able to enter the correct values under time pressure.	Computing capacity was dimensioned to meet demand 99.9% of times due to cost pressure and efficiency requirements.	Website was tested and optimized for three most popular browsers and three most popular operating systems. Cost and time pressure does not allow extensive testing.

the first example, albeit in a reversible manner. In the fourth example, the concept of stress does not apply as the failure is due to a configurational incompatibility.

Quantitative Reliability Concepts The previous Section summarized the most important qualitative features of failures. To quantify and communicate the reliability of systems, several mathematical concepts based on continuous probability functions have been introduced.[1]

The probability that a system **S** is functional at time t is given by its reliability function $R(t) \in [0, 1]$. Likewise, the probability of having failed up to time t is given by the cumulative probability of failure (cdf),

$$F(t) = 1 - R(t). \tag{3.1}$$

For a fleet (also called population) of n identical systems, the cumulative probability of failure can be approximated by the ratio of failed systems, $F(t) \approx \hat{F}(t) = n_f(t)/n$, with $n_f(t)$ being the number of failed systems at time t.

If a system or component has multiple failure mechanisms, they can be aggregated to calculate the combined failure behavior. For independently competing failure mechanisms (i.e. one failure does not trigger another failure), the cumulative probability of failure (cdf) of a system with M different failure mechanisms, each described by separate cdfs $F_j(t), j = 1, ..., M$, is given by [37]

$$\bar{F}(t) = \prod_{j=1}^{M} F_j(t). \tag{3.2}$$

The probability that a system fails within a time increment $[t, t + dt]$ is given by its failure probability density (pdf) $f(t)dt$, which is the time derivative of the cumulative probability of failure,

$$F(t) = \int_{-\infty}^{t} f(t)dt. \tag{3.3}$$

The failure probability density can be approximated by generating a normalized histogram of the time-to-failure T of each system in a fleet.

The hazard rate $h(t)$ describes the rate of failures per time per functional systems,

$$h(t) = \frac{f(t)}{R(t)}. \tag{3.4}$$

It allows to distinguish three different failure behaviors: A decreasing, constant or increasing hazard rate. Decreasing hazard rates may occur when manufacturing errors lead to failures at the beginning of system use. This is called infant mortality. Systems with infant mortality can be screened before they are put into operation. This, so-called, burn-in removes systems from the population that would fail early. Some system exhibit constant

[1]The explanations below follow the contents of standard reliability textbooks [8, 6].

failure rates, especially due to failures of non-physical nature - e.g. improper use. Most systems will degrade and wear-out after some usage time, which leads to an increasing hazard rate.

The first moment of the failure probability density function (pdf),

$$E[T] = \mu = \int_{-\infty}^{+\infty} t f(t) dt, \tag{3.5}$$

is the Mean Time To Failure (MTTF) [8]. Naturally, the expressiveness of the mean is limited for systems with non-constant failure behavior over time. It is possible to use higher moments of the pdf to describe the behavior more accurately, or resort to parametric on non-parametric models to quantify system reliability, which are covered later in this Section.

The evolution of input, output and environment properties, as well as loads and stresses of systems over time can be expressed as (vector-)valued functions, $\mathbf{I}(t)$, $\mathbf{O}(t)$, $\mathbf{E}(t)$, $\mathbf{L}(t)$ and $\xi(t)$, respectively.

Repairable Systems The introduced quantitative reliability concepts apply to non-repairable systems. For repairable systems, the methods have to be extended to model consecutive failures in time and the repair and maintenance activities.

The MTBF is the Mean Time Between Failures [8]. It can be calculated by averaging the times a system works between consecutive failures. It can include maintenance activities, which reduce the effective number of failures. Therefore, it shall not be confused with the MTTF. The average duration of repair is expressed as Mean Time To Repair (MTTR) [8]. The system availability is given by,

$$A = \frac{time\ functional}{time\ functional + time\ nonfunctional} = \frac{uptime}{uptime + downtime}. \tag{3.6}$$

It asymptotically converges to

$$A_\infty = MTBF/(MTBF + MTTR). \tag{3.7}$$

Note that operational interruptions due to planned scheduled maintenance do not count as downtime. Downtime occurs when the system is not functional despite being expected to function.

Quantitative Reliability Modeling Reliability modeling allows to predict the system behavior as a function of any relevant factors during a system life cycle. Relevant factors usually include the operating conditions, manufacturing techniques, design choices, and component supplier selection of a system but may also consider less tangible factors such as the logistics and storage history of systems, the experience of maintenance teams, etc. For the sake of brevity, all potentially relevant factors during a system life cycle can be denoted as $\mathbf{X}$ and the reliability metrics describing system behavior as $\mathbf{Y}$. Then, the problem of reliability modeling can be expressed as,

$$\mathbf{Y} \approx \Phi(\mathbf{X}), \tag{3.8}$$

with $\Phi(\cdot)$ being the reliability model, often using a probabilistic formulation. The more accurate and the earlier in a system life cycle a the reliability model is available, the greater its potential value for driving correct decisions and cost savings, according to Figure 1.1. Finding accurate reliability models and using them for deriving general permanent reliability improvements within organizational contexts is the focus of this thesis. There are different strategies to obtain such models. A distinction is commonly made between data-driven, knowledge- (also model- or physics-) driven and hybrid approaches.

In a data-driven approach, the reliability model $\Phi(\cdot)$ is automatically inferred from observed data ($\mathbf{X}$ and $\mathbf{Y}$) using statistical and ML methods. This will be discussed in detail in a separate Section on ML techniques (see Section 3.3). For example, the profile depth of a car tyre can be measured at different mileages. Then, $\hat{X}$ would be the recorded mileage, $\hat{Y}$ would be the corresponding profile depth, and the reliability model $\hat{Y} \approx \Phi(\hat{X})$ can be obtained by regression. The obtained model can predict profile wear based on mileage. However, the model implicitly assumes a certain type of tyre, car and usage profile. I.e. it cannot be used to predict tyre wear for another kind of tyre, car or usage condition. It is only valid under the conditions in which the training data was generated. This is one of the major limitations of data-driven methods.

In the knowledge-driven approach, the reliability model $\Phi(\cdot)$ is built from first principles, based on the a priori knowledge about the system and the problem domain. For the car tyre example, a model of tyre wear can be derived with physical knowledge. It could be based on the mechanical properties of the rubber a, the weight (distribution) of the car b, the usage conditions c, and the strength of the car's engine d. The model could take the form $Y \approx \Phi(X; a, b, c, d)$. The set $a, b, c, d = \theta$ are called parameters of the model. They can be either be known from physics and domain knowledge or derived in experiments. Such a model can be used for different tyres, cars, and usage conditions as they are explicitly modeled. Hence, in principle it can be considered superior to the data-driven model of tyre wear. However, in many realistic settings, knowledge-driven models are either not available or inaccurate.

In practice, modeling methods contain both data and knowledge-driven components and can be considered hybrids. The parameter values and the model structure can be known in advance or need to be determined from measurement data. Depending on the ratio of data and knowledge-driven components in a model, they are classified respectively. The modeling approaches in Chapter 6 and 8 are mostly data-driven, whereas the method in Chapter 7 is a hybrid method.

A particularly useful and popular function for modeling reliability problems is the two-parameter Weibull distribution [38]. It models the reliability of a system with two parameters,

$$R(t) = \exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right]. \tag{3.9}$$

Its first parameter, η, characterizes the lifetime t at which 63.2% of the population of systems has failed. It is called characteristic lifetime or scale parameter. Its second parameter, β, allows to model an increasing, constant, or decreasing hazard rate when setting $\beta > 1$, $\beta = 1$, or $\beta < 1$, respectively.

Due to its simplicity, it can be useful to quickly identify whether a population of systems exhibits unusual hazard rates. However, it cannot model arbitrary fault time distributions due to its limited set of parameters. Generally, it is advisable to visualize the failure characteristics of data sets before assuming that they follow any specific parametric model.

Maintenance Strategies and Fault Tolerant Systems Maintenance is defined as any activity intended to retain or restore a system in or to a specified state in which the system can perform its required purpose [39]. This includes tests, measurements, replacements, adjustments, and repairs. It aims to maintain or increase the reliability of a system during its operational lifetime.

The historically most common approach to maintenance is to run a system until it fails and return it to a functional state thereafter. This is called reactive maintenance. Since systems may have very high downtime costs, preventive maintenance techniques have been developed to avoid failures. E.g., drive belts could be regularly replaced to avoid failures. Nevertheless, for certain systems in non-critical applications, reactive maintenance can be the most cost-effective strategy. A trivial example is a living room light bulb for which a regular replacement before a failure occurs would only increase the operational cost and bear almost no practical advantage.

In the common time or cycle-based preventive maintenance method, maintenance is carried out according to a fixed schedule. For example, the oil in a combustion engine is replaced after a fixed mileage and some airplane checks are performed after a certain amount of hours in flight. Such a maintenance strategy can increase the system lifetime and prevent unplanned breakdown. However, maintenance plans are often very conservative and result in unnecessarily frequent interventions.

It is important to know the hazard rate of the system for which maintenance is carried out. For systems with decreasing hazard rates, preventive maintenance can lead to more frequent faults because the new replacing systems exhibit a higher failure rate than the replaced ones. When hazard rates are constant, preventive replacements do not change the frequency of faults, but increase the cost.

To address these limitations, condition-based maintenance has been introduced. The idea is to measure or infer the health of a system and perform planned maintenance shortly before the system is about to fail. It promises to increase the reliability of systems in comparison to reactive maintenance by repairing or replacing systems before they fail. In comparison to preventive maintenance, it promises to decrease the maintenance costs as unnecessary actions are not carried out.

Although superior in theory, the main drawback of this method is that it requires the condition of a system to be measurable and failures predictable at a cost lower than the promised savings in comparison to reactive and preventive maintenance. This is the case for systems such as car tyres, for which the profile depth can be easily measured at a cost much lower than the tyre. However, for low cost electronic components, the cost of measuring their condition may exceed their value by far.

Condition monitoring may still be justifiable if the failure of such low cost components causes a high financial loss. However, a better strategy might be to employ some form of redundancy, which ensures a functional system despite failures of some of its components. This concept is called fault tolerance and it relies on four principles [40]: Redundancy, Fault Isolation, Fault Detection And Notification, and On-Line or Scheduled Repair. Redundancy means that the functionality of a system is distributed over several sub-systems. Then, the overall system can carry out its function despite failure of one or several sub-systems. Fault isolation requires that a failure in one sub-system cannot propagate to other sub-systems, causing a chain of failures. Protective devices, local separation, and variation of sub-systems can ensure this. Fault detection and notification implies that faults do not remain undetected when a sub-system fails and that a repair team gets notified. Finally, after the repair team gets informed, it has to carry out the repair of the failed sub-system either during operation or in a scheduled maintenance stop.

This approach allows to achieve close to 100% availability. Nevertheless, in practice such fault tolerant systems usually have design trade-offs which renders them vulnerable to certain attacks or simultaneous failures of several sub-systems. However, the probability of occurrence of such events can be reduced to a minimum for real systems. E.g., flight control computers (See e.g. [41, 42]) conform to these principles and achieve quasi-continuous availability in operation. A closer look at the robustness of biological systems reveal similar principles at even higher levels of sophistication [43].

Cost models Cost models have been developed to quantify and compare the operational cost of different maintenance and operational strategies. This thesis uses an adaptation of the models proposed by Vachtsevanos et al [44] for electronic and electrical hardware systems, which are the primary subject of investigations. It is composed of the cost of the equipment (design, development, material, and production costs) c_{eq}, the cost of repair and maintenance activities C_r, and the cost due to downtime C_d. Adding these costs, the overall life cycle cost can be expressed as,

$$C = C_{eq} + C_r + C_d, \tag{3.10}$$

in a general form. Assuming an average cost per repair of internal failures c_r and an average cost per unplanned downtime event c_d, the cost can be expressed as

$$C = n_{eq}c_{eq} + n_r c_r + n_d c_d, \tag{3.11}$$

with n_r being the number of repairs, n_d being the number of downtime events during the system lifetime, and n_{eq} the equipment cost factor to consider redundant configurations or additional costs for condition monitoring. The formula can be generalized to account for different failure modes and corresponding repair actions.

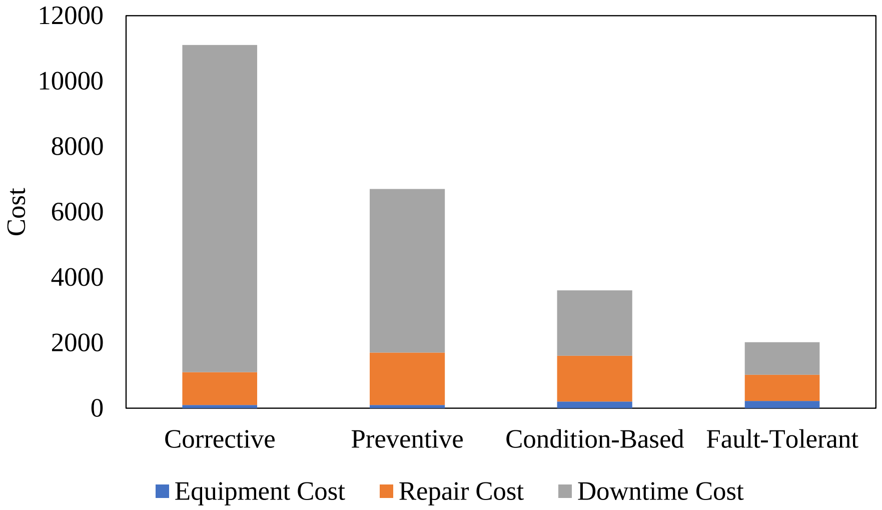
The expression can be exemplified by comparing the cost of reactive, preventive, condition-based maintenance, and fault tolerance solutions for a hypothetical power converter example. In corrective maintenance, a system is run until it fails. Hence, the number of downtime occurrences equals the number of repairs, $n_{r,corr} = n_{d,corr}$. For preventive maintenance, the system will be repaired before it fails. However, sometimes the system may fail during operation, $0 \leq n_{d,prev} \leq n_{d,corr}$. Still, the number of repairs will be larger than the number of downtime occurrences, $n_{r,prev} > n_{d,prev}$. For condition-based maintenance, the situation is similar to preventive maintenance. However, due to condition inspection, the timing of maintenance is expected to be more precise. Hence, less downtime occurrences and less repairs can be expected. The condition monitoring requires additional equipment investment for sensors or routine inspections (here the inspection costs are seen as part of equipment costs). For fault tolerance, downtime occurrences can practically be eliminated. Load sharing on several redundant systems can lead to longer lifetimes due to lower stress levels. However, the required redundancy leads to additional equipment cost.

Figures 3.1a and 3.1b show the life cycle cost for different maintenance strategies for two hypothetical power converters with low ($c_{eq,a} = 100$) and high ($c_{eq,b} = 10000$) equipment cost, respectively. The y axis shows the life cycle cost for the four different operational strategies; corrective, preventive, condition-based maintenance, and fault-tolerance, denoted on the x axis. The color code marks the cost contribution due to equipment costs (blue), repair cost (orange), and downtime cost (grey). In both cases the cost of repair c_r is 100, the cost of a downtime occurrence c_d is 1000, and the cost of condition monitoring is 100. However, depending on the equipment cost, the relative cost of the different operational strategies varies greatly. Hence, choosing the correct operational strategy depends on various cost factors as well as the failure behavior of the considered system.

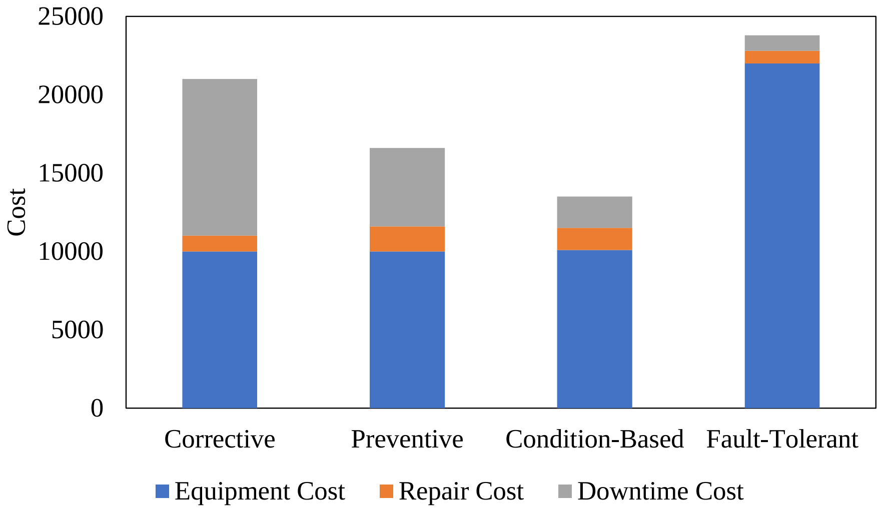
Table 3.2 lists the assumed numbers of repairs, downtime occurrences, and the equipment cost (including condition monitoring and redundancy) in each row for the four different operational strategies in each column. The equipment cost is the only factor that changes between the two scenarios. For low equipment costs, redundancy is the most cost-effective solution. For high equipment costs, condition-based maintenance is more cost-effective in this hypothetical example. Similar relations are observed in realistic settings [24, 45] and in the investigated scenario in Chapter 7.

Reliability Simulation The operation of a system can be simulated to evaluate different operational and maintenance strategies and their associated costs. This requires a reliability model (equation 3.8), an operational cost model (equation 3.11), and a simulation engine. Due to the probabilistic nature of failures and the statistical concepts describing them, Monte Carlo simulations [46, 47] are often employed as simulation methods. The principle is to carry out multiple simulations with parameters drawn from their probability distribution. The results of all simulations are collected and their statistics calculated.

Thereby, different parameters and scenarios can be studied over the expected life cycle of a system. Based on the results, the set of parameters with the best outcome in terms of life cycle cost or reliability can be recommended to decision makers. Further details of



(a)



(b)

Figure 3.1: Life cycle cost for different maintenance strategies for a hypothetical example with (a) low equipment cost and (b) high equipment cost. For low equipment cost, fault tolerance is the most cost-effective solution. For high equipment cost, condition-based maintenance is the most cost-effective solution.

Table 3.2: Example of cost parameters for different maintenance strategies for two power converters (a and b) with different equipment costs.

	Corrective	**Preventive**	**Condition-Based**	**Fault Tolerant**
n_r	10	16	14	8
n_d	10	5	2	1
$C_{eq,a}$	100	100	200	220
$C_{eq,b}$	10000	10000	10100	22000
C_a	11100	6700	3600	2020
C_b	21000	16600	13500	23800

such a simulation approach are provided in Chapter 7.

Established Reliability Methods during System Life Cycles The system life cycle has been introduced in Chapter 1, consisting of concept, design, production, field use, and end-of-life. Throughout the life cycle, engineering decisions are made and project costs are committed. The later an error, requiring a change of the system, is detected, the higher its cost. Hence, reliability issues have to be avoided by ensuring that correct decisions are made throughout.

Ideally, reliability methods guide engineering decisions from the very beginning of a system life cycle. At the same time, in early life cycle stages, limited knowledge about the system characteristics and usage are available. Successful reliability methods effectively manage this conflict by providing systematic ways to make the required knowledge for engineering decision support available early in a life cycle. In the following, a selection of such methods, based on the author's experience, common practice at CERN, and literature [6, 8, 48], is presented. It is pointed out that depending on established procedures, the optimal choice of reliability methods and used definitions may vary.

During the concept phase, the system specifications, (intended and unintended) usage, reliability requirements, as well as how to measure them, are defined. Alternative system concepts can be compared based on high-level reliability modeling and prediction. When predecessors are available, their strengths and weaknesses are assessed. Good practice is reused for the new system, whereas weak aspects are eliminated or improved.

During the design phase, more detailed information about the system emerges. Failure-Modes-and-Effect-Analysis (FMEA) [8] is carried out to identify and prioritize potential weaknesses of a system and mitigation measures based on pooling expert knowledge and experience. Based on the FMEA output, fault tolerance schemes, reliability testing, or specific maintenance strategies are further investigated for their suitability to mitigate certain identified risks. Furthermore, groups of experts carry out in-depth design reviews. Towards the end of the design phase, prototypes and their associated data recording mechanisms are available. They allow functional and stress tests, such as Accelerated Life Testing (ALT) [49]. Weaknesses are identified and resolved before a system enters mass production.

During the production phase, various quality methods ensure conformance to production specifications. Functional (reception) tests can assure correct handling during all phases of production and assembly.

During field use, operation is monitored and failures are reported. Operational faults need to be analyzed by project stakeholders and mitigation measures identified. Repair and maintenance is carried out in either a reactive, preventive or condition-based manner. Depending on the behavior of the system, maintenance and operation strategies can be adapted.

During end-of-life, reusable parts of the system are identified. Strengths and weaknesses of the system are analyzed and communicated to future systems' project teams with the goal of achieving a continuous improvement of reliability and preventing repeated mistakes. As usually a range of comparable systems are managed at different life cycle stages at the same time, insights obtained from one system can be utilized in other systems. Hence, communication of reliability insights to other project teams is advisable at all life cycle stages.

3.2 Data, Information and Knowledge

The terms data, information and knowledge are used extensively throughout this thesis. To avoid confusion, the definition by Liew [50] is followed:

- "Data are recorded (captured and stored) symbols and signal readings. Symbols include words (text and/or verbal), numbers, diagrams, and images (still &/or video), which are the building blocks of communication. Signals include sensor and/or sensory readings of light, sound, smell, taste, and touch. As symbols, 'Data' is the storage of intrinsic meaning, a mere representation. The main purpose of data is to record activities or situations, to attempt to capture the true picture or real event. Therefore, all data are historical, unless used for illustration purposes, such as forecasting.

- Information is a message that contains relevant meaning, implication, or input for decision and/or action. Information comes from both current (communication) and historical (processed data or 'reconstructed picture') sources. In essence, the purpose of information is to aid in making decisions and/or solving problems or realizing an opportunity.

- Knowledge is the (1) cognition or recognition (know-what), (2) capacity to act (know-how), and (3) understanding (know-why) that resides or is contained within the mind or in the brain. The purpose of knowledge is to better our lives. In the context of business, the purpose of knowledge is to create or increase value for the enterprise and all its stakeholders. In short, the ultimate purpose of knowledge is for value creation."

Liew explains further that the source of data and information lies in activities and situations. In the case of reliability studies, activities could be repairing a system and situations could be the condition leading to a fault of a system. These activities can be captured and stored in some database, which leads to data, and/or a human being can absorb and understand the activities and situations, recognize relationships and derive desirable actions, which leads to knowledge.

In this thesis, methods are developed to effectively combine specialized data (of engineered systems as recorded in databases) with specialized knowledge (as internalized by system experts) to extract information and new knowledge to improve the reliability of systems cost-effectively.

3.3 Artificial Intelligence and Machine Learning

In a previous paragraph on reliability modeling, the data-driven approach was introduced. It is based on inferring the relationship between relevant factors during a system life cycle $\mathbf{X}$ and reliability metrics $\mathbf{Y}$ from observed data using ML. ML is a sub field of AI. AI is any kind of intelligence demonstrated by machines. ML is the ability of algorithms to improve from experience, commonly encoded in data.

Supervised Machine Learning When the data is available in two sets, e.g. system monitoring data $\mathbf{X}$ and reliability metrics $\mathbf{Y}$, and the goal is to learn a relation $\mathbf{Y} \approx \hat{\mathbf{Y}} = \Phi(\mathbf{X})$ between the input data $\mathbf{X}$ and the target or output data $\mathbf{Y}$, the task is called supervised learning. When the target data is discrete, the problem is called classification and when the target data is continuous, the problem is referred to as regression. The following paragraphs give a practical introduction to the most relevant aspects of applying ML. Readers are forwarded to the literature for a more detailed treatment. E.g. the books by Hastie et al [51] and Geron et al [52] are a good starting point and form the basis for the following paragraphs.

Regression Contrary to the knowledge-driven approach, where a model $\mathbf{Y} \approx \hat{\mathbf{Y}} = \Phi(\mathbf{X})$ is derived from first principles, in the data-driven approach, it is learned from a training data set $\mathbf{Y_{train}}, \mathbf{X_{train}} = (y_1, ..., y_N)_{train}, (x_1, ..., x_N)_{train}$, with N being the number of samples in the data set.

This is exemplified on a hypothetical data set of car tyre profile wear. Figure 3.2 shows the training data set. The blue dots depict the measured profile depth (y axis) as a function of the mileage (x axis). The only feature of the data set is the mileage measured in kilometers and its target variable is the measured profile depth, which should be modeled. In the language of statistics, the mileage would be the independent variable and the profile depth the dependent one.

The data was obtained by performing profile depth measurements for several cars on an irregular basis. It is known that the data is collected for a single type of tyre. Only the front right wheel was measured. The car types are not known. There are only a few

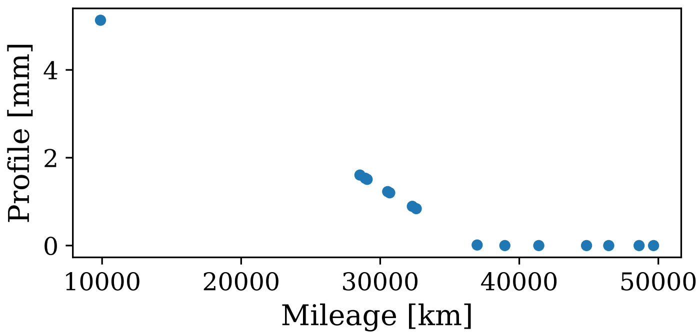


Figure 3.2: Raw collected data of tyre wear.

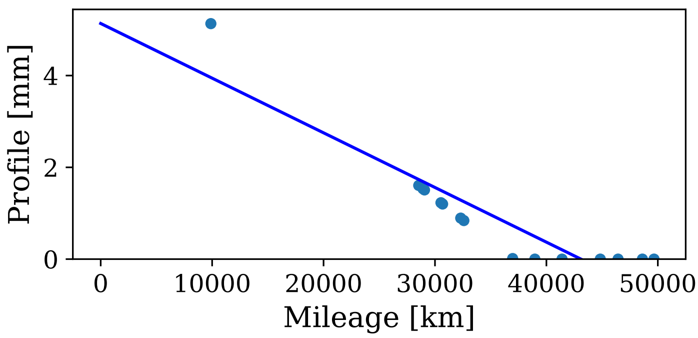


Figure 3.3: Linear fit to raw data.

data points, which is common for reliability problems. Although many ML techniques have been developed for applications with large data sets, several methods perform well on small data sets as well. In the following, a simple example of such a ML technique is introduced and the common steps in a ML project are carried out.

One of the simplest models to express the observed relation is a linear model, $\hat{Y} = \theta_0 \cdot X^0 + \theta_1 \cdot X^1 = X^T \cdot \theta$. Its parameters $\theta = (\theta_0, \theta_1)$ are obtained by minimizing the squared error, $SE(\theta) = \sum_{i=1}^{N}(y_i - x_i^T \cdot \theta)^2 = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$. The solid blue line in Figure 3.3 shows the best fit to the data obtained by the described linear regression principle.

Data Cleaning and Validation Notably, the function misses the evolution of the data. This is due to the many profile depth values that are zero. Apparently, some tyres are still used despite not having any profile left. However, this is not the process that should be modeled. Therefore, the data points with a profile depth of zero are removed from the data set.

Such preliminary visualizations and tests are referred to as data cleaning and validation. It is one of the first and most important steps in a machine learning project as ML models can only be as accurate and valid as the data that describes the process of interest.

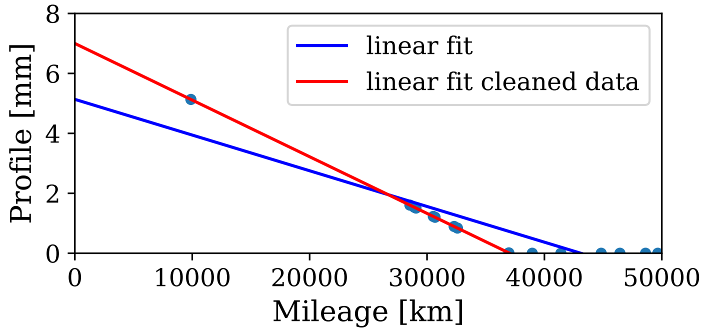


Figure 3.4: Linear fit after cleaning data. Fit to data before cleaning captured wrong trend.

Regression Metrics The red solid line in Figure 3.4 shows the best linear fit to the data after cleaning it by removing data points with a profile depth of zero. Based on visual inspection, the trend is better captured using the cleaned data set only. To quantify the error between the model and the data, error metrics are used, such as the squared error (as used for regression above),

$$SE = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{3.12}$$

the mean-squared-error,

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{3.13}$$

or the mean-absolute-error

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|. \tag{3.14}$$

For the model fits in Figure 3.3, an MSE of 0.267 is obtained on the training data set for the linear model before cleaning. After cleaning the MSE drops to 0.038, as is also visually evident in Figure 3.4, which shows the fit after cleaning the data. This confirms that the fit has improved after cleaning the data.

Overfitting The error obtained on the training data is not a good indicator for the quality of a model. If a high order polynomial model is fitted to the training data, it achieves an MSE of 0.006, which is better than the linear model.

Looking at the fit in Figure 3.5, it is obvious that it does not accurately capture the relation between mileage and profile degradation. The polynomial fit oscillates strongly between the data points. This process is called overfitting. The opposite extreme would be to choose a constant function, e.g. the mean, $\hat{Y} = 1/N \sum_{i=1}^{N} y_i$, and is called underfitting.

For such a simple example with only one feature it is easy to pick the right complexity of a function by visualizing the fitted curves. However, for high-dimensional problems, it

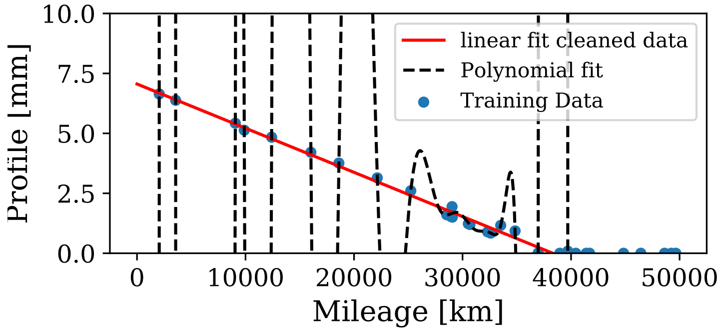


Figure 3.5: High order polynomial function achieves lower MSE than linear fit. However, it does not capture the trend correctly: Overfitting

is not possible to judge based on visual inspection. A general approach to select the most appropriate model for a data set is presented in the following.

Model Comparison and Performance Estimation To estimate the performance of a model, the available data is split in several sets. Even before the data is explored and inspected, a test set is separated and not used until the final evaluations of the model.[2] Only when no information about the test set is available during model development, an unbiased estimate of model performance can be obtained. All the data exploration, cleaning, and model development is carried out with the remaining data set. Since often several different models (linear model, polynomial model of third order, and polynomial model of n-th order) are trained, they need to be compared. Therefore, the remaining data set is further split in training sets and evaluation sets.

A popular choice for small data sets is K-fold cross-validation. It splits the data set into K equally sized parts or folds. Then each of the K folds is used for model evaluation and the remaining K-1 set for training. Thereby, mean and variance of K validation errors can be obtained, which allows a comparison of different models.

For data without sequential order or time dependence, the data can be shuffled before being split in different sets. If the data is sequential or time-dependent, the order of the data should be maintained for unbiased performance estimates of the trained models. E.g., when the data are time series of the extend of a glacier from 1980 to 2010, the test set should be the time series after a specific year and the training and validation sets should be the time series before that specific year. Shuffling the data before splitting would introduce a bias.

Bias More data, $\mathbf{Y_{new}}, \mathbf{X_{new}}$, has been collected for the car tyre example. It is plotted together with the formerly used training data in Figure 3.6. Apparently, the new data shows

[2]The previously carried out steps should have only been carried out after a test set has been separated.

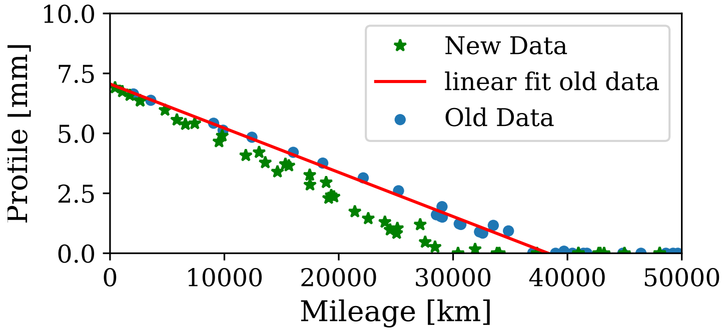


Figure 3.6: Evaluation of linear fit on cleaned data on newly arrived test data. Apparently, data collection for initial data was biased.

a different relation between mileage and profile wear in comparison to the previously used old data.

Further investigations reveal that the old data has been collected from the countryside with particularly little traffic and no highways, whereas, the new data has been collected in a metropolitan area. It seems that the linear model obtained on the data from the countryside has a very limited validity outside the countryside.

This is an example of data collection bias. It reflects that data-driven models are limited by the quality of the data they are trained on. It can be addressed by careful attention during data collection and precise documentation of the limits of data-driven models. The topic is further discussed by Baer [53].

Explainable AI Besides predicting target variables, ML models can also be used to gain insight into problems. Formally, Explainable AI aims to make predictions by AI solutions be understood by human experts [25]. For the car tyre example, a more detailed understanding of profile degradation and its influencing factors is required.

Based on a limited a priori physical understanding, it is decided to additionally collect the strength of the vehicle, as well as the drivers' behavior using acceleration monitors fitted to the cars. Hence, the data has now three features and one target variable: the engine strength, the driver behavior (average acceleration normalized to one), the mileage, and the measured profile depth. The data was collected for 20 drivers. Their profile depth and mileage were measured 15 times for each driver. This results in a data set of dimension $(20, 15, 4)$.

To obtain an unbiased estimate of the model performance, a test set of five drivers is separated from the collected data. All the following steps are only carried out on the remaining data set. Figure 3.7 shows the input and output data after data items with a profile depth of zero were removed as for the previous example. On the diagonal, histograms of each feature are plotted. On the off diagonal plots, each variable is plotted as functions of each other variable. There is a clear correlation between mileage and profile depth. For other variables, no clear dependence is visible.

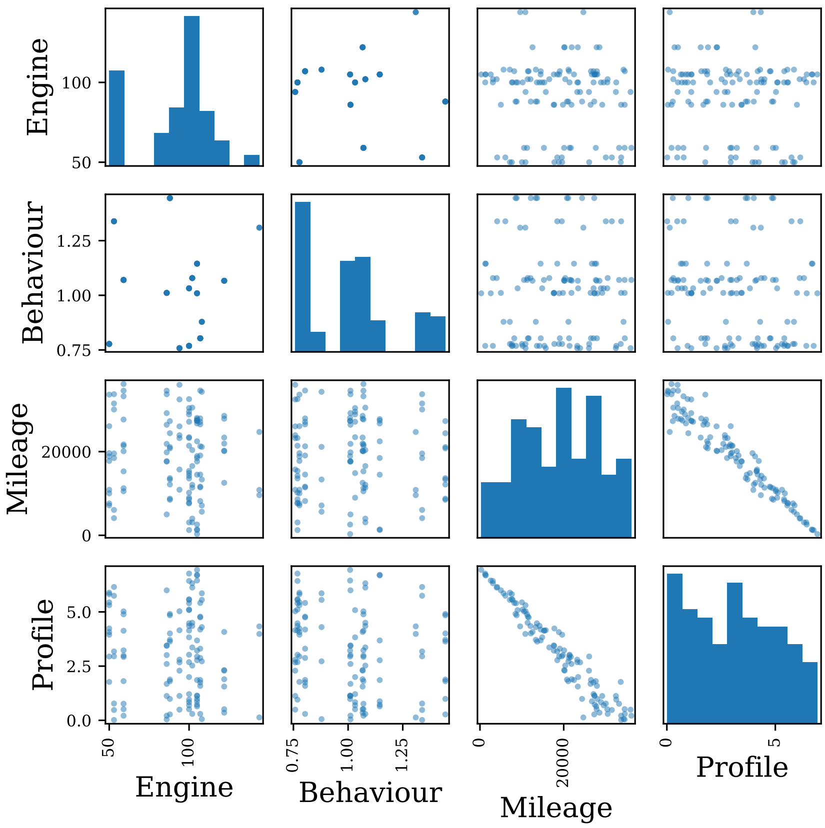


Figure 3.7: Scatter plot of new extended multivariate data set, containing engine strength, driver behavior, driven mileage, and tyre profile depth as variables. Histograms of each variable the diagonal. Scatter plots of all combinations of any two variables on off-diagonals. Clear dependence only visible for profile depth as function of mileage driven.

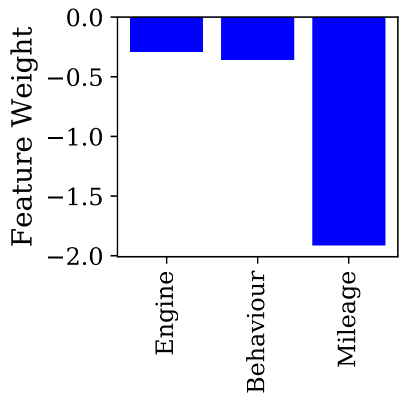


Figure 3.8: Feature weights (θ_1) of the linear multivariate model fitted to training data. Learning a model based on all features improves the predictive performance of the model. Hence, all features are considered relevant. The mileage is the most important feature.

A first attempt to quantify the dependence is to fit a multivariate linear model, $\hat{Y} = \theta_0 + X^T\theta_1$ to the data and inspect its parameters θ_1, which reveal the relative importance of the features. When trained on the data set, the multivariate model has an MSE of 0.09 on the test data set. The univariate model, which only considers the mileage as feature, achieves an MSE of 0.27. Hence, the model improves when using the additional features.

The relative importance of each feature, θ_1, is plotted in Figure 3.8. The importance (feature weight) is plotted on the y axis and the features on the x axis. The mileage remains the most important feature, but the driving behavior and engine strength are relevant too. However, based on the feature weights it cannot be concluded that driving behavior and engine strength are among the causal factors of tyre wear. This can only be judged by having a physical understanding of the processes leading to tyre degradation. Still, the outcome of the explainable model can guide the search for the causal physical processes.

This trivial example shall demonstrate the idea of explainable AI. In realistic settings, problems may have thousands of features and ML models are nonlinear and have thousands of parameters. In such a so-called black box scenario, it is much more difficult to interpret predictions of the model. The usefulness of explanations can be assessed within the organizational context that they are used in. They are useful if they help stakeholders in effective and timely decision making. Holzinger et al [54] proposed a system causability scale to evaluate the usefulness of explanation methods in use. Samek et al give a general overview of the recent state of the art in explainable AI [55]. An application of such methods is presented in Chapters 6 and 8.

Classification The car tyre problem can be reformulated by defining a profile depth of 0.5 mm as the lower limit for an acceptable tyre. This transforms the binary variable from being continuous to being discrete. Instead of the profile depth, a model can predict whether the car tyre is acceptable or not. This is a (binary) classification problem. The

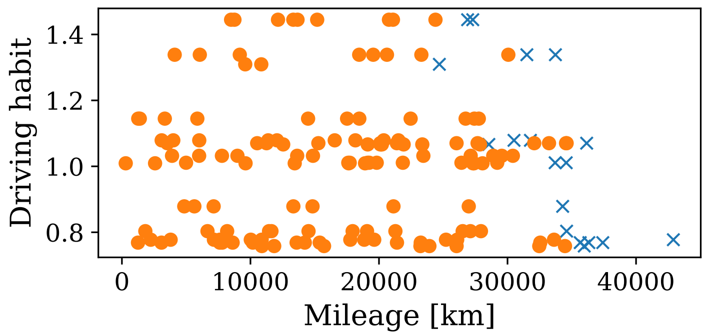


Figure 3.9: Reformulation of problem as classification problem by defining a threshold for acceptable car tyre profile depth. A classifier aims to find a separation boundary between class 'acceptable' (orange dots) and 'not acceptable' (blue crosses).

transformed data of the car tyre example is plotted in Figure 3.9 in the plane spanned by the mileage and driving behavior features. The orange dots mark the acceptable profile depths and the blue crosses not acceptable profile depths. There is no clear separation boundary visible between the two classes, which makes it difficult to obtain an 'accurate' classifier based on the mileage and driving behavior features.

Classification Metrics Classification problems require different metrics than regression problems. A frequently used metric is the accuracy. It is the fraction of correct predictions among all predictions made by a model,

$$accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions}. \tag{3.15}$$

However, the car tyre data set can be used to show a major drawback of the accuracy metric. A trivial model that always predicts the tyre to be acceptable, achieves an accuracy of 0.94 for the data collected. This appears to be very good, but is clearly the wrong model. The reason for the high accuracy is that the 'acceptable' class is far more frequent in the data set than the 'not-acceptable' class. Such a situation is called imbalanced data set.

The so-called confusion matrix is a more robust way to measure classification performance. It counts the number of true positive (TP, not acceptable tyre classified as not acceptable), true negatives (TN, acceptable tyre classified as acceptable), false positives (not acceptable tyre classified as acceptable), and false negatives (not acceptable tyre classified as acceptable) classifications in a matrix.[3] The trivial model, which would classify all tyres as acceptable, would achieve 0 TPs, 136 TNs, 8 FPs, and 0 FNs. Despite not having classified any worn tyre correctly, this yields a high accuracy.

Precision, recall, and F1 score are better suited metrics for imbalanced problems. Pre-

[3]Note that here positive corresponds to the 'not acceptable' class, which might seem counter-intuitive.

cision is defined as the ratio of TPs over all positively classified items,

$$precision = \frac{TP}{TP + FP}. \tag{3.16}$$

Recall is the fraction of TPs selected from all actually positive items,

$$recall = \frac{TP}{TP + FN}. \tag{3.17}$$

The harmonic mean of precision and recall is the F1 score,

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{3.18}$$

The trivial classifier would achieve a precision, recall and F1 of 0.

Imbalanced Learning In reliability problems, it is common to have imbalanced data sets, e.g. less examples of failed than functioning systems. It is often more important not to classify an erroneous system as functioning (FN) than vice versa (FP). In such cases, class weights can be assigned to reflect that importance. Alternatively, so-called re-sampling strategies can be employed. Thereby, either the more frequent class can be reduced (down-sampled) or the less frequent class can be expanded (up-sampled) until the desired class balance is obtained. In the simplest case, random removal or duplication of data items is employed. More complex sampling strategies, such as SMOTE [56], synthetically generate data items which are similar to existing data items by using certain generation rules. For data accessible to human intuition, such as images or sound, it can easily be judged whether such rules are appropriate. E.g., for an image of a dog, a slight rotation or stretching of the dog does not turn it into a cat. However, for more complex data, such as system monitoring data, it cannot be judged upfront whether certain transformations of the data change its information content. Therefore, advanced sampling strategies need to be employed with caution in complex data scenarios. An overview of strategies for imbalanced data sets is provided by He et al [57] and an example of a sampling strategy for complex system monitoring data is provided in Chapter 6.

Learning Summary

Reliability Engineering is the technical discipline which aims to improve the reliability of systems. Recent developments in data science have the potential to lead to methods for the cost-effective reliability optimization of systems.

Chapter 4

Literature Review

To gain an understanding of the state-of-the-art in the domain of data-driven reliability optimization, a literature review is carried out. This chapter addresses the relevant literature shared by all research questions introduced in Chapter 1. Literature uniquely related to each of the three scenario-specific research questions are presented separately in Chapters 6-8, respectively.

An overview of the relevant research areas is given in Section 4.1. This includes the area of analytics, reliability optimization, and prognostics. The section concludes with a definition of data-driven reliability optimization and discusses its relation to the other relevant research areas mentioned.

In Section 4.2, current limitations of data-driven reliability optimization are discussed with a focus on the development, implementation, and adoption of such methods in organizational contexts.

4.1 Literature Overview

Data-driven reliability optimization as technical term is not frequently used in the reliability studies literature. Therefore, this section provides an overview of related research areas as well as a definition and demarcation of data-driven reliability optimization to other relevant research areas, such as prognostics in reliability engineering and the multidisciplinary analytics field.

Analytics Nelson [58] defines analytics as 'a comprehensive, data-driven strategy for problem solving'. Its precise definition is under debate [59]. Generally, it can be viewed as the connecting tissue between data and decision making and is a sub-field of the vaguely defined data sciences.

The topic is divided in descriptive, predictive and prescriptive analytics, which aim to describe, predict or provide ways to change the outcome of a process, respectively. Analytics builds upon use of mathematics, statistics, data, and expert knowledge to create any form of benefit.

Reliability Optimization Reliability optimization aims to optimize an objective function given decision variables and constraints. The objective function usually represents reliability or cost. The decision variables can be tuned to maximize the objective function. Examples include system configuration, component choice, and redundancy allocation. The constraints represent boundary conditions, such as physical, cost, or reliability constraints. [60]

Reliability optimization has been established in the mid of the twentieth century. It has evolved from static and exact to dynamic and approximate solutions for optimization problems to better reflect practical needs. Future challenges involve the integration of continuous streams of data in modern interconnected systems, the ability to adapt to changing systems and environments, and the accounting for uncertainties in the optimization process. [60, 61]

Prognostics Prognostics in engineering sciences is defined in ISO13381-1 as 'an estimation of time to failure and risk for one or more existing and future failure modes' [62]. As such, it allows to perform predictive maintenance which leads to reduced operational costs of systems.

Defining Data-Driven Reliability Optimization Data-driven reliability optimization is the application of predictive and prescriptive analytics to reliability studies. Its goal is to increase the reliability of systems and decrease their life cycle costs. It shares the goals of reliability optimization and tackles its aforementioned open challenges with modern analytics approaches.

(Data-driven) Prognostics can be classified as data-driven reliability optimization with a narrower objective of forecasting the precise end of life of certain systems and provide strategies to deal with it. As explained in Chapter 1, the biggest potential for reliability improvement and life cycle cost savings is at the beginning of a system life cycle. However, prognostics acts at the end of the life cycle, which limits its potential benefits.

Data-driven reliability optimization aims to enable problem solving and decision making at all stages of the system life cycle, particularly at early stages when the potential benefits are largest.

Quantitative data-driven reliability optimization uses quantitative methods to achieve these goals. It is the primary topic of this thesis.

4.2 Existing Limitations of Data-Driven Frameworks for Reliability Optimization

In this section, current limitations of data-driven reliability optimization are discussed with a focus on the development, implementation, and adoption of such methods in organizational contexts. A large fraction of available literature on data-driven reliability optimization is focused on the sub-fields of prognostics and predictive maintenance. Since the general approach of prognostics overlaps with that of data-driven reliability studies, except for having narrower objectives, it is possible to generalize limitations from prognostics and predictive maintenance literature. Hence, the limitations outlined below are based on a meta-review of survey and review articles in the fields of prognostics and general reliability engineering. These limitations are the inability of current methods to deal with the complexity and uncertainty of realistic settings, that infrastructures cannot meet the minimal requirements current methods pose, that organizations can not meet minimal requirements of current methods, and that the focus of current methods on predicting remaining useful life is too narrow.

Complexity and Uncertainty in Realistic Reliability Problems Zio [63] states system modeling and representation, model quantification, and uncertainty-quantification of model and system behavior as the main challenges of reliability engineering. Increasing functional requirements of modern systems often lead to systems at the frontier of technology and complexity. Failures can always occur as emergent behavior when multiple such complex systems interact in dynamic environments. Good model representations and quantification for such failure behavior are almost impossible to provide. Reliability optimization techniques have to continually evolve to match the complexity requirements of modern interconnected systems [60].

Data-driven reliability optimization methods struggle to keep up with the pace of technological developments. Existing studies are mostly carried out on a component level, often considering only single failure modes. However, in reality systems are composed of many components and are affected by multiple, often overlapping failure modes and mechanisms [17, 64, 65]. Many studies do not account for the dependence of failure mechanisms on multiple factors [66]. Dynamic environments and unforeseen inputs can often not be robustly handled by data-driven methods [19]. Degradation does not only happen during operation of systems. Systems are exposed to stresses during transport, storage and installation, which can remain unobserved by monitoring systems.

The lack of suitability of developed methods for realistic problem settings is exemplified in an informal study by Hodkiewicz et al [16]. Of all 64 published papers in IEEE Transactions on Reliability in 2013 which proposed reliability models, they found that only in 7 papers it was attempted to validate the developed methodologies with field data including any description of the data collection.

Uncertainty in reliability problems arises from several sources. The limited knowledge of the future usage of systems poses an irreducible uncertainty. Reducible uncertainties arise due to the limited available data of system degradation which is expensive to generate, the limited understanding of failure mechanisms, and the limited availability of accurate modeling options [64]. Therefore, an uncertainty-quantification and propagation throughout all steps in reliability modeling is essential. Some methods are suited for handling of uncertainty whereas others, such as neural networks, are less suited [67, 17]. The limitations of existing data-driven methods make industrial success stories rare [19, 17].

Infrastructures cannot Meet Minimal Requirements of Methods Proposed in Literature Methods proposed in the literature frequently require a well-functioning monitoring of equipment and existing data sets of run to failure data. However, many organizations cannot meet these requirements [17]. Especially the required sensors and associated data acquisition systems for monitoring are expensive to install in existing and new machinery. Moreover, dedicated sensing equipment may be unreliable itself, can require maintenance, and needs regular calibration. Given that the expected benefits of such investments are not certain, many organizations do not install dedicated sensor systems for reliability purposes [20, 21].

However, most machinery already log operational data through their control systems. This data may provide condition monitoring and reliability information at no additional cost. Some successes using such data have already been reported [23].

Organizations can not Meet Minimal Requirements of Methods Proposed in Literature Organizations face severe challenges when choosing and implementing data-driven reliability optimization methods. Existing studies have identified several reasons. Sikorska et al [17] and An et al [68] criticize that existing review papers focus on mathematical aspects of different methods instead of the value of the methods for reliability optimization in the respective domain context. Furthermore, Sikorska et al suggests that approaches proposed in the scientific literature are not developed by problem solvers but by mathematicians with the desire to fit models to problems [17]. Elattar et al [69] and Nguyen et al [65] point out that there is a lack of standardized approaches which would help practitioners navigate the vast choice of options.

Tiddens et al [22] showed that practitioners choose the methods for reliability optimization based on the experience of project stakeholders or other companies and availability of ready-to-use implementations instead of systematically choosing an appropriate approach from the beginning. This often leads to an expensive trial and error approach. Furthermore, objectives and needs of implementing data-driven methods are not defined upfront.

Another frequently encountered challenge is the provision of data, which can meet the requirements of the developed methods and subsequently support effective decision making. Hodkiewicz et al [16] points out that only a fraction of organizations can provide the data for basic decision making. They propose a universal metric to measure data fitness for purpose and organizational incentives to improve data quality [70, 71]. Tiddens et al observed that the quality of data within an organization improves with experience in dealing with data-driven methods [22].

Other problems in data collection are the lack of standardized ways for data collection [20, 68] and the lack of knowledge of failure modes which should define the relevant data to collect. Besides the inability of organizations to collect meaningful data, the general scarcity of useful data leads to few scientific methods being evaluated on data sets from the field [16]. An et al encourages the sharing of data sets [68] as the existing benchmark data sets are limited in their usefulness, as is also pointed out by Eker et al [72].

Narrow Focus on Predicting Remaining Useful Life Data-driven prognostics, focus on the accurate prediction of the remaining useful life of components. As pointed out by Sun et al [21], prognostics can provide many additional benefits across the system life cycle. Especially when it can be used to improve the next generation of systems at early life cycle stages, the expected cost benefit is largest.

Such opportunities are partially identified by research in reliability optimization. However, latest developments in dynamic data availability, algorithmic capabilities and practitioners needs have only been partially exploited towards the goal of cost-effective data-driven reliability optimization [60, 61].

Chapter Learning Summary

Existing challenges for data-driven reliability optimization methods include that methods are not fit for the complexity realistic settings, require data that can hardly be provided, and are not sufficiently considering organizational objectives and contexts.

Chapter 5

Methodology

5.1 Overview

The goal of this thesis is to resolve existing practical limitations of data-driven reliability optimization methods in organizational contexts to unlock their potential for cost-effective reliability improvement. The general approach to achieve this goal involves three steps:

1. The first step is to identify the state-of-the-art and its limitations. This is covered in Chapter 4.

2. The second step aims to improve upon the state-of-the-art. This is done by understanding the limitations of existing methods and proposing a general methodology for the development and implementation of data-driven reliability optimization methods that address these limitations. This general methodology is introduced in Section 5.2. The methodology is then used to develop reliability optimization methods for three realistic scenarios in Chapters 6-8. These three scenarios correspond to the three research questions introduced in Chapter 1.

3. The third step aims to (1) verify that existing practical limitations have been addressed for the three realistic scenarios and that the proposed general methodology is useful, (2) identify missing links to use the developed methods from Chapters 6-8 for cost-effective reliability optimization in organizational contexts, and (3) derive a generalized framework by combining previous findings, which addresses the Umbrella RQ introduced in Chapter 1. The methodology for these three sub-steps is described in Section 5.3 and executed in Chapter 9.

5.2 Constructive Methodology

The success of an analytics project can be facilitated by following established implementation guidelines. CRISP-DM is the Cross-Industry Standard Process for Data Mining [73]. It is the most widely used guideline for data mining projects across application fields

[74, 58]. It separates the overall project into six phases - Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. Steps can be repeated, the order does not have to be followed strictly, and the implementations can be updated continuously.

The CRISP-DM methodology provides guidelines that are expected to address most of the mentioned difficulties of data-driven reliability optimization projects. Hence, it is chosen for the development and implementation of methods in this thesis. However, it is slightly modified to address the mentioned difficulties better. The modifications are described below.

Based on the findings of the literature review in Section 4.2, it is concluded that many data-driven projects in reliability studies fail due to insufficient amount and quality of data, unclear objectives, and inappropriately selected methods. Insufficient data cannot be quickly fixed, as reliability data collection is usually a costly and lengthy process. Therefore, an iterative implementation process would come to a halt after it has been discovered that sufficient data does not exist and cannot be obtained under the project effort and time constraints.

To avoid such unplanned project failures, utmost importance is paid to initial phases of the CRISP-DM methodology of Business and Data Understanding. Moreover, a third phase of Model Understanding is added in projects of this thesis. It ensures that the right modeling approach is chosen with respect to the required decision objectives and available data.

These three initial phases verify the feasibility of a data-driven reliability optimization project in a structured manner, before any implementations take place. The project implementation, consisting of Data Preparation, Modeling, Evaluation and Implementation phases, is only executed after the feasibility of the project and its requirements have been established.

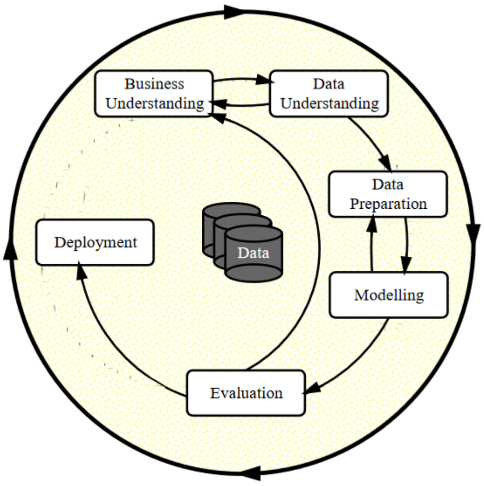
Figures 5.1a and 5.1b show the original and the slightly modified methodology with all its phases, respectively. In Figure 5.1b , the separation between project assessment and project implementation is more emphasized to reflect that project implementation is only executed after project feasibility has been properly assessed. Moreover, the Model Understanding phase is added.

Each of the phases is described in more detail below. They are guided by the CRISP-DM recommendations but slightly adapted for data-driven reliability optimization projects.

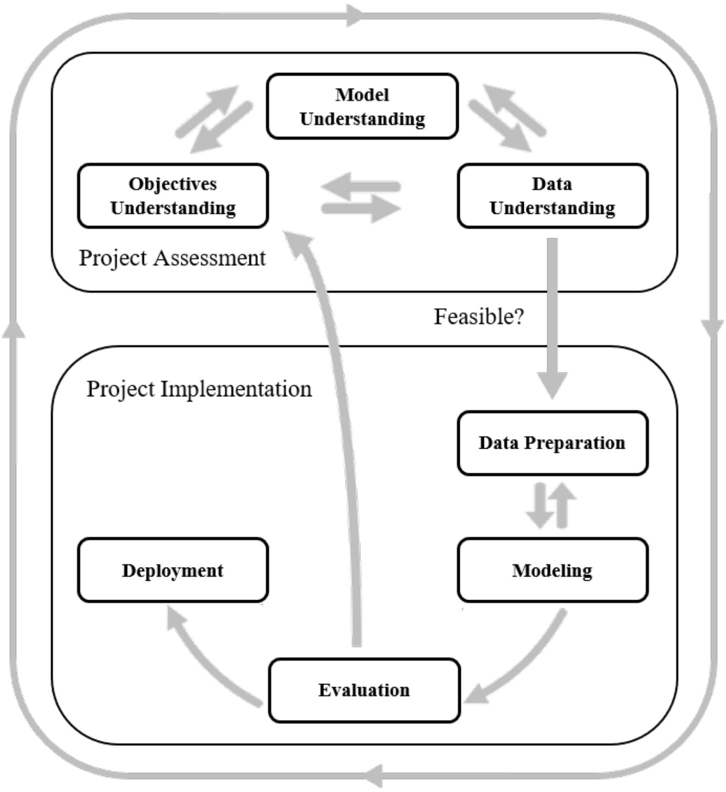
5.2.1 Project Assessment

Project Assessment ensures that a data-driven reliability optimization project can be carried out before any implementations begin. It is structured as follows.

Assess Objectives and Decision Variables - Business Understanding The first step is to identify the objective and the means, i.e. the decision variables, by which it can be achieved. E.g., the objective could be to improve reliability through choosing a reliable microcontroller and the decision variables three different suppliers of microcontrollers. The



(a)



(b)

Figure 5.1: (a) The original CRISP-DM methodology. [73] (b) Adapted CRISP-DM model.

identification of objectives and decision variables has to be carried out together with all relevant stakeholders. This avoids misunderstandings and ensures that all decision variables are covered. The objective should be converted into a measurable quantity, such as cost, reliability or a combination thereof. The method to estimate the degree of achievement of the objective should also be agreed on.

Assess Analysis and Modeling Methods to Support Objectives - Model Understanding The appropriate methods to achieve the defined objective need to be selected. Matching the desired optimization objective with the method output narrows down the choice of potential methods. Then, a literature review helps to select the best method among the remaining pool of methods.

E.g., to choose between different suppliers for a microcontroller, one could compare their specification sheets, evaluate the past experience with the different suppliers, or perform a reliability test. For the microcontroller example, a literature review could reveal that the specification provided by the suppliers and evaluating past supplier experience are insufficient or misleading. Hence, reliability testing has to be carried out.

When no suitable method for the considered objective is available, existing methods can often be modified to address the considered objective. A literature review can help identify potential modifications and methods.

Assess Data Availability and Quality - Data Understanding The pre-selected methodologies require inputs in form of data and knowledge. At this stage, it is assessed whether the required data and knowledge is either already available or can be collected.

Based on recommendation from Hodkiewicz et al [16], data availability and quality is assessed with an hierarchical framework. For each method the minimum and optimal data and knowledge requirements are known. Then the actual data and knowledge availability is compared against its requirements. If the actual data fulfills the minimum requirements of a method, the approach can be implemented. If not, it has to be decided if it is possible to collect the missing data and knowledge or to exclude the method from the pool.

E.g., for the microcontroller supplier selection example, the necessary data from a quantitative reliability test needs to be obtained by setting up and running the reliability test. Depending on the project budget this may be feasible or not.

Feasibility Check The coherence of objectives, means, methods, and available data and knowledge are confirmed before implementing and executing the data-driven optimization method. Sufficient software, hardware, and time resources for implementing the modeling strategies need to be available.

5.2.2 Project Implementation

The steps of the project implementation are described in detail in the methods Chapters 6-8. Here an overview is given.

Data Preparation The missing data and knowledge is collected by

- running experiments,
- interviewing system experts, and
- literature research.

Readily available data is cleaned, validated with system experts to ensure data quality meets its requirements, and stored in an accessible format.

Modeling The modeling implementation approach strongly depends on the chosen modeling strategies and is described in detail in later Sections. It is beneficial to employ several methods in parallel and compare their outputs. Whenever a novel method is implemented, all functionalities are first verified on a known test problem. When the novel method passes this test successfully, it is applied to the actual problem.

Evaluation Conformance to requirements and organizational objectives need to be ensured. Validation ensures that the implementation and its outcomes serve the organizational needs. This is carried out in the evaluation phase.

Deployment and Decision Making Successful implementation of the optimization method allows to determine the decision parameters which lead to the best outcome. Project stakeholders are informed about the recommended decisions and its justification.

Follow-Up and Iterative Improvement To validate that the suggested decisions are actually leading to the desired outcome in the long term requires follow-up after implementation. Feedback from follow-up can be used to further improve the developed methodology. Moreover, data-driven projects often produce additional insights which trigger new projects. Therefore, the developed frameworks should not be treated as static solutions but continuously evolved and adapted. Successful implementations can often be reused for related projects due to the modular structure of data-driven frameworks. This is represented by the outer circle in Figure 5.1b.

5.2.3 Limitations of the Constructive Methodology

The research for this thesis has been carried out over three consecutive years. It was originally framed as initial exploratory research into data-driven reliability optimization methods applied to power converters at CERN. A list of limitations results thereof.

Long term validation of reliability optimization often needs follow up which surpasses a three year period. Therefore, whenever possible, the methods were constructed as if they were implemented a years ago. Using data up to a years ago, the method was developed, and using data from a years ago until the actual date of the research being carried out, the method was validated. The specific choice of a depends on the project horizons.

As a result, additional data collection was often limited because it would have concerned historic systems and problems for which data could often not be generated at a later point in time. A positive side effect is that thereby the used data is representative for the quality and amount of pre-existing data sets in organizations.

The methods in this thesis are novel and developed in a field of research which struggles to showcase implementation success stories (as outlined in Section 4.2). Therefore, the final implementation of decisions derived by the methods in this thesis could only be partially carried out or not at all. However, by means of simulation, verification and validation it was possible to answer the arising 'what if' questions.

5.2.4 Structure of Constructive Chapters

The constructive Chapters 6-8 have a uniform Section structure as given below. They were carried out using the proposed modified CRISP-DM method. However, for the ease of reading, the order of presentation does not strictly follow the order of how the steps were carried out in practice.

1. Scenario Description and Problem Definition/Objectives Understanding: An introduction outlines the overall scenario setting, its relevance, and the need for the specific reliability optimization problem. It ends with a short scenario description and the required outcomes of a data-driven reliability improvement project for informed decision making.

2. Related Work and Methods Selection/Model Understanding: A literature review addresses related projects and the modeling approaches therein. Based on the findings the best-suited method and modeling strategy is determined.

3. Methodology/Modeling: The proposed methodology is described in detail.

4. (Optional) Modeling Verification: Whenever newly developed methods have been developed, they are verified on test problems.

5. Data Requirements and Availability/Data Understanding and Feasibility Check: The availability of data and knowledge is assessed. It is ensured that the data requirements are fulfilled. This Section represents the Data Understanding phase and the final feasibility check before implementation. Note that for the sake of presentation it is not always reported before the implementation steps are described.

6. Numerical Experiments/Data Preparation, Model Implementation, and Results Evaluation: The proposed methodology is implemented and applied to the considered reliability optimization scenario. Results are presented and their implications discussed. Based on the results, optimal decision making is proposed.

7. (Optional) Discussion: Lessons learned from the data-driven reliability optimization project are stated.

8. A summary, conclusion and research outlook closes each chapter.

5.3 Evaluative Methodology

Based on the constructive methodology, three representative use cases of data-driven reliability optimization are addressed. They reflect actual conditions of field data availability in organizations and are implemented using the latest analytics and modeling tools. The evaluative methodology combines the findings of the constructive chapters to answer the Umbrella RQ. It is composed of the following steps:

1. A critical assessment of the previous three constructive chapters to verify that the tailored CRISP-DM methodology helps to overcome practical limitations of data-driven reliability optimization methods. It consists of two parts: An evaluation whether the practical limitations have been addressed successfully in each of the constructive methods of Chapters 6-8 and study of the usefulness and potential improvements of the tailored CRISP-DM methodology.

2. An identification of missing steps to use the developed methods from Chapters 6-8 for cost-effective reliability optimization in organizational contexts. Specifically, the optimal timing of each constructive method within a system life cycle is discussed to maximize their effectiveness and suggestions are made for the effective collection and provision of high-quality reliability data.

3. Finally, all the previous findings are combined to provide a cost-effective data-driven reliability optimization framework for complex engineered systems, which addresses the Umbrella RQ.

5.3.1 Limitations of the Evaluative Methodology

Single best practices cannot exist due to the variety of organizational settings and reliability challenges. Therefore, the validity of generalizations of the findings is the main concern.

The methods have been validated across different scenarios and techniques in a particle accelerator domain. Moreover, they are based on methods that have been applied in various other domains. Hence, most of the findings should apply to a wide range of data-driven reliability optimization problems in the particle accelerator domain and likely in other related domains as well. Nevertheless, it is always recommended to check the compatibility of assumptions when reapplying the findings in new organizational contexts and application domains.

Chapter Learning Summary

Reliability optimization methods are implemented for three use cases using a general methodology which promises to resolve existing limitations. The methodology assesses project feasibility before implementation by cross-checking project objectives with available methods and data.

Chapter 6

Data-Driven Discovery of Failure Mechanisms

This scenario concerns a situation in which expert knowledge on the failure behavior of the studied system is limited. E.g., this may occur when the system is new and experts have not accumulated sufficient operational knowledge yet, when the system is very complex and some of its interactions cannot be foreseen despite best efforts, or when separate groups of specialists manage each of its separate sub-systems and interactions between the sub-systems are not sufficiently investigated.

In all such scenarios, the relevant failure mechanisms are mainly encoded in the operational data logged by the system. It may appear in the form of fault, alarm, and error codes as well as monitoring signals, such as temperatures, pressures, positions, operational settings, and configurations. Complex systems log many such variables at high sampling rates to capture the dynamics of the monitored systems. The volume of the data stream makes manual analysis challenging for human operators. An automated data analysis tool to predict faults and identify the relevant mechanisms would help system operators and experts to solve arising reliability issues faster.

This chapter presents such a method and, thereby, addresses RQ1. It is based on the latest techniques in the fields of explainable AI and deep learning. The method learns the system behavior from logged operational time series data. It can predict and explain system faults by highlighting the monitoring signals, which contribute the most to the failure. System experts can then react faster to arising reliability problems as they can focus their attention on the few relevant sub-systems.

This chapter and the methods and findings therein are based on previous publications [75, 76].

6.1 Scenario Description and Problem Definition

Despite the increasing complexity of technical systems, demands on their reliability are continuously rising. Reliability methods have to evolve with technological developments to meet such demands. Whilst errors in simple systems could be analyzed manually by experts, increasingly complex and interconnected systems render a manual analysis impossible. This stems from to the overwhelming amount of potentially relevant failure mechanisms and precursors that a human expert cannot take into account simultaneously.

Modern particle accelerators are an example of such complex systems. Their failures and anomalies cannot be fully captured by analytical models for several reasons:

- Accelerators are composed of highly specialized equipment which is built in low volumes and for which reliability models are rarely available.
- They can be composed of many thousands of such sub-systems, each recording large amounts of heterogeneous data.
- The operational configurations and modes normally change over time, which impacts the accelerators' reliability and operational margins (e.g. the operational margins may greatly vary when protons or ions are accelerated).
- Additionally, constant maintenance, upgrades, tuning of settings, etc. renders a modern particle accelerator a constantly evolving system.

Considering all these factors, traditional analytical modeling approaches are often inadequate and not practicable.

However, operational data, e.g. fault, alarm, and error codes, as well as monitoring signals, such as temperatures, pressures, positions, operational settings, and configurations, are usually abundant. They are logged at a rate and dimension that human operators cannot analyze in a timely manner. An automated data analysis, which helps operators in decoding the relevant failures and mechanisms from the abundant data, is required in such a setting. Such automated methods must handle heterogeneous data formats, be applicable to raw logging data, generalize from a few logged faults, and scale to hundreds of input signals. Such a data-driven prognostics and diagnostics framework is useful if it provides advance prediction of faults and the most relevant factors that cause them. With this information, system operators can mitigate or remove failure conditions and increase the system availability.

The predictive performance of such a framework can be assessed with standard classification metrics, such as the F1 score as introduced in Section 3.3. As predictions are generally more useful the earlier they are available, the lead-time of the prediction is relevant too. The quality of the relevant failure precursors and mechanism explanations can be assessed in surveys with system operators and experts in realistic usage settings, as e.g. proposed by Holzinger et al [54].

6.2 Related Work and Method Selection

Fault prediction methods based on monitoring data have been studied in the fields of system health management [77], prognostics and diagnostics [78, 79, 80], and predictive maintenance [81]. They are usually classified as model-driven, when analytical models of the system and expert knowledge is employed, or data-driven, when data is used to identify the system behavior. As model-driven approaches are infeasible in the considered scenario, only data-driven methods are discussed further.

Overview of Existing Methods

Data-driven methods are usually separated by their modeling approach in classical ML (e.g. k-nearest neighbor, SVM, decision tree), deep learning (e.g. deep convolutional neural networks, deep belief networks, recurrent neural networks) and probabilistic reasoning (e.g. Gaussian Processes, Hidden Markov models, Bayesian Graphical Networks) techniques.

Alternative classifications are based on the application field (e.g. mechanical, electrical, software), the system complexity (e.g. component, assembly, system, system of systems), the model learning approach (un-supervised, semi-supervised, supervised), and the availability and kind of data (binary, discrete, numeric, text, univariate, multivariate). The interpretability and explainability of methods is usually not a classification criterion and there are few works treating such aspects explicitly. However, as e.g. highlighted by Abdul et al [82], this is an important aspect of predictive methods in general and specifically for the concerned scenario. In the following Sections, existing relevant work is grouped by the choice of modeling approach and explainability is discussed for each work. These groups are SVM, Granger Causality, Association Rule Mining, Probabilistic Reasoning, and Deep Learning-based methods.

SVM For classical ML approaches SVMs are commonly used. Fulp et al [83] and Zhu et al [84] used SVMs to predict hard drive failures based on hand-crafted features of the system health. Leahy et al [23] used manually generated features based on data from Supervisory Control and Data Acquisition (SCADA) systems to predict failures of wind turbines. Fronza et al [85] tackled a systems of systems problem in which failures in large software systems were predicted. Although SVMs would allow interpretation of the learned models to gain further understanding of the failure mechanisms, it is not closer investigated in any of the works mentioned. Partially this is explained by the fact that the failure mechanisms are already understood when the methods are applied. Hence, the methods aim to predict the precise timing of a known failure mechanism instead of discovering the mechanisms at work.

Granger Causality Qui et al [86] developed a method based on L1 regularized Granger causality to identify root causes of anomalies in industrial processes. Their method is interpretable, scalable, and robust to meet industrial requirements. However, it is not used to predict faults in advance.

Association Rule Mining Vilalta et al [87] and Serio et al [88] employ association rule mining to infer failure mechanisms in complex infrastructures from logged data. The extracted rules are easy to interpret for machine experts. Vilalta et al identify anomalies in computer networks. The common class imbalance is solved by solely using the minority class data (i.e. failure data). Good accuracy, but also limits of practical applications, are reported. Serio et al carried out the only related work in the particle accelerator domain. Expert verified fault association rules between sub-systems were extracted and reported. However, time dependence between events is not considered and failure predictions are not carried out.

Probabilistic Reasoning Methods Within the class of probabilistic reasoning techniques, Mori et al [89] performed a root cause diagnosis for industrial processes using a Bayesian graphical model. Interpretable and accurate results were reported. Liu et al [90] mix probabilistic modeling and deep learning by combining ideas of state space modeling with Restricted Boltzmann Machines or Deep Neural Networks. Their scalable approach identifies root causes of anomalies in industrial processes with high accuracy.

Deep Learning Methods Saeki et al [91] use deep learning methods to classify anomalies of wind turbine generators from spectral data. In a test environment, their visual explanation technique highlighted the same failure precursors as a group of human experts. However, the authors note that the used data was not representative for realistic industrial scenarios. Amarsinghe et al [92] used a deep neural network to identify Denial of Service attacks on computing networks. The method highlights the most relevant inputs for its classification decisions with Layer-wise Relevance Propagation (LRP), which has been previously been introduced by Bach et al [93]. High classification accuracies and intuitive explanations were reported. The method uses hand-crafted features based on the raw data. Bach-Andersen et al [94] detect early fault precursors for wind turbine ball bearings based on raw spectral data. Among a comparison between logistic regression, fully connected neural networks, and deep convolutional neural networks, the latter performed best. Insights into the failure behavior are obtained by applying a visualization method to higher level layers of the deep network. Accurate results, robustness to class imbalance, and scaling to high dimensions is reported.

Discussion and Method Selection

Based on the previous research, a combination of deep learning and LRP appears to be promising for fault prediction and explanation in particle accelerators. They offer superior performance, scale well, and are able to handle raw data. The study of Bach-Andersen et al represents a good starting point. However, it uses a data structure, which is not representative for the particle accelerator domain and it has been used in a different field of application. Additionally, the LRP method has been more widely applied and tested [95, 96] than the visualization technique used in Bach-Andersen et al or other explanation techniques, such as class activation maps [97] or LIME [98].

In terms of modeling approaches, deep convolutional networks appear promising based on the literature. A recent and extensive review of deep learning models for (multivariate)

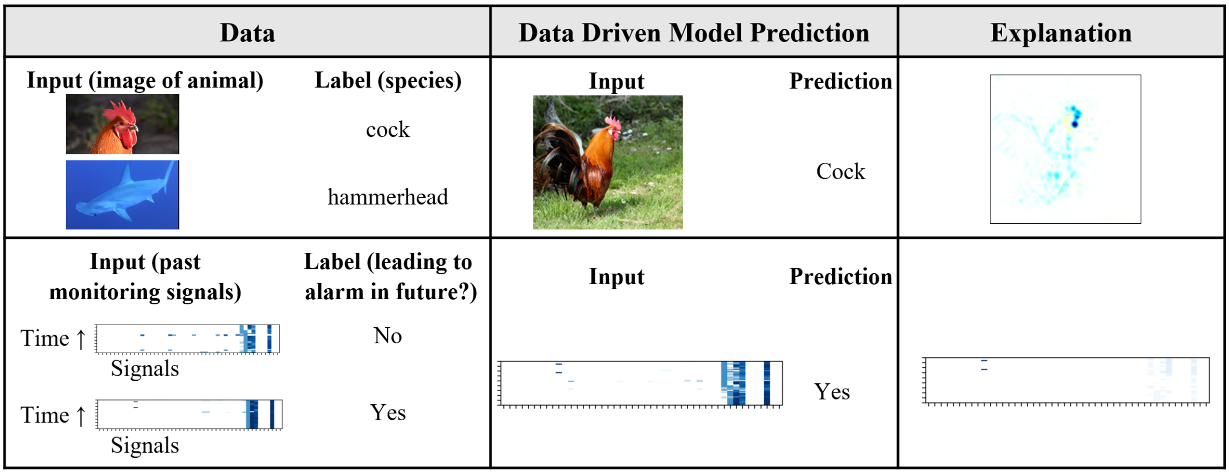


Figure 6.1: Upper Row: ML algorithms are able to identify animal species based on labeled images. Explanation techniques help to understand which pixels contribute the most to assign a certain species to an input image. [93] Lower Row: Logged time series are accumulated during the operation of a particle accelerator. A sliding window approach extracts a data set consisting of inputs characterising the relative past behavior of the system and outputs indicating if a specific alarm or fault occurs in the relative future, which is shown in the left cell (Data). This generates a supervised training data set without manual labeling effort. Based on this data set, a model can be learned to predict certain system alarms and faults, which is shown in the middle cell (Data Driven Model Prediction). LRP can then be used to highlight the most relevant input signals in the past that precede a fault in the future, which is shown in the lower right cell (Explanation). It highlights that only two alarms (darker blue) are relevant for the fault. [75]

time series classification by Fawaz et al [99] confirms that deep convolutional architectures outperform other methods across a variety of applications settings at a reasonable computational burden. Wang et al [100] reported similar results for univariate time series earlier.

Hence, deep convolutional neural networks are chosen as primary modeling method of failure phenomena in particle accelerators. For explanations of relevant failure precursors LRP is selected.

Deep neural networks are general approximators [101]. Hence, they are in principle capable of handling the variety of data sources without manual feature extraction and accurately modeling failure phenomena in particle accelerators. In this work, the convolutional networks are compared against classical ML approaches which serve as benchmark solutions.

Proposed Approach

The proposed approach is illustrated in Figure 6.1. Logged time series are accumulated during the operation of a particle accelerator. These consist of fault, alarm or anomaly signals, and monitoring signals. A sliding window approach extracts a data set consisting of inputs characterising the (relative) past behavior of the system and outputs indicating if a specific alarm or fault occurs in the (relative) future, which is shown in the lower left cell of the figure. This generates a supervised training data set without manual labeling effort. Based on this data set, a model can be learned to predict certain system alarms and faults, which is shown in the lower mid cell of the figure. LRP can then be used to highlight the most relevant input signals in the past that precede a fault in the future, which is shown in the lower right cell of the figure. It highlights that only two alarms (darker blue) are relevant for the fault. With this information, system experts can focus their attention and find solutions to arising reliability problems faster. The effective use of such a method can help to increase the availability of complex systems.

For example, certain magnet power converters steer particle beams based on feed-back from beam position monitors. Faulty beam position monitors could lead to noisy beam position measurements which could trigger a preventive shut down of a power converter. This can lead to the interruption of operations of a whole particle accelerator.

A predictive method could forecast such a preventive shut down. However, without additional information about the forecast, system experts have to manually search for the potential root cause. Considering the sheer amount of possibly relevant signals, they might not identify the mechanism before the preventive shut down happens. However, LRP could highlight a faulty beam position monitor as the most relevant precursor, because its noisy measurement disturbs the current regulation loop of the power converter. Based on this automatically generated hint, experts can simply replace the faulty beam position monitor and avoid an unplanned interruption of operations.

6.3 Methodology - Explainable Deep Learning Models for Failure Mechanism Discovery

Definitions and Overview

An infrastructure composed of multiple sub-systems is studied. Its behavior is monitored over time in a range of N observables, including event and alarm logs, continuous and discrete monitoring signals, and operational commands and settings. This forms a multivariate time series, $\mathbf{S} = \{\mathbf{S}_{i,t} : i \in [1 : N] \text{ and } t \in \mathbb{N}\}$. The range of alarms, faults and anomalies, which should be predicted are contained within this range of observable signals.

A relation between (relative) past and future behavior of the infrastructure can be approximated by an autoregressive model. It can only be approximated, since alarms and failures may appear without any advance indicators (precursors). Even in situations with precursors, they might not be captured by the monitoring systems. In a time discrete

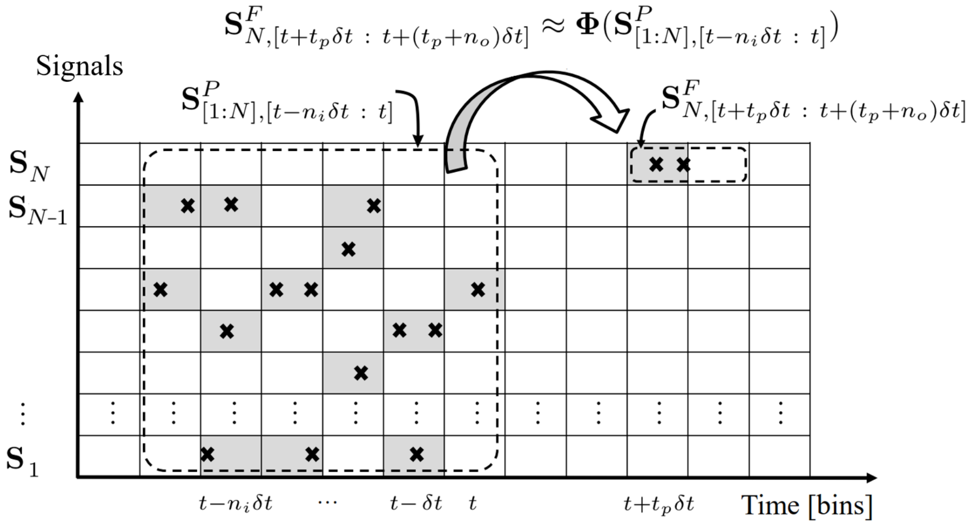


Figure 6.2: Time discrete model formulation. The x-axis represents discrete time and the y-axis monitoring signals of the investigated infrastructure. Crosses mark events that could be faults, alarms, changes in monitoring values, etc. Events of the signal $\mathbf{S}_N$ represent infrastructure faults that the model $\mathbf{\Phi}(\cdot)$ predicts. [75]

formulation, the approximate autoregressive model takes following form,

$$\mathbf{S}^F_{N,[t+t_p\delta t\ :\ t+(t_p+n_o)\delta t]} \approx \mathbf{\Phi}(\mathbf{S}^P_{[1:N],[t-n_i\delta t\ :\ t]})$$

with

- $\mathbf{S}^F_{N,[t+t_p\delta t\ :\ t+(t_p+n_o)\delta t]} = 1$, if a failure occurs between time $t + t_p\delta t$ and time $t + (t_p + n_o)\delta t$, and zero otherwise,
- $\mathbf{S}^P_{[1:N],[t-n_i\delta t\ :\ t]}$ being finite histories of observed signals covering the time steps $t-n_i\delta t$ to t and being considered as possible precursors,
- δt being the discretization time,
- t_p the prediction- or lead-time,
- n_o the number of time steps chosen to capture the future failure behavior,
- n_i the number of discrete time steps chosen to capture the history of the observed signals and
- $\mathbf{\Phi}$ an autoregressive model.

The model and its variables are illustrated in Figure 6.2. The x axis shows the discretized time and the y axis the various monitored signals.

In all cases, except very simple systems, the autoregressive model $\mathbf{\Phi}(\cdot)$ cannot be obtained from first principles. Hence, the model is learned from observed historic data $\mathbf{S}$, accumulated during operations of the infrastructure. This is done in a supervised learning setting by providing pairs of input data, $\mathbf{S}^P_{[1:N],[t-n_i\delta t\ :\ t]}$, and output data, $\mathbf{S}^F_{N,[t+t_p\delta t\ :\ t+(t_p+n_o)\delta t]}$. Different learning algorithms can be applied to the supervised training data set.

A trained model can predict the future system behavior if new observed input data is provided. However, it would predict the occurrence of a failure without the required information to prevent that failure. This required additional information is provided in the form of a relevance measure, $\rho(\mathbf{S}^P) \in \mathbb{R}^{(n_i,N)}$. It indicates the relevance of each input signal at each discrete time step. If a failure is predicted, the input signals contributing the most to the prediction of a failure will be assigned the highest relevance values. This allows system operators and experts to focus their attention and remove failure conditions before they lead to faults.

It has to be pointed out that the framework does not identify causal relations but only temporal precedence of correlated precursors of failures [102]. However, such information helps experts to establish causal models. Such findings can be integrated in model-driven system characterizations, such as presented in Chapter 7.

The methods can be used both in on- and offline analysis. As online tool, it acquires data from the infrastructure in real time as input and provides predictions and explanations of imminent failures continuously. System operators can then use this information to carry out planned maintenance before the failures happen in an uncontrolled way. This requires a lead-time, $t_p > 0$, to give system experts sufficient time to react.

As offline tool, complex failure mechanisms that already occurred can be explained using the input activation function. System experts can then modify the infrastructure so that the failure mechanism cannot occur again. As offline tool, no lead-time for predictions is required.

ML Pipeline

In the following, the procedure to derive the autoregressive model $\mathbf{\Phi}(\cdot)$ from observed data is detailed. It consists of data collection, model selection and evaluation, subsampling strategies, input data filtering and normalization, training of models through learning algorithms, and the explanation of their predictions. The procedure is summarized in a pseudoalgorithm at the end of this section.

Data Collection Observable signals $\mathbf{S}$ from the investigated infrastructure are stored in a data-set $\mathcal{D}$ in time series format. The specific signals are selected so that the relevant failure precursors and faults or alarms are contained in the data. Often this will be based on expert recommendation. Further details of the data collection are provided in the use case Section 6.5.

Model Selection and Evaluation The formulation of the autoregressive model includes a range of parameters, e.g. $[\delta T, n_i, n_o, t_p]$ (some will be introduced later in this Section).

These need to be optimized for the specific prediction task. This is carried out through an exhaustive grid search within a K-fold validation strategy [103]. Contrary to the widely used cross-validation, the temporal order of the folds is never mixed. Instead, the training set is continuously expanded and the validation set shrunken.

Formally, the full data set, $\mathcal{D}$, is split in a training set, $\mathcal{D}_{train}$ up to time t_{split}, and a final test set, $\mathcal{D}_{test}$ after time t_{split}. K further folds are obtained by splitting the training set into sub-training sets and validation sets at subsequent split times $t_{sub-split,k}$, $k = 1, ..., K$.

Subsampling In most cases, failures and alarms occur rarely in infrastructures. Hence, the output data $\mathbf{S}_N^F$ contains few failure examples (class '1') and many examples without failures (class '0'). This so-called class imbalance depends on the number of faults in the data as well as the model formulation parameters, such as the discretization time δt or the number of time steps that capture failures n_o. For the considered use case, an imbalance of up to $1 : 10^4$ is observed. Such strong imbalances lead to difficulties when using the training algorithms. Hence, the classes need to be more balanced.

To achieve that, several sampling methods are applied. Random subsampling of the majority class ('0') is applied until a pre-set target ratio, $p_{0,targ} = freq(cl_0)/freq(cl_1)$, is obtained.

Data items n_{cov} time steps before and after each class '1' example are added as it increases the 'contrast' in the vicinity of class '1' occurrences. This leads to improved classification performance and can be considered as an upsampling strategy.

A choice of the output window length, $n_o > 1$, leads to n_o times oversampling of class '1' items. This can result in an improved classification performance at the cost of a decreased certainty of the timing of predicted faults [76].

Input Filtering and Normalization Input signals, which do not contain statistically significant information, are automatically removed. This includes input signals with less than α_{min} values non-equal to zero and signals with a variance smaller or equal to σ_{min}. $\alpha_{min} = 4$ is chosen as it represents the minimal number of data items from which any of the algorithms could discriminate a pattern[76]. σ_{min}=0, is selected to remove constant signals. All inputs are normalized to the range $[0, 1]$.

Model Learning Algorithms Below, the different algorithms to learn the autoregressive model $\mathbf{\Phi}(\cdot)$ from observed data in a supervised fashion are discussed. The problem is a binary classification as the output variables can be either '0' or '1'. Both deep learning and classical ML algorithms are used and compared.

Recent studies of deep learning for multivariate time series classification [99, 100] found that deep fully convolutional networks reach state of the art performance while being easier to train than recurrent neural networks. They are chosen as main modeling strategy and are compared against SVM, Random-Forest, and k-Nearest-Neighbor classifiers, which are chosen due to their past successes and wide usage. Each of the used algorithms is explained in more detail below:

- FCN: The architecture was proposed by Wang et al [100]. It is made of three blocks with three layers in each: a convolutional layer, a batch normalization layer [104], and a ReLU activation layer. A Global Average Pooling layer (GAP) averages the output of the last block over the whole time dimension. The GAP layer is connected to a softmax classifier. Each convolutional layer has a stride of one with zero padding for conservation of the input data shape. Each of the three convolutional layers contains 128, 256, and 128 filters with a length of 8, 5, and 3, respectively. In comparison to the implementation by Fawaz et al [99], the number of training epochs is set to 2000. The optimization is stopped earlier when the validation loss is not decreasing by more than 0.001 within 200 epochs. The loss function is defined as categorical cross entropy. The model achieved the highest accuracy across 13 different multivariate time series classification tasks in the study by Fawaz et al [99]. Hence, it was chosen as the main architecture.

- FCN2drop: Dropout regularization is recommended in situations with few training data to avoid overfitting. It is expected to lead to a performance gain for the considered scenario with very few class '1' data. The FCN architecture is taken with dropout applied to the second convolution layer and the GAP layer. The dropout probability is set to $p_{drop} = 0.5$.

- FCN3drop: The FCN architecture is taken with dropout applied to the second and third convolution layer and the GAP layer. The dropout probability is set to $p_{drop} = 0.7$.

- tCNN: Zhao et al [105] proposed a network consisting of two convolutional layers with 6 and 12 filters each. A fully-connected layer with sigmoid activation function connects to it. The mean-squared error is used as loss function instead of cross-entropy. The same early stopping criterion as for FCN is added to the implementation of Fawaz et al [99].

- SVM: As reference classifier, a support vector machine with linear kernel functions is used. The default implementation from the sklearn package [106] showed best performance across tasks and is used.

- RF: The random forest classifier is a meta classifier composed of multiple decision trees. The default sklearn implementation [106] is used. The parameter for the considered number of features for optimal splitting is changed to the square root of numbers of features.

- kNN: The k-Nearest-Neighbor classifier is used with default parameters from the sklearn package [106] except for selecting $n = 7$ neighbors.

Parameters for the classical methods (SVM, RF, kNN) were selected based on recommendations from the scikit-learn user guide [106] and a set of preliminary tests with data similar to those from the use case. Classical methods require one dimensional input data.

This is achieved by flattening the 2D input data during training and predictions. Some spatial correlation information is lost during the flattening. The deep architectures do not suffer from that as they can directly use the 2D data.

The performance of the classifiers is measured in terms of accuracy and F1 scores on the validation and test sets. The accuracy is reported together with the fraction of the majority class in the test data. This allows to assess if the classifier performs better than a trivial predictor that always predicts the majority class. The F1 score is a suitable performance metric for the class imbalance situation. Results are usually reported with the prediction lead-time t_p as the time between fault prediction and actual fault influences the usefulness of the prediction.

Explaining Predictions

The method quantifies the relevance of each input at each time step, $\rho(\mathbf{S}^P) \in \mathbb{R}^{(n_i,N)}$, towards the classification output. This helps system experts identify the relevant failure precursors. The input relevance is plotted in color maps. Darker colors signify higher relevance.

LRP provides relevance measures for deep neural networks by propagating the classification output backwards through the layers of a neural network. Neurons, which contribute more to a subsequent layer, pass back more relevance. This techniques achieves best-in-class explanations [96]. A publicly available toolbox is used for implementation [107]. Different rules can be chosen. In preliminary tests, comparing Gradient x Input[95], LRP-0, and LRP-ϵ rules [93], LRP-0 demonstrated minimally better filtering of irrelevant failure precursors.

For the SVM classifier the input relevance can be accessed through the input feature weight vector [106]. For kNN and RF the input relevance was not evaluated as they were only used as classification benchmarks.

The quality and usefulness of an explanation largely depends on the embedding of the method in actual usage settings. The overall explanation process can be assessed by evaluating user experience, e.g., as proposed by Holzinger et al [54].

Since the presented method is a proof of concept at the time of writing, such an assessment cannot be fully carried out. Instead, a simplified evaluation based on three criteria (derived from [54]) is used. The criteria are the completeness of the provided explanation factors, the ease of understanding, and the degree of causality within the studied processes that can be derived from the explanation [75]. They will be discussed for each of the experiments in Section 6.5. The timeliness of an explanation would be an interesting factor as well. However, without an actual online usage setting it cannot be estimated reliably.

6.3.1 Pseudoalgorithm

The overall ML approach is summarized in by pseudoalgorithms below:

Pseudoalgorithm illustrating the overall process:

1. $\mathcal{D}$ `load monitoring time-series data`

2. Split data $\mathcal{D}$ in training $\mathcal{D}_{train}$ and testing set $\mathcal{D}_{test}$
3. (Model selection:)
   For: varying parameters
   $[p_{0,targ}, n_{cov}, \delta T, n_i, n_o, t_p, t_{sub-split}]$ **Do:**
   (a) Perform sub-algorithm (see below) $\leftarrow [\mathcal{D}_{train}, p_{0,targ}, n_{cov}, \delta T, n_i, n_o, t_p, t_{sub-split}]$
   (b) Store obtained performance metrics $\mathcal{P}_{train}$ and input activation $\rho(\mathbf{S}^P)_{train}$ on hold-out sets
4. (Model testing:)
   For: best performing model
   $[p_{0,targ}, n_{cov}, \delta T, n_i, n_o, t_p, t_{split}]_{optimal}$ **Do:**
   (a) Perform sub-algorithm $\leftarrow [\mathcal{D}, p_{0,targ}, n_{cov}, \delta T, n_i, n_o, t_p, t_{split}]_{optimal}$
   (b) Store obtained performance metrics $\mathcal{P}_{test}$ and input activations $\rho(\mathbf{S}^P)_{test}$ on test set
5. Evaluate consistency between $[\mathcal{P}_{train}, \rho(\mathbf{S}^P)_{train}]$ and $[\mathcal{P}_{test}, \rho(\mathbf{S}^P)_{test}]$ for optimal parameters

Sub-algorithm:

1. Get parameters $\leftarrow [\mathcal{D}, p_{0,targ}, n_{cov}, \delta T, n_i, n_o, t_p, t_{sub-split}]$
2. Transform time series data to pairs of input data $\mathbf{S}^P_{[1:N],[t-n_i\delta t\ :\ t]}$ and output data $\mathbf{S}^F_{N,[t+t_p\delta t\ :\ t+(t_p+n_o)\delta t]}$.
3. Split data in sub-training set(s), $\mathcal{D}_{sub-train}$, and sub-testing set(s), $\mathcal{D}_{sub-test}$ at defined splitting time(s) $t_{sub-split}$.
4. Perform sub-sampling on training set.
5. Perform input filtering and normalization
6. **For:** all target signals and classifiers **Do:**
   (a) Train predictive model $\mathbf{\Phi}(\cdot)$ on $\mathcal{D}_{sub-train}$
   (b) Evaluate performance metrics $\mathcal{P}_{sub-test}$ and input activation $\rho(\mathbf{S}^P)_{sub-test}$ on filtered and normalized sub-testing set
7. Collect and report performance metrics and input activation.

6.4 Data Requirements and Availability

The data requirements of the proposed method and the actual availability of data are assessed. Since the method is developed for a scenario with limited a priori expert knowledge on the system behavior, the data requirements focus on the available logged time series.

Data Requirements The data can be distinguished in input data - time series characterising the 'past' machine behavior - and output data - 'future' events within the logging time series that should be predicted. There are no particular data format requirements for the input data as long as it can be encoded numerically. However, to achieve a good predictive performance, the input data should contain the relevant precursors for the future event to be predicted. The input data may also include past observations of the output data (i.e. if you want to predict future occurrences of alarm x, past occurrences of x may be used as input).

The requirements for the output data are that the time series are discrete. In most cases, failures are discrete binary events and automatically satisfy this condition. For the method to show good performance there should be at least four examples of fault or alarm occurrences in the data. This represents the absolute minimum for which the framework is able to detect patterns [76]. However, in that case the validation process relies on human judgement of the provided predictions and explanations. To carry out proper validation techniques, such as cross-validation, faults or alarms should occur more than ten times in the data. Generally, the more alarms are present in the data the more reliable the method works. When a specific alarm or fault signal represents a single failure mechanism, the interpretation of failure explanations is simplified. However, the method does also work when different failure mechanisms are grouped into a single alarm signal.

Modifications of the studied infrastructure may introduce or remove certain failure modes. If a method is trained on data lacking a failure mode, which is later introduced, it will likely not be able to adapt its predictions. This is addressed by the chosen validation process to include the effects of infrastructure modifications for unbiased performance estimates. Hence, changes in the infrastructure that may lead to predictive performance degradation are discovered before the method is deployed to an operational setting.

Data Availability The actual data availability is assessed for the so-called Proton Synchrotron Booster (PSB) at CERN as introduced in Section 2.2.2. The method is later applied to this use case. The Monitoring and condition data, operational configurations and settings, and failure and alarm data are continuously logged since 2014. Data between January 2015 and December 2017 is selected for testing the method. It contained sufficient alarm data and the infrastructure did not substantially change within that time period.

For the input data, alarms, interlock, and beam destination signals are used. The choice was based on expert recommendation. They are expected to most likely contain failure precursors. However, the data logging systems were not designed for prognostic purposes [108]. Hence, it can only be guaranteed that the input data satisfies the minimal requirements. Whether failure precursors are actually present in the input data can be assessed with the explanation output provided by the proposed method.

For the output data, two power converter failure modes are considered: malfunction of a power converter controller and failure of a current measurement device. Both occurred more than ten times in the considered time frame. It is not known whether one or multiple failure mechanisms lead to each of the failure modes. Again, it can only be guaranteed that the output data satisfies the minimal requirements.

Discussion Overall, the framework has very low minimal requirements for the data. If the data satisfies recommended requirements (precursors, number of fault occurrences, separate mechanisms), results are expected to be better and easier to interpret. Still, for data with minimal requirements the method should provide useful results. For the considered use case, the data availability and quality is acceptable but not optimal. The resulting performance of the method for this case is discussed in detail in Section 6.5.

6.5 Numerical Experiments

The proposed approach is first validated on synthetically generated data and then tested on PSB particle accelerator data. Publicly available implementations are provided[1].

6.5.1 Synthetic Data Experiments

To validate the approach it is applied to synthetically generated data. Using synthetic data allows to assess if the framework predicts and identifies the manually created failure mechanisms correctly.

Noise Robustness The first experiment tests from how many input signals the correct failure precursors can be isolated whilst fewer than ten failure examples are present. For this test, a infrastructure is simulated by n_{rand} systems randomly firing alarms and one system S_p that produces two subsequent failure precursors which are followed by a critical failure of the infrastructure S_s^F. The pattern and its timing parameters are shown in Figure 6.3. The time evolves in the horizontal direction and the different signals are listed vertically. The S_p signal contains deterministic precursors. Two following signals cause a fault signal S_s^F. A range of randomly activated noise signals, $S_{R_1}, S_{R_2}, ...$, represent non-relevant parts of the infrastructure.

The problem parameters are a time $t_{br} \sim \mathcal{N}(\mu = 14.61\ d,\ \sigma = 14.61\ d)$ between randomly firing precursors, $S_{R_l},\ l = 1, 2, ..., n_{rand}$, a time $t_{bp} \sim \mathcal{N}(\mu = 1\ d,\ \sigma = 1\ d/24)$ between deterministic precursors S_p, a time $t_{pe} \sim \mathcal{N}(\mu = 10\ d/24,\ \sigma = 1\ d/24)$ between deterministic precursors S_p and infrastructure failures S_s^F, and a time $t_{ep} \sim \mathcal{N}(\mu = 36.525\ d,\ \sigma = 36.525\ d)$ between infrastructure failure S_s^F and deterministic precursors S_p with d being a day of 24 hours.[75] The data covers a time range of 2.7 years. $n_{rand} = [2^0, 2^1, ..., 2^9]$ randomly firing systems are added.

The method is applied with following parameters: Sampling times $\delta t = [2\ h, 3\ h]$ (h for hours), input range $n_i = 40$, lead-time $t_p = 0$, output range $n_o = [1, 2, 3, 4]$, sub-sampling target ratio $p_{0,targ} = 0.8$, and class '1' neighborhood coverage $n_{cov} = 2$. The data is split at a time t_{split} so that 80 percent of the data-set are used for training and model-selection and

[1] `https://github.com/lfelsber/alarmsMining`

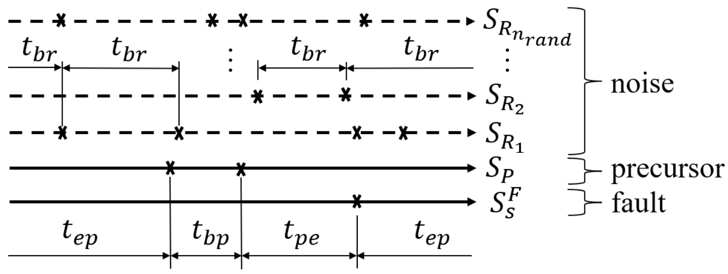


Figure 6.3: Parameters of synthetic pattern. The S_p signal contains deterministic precursors. Two following signals cause a fault signal S_s^F. A range of randomly activated noise signals, $S_{R_1}, S_{R_2}, ...$, represent non-relevant parts of the infrastructure. [75]

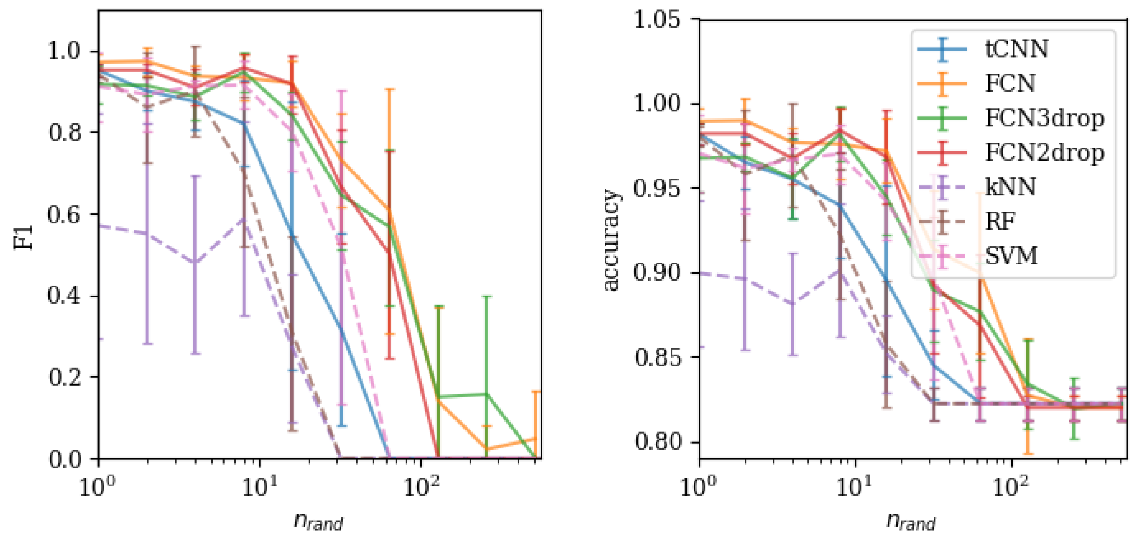
20 percent for final testing. A 7-fold validation is used for model selection. Sub-splitting times $t_{sub-split}$ are chosen to have $[50, 55, 60, 65, 70, 75, 80]$ percent of training data and $[50, 45, 40, 35, 30, 25, 20]$ percent for validation. Only 7 (13) critical failures were present in the training data of the validation folds (the whole data-set) on average. This represents a realistic number of observed failures in historic data.

The results for $\delta t = 3h$ and $n_o = 2$ are presented as with these hyper-parameters good results across classifiers are obtained. The F1 score and accuracy are plotted as a function of the number of randomly firing signals, n_{rand}, in Figures 6.4a and 6.4b. For higher numbers of random signals the classification performance is decreasing for all classifiers. The FCN-based networks perform better overall. For up to fifty random signals, the patterns can be predicted from only seven faults in the training data with an acceptable performance for the FCN-based architectures.

Recovering Fault Tree Structure The second experiment tests whether faults due to interactions of multiple sub-systems may be predicted and explained. For this experiment, data from two interacting sub-systems, S_{P_1} and S_{P_2}, and four additional non-interacting sub-systems, $S_{R_{1-4}}$, are generated. An infrastructure fault, S_b^F, happens when simultaneous failures of the two interacting sub-systems fulfill a Boolean AND, OR, or XOR condition. The parameters of the timing and delays of the signals are shown in Figure 6.5.

The data was generated with parameters $t_{br} \sim \mathcal{N}(\mu = 23\ min,\ \sigma = 24\ min)$ and $t_{pe} = 71\ min$, $t_{ep} = 120\ min$. The framework is applied with following parameters: sampling time $\delta t = 12\ min$ (min for minutes), input range $n_i = 5$, lead-time $t_p = 5$, output range $n_o = 1$, sub-sampling target ratio $p_{0,targ} = 0.8$, and class '1' neighborhood coverage $n_{cov} = 2$. The data set is split in equally sized training and testing sets. No K-fold validation is performed, as no hyperparameters need to be selected in this case. Less than 20 critical failures are contained in the input data. Deep networks and traditional methods consistently reach $F1 > 0.97$. The explanation results for the FCN2drop network are discussed in the following.

The input activations for the AND, OR, and XOR scenario are illustrated in Figure 6.6. The three lines represent three randomly selected system snapshots shortly before a critical



(a) (b)

Figure 6.4: Dependency of the predictive performance on the number of randomly firing signals. The line depicts the mean and the error bar plus and minus one standard deviation calculated over the 7 validation sets. Solid lines represent deep models, which perform on average better than the classical models (dashed). (a) The F1 score. Note that the predictors level out at 0.0 for large n_{rand}, which is the binary F1 score when always predicting the majority class ('0'). (b) The accuracy. The predictors level out at 0.82 for large n_{rand}, which is the accuracy when always predicting the majority class ('0'). [75]

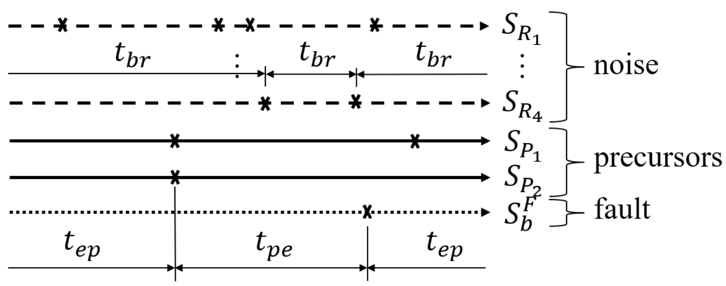


Figure 6.5: Parameters of synthetic pattern. An infrastructure fault, S_b^F, happens when simultaneous failures of the two interacting sub-systems, S_{P_1} and S_{P_2}, fulfill a Boolean AND, OR, or XOR condition. Four additional non-interacting sub-systems, $S_{R_{1-4}}$, randomly trigger alarms that do not lead to a fault. [75]

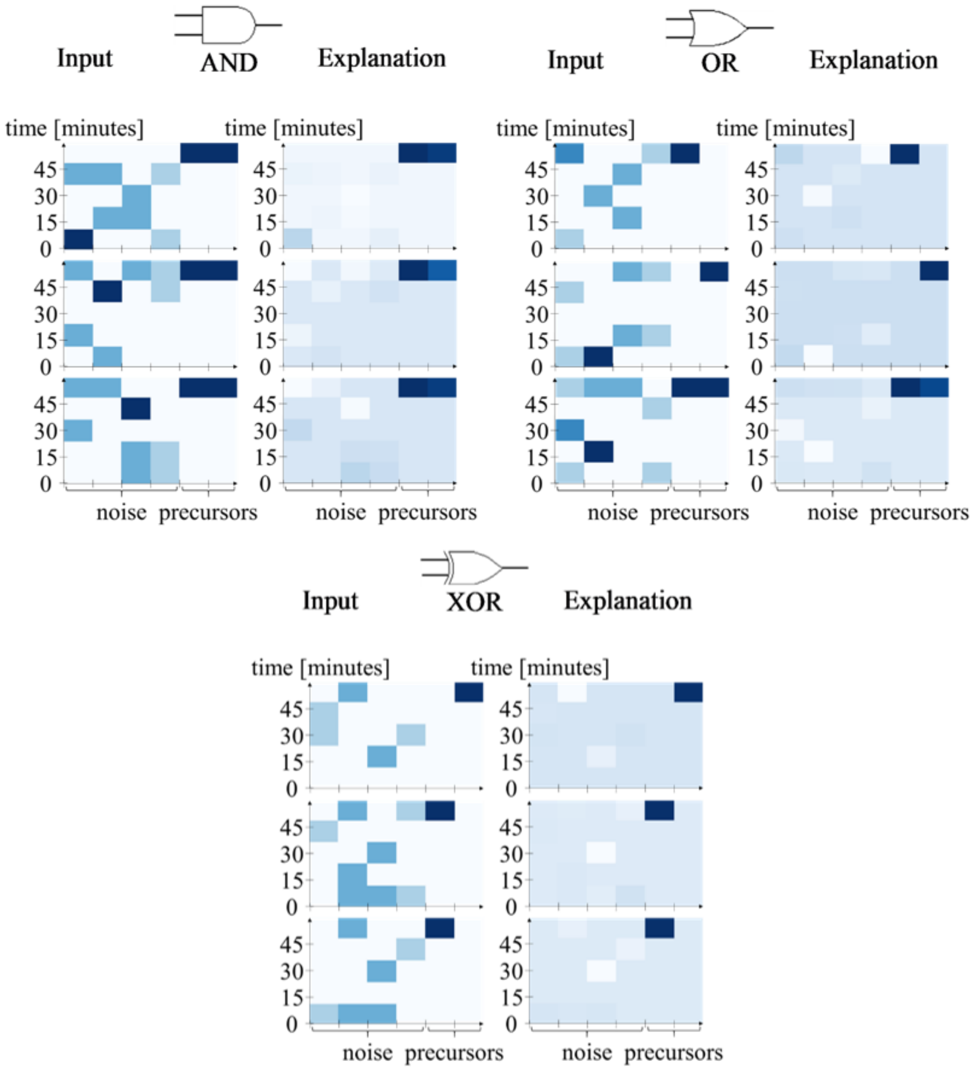


Figure 6.6: Illustration of AND, OR and XOR fault logic extraction. Left columns show three randomly selected input windows before failure occurrence. Right columns show the relevant precursors obtained with the FCN2drop network (darker colors indicate higher relevance). Comparing the relevant precursors (right columns), allows to distinguish different Boolean rules and recover the fault logic of the system. [75]

error occurred. Left columns are the unfiltered input data and right columns show the relevant inputs after applying the LRP method. For all scenarios the relevant precursors, S_{P_1} and S_{P_2}, are correctly highlighted by the LRP method. By comparison of the different snapshots, the Boolean logic can be inferred successfully. However, this requires several fault behaviors to be available at the same time.

The results confirm that different types of system interaction can be inferred from the explanation output (left columns). The raw inputs (right columns) would not provide sufficient information to do so. With respect to the proposed three criteria assessment, the explanation can be considered complete if several fault behaviors are observed. For two interacting systems, the interaction mechanism is easy to understand. The degree of causality that can be derived from the explanation cannot be assessed for this synthetic example.

Further synthetic data experiments with the proposed method have been carried out in [76]. Among the main findings are that an increase of n_o by one or two may improve accuracy when the delay between precursors and faults exhibits high variance, that the input relevance generally highlights precursors with lower timing variance stronger than such with higher variance, and that patterns can be identified from only four examples in extreme cases.

6.5.2 Particle Accelerator Data Experiments

In the following experiments, the method is applied to monitoring data sets (henceforth called 'real data') from the PSB accelerator at CERN as described in the data availability Section 6.4. It is assessed whether the method can predict faults in advance from the monitored data and is able to extract relevant precursors, which explain the predicted failure mechanisms.

As mentioned, the data set was not recorded for such data analysis purposes. It mostly served diagnostic purposes. Moreover, the infrastructures are continuously worked on.

The following data are taken for the analysis:

- Data of the LASER alarm database [109] contains logged alarms. They have a list of attributes including system name, fault code, and priority. For the chosen data, the priority take values 2 and 3. Priority 2 alarms are warnings, whereas priority 3 alarms are faults leading to the shutdown of the affected system. An alarm begins with a rising flag and ends with a falling flag. Only rising flags were used in the analysis. Data from a group of eight power converters are chosen. For the input data representation, all fault codes of a system were grouped. This results in eight input signals. The choice of target data is closer described in subsequent Sections.

- 27 Interlock signals were added based on expert recommendation. They record internal and external disturbances, which can lead to an interruption of accelerator operation.

Table 6.1: Performance metrics for mixing synthetic and real data experiments. $\mathbf{frac_{maj}}$ stands for the fraction of the majority class and is shown as reference for the accuracy of a trivial predictor always predicting the majority class. v and $\sigma_{\mathbf{v}}$ stand for the mean and standard deviation over the 7 validation folds, respectively, and t for results on the test set. [75]

	tCNN			**FCN**			**FCN3drop**			**FCN2drop**			**kNN**			**RF**			**SVM**			**frac maj**		
	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t
F1_cl1	0.00	0.00	0.80	0.89	0.11	0.93	0.79	0.22	0.80	0.92	0.05	0.93	0.12	0.16	0.20	0.00	0.00	0.00	0.73	0.15	0.71			
acc	0.83	0.03	0.95	0.97	0.03	0.98	0.93	0.06	0.95	0.97	0.02	0.98	0.84	0.03	0.87	0.83	0.03	0.89	0.91	0.05	0.94	0.83	0.03	0.89

- The beam destination variable provides information on operational settings of the accelerator. The eight beam destinations are one-hot-encoded and used as additional input data.

Overall, the data consists of 8 LASER alarm signals, 27 interlock signals, and 8 beam destination signals. Time intervals in which the PSB is not operational are omitted, as the data is not representing operational conditions.

Mixing Synthetic and Real Data In a preliminary step, it is tested if synthetic patterns can be isolated from the real world data. In that sense, the noise robustness experiment is repeated with the noise channels being replaced by the real world data described above.

The following framework parameters are used: sampling times $\delta t = [2\ h, 3\ h]$ (h for hours), input range $n_i = 40$, lead-time $t_p = 0$, output range of $n_o = [1, 2, 3, 4]$, subsampling target ratio $p_{0,targ} = 0.8$, and class '1' neighborhood coverage $n_{cov} = 2$. The model validation strategy is the same as for the noise robustness experiment.

Parameters $\delta t = 3\ h$ and $n_o = 3$ achieved high F1 scores and accuracy and are shown in Table 6.1. The columns indicate the different models that were trained. $\mathbf{frac_{maj}}$ stands for the fraction of the majority class and is shown as reference for the accuracy of a trivial predictor always predicting the majority class. v and $\sigma_{\mathbf{v}}$ stand for the mean and standard deviation over the 7 validation folds, respectively, and t for results on the test set. The lines show the F1 (denoted F1_cl1) and the accuracy (denoted acc).

Only 7 faults are present in the training data. Despite that, the FCN network achieves an F1 score close to 1. This indicates that a well defined failure pattern in real data could be detected by the FCN network from less than ten training examples.

The input activation plots for the FCN network and the SVM are presented in Figure 6.7a. Both methods correctly identify the synthetic precursors from the 45 signals. In terms of the three assessment criteria of the quality of explanations, they can be considered complete and easy to understand. The causality cannot be assessed for the synthetic case.

Real Data As final test, the method is applied to the real data set. It is tested whether it can predict and explain two types of power converter alarms with priority 3: fault code

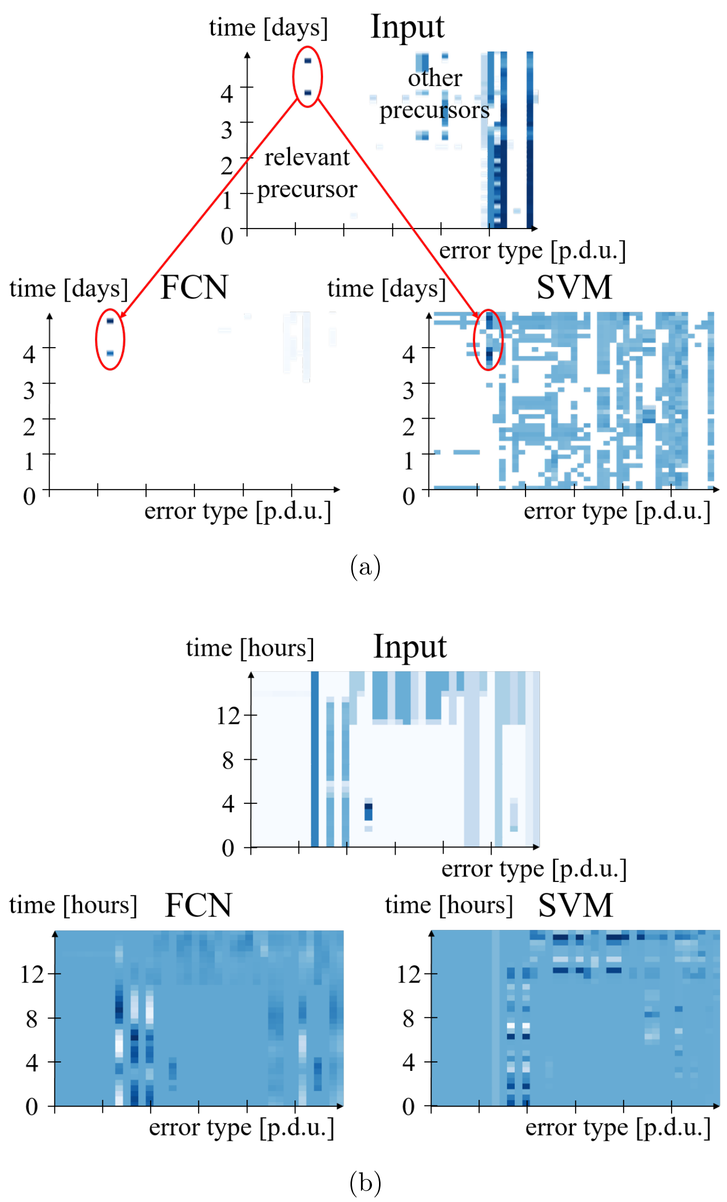


Figure 6.7: (a) Upper: Input window for a single fault prediction example. Red ellipse highlights the correct precursors. Lower left: Correctly identified relevant failure precursors by FCN network. Lower right: Correctly identified failure precursors by SVM network. (darker colors in the heatmap signify higher relevance). (b) Input relevance for real data input window with $\delta t = 30\,min$, $n_i = 32$, $t_p = 0$, $n_o = 4$ and $\mathbf{S}_{F_0}$. The SVM assign relevance to more signals than the FCN. System experts could identify that certain combinations of external interlock signals and operational modes lead to infrastructure failures. [75]

Table 6.2: Performance metrics for real data experiments. $\mathbf{frac_{maj}}$ stands for the fraction of the majority class and is shown as reference for the accuracy of a trivial predictor always predicting the majority class. v and $\sigma_{\mathbf{v}}$ stand for the mean and standard deviation over the 7 validation folds, respectively, and t for results on the test set. [75]

	$S_{F.i}$	n_i	t_p	FCN			FCN3drop			FCN2drop			tCNN			kNN			RF			SVM			frac maj		
				v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t
acc	4	16	0	0.95	0.03	1.00	0.95	0.03	1.00	0.95	0.03	1.00	0.91	0.06	0.92	0.90	0.02	0.92	0.90	0.04	0.92	0.89	0.04	0.92	0.87	0.04	0.92
F1	4	16	0	0.81	0.12	1.00	0.84	0.09	1.00	0.84	0.09	1.00	0.36	0.50	0.00	0.68	0.10	0.67	0.35	0.33	0.00	0.13	0.30	0.67			
acc	7	16	0	0.95	0.03	1.00	0.95	0.03	0.83	0.92	0.06	0.96	0.87	0.03	0.92	0.91	0.03	0.92	0.87	0.03	0.96	0.89	0.04	0.96	0.86	0.03	0.92
F1	7	16	0	0.84	0.09	1.00	0.84	0.09	0.33	0.71	0.27	0.80	0.00	0.00	0.00	0.74	0.03	0.67	0.00	0.00	0.67	0.17	0.38	0.80			
acc	4	32	0	0.98	0.03	1.00	0.97	0.02	0.92	0.98	0.03	1.00	0.98	0.03	0.92	0.89	0.01	0.92	0.88	0.03	0.92	0.85	0.04	0.96	0.88	0.04	0.92
F1	4	32	0	0.94	0.09	1.00	0.90	0.05	0.00	0.92	0.11	1.00	0.93	0.10	0.00	0.57	0.09	0.67	0.18	0.25	0.00	0.33	0.10	0.80			
acc	7	32	0	0.97	0.03	1.00	0.95	0.03	0.92	0.97	0.02	0.92	0.93	0.05	0.92	0.87	0.04	0.96	0.88	0.03	0.92	0.85	0.04	1.00	0.86	0.03	0.92
F1	7	32	0	0.89	0.07	1.00	0.84	0.09	0.00	0.89	0.07	0.00	0.68	0.39	0.00	0.53	0.14	0.80	0.46	0.29	0.00	0.38	0.21	1.00			
acc	4	16	1	0.94	0.06	1.00	0.92	0.03	1.00	0.97	0.03	1.00	0.88	0.05	0.92	0.91	0.03	0.92	0.88	0.03	1.00	0.94	0.02	0.96	0.86	0.04	0.92
F1	4	16	1	0.75	0.26	1.00	0.74	0.06	1.00	0.89	0.10	1.00	0.16	0.36	0.00	0.74	0.06	0.67	0.08	0.18	1.00	0.66	0.24	0.80			
acc	7	16	1	0.95	0.01	1.00	0.91	0.03	1.00	0.96	0.01	1.00	0.86	0.03	0.91	0.91	0.03	0.96	0.86	0.03	0.91	0.94	0.02	1.00	0.83	0.00	0.91
F1	7	16	1	0.85	0.06	1.00	0.77	0.06	1.00	0.88	0.01	1.00	0.00	0.00	0.00	0.76	0.04	0.80	0.00	0.00	0.67	0.67	0.25	1.00			
acc	4	32	1	0.92	0.03	1.00	0.93	0.02	0.92	0.96	0.01	0.96	0.88	0.04	0.92	0.88	0.02	0.96	0.91	0.05	0.92	0.87	0.03	0.96	0.86	0.03	0.92
F1	4	32	1	0.62	0.18	1.00	0.68	0.19	0.00	0.84	0.06	0.80	0.16	0.36	0.00	0.55	0.12	0.80	0.72	0.18	0.00	0.40	0.15	0.80			
acc	7	32	1	0.92	0.03	1.00	0.96	0.01	0.91	0.97	0.02	0.96	0.88	0.04	0.91	0.86	0.05	0.96	0.89	0.05	0.91	0.84	0.04	1.00	0.84	0.02	0.91
F1	7	32	1	0.68	0.23	1.00	0.86	0.04	0.00	0.90	0.05	0.80	0.17	0.38	0.00	0.53	0.14	0.80	0.33	0.45	0.00	0.38	0.21	1.00			

$\mathbf{S}_{F_4}$ (malfunction of a power converter controller) and $\mathbf{S}_{F_7}$ (failure of a current measurement device).

Sampling times $\delta t = [10\ min, 30\ min, 2\ h]$ (h for hours, min for minutes), input ranges $n_i = [16, 32, 64]$, lead-times $t_p = [0, 1]$, output ranges of $n_o = [1, 2, 4, 16]$, a sub-sampling target ratio $p_{0,targ} = 0.8$, and a class '1' neighborhood coverage $n_{cov} = 2$ are used as parameters. The model validation method of the previous experiment is kept.

Table 6.2 shows the results obtained with parameters $\delta t = 30\ min$, $n_i = [16, 32]$, $t_p = [0, 1]$, and $n_o = 4$ as these led to good results. The FCN-based networks achieve an F1 score close to 1 both for an offline use without lead-time of predictions ($t_p = 0$) and for an online use with lead-time ($t_p = 1 = 30\ min$). These results were obtained from as few as 17 fault examples on average in the training data for validation and testing. Applying dropout does not have a strong impact on performance. The results of the tCNN network strongly depends on the problem parameterization. Its F1 score ranges from 0 to 1. The performance metrics of the classical ML methods are significantly outperformed by the FCN-based architectures.

For the two targeted failure modes, the method shows a satisfactory predictive performance. However, tests on other failure modes, which are not presented here, showed that often no predictive patterns can be identified. The promising results from synthetic tests indicate that this is most likely due to an absence of well defined failure precursors in the selected and available input data. This is not surprising as the data logging system was not designed for prognostics applications and many faults occur without precursors.

An example of an input relevance plot for a fault is shown in Figure 6.7b. In comparison to the synthetic experiments, the fault mechanism is not known a priori. The input

Table 6.3: Performance metrics on the PEMS dataset. $\mathbf{frac_{maj}}$ stands for the fraction of the majority class and is shown as reference for the accuracy of a trivial predictor always predicting the majority class. v and $\sigma_\mathbf{v}$ stand for the mean and standard deviation over the 7 validation folds, respectively, and t for results on the test set. Results for $\delta t = 10\ min$ are omitted for brevity.

		tCNN			**FCN**			**FCN2drop**			**kNN**			**RF**			**SVM**			**frac maj**		
	t_p	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t	v	σ_v	t
acc	2	0.82	0.02	0.85	0.83	0.02	0.91	0.82	0.02	0.89	0.76	0.01	0.79	0.91	0.02	0.96	0.83	0.01	0.85	0.64	0.01	0.64
F1	2	0.74	0.03	0.79	0.74	0.03	0.87	0.74	0.03	0.84	0.69	0.01	0.74	0.87	0.04	0.94	0.75	0.03	0.76			
acc	4	0.81	0.01	0.86	0.82	0.02	0.90	0.80	0.01	0.91	0.75	0.01	0.76	0.91	0.03	0.97	0.80	0.01	0.86	0.65	0.01	0.63
F1	4	0.73	0.02	0.80	0.74	0.02	0.85	0.70	0.02	0.87	0.68	0.00	0.70	0.87	0.04	0.95	0.71	0.01	0.79			
acc	8	0.82	0.02	0.85	0.83	0.02	0.89	0.82	0.02	0.88	0.78	0.01	0.78	0.91	0.02	0.95	0.82	0.02	0.86	0.65	0.01	0.64
F1	8	0.73	0.02	0.78	0.73	0.04	0.84	0.71	0.01	0.81	0.68	0.01	0.67	0.86	0.03	0.93	0.73	0.01	0.79			

relevance highlights several signals at the same time. When discussing the results with system experts, it was noticed that interpretation of input activation plots is increasingly difficult when several signals are highlighted. Nevertheless, with the help of the input activation plot and consulting additional logbooks and databases, the range of likely failure mechanisms could be greatly reduced.

With respect to the three explanation quality criteria, the following is observed: The explanation was incomplete for this case as additional information sources needed to be consulted. However, with the provided input activation plots the consultation of other sources is more focused. The ease of understanding of the explanations decreases with the number of highlighted precursors. The explanations help to shrink the pool of postulated causal chains, but some uncertainty remains. Overall, system experts expressed that the input activation provides useful information and that it can be a promising approach for the operation of future particle accelerators.

6.5.3 Further applications

The presented method can be used for any discrete event prognosis and explanation in any domain. To exemplify that, the framework is used to predict and explain traffic jams in the San Francisco bay area. Data on highway occupation data was collected by the California Department of Transportation with 963 sensors storing car traffic occupation measurements, $O_h \in [0, 1]$, in ten minute intervals [110].

A highway occupation $O_h > 0.5$ is defined as traffic jam (represented by class '1') to obtain a binary target variable. Based on traffic data from October 3, 2008 to March 3, 2009 models are trained. Of the 963 sensors, for which more than 300 but less than 5000 class '1' occurrences were measured (to select sensors for which traffic jams are rare events), three were randomly chosen as target data. As input data, the 50 most correlated sensors for each target sensor were chosen. The weekday was hot encoded as additional input as traffic dynamics may change depending on the weekday. The target sensor was removed from the input, as the goal is to explain traffic jams based on the past traffic in other locations.

The framework is applied to the data with sampling times $\delta t = [10\ min, 30\ min]$, an input range $n_i = 10$, lead-times $t_p = [0, 1, 2, 4, 8]$, an output range of $n_o = [1, 2, 3]$, a sub-sampling target ratio $p_{0,targ} = 0.5$, and a class '1' neighborhood coverage $n_{cov} = 2$. The same model validation strategy is chosen as for the synthetic data experiment. All classifiers were trained and evaluated.

In Table 6.3, the results are shown for traffic sensor 401390 which is located on the Grove Snaffer Fwy in Oakland (coordinates 37.827454, -122.267769). Note that traffic jams are predicted with F1 scores close to 1 up to four hours in advance. In this experiment with continuous, numeric data, the RF, FCN and FCN2drop classifiers show the best performance. The successful application of our modelling approach to an application from a completely different domain gives confidence in the general validity of the results obtained in previous experiments and underlines the universality of the employed methods.

6.6 Chapter Summary, Conclusions and Outlook

A framework to identify and predict failure mechanisms for situations with limited a priori knowledge on the system behavior is presented. It can handle data sets encountered in realistic settings, such as in particle accelerators, containing multivariate signals, raw data of heterogeneous types, and few observed failures. It uses deep convolutional neural networks to extract predictive failure patterns from historic monitoring data. This allows prediction of faults in real time.

To help system experts and operators interpret the predictions, the most relevant failure precursors are highlighted by Layer wise Relevance Propagation (LRP). Thereby, reaction times to arising problems can be shortened and recurring failure conditions can be removed. This has the potential to improve the availability of future particle accelerators.

The method is validated on synthetic data. It can correctly identify predictive patterns from as few as seven examples for up to 50 input signals. The type of interactions between multiple systems that lead to failures can be reconstructed with the explanations provided by the method.

For data from the PSB accelerator at CERN, the framework predicts power converter faults with F1 scores close to 1. It can do so at a bare minimum of data preprocessing, from fewer than 20 failure examples, and without manual labeling effort of the data. The failure explanations help system experts to narrow the pool of potential failure causes and speed up troubleshooting. The results are encouraging because the historical data of the use case was not collected for predictive purposes. Modern systems collect increasingly more monitoring signals containing failure precursors. This facilitates the application of the introduced method further.

The following research could increase the effectiveness of the proposed approach further:

- Few shot and transfer learning approaches might learn predictive failure patterns from even fewer failure examples.

- Complex infrastructures are evolving systems. Re-learning schemes allow to update predictive models continuously. This renders the approach fit for the long term usage of such a method in evolving infrastructures.
- Using logarithmic time scales on the input data could capture failures happening over particularly long time scales, such as wear-out phenomena.

The method is generally useful to predict and explain discrete events in time series. For applications outside the particle accelerator domain, it only needs adaption of the input- and output data provision and the grid of optimization parameters. Potential examples include identifying root causes of stock market crashes, climate phenomena, or transmission of diseases. Within the field of reliability studies, the method has demonstrated to be useful for the operation of complex particle accelerators and, hence, can be a promising tool for other complex systems.

Chapter Learning Summary

Explainable Deep Learning can help improve the operations of particle accelerators. This requires that fault data as well as data that might contain fault precursors are logged during operation. Data logging objectives should be expanded from explaining faults that occurred to hinting at faults that will occur.

Chapter 7

Data and Knowledge-Driven Parametric Model-Based Reliability Optimization

In the previous chapter the modeling was mainly data-driven as expert and knowledge-based models of the studied system were hardly available. Contrary to the previous chapter, below a scenario is studied in which a priori knowledge on the failure behavior of a system is already available. Such knowledge could be the output of classical reliability analysis carried out by system experts.

In such scenarios information is available in both quantitative (e.g. fault time distribution) and qualitative (expert knowledge on causes of faults) form. This requires flexible, data-effective modeling strategies capable of exploiting all available knowledge. Purely data-driven methods, such as in the previous chapter, are ineffective in encoding qualitative expert knowledge as their model structure is not flexible enough and their model parameters do not directly translate into a physical meaning. A parametric modeling strategy can include both quantitative and qualitative knowledge effectively and transparently. Hence, they are better suited for such scenarios.

Below, such a parametric reliability modeling strategy including a simulation concept is presented, which addresses RQ2. Each of its modules is based on well established reliability concepts which allows translation to and from expert knowledge. This also has the benefit to assess whether such models can predict system behavior under novel operating conditions and even for re-designed systems. In combination with the simulation engine, optimal designs, operational strategies, and maintenance schedules with minimal life cycle cost, can be determined.

The methods, findings and some of the illustrations of this chapter have been published previously in [111, 112].

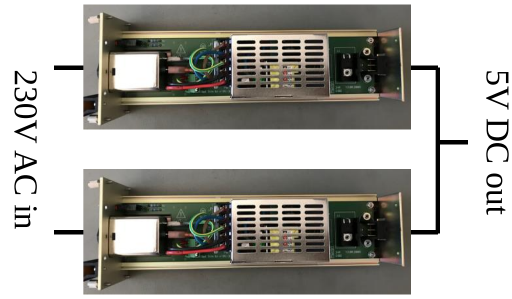


Figure 7.1: Functional diagram of the redundant switch mode power converter with two identical units. [112]

7.1 Scenario Description and Problem Definition

This scenario is concerned with the optimization of a redundant AC-DC power converter operation. It powers machine protection systems at CERN. Hence, it has stringent reliability requirements. A fault tolerant solution was implemented. It consists of two identical units which are operated in a 1-out-of-2 (1 oo 2) parallel redundancy as illustrated in Figure 7.1. It shows the 230V AC (alternating current) input on the left, which powers each of the redundant units which transform the voltage and current to 5V DC (direct current) output. The output is then combined as illustrated in the right of Figure 7.1. The redundant configuration implies that if one of the converters fails, the operation can be continued by the second converter alone. Once one of the converters fails, the faulty unit can be replaced during scheduled breaks of LHC operation. Thereby, continuous availability is aspired.

In fact, close to 400 of the considered redundant powering solutions have demonstrated continuous availability over more than ten years in practice. However, during operations in 2018, an increase of failures in the redundant units was observed. This led to two expert groups independently carrying out Weibull reliability analysis (see [113, 114]) based on the recorded data of the complete fleet of power converters. The objective was to determine if the whole fleet of systems requires replacement within the coming years of operation.

During the investigations, additional relevant questions arose, which could not be answered by the Weibull analysis. These are:

- How does the load need to be distributed between the 1 oo 2 redundant units to achieve the lowest probability of unavailability and lowest life cycle cost based on the data recorded?
- How can an optimal load sharing strategy be derived for any k oo n redundant system under a certain operating condition based on its recorded data?
- How can the operation of a redesigned future system under a certain operating condition be optimized based on recorded data of its predecessor?

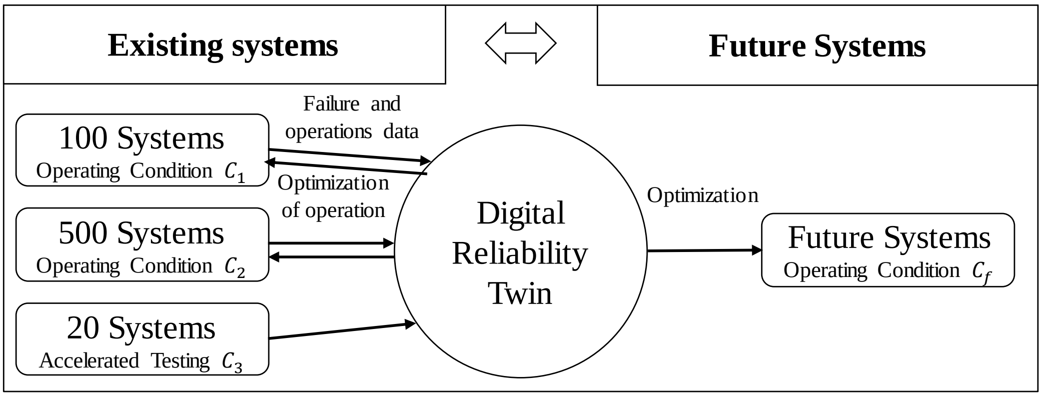


Figure 7.2: Overview of the approach. Data from a system operating at different conditions is combined to form a digital reliability twin, which can be used to optimize existing and future systems under different operating conditions. [112]

In this chapter, a generic approach is developed to address these questions based on the data availability for the power converter example. The approach to address these questions is illustrated in Figure 7.2: A transparent reliability model (Digital Reliability Twin in the center of Figure 7.2) is learned with data and knowledge from existing systems in different operating conditions (on the left in Figure 7.2). Then, the model can be used to optimize new operational scenarios of existing and future systems (on the right in Figure 7.2).

The metric for these optimizations is the life cycle cost Equation 3.11. The equipment cost is the cost of the (redundant) power converter, the repair cost is the cost of replacing one of the redundant units, and the downtime cost is the cost due to interruption of the operation of the powered system. As for all reliability optimization techniques, the earlier available in the system life cycle and the more precise the results are, the bigger is their potential value for decision making. Moreover, due to the scarcity of operational failures, the method has to be suitable for small data regimes. Hence, the developed approach should be data-effective and able to handle uncertainties due to limited data.

7.2 Related Work and Methods Selection

This section discusses the relevant work for the considered approach. These are the digital twin framework and degradation models of load-sharing systems.

Digital Twins The desired objective requires an approach, which integrates data collection, data analysis, modeling, simulation, and cost assessment. A framework, which combines all these steps is the digital twin.

The term digital twin has been coined by Glaessgen et al [115]. They describe it as 'an integrated multi-physics, multi-scale, probabilistic simulation of a complex product and uses the best available physical models, sensor updates, etc., to mirror the life of its corresponding twin...'. With respect to reliability, they note that 'the digital twin will increase the reliability of the [system] because of its ability to continuously monitor and mitigate degradation and anomalous event'. This means that they perceive a prognostics solution to increase the system reliability using the digital twin approach.

The limitations of existing prognostics solutions have been discussed in Chapter 4. These include lack of appropriate data and models to predict system behavior accurately. As the digital twin would rely on such models to increase system reliability, it suffers from the same limitations.

Documented implementations of the digital twin paradigm for reliability purposes are scarce. Tuegel et al [116] and Cerrone et al [117] show how existing methods can be extended with ideas of the digital twin in the aeronautics industry. Reifsnider et al [118] propose a digital twin-based method for composite materials. Gabor et al [119] and Alaswad et al [120] discuss implementation options and potential advantages of digital twins.

Despite the practical implementation challenges of the digital twin, its integrated data collection, modeling and simulation concept is appealing. It aims at coherence between the data collection, usage and the desired decision objectives - similar to the project assessment stage (see Section 5.2) used for implementations of methods in this thesis.

To overcome the disparity between the proposed high fidelity modeling and high resolution sensor updates and the actual availability of reliability models and data in various fields of industries, a more conservative modeling approach can be a first useful step. Methods based on accelerated testing, such as pioneered by Nelson [121, 122, 123], are more appropriate for the data availability considered in this and many other realistic scenarios. They can be combined with a simulation engine and a cost model to achieve cost-effective reliability optimization.

Load Sharing Degradation Analysis and Modeling In the following, relevant existing work in terms of reliability analysis and modeling of load-sharing systems are discussed as it forms the core of a digital twin approach for the considered scenario.

The main challenge in modeling the load sharing of redundant systems is that the loads and stresses within the redundant units are interdependent. A failure in one of the units influences the loads of all other units. Hence, even for applications with constant loads, the load profiles within redundant units are dynamic. Several methods have been proposed to model such dependencies and are discussed below.

Early modeling approaches were limited to exponential failure rates (constant hazard rate) and 1 oo 2 systems, due to a lack of detailed reliability models and insufficient computational performance [124]. Such modeling is too inaccurate to be useful in realistic settings. In such models, the failure probability of the system does only depend on its age and instantaneous load. However, empirical studies showed that the entire load history is relevant for determining the instantaneous failure rate [125, 126, 123].

Over time, more accurate modeling approaches based on the Weibull failure distribution (non-constant failure rates) and the cumulative exposure model [121] were developed [127] for 1 oo 2 systems. Such approaches consider the load history.

More recently, efficient computational procedures to extend such modeling to large k oo n systems have been produced [128]. Multiple failure modes within redundant units were modeled by Huang et al [129]. Modeling approaches which allow the incorporation of qualitative knowledge about the failure mechanism have been proposed by Yang et al [130]. Some methods even consider specific behaviors of redundant units, such as delays due to rebooting [131].

Despite the many variations of reliability modeling of load sharing systems, no approach for a combination of non-constant hazard rates, handling the effect of load histories on multiple failure modes, non-balanced load sharing, and propagation of parametric uncertainty has been reported so far. However, these are requirements to model the considered power converters scenario realistically.

Since most presented approaches are based on methods of accelerated life testing [121, 123], the existing modeling approaches can be extended to arrive at the desired modeling. Combined with a simulation approach and the cost model, as suggested by the digital twin approach, the objectives outlined in the previous section can be addressed. The details will be explained in the next Section.

7.3 Methodology - Parametric Digital Reliability Twins

7.3.1 Overview

The overview of the proposed methodology is given in Figure 7.2. A quantification of system reliability, here called Digital Reliability Twin (in the center of the figure), is learned from existing system under different operating conditions (in the left of the figure) using methods from Accelerated Life Testing (ALT). The models can be continuously updated, once new data is generated during operations. The model can be used to optimize the operation of existing and future systems in new operating conditions (in the right of the figure).

The execution of the approach is based on four steps. These are data collection, digital twin synthesis, simulation, and evaluation and decision making. These steps will be carried out for a use case in Section 7.4.

Below, the elements of the Digital Reliability Twin are introduced. Firstly, the quantitative model behind the Digital Reliability Twin and the required reliability modeling backgrounds and definitions are presented. After that, a Monte Carlo engine to simulate the system over its life cycle is discussed. Then, the cost model for life cycle cost evaluation and decision making is introduced. Finally, the data collection requirements and actual data availability and quality is assessed.

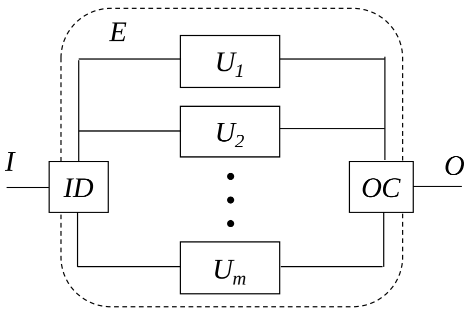


Figure 7.3: Redundant system illustration.

7.3.2 Reliability Modeling of Load Sharing in Redundant Systems

The quantitative model of the Digital Reliability Twin and the required reliability modeling backgrounds and definitions are presented as follows:

1. The relevant definitions for redundant systems are introduced.
2. The modeling of multiple failure modes by Weibull distributions is discussed.
3. The acceleration factor modeling is outlined to describe the effect of system loads on stresses or failure drivers.
4. The cumulative exposure model is introduced to describe degradation under variable stresses.
5. The load sharing of multiple redundant units is discussed.
6. All the previous modeling steps are combined into a hierarchical load sharing model.
7. The hierarchical load sharing model is validated on a benchmark problem from the literature.
8. A summary of the modeling approach is provided.

Definitions Figure 7.3 illustrates a redundant system S. It is composed of an Input Distributor (ID), Output Collector (OC), and n redundant identical units, $U_1, .., U_n$. A system boundary (dashed line) separates it from an operational environment E. The operating conditions are referred to as the collection of environment, input and output, $\mathbf{C} = [E, I, O]$. The system is considered available, if it delivers the specified output O whilst the environment and the input are within specifications. It is faulty if it fails to do so.

Multiple Failure Modes in Redundant Units A failure in a k oo n redundant system occurs when more than $k-n$ units are simultaneously in a failed state, the input distributor is faulty, or the output collector has a failure. In the considered scenarios, the input distributor an the output collector are extremely simple and robust designs. Hence, they are assumed to have no failures.

The reliability of a single Unit U_i at a time t is described by its reliability function, $\bar{R}_i(t)$. Its probability of being faulty at time t is given by $\bar{F}_i(t) = 1 - \bar{R}_i(t)$. Different failure mechanisms j may lead to the failure of a unit. They are modeled as competing risks [37], $\bar{F}_i(t) = 1 - \prod_{j=1}^{M}(1 - F_j)$.

The failure probability due to a single failure mechanism can be modeled by several statistical models [8, 6]. The two parameter Weibull distribution is chosen as it is frequently used in reliability studies and commonly known by system experts. It is given by, $F_j(t; \eta_j, \beta_j) = 1 - e^{(-t/\eta_j)^{\beta}_j}, t > 0$. η_j is the characteristic lifetime (with the property $F_j(t = \eta_j) \approx 0.63212$) and β_j is the shape parameter that indicates a decreasing ($\beta_j < 1$), constant ($\beta_j = 1$), or increasing ($\beta_j > 1$) failure rate with time for a failure mechanism j.

Acceleration Factor Modeling A failure mechanism is activated by a failure driver, $\xi_{\mathbf{j}}$. For example, increasing temperature leads to a faster evaporation of the electrolyte in capacitors. This degrades its capacity. For this example, the failure driver is the temperature and the failure mode is capacity degradation. The stress $\xi_{\mathbf{j}}$ results from system operating conditions, $\mathbf{C}$. Their relation can be expressed by empiric or analytic models, $\xi_{\mathbf{j}} = \Gamma_j(\mathbf{C}; \mathbf{\Lambda})$, with parameters $\mathbf{\Lambda}$. For the capacitor case, such a model would express the capacitor temperature as a function of the operating current, which causes heat dissipation.

Acceleration factor modeling allows to quantify the relationship between the failure driver and the time to failure. With a two-parameter Weibull distribution, the acceleration factor is expressed as, $AF_j(\xi_{\mathbf{j}}, \xi_{\mathbf{j,ref}}; \mathbf{\Theta}) = \eta_{j,ref}/\eta_j$. $\eta_{j,ref}$ is the characteristic lifetime at a reference operating condition[1], $\xi_{\mathbf{j,ref}} = \Gamma_j(\mathbf{C_{ref}})$. Parameters $\mathbf{\Theta}$ define the kind of acceleration model. The acceleration factor model is often expressed by exponential- or power-laws, e.g. $AF_j(\xi_j, \xi_{j,ref}; \Theta) = (\xi_j/\xi_{j,ref})^{\Theta}$. [49, 112]

Cumulative Exposure Model To consider the degradation of the redundant units due to their non-stationary load history, the cumulative exposure model is used [121]. It is based on acceleration factor modeling by summing the load history dependent acceleration factor over time: $AF\left(\xi_{\mathbf{stress}}(t), \xi_{\mathbf{ref}}\right)$. Thereby, an effective system age τ is obtained,

$$\tau(t) = \int_{t'=0}^{t} AF\left(\xi_{\mathbf{stress}}(t'), \xi_{\mathbf{ref}}\right) dt'. \tag{7.1}$$

A system with a failure probability characterized by a two parameter Weibull distribution under reference conditions, $t_{fail,ref} \sim Weibull(\eta_{ref}, \beta_{ref})$, will fail once the effective system

[1]Note that we assume uniform acceleration, i.e. $\beta_{j,ref} = \beta_j$. The methodology can be extended to non-uniform acceleration. However, a changing β_j indicates a changing failure mode and should be divided in separate failure modes.

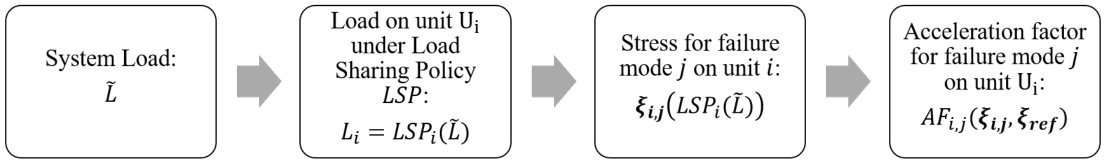


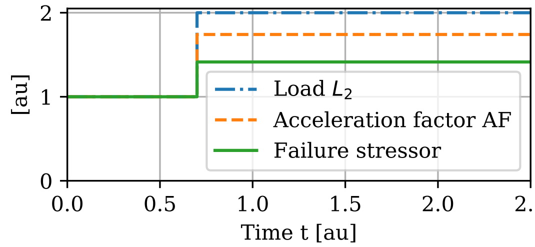
Figure 7.4: Illustration of the proposed hierarchical load-sharing modeling strategy.

age exceeds the the reference failure time drawn from the Weibull distribution, $\tau(T) > t_{fail,ref}$. Thereby, the effect of arbitrary load histories can be assessed when the system failure probability under a reference operating condition, as well as its acceleration factor model, are known.

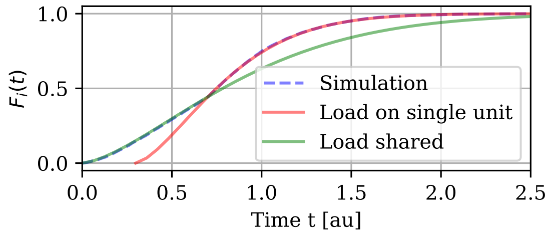
Load Sharing Modeling So far, a relationship between the failure probability of the redundant units and its operating conditions has been established. The missing piece is the relation between the system load $\tilde{L}$ and its influence on the operating conditions. Specifically, the load on each of the redundant units, L_i, has to be determined as a function of the system load and the available redundant units. This is modeled by so-called load sharing policies, $L_i = LSP_i(\tilde{L})$, which distribute the system load among the functional redundant units. A very common load sharing strategy is balanced load sharing, which is also used for the power converter example. Two other sharing strategies are studied: Hot-Spare operation, in which some of the units are in stand-by mode without any load, and imbalanced load sharing, in which the load is distributed unequally among the functional units.

For the considered 1 oo 2 scenario, $LSP_{1:1}$ stands for balanced load sharing with half of the load on each unit, $LSP_{1:0}$ for hot spare operation with the whole load on one unit, and $LSP_{1:2}$ with one third of the load in one unit and two thirds of the load on the other unit.

Hierarchical Load Sharing Model A combination of all the introduced concepts in a hierarchical way leads to a modeling approach to support the decision objectives for the considered scenario. Figure 7.4 shows the resulting hierarchical modeling approach. The system load $\tilde{L}$ is the relevant operating condition at the highest modeling level. A load sharing policy LSP distributes the system load among the redundant units. A parametric model Γ determines the stress ξ for failure mode j on unit i as a function of the unit load L_i. The stress $\xi_{i,j}$ determines the acceleration of the degradation through the acceleration factor $AF_{i,j}$. Finally, the failure time of the unit is determined through the cumulative exposure model and its Weibull failure probability at reference load. Combining all ex-



(a)



(b)

Figure 7.5: Application of the proposed load-sharing and the cumulative exposure model [121] to a 1 oo 2 redundant device: (a) Simulation parameters over time. (b) Illustration of the failure probabilities for unit two over time. The blue dashed line corresponds to the simulated failure probability for the described scenario. The green and red line depicts the failure probability for half or full system load, respectively. Note the increasing failure rate at $t_c = 0.7$. The time difference between the green and red line can be evaluated analytically as $t_s = t_c(1 - 1/AF(\tilde{L}^{0.5}, (\tilde{L}/2)^{0.5})$ [132]. [111]

pressions, the failure probability for a unit can be expressed as

$$\tilde{F}(t, \mathbf{C}; \eta_{j,ref}, \beta_j, \mathbf{\Lambda}, \mathbf{\Theta}) = \\ 1 - \prod_{j=1}^{M} \left[1 - \exp\left(\frac{-\int_0^t AF_j(\mathbf{\Gamma_j}(\mathbf{C}(\mathbf{t'}); \mathbf{\Lambda}), \xi_{\mathbf{j,ref}}; \mathbf{\Theta}) dt'}{\eta_{j,ref}} \right)^{\beta_j} \right]. \quad (7.2)$$

Model Validation The validity of the model is assessed by reproducing the results of variable load failure behavior as reported by Pozsgai et al [132]. It concerns a 1 oo 2 redundant system with a single failure mode, $t_{fail,ref} \sim Weibull(\eta_{ref} = 1, \beta_{ref} = 1.5)$, a power-law acceleration factor, $AF(\xi_{stress}, \xi_{ref}) = (\xi_{stress}/\xi_{ref})^{1.6}$, a nonlinear load-stressor relationship, $\xi_{i,stress} = L_i^{0.5}$, and a balanced load-sharing policy, $L_1 = L_2 = \tilde{L}/2$. The operation of unit 2 is simulated from time $t = 0$ to $t = 4$. At time $t_c = 0.7$, unit 1 is shut down. Hence, for $t > t_c$ unit 2 supports the full load. The resulting load, acceleration factor, and stress profile (y axis) for unit 2 are shown as a function of time (x axis) in Figure 7.5a. The discontinuity of the profiles occurs at $t_c = 0.7$, when unit 1 is shut down.

The simulation is carried out by drawing reference failure times, $t_{fail,ref} \sim Weibull(\eta_{ref} = 1, \beta_{ref} = 1.5)$, from the Weibull distribution above. The unit operates until the effective age τ exceeds the drawn reference failure time, $\tau(t) > t_{fail,ref}$. The effective age is calculated by the cumulative exposure model under the given load. By recording the time t for repeated experiments, the cumulative failure probability $F_2(t)$ under the variable load can be determined. The resulting cumulative fault distribution under the variable load (dashed line) is shown in Figure 7.5b together with the cumulative failure probabilities when assuming the two constant load settings (red and green lines). The first constant load setting assumes a shared load up to $t_c = 0.7$. Thereafter, the second load setting assumes that all the load is on a single unit. The cumulative fault distribution under the variable load follows the failure probability of the shared load setting up to $t_c = 0.7$ and continues along the failure probability of the single unit setting thereafter. The results are identical to those of Pozsgai et al [132], which validates the chosen modeling approach.

Model Summary Despite its overall complexity, the hierarchical model is composed of simple layers of widely used functions in reliability studies. This has the advantage that system experts are more likely to be familiar with it and their knowledge can be translated into parameter values. When their knowledge is not sufficient, parameters can be obtained by consulting scientific literature in which these functions are often used. Finally, the parameters can also be obtained from system failure data through parameter estimation techniques.

For the common situation of partial knowledge of parameter values, it is even possible to mix expert knowledge, related scientific work, and operational data. E.g., in a Bayesian parameter inference scheme, prior distributions on parameter values can be determined with experts and posterior estimates obtained from the available data. This ensures that all available data is used effectively.

Since the operating conditions are explicitly modeled, it is possible to study the system behavior for new operating conditions as long as the acceleration factor and load to stress relationship is valid. Thereby, the operation of the system can be optimized for new operational environments.

As the fault mechanisms are modeled separately, it can be assessed how redesigns of the system affect reliability. Failure modes can be removed or modified individually if a redesign affects them. This allows to optimize the operation of redesigns of existing systems to a certain extent. Moreover, the model addresses the requirements of a combination of non-constant hazard rates, handling the effect of load histories on multiple failure modes, and non-balanced load sharing. The propagation of distribution parameter uncertainty is covered by using an appropriate simulation strategy, which is explained in the following section.

7.3.3 Simulation Engine

The presented hierarchical model allows to determine the failure time of a redundant unit as a function of the system load history. To determine the optimal operational policy of the system, a simulation engine needs to be added. It consists of the following elements:

- Core Model: It simulates the operation of a single system S stochastically for a lifetime, which includes faults, repairs and the load sharing strategy. The corresponding state diagram for a redundant system, such as the power converter, is shown in Figure 7.6. It contains two system states, 'Operation' and 'Down', which indicate if the overall system is available or not. The simulation starts by drawing lifetimes $t_{i,j}$ from the Weibull distributions for each unit i and each failure mechanism j. Whilst in 'Operation', the simulated time t is evolving and damage is accumulated through

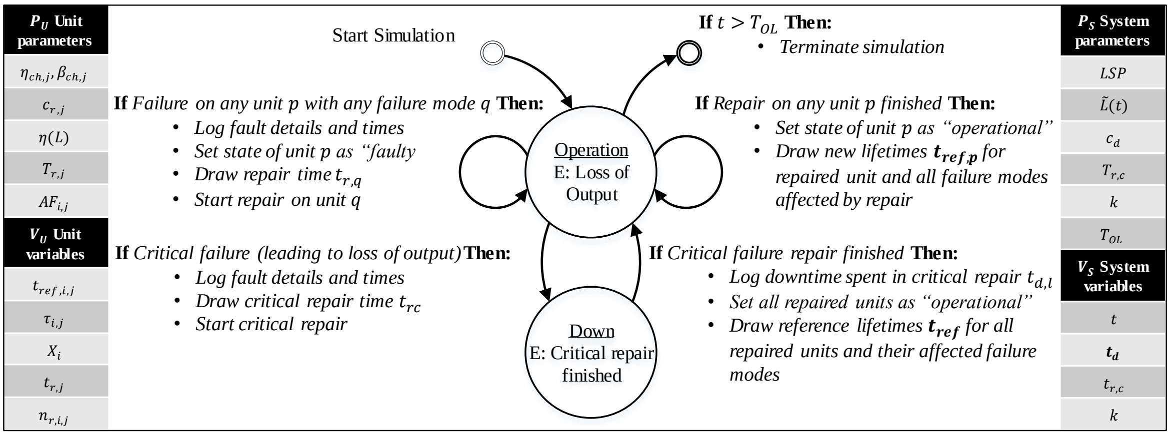


Figure 7.6: Illustration of the core level simulation approach: Simulation parameters are defined as inputs for the simulation; Simulation variables are written during run-time of the simulation; (Middle): State diagram of the proposed simulation strategy and the state transition conditions. [111]

the system loads as specified in the hierarchical model. Once a unit p fails due to a mechanism q, a repair is initiated. It is finished after a repair time $t_{r,q}$, which is drawn from a repair time distribution $T_{r,q}$. During the repair, the load on the remaining functional units is distributed according to the specific load sharing policy LSP. If $n-k-1$ (i.e. a critical amount of) additional units fail before unit p is repaired, the system stops 'Operation' and goes into the 'Down' mode. This initiates a critical repair with a duration $t_{r,c} \sim T_{r,c}$. Once the critical repair is finished, the system returns to the 'Operation' state. The simulation terminates when the simulated time exceeds the operational lifetime of the system, $t > T_{OL}$. After the simulation has finished, all faults and fault times, repairs, and downtime events are reported.

- Core Initializer: The presented hierarchical model is stochastic. That means that a single evaluation of the model would be a sample of a random variable. To get a more complete characterization of the model behavior, many evaluations need to be obtained to observe the distribution of the random variable. The Core Initializer executes many such evaluations of the core model with the same initialization parameters. The results of all simulations can be expressed as distributions and statistics thereof (e.g. the mean and variance). The required number of evaluations for such a Monte Carlo-based simulation can be estimated by monitoring the convergence of statistics of the reported distributions.

- UQ Initializer: The modeling and simulation parameters are often uncertain due to limited data and knowledge. To perform an end-to-end uncertainty-quantification, the Core Initializer is executed multiple times with model and simulation parame-

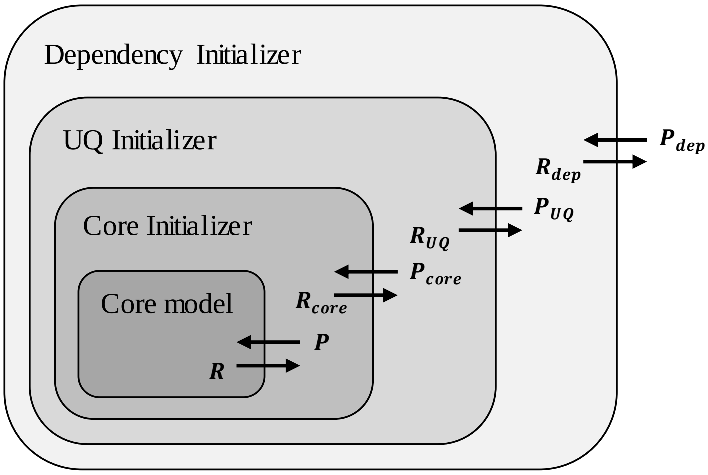


Figure 7.7: Layered simulation approach. [112]

ters drawn from their respective distributions. As before, the number of necessary executions can be determined by monitoring the statistics of the results.

- Dependency Initializer: The last layer allows to study the influence of different operational parameter combinations to assess the influence of different operational strategies. E.g. different load sharing policies, different loads, or different maintenance strategies can be compared.

The overall simulation approach, consisting of the four layers described above, is illustrated in Figure 7.7. The outer layers pass parameters P to the inner layers, whereas the inner layers report simulation results R to the outer layers.

7.3.4 Cost Model

Based on the simulation results (or their statistics), the life cycle cost can be evaluated using equation 7.3,

$$C = n_{eq}c_{eq} + n_r c_r + n_d c_d, \tag{7.3}$$

with n_r being the number of repairs, n_d being the number of downtime events during the system lifetime, and n_{eq} the equipment cost factor to consider additional costs due to redundancy. For the specific optimization objective, the equipment cost c_{eq} is not relevant as it does not change depending on the chosen load sharing policy. Hence, it will be omitted here. The number of repairs and downtime events are provided by the simulation. The repair and downtime cost need to be determined for the considered use case. This will be discussed in the following section.

7.3.5 Data Requirements and Availability

The data assessment is carried out by first stating the data requirements for the proposed modeling approach. Then the requirements are cross-checked with the actually available data.

Data Requirements The envisaged modeling consists of three elements:

1. the hierarchical system failure behavior quantification,
2. the simulation over a system life cycle, and
3. the life cycle cost equation.

The hierarchical model itself is composed of a load sharing policy LSP, a relationship Γ between unit load L_i and failure mechanism stress $\xi_{i,j}$ with parameters Λ, an acceleration factor model AF relating between a reference stress ξ_{ref} and the actual stress, and a two parameter Weibull distribution characterizing the unit failure probability over time at reference stress for each failure mode j.[2]

The simulation model requires a parameterized hierarchical model and a system operation parameters. These are the operational lifetime T_{OL}, repair time (distributions) T_r, critical repair time (distributions) $T_{r,c}$, the system load $\tilde{L}(t)$, other operating conditions $\mathbf{C}(t)$ (temperature, humidity, etc.), and the configuration of the redundant system (k oo n).

Finally, the cost model is based on knowledge of the (average) repair and downtime cost. The cost of the equipment is not relevant in the considered optimization setting.

Data Availability Based on the outlined data requirements, the feasibility of the proposed modeling approach will be evaluated by assessing the actual data availability and quality.

The parameters of the hierarchical model are taken from expert reliability analysis on operational data from more than ten years, totaling 58 Mu-h (Mega unit-hours). The data has been recorded by a system expert who has been in charge for the whole operational period. The data is well documented and consistent [113]. Two independently carried out investigations on the data identified the same Weibull parametrization of the failure behavior [114]. The acceleration model parameters and the load-stress relationship model were developed in additional studies [133, 114] which were presented to expert panels. Therefore, the data availability and quality can be considered sufficient. The only limitation is that it was noted that due to the limited data availability, the parameter estimates are rather uncertain. However, no systematic uncertainty-quantification was performed so far. To address this issue, an ad-hoc sensitivity analysis is performed by assuming that the

[2] When limited knowledge about the specific load-stress relationship is available, it can be skipped by setting $\Gamma = L_i$.

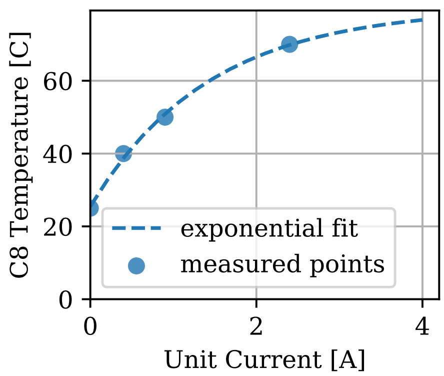


Figure 7.8: Experimentally determined relationship between temperature and load on unit $\xi_j(L_i) = T(I)$. [112]

parameters follow normal distributions with a mean given by the expert-estimated values and a variance of 10% of their mean.

Many of the required simulation parameters are provided in the expert reliability analysis [114, 113]. Only the repair time distributions needed to be gathered separately. This could be done by interviewing the responsible expert for the repair of the system. The repair costs were taken from the expert analysis.

It was noticed that the data collection is simplified by the availability and project participation of the system designer and system maintainer. Moreover, during system design a structured data collection procedure was implemented already. It facilitates the coherent collection of data.

Overall, the available data satisfies the requirements for method implementation. Nevertheless, the results are interpreted with caution, as the uncertainty of the parameters is not estimated systematically. In the following Section, the implementation of the proposed approach and its results for the considered redundant power converter are presented.

7.4 Numerical Experiments

In this Section, the implementation of the described methods and its results for the use case are presented.

The redundant powering solution is composed of two identical 1 oo 2 redundant switch mode power converters as shown in Figure 7.1. The converters have been operated at the LHC for more than 10 years during which 59 faults have been recorded. For each fault, failure times $t_{i,j}$, modes j, and operating conditions $\mathbf{C}$ are known [113]. The $n = (290, 45, 58)$ systems have been operated at constant loads of $\tilde{L} = (0.4, 0.9, 1.2)A$, respectively. The input voltage is 230 V AC and the output voltage is 5 V DC.

The reliability investigations [113, 114, 133] identified three failure modes and their associated Weibull distributions and acceleration factor model. These are summarized in

Table 7.1: Failure mode parameters: The parameters for the acceleration factors for failure mode 1 and 2 were obtained from operational failure data at different constant loads [133, 114]. The acceleration factor for capacitor wear-out was taken from literature [135, 136, 137], whereas, the function relating the temperature for capacitor wear with the current of the unit was obtained experimentally [114] as shown in Figure 7.8.

j	Description	ξ_j	Γ_j	AF_j	$\eta_{j,ref}[d]$	$\beta_{j,ref}$	$\xi_{j,ref}$
1	Fuse wear[134]	$I[A]$	P_{unit}/U	$\left(I/I_{ref}\right)^{1.0}$	19219	1.16	$1.2A$
2	Not investigated	$I[A]$	P_{unit}/U	$\left(I/I_{ref}\right)^{0.6}$	76768	0.9	$1.2A$
3	Capacitor wear[135]	$T[K]$	$T = 55\left(1 - e^{-0.7I}\right) + 298$	$\exp\left(\frac{0.94}{k_b}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right)\right)$	4200	8.3	330K

Table 7.1. Each line corresponds to a failure mode and shows its parameters and characteristics in terms of failure mode number j, stress ξ_j, load to stress relation Γ_j, acceleration function AF_j, characteristic lifetime $\eta_{j,ref}$ in days $[d]$, beta factor $\beta_{j,ref}$, and reference stress $\xi_{j,ref}$. The first failure mode is fuse wear, likely due to repetitive heating and cooling of the AC input [134]. A power law acceleration depending on the electrical current amplitude is identified empirically. The second failure mode is less frequent. Hence, its specific mechanism was not identified but only quantified. It is characterized by current amplitude dependent power law acceleration. Both failure modes show a relatively constant hazard rate ($\beta \approx 1$). The third failure mode shows a strong wear-out behavior ($\beta > 1$) and was investigated closer [114]. It was revealed that a transient voltage suppressor is heating a nearby electrolytic capacitor. The resulting accelerated electrolyte evaporation leads to capacitance degradation. This triggers the failures. The heating was empirically characterized as shown in Figure 7.8 [114]. It shows the temperature of the affected capacitor C8 (y axis) as a function of the unit current (x axis). The blue dots are the experimentally measured temperatures. The dashed line is an exponential function which fits the data trend accurately. The temperature dependent acceleration model for this specific type of electrolytic capacitor was derived by Parler et al [135].

The chosen simulation parameters are discussed in the following. The operational lifetime is set to $T_{OL} = 5000\ d$ (d for days) which corresponds to a lifetime of close to 14 years. The repair after a unit has failed is carried out by a replacement of the unit. The replacement itself is very quick. However, the repair team needs to wait to be granted access to the LHC for non-critical repairs. This can take several days. Based on an expert interview, the repair time distribution is best modeled by a rectified Gaussian distribution, $t_{r,j} = t_r \sim \mathcal{N}^R(\mu = 5\ d, \sigma = 1.5\ d)$. The average cost of a repair is $c_r = 150\ CHF$ [113]. When both redundant units fail simultaneously, the LHC stops and a critical repair can be carried out immediately. It is modeled by a delta distribution, $t_{r,c} \sim T_{r,c} = \delta(3.5\ h)$ (h for hours). The cost of the repair of both units is 300 CHF. Much higher is the cost due to the associated downtime event, which was estimated to be $c_d = 100000\ CHF$ on average.

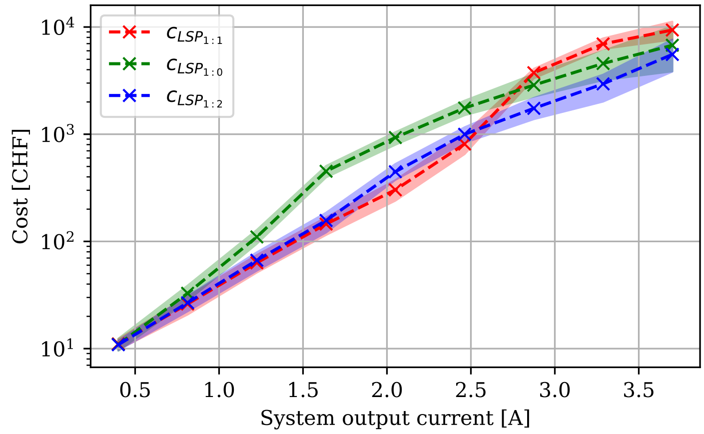
As mentioned before, the systems are operated at three load levels, $\tilde{L} = (0.4, 0.9, 1.2)A$. In the simulation the load level will be varied from $\tilde{L} = 0.4A$ to $3.7A$. Three different load sharing policies are studied: balanced load sharing $LSP_{1:1}$, hot-spare operation $LSP_{1:0}$, and imbalanced load sharing $LSP_{1:2}$. Depending on the parameter combination, up to 10^7 evaluations of the Core Model have been executed to obtain stable estimates of the number of repairs and downtime statistics. Hundred evaluations were carried out on the Core Initializer Level for robust estimates of the uncertainty bounds.

Results The simulated life cycle cost (y axis) as a function of the load sharing policy (balanced load sharing $LSP_{1:1}$ in red, hot-spare operation $LSP_{1:0}$ in green, and imbalanced load sharing $LSP_{1:2}$ in blue) and the system output current (x axis) is shown in Figure 7.9a. The dashed line represents the mean. The shaded area shows the 95% highest probability density interval. The expected mean of the number of repairs and the number of downtime events recorded during the simulation are shown in Figure 7.9b with dash-dotted and solid lines, respectively. Again, the shaded area shows the 95% highest probability density interval. The cost is a linear combination of the the number of repairs and downtime occurrences. Due to the cost discrepancy of a factor of thousand between the repair and the downtime costs, the number of downtime events dominates the overall cost.

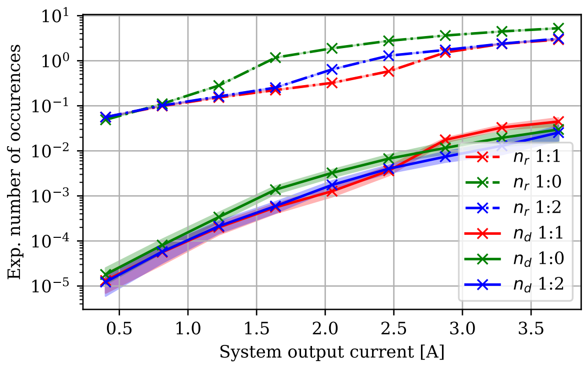
Evidently, the system behavior changes with output load. For low currents ($I < 1\ A$), the cost is similar for all load sharing policies. For intermediate currents ($1\ A < I < 2.5\ A$), balanced load sharing has the lowest cost and hot spare operation results in the highest cost. For high currents ($2.7\ A < I$), balanced load sharing has the highest cost and imbalanced operation results in the lowest cost. Currents, $2.5\ A < I < 2.7\ A$, are in a transition period for which the cost of balance load sharing rapidly increases relative to the other load sharing policies.

Figure 7.9c gives further insight in the frequency of the different failure modes in the imbalanced load sharing scenario. The expected number of failures for each failure mode and unit (y axis) is plotted as a function of the system output current (x axis). Failure modes 1 to 3 are represented by different colors. Generally, the first unit (solid lines), which carries two thirds of the load, has a higher number of failures than the second unit (dashed lines), which only carries one third of the load. Failure modes one and two increase steadily with the output load. Failure mode three is negligible for low currents but dominant for high currents.

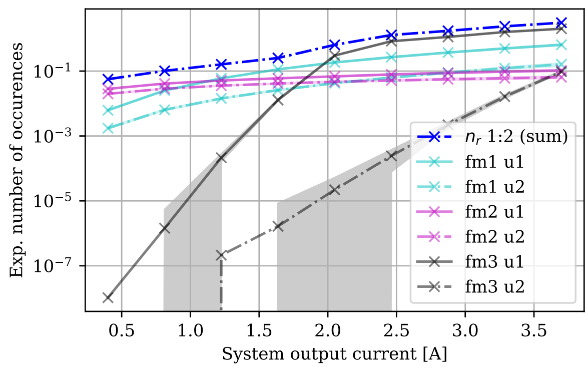
Discussion The system shows non-trivial failure behavior. Depending on the output load, different load sharing policies are cheaper in terms of life cycle cost. Balanced load sharing leads to lowest costs in low and intermediate current scenarios, whereas for higher currents it results in the highest costs. To explain the behavior it helps to remember that the third failure mode (capacitor wear) has a very strong wear-out behavior while the first two failure modes are constant (random) in time. The third failure mode is dominant for high currents. Hence, with higher currents, the units develop a stronger wear-out characteristic. This means that if the load is shared equally among the units, there is a higher chance of simultaneous wear-out and failure, which leads to a downtime event. If the load is not shared equally (or not at all), simultaneous wear-out is much less likely.



(a)



(b)



(c)

Figure 7.9: Final simulation results; Lines represent the expected means and shaded areas the 95% highest probability density intervals: (a) The system cost C in CHF for different load-sharing policies at different system loads (currents). (b) The total number of repairs n_r, and the number of losses of system output n_d for different load-sharing policies at different system loads (currents). (c) The expected number of failures per failure mode for the $LSP_{1:2}$ load-sharing scenario; 'fmx uy' stands for failure mode x on unit y.

That is the explanation for the higher number of downtime events for the balanced load sharing strategy for increased system loads. Such a behavior could only be identified due to the detailed modeling of the failure modes, their mechanisms, and the dynamic load sharing. In a purely data-driven analysis without separation of failure mechanisms, such insights would have been overlooked easily.

The width of the 95% intervals is determined by the parameter uncertainty and the stochastic nature of the problem. The contribution of the stochasticity could be reduced by increasing the number of system evaluations. Due to the low probability of simultaneous failures in the two units for low output loads the statistics of the results are uncertain despite up to 10^7 Monte Carlo evaluations. The contribution of the uncertain input parameters is due to ad-hoc assumption of their uncertainty based on expert judgement. Hence, the actual uncertainty bounds might differ.

Decision Recommendation The original decision objective was to determine the optimal load sharing policy for the power converter system. Based on the simulation results, it is concluded that the optimal load sharing strongly depends on the output load. For the three operational currents, $I = (0.4, 0.9, 2.4)A$, balanced load sharing seems to result in the lowest cost. However, for high currents it might develop into a costly solution. Imbalanced load sharing seems to offer a low cost for all currents. Hence, considering the parametric uncertainty and the limited data, the safest strategy is imbalanced load sharing.

7.4.1 Further Applications

As initially suggested in Figure 7.2, the parameterized model of a single unit can be used to optimize operations of the existing system under new operating conditions or even for redesigned systems. In the example above, the objective was to optimize operations of the existing system. In a follow-up work [112], the same model has been used to optimize the operation of a future system. The use of a single power converter unit in a non-redundant manner was studied for a less critical application with lower downtime cost. The objective was to determine the optimal maintenance strategy. The conclusion was that for low currents, reactive maintenance leads to lower life cycle costs. For higher currents, a preventive maintenance is advisable. The optimal timing of preventive replacements could also be determined for different output loads.

Re-designs of the units could be assessed as well. E.g., it could be decided to use different fuses. Then, the models for the second and third failure mode could be recycled. Potential new failure modes due to the different fuse would have to be identified separately.

If the whole system failure behavior was encoded in a neural network instead of the proposed hierarchical model, it would be very complex, if not impossible, to assess which parameters of the models can be reused if the associated system is re-designed. Furthermore, the parametric hierarchical model allows to derive more generally applicable insights into system degradation. An example is that it could be identified that the optimal load sharing for redundant units depends on their hazard rate characteristics. This illustrates the advantage of a transparent parametric modeling approach.

7.5 Chapter Summary, Conclusions and Outlook

A parametric modeling approach for reliability optimization projects with mixed availability of qualitative and quantitative data has been presented. At its core is a hierarchical model composed of relatively simple and commonly used reliability models, such as the Weibull and acceleration factor model. This modeling approach allows to infer model parameters in a combined manner from expert knowledge, scientific literature, and directly from data. The hierarchical model is combined with a simulation engine and a life cycle cost model. It allows to study the system behavior under previously unseen operating conditions and assessment of the reliability of future redesigns of the system.

The proposed method has been applied to a redundant powering solution to determine the load distribution among the redundant units with the lowest life cycle cost. The results show that the optimal load sharing solution depends on the output load, which influences the hazard rate characteristics. Balanced load sharing has low costs for low output load. For high output currents, balanced load sharing leads to the highest costs as the units have a higher risk of simultaneous failure. This may result in high downtime costs. The imbalanced load sharing results in low costs for all operating loads. Hence, based on the limited certainty of available data and knowledge, it is recommended as the safest load sharing policy.

The parametric hierarchical modeling strategy demonstrated to be useful in settings of mixed qualitative and quantitative information. The same system degradation model can be used to optimize systems and system derivations under current and future operating conditions. Such optimizations would be less effectively carried out with purely data-driven modeling strategies.

Chapter Learning Summary

Expert knowledge and fault data can be combined to improve reliability models. Such models can be used to improve system operation for new operating conditions and redesigned systems.

Chapter 8

Data-Driven Discovery of Organizational Reliability Aspects

The previous two chapters concerned scenarios with a focus on technical aspects of reliability. Chapter 6 aims at discovering fault mechanisms and using them to predict imminent failures. Chapter 7 provided a systematic way to combine expert knowledge and data to develop quantification of failure behavior under different operational settings. Both chapters use information of the system when in an operational state, encoded either in data, documentation, or expert knowledge. To improve the reliability of the systems, suitable technical modifications, e.g. optimal component choice or design, can be derived from the results of applying the methods.

However, the reliability of a system is also affected by aspects such as project management, outsourcing of fabrication, or level of experience of the involved project stakeholders. The methods in the previous chapters do not provide information on the relevance of such aspects for system reliability.

This chapter introduces a method to address such aspects to address RQ3. Based on the field reliability of systems and relevant information of their life cycle, a multivariate parametric model is derived and subsequently trained. It can be used to predict the reliability of future fielded systems and extract the most relevant factors influencing it. This information allows effective organizational decision making early in a system life cycle, which maximizes cost-effective reliability improvement.

This Chapter and the methods and results therein are based on previously published work [138].

8.1 Scenario Description and Problem Definition

It is desirable to be able to predict the reliability of a system in advance to check whether they are expected to fulfill the specified reliability requirements, compare design alternatives, determine the required amount of spares, or calculate expected warranty cost.

The reliability of a system in the field is influenced during all phases of its life cycle.

All technical, human, and organizational processes may have a positive or negative impact on the system reliability. E.g., particle accelerators can take decades from the first idea to their operation. They will be conceived, designed, built, tested, commissioned and operated by generations of engineers with diverse backgrounds. The tasks of each of them may influence the achieved performance of the accelerator during operation.

Modeling of all these processes is impractical. Instead, common reliability prediction methods restrict the modeling to limited aspects of a system (e.g. design, component choice) or phases during the life cycle (e.g. manufacturing, testing). A summary of existing methods is provided in Section 8.2. Since these methods only model certain aspects, their reliability predictions may be misleading, if the modeled aspects do not contain other relevant aspects within a system life cycle. Moreover, they cannot provide a systematic way to quantify the uncertainty of their predictions since potentially important life cycle aspects are not considered.

To overcome this limitation, a statistical model of field reliability is learned for the whole system life cycle based on data from existing comparable systems as illustrated in Figure 8.1. So-called *quantitative reliability indicators* are collected for a group of existing systems (grey box). When their field reliability is known (green box), multivariate regression can be used to establish a relation (orange box) between the quantitative reliability indicators and the achieved field reliability. Such a model allows to predict the field reliability of new systems and to identify the relevant factors during a system life cycle. Using appropriate regression methods, the uncertainty of predictions and influencing factors can be quantified.

To measure the predictive performance of the methods common regression metrics can be used. They can be compared against the error of traditional reliability prediction methods. Few studies have systematically evaluated the discrepancy between predicted and actual reliability of systems. Jones et al [139] conclude that reliability predictions can deviate by orders of magnitude between different traditional methods and the actual field reliability. These errors emerge because the used methods model the reliability based on the selection of components and ignore other relevant aspects, such as design considerations, manufacturing, or supplier selection.

Such imprecise reliability predictions may even mislead design choices. Hence, reliability prediction has to be considered highly stochastic and difficult and an uncertainty quantification is essential for such problems.

In this chapter, a reliability prediction use case of power converters in particle accelerators is studied. It is shown that with the proposed approach the field reliability can be predicted accurately with few reliability indicators, which are available early in a system life cycle. This allows to use the approach early in a system life cycle when it is potentially most valuable. The workload is greatly reduced in comparison to other reliability prediction methods, as the modeling does not need to be carried out manually because it is automatically inferred from the available data. With appropriately chosen reliability indicators and regression methods, the influence of each reliability indicator is quantified. It is shown that non-technical factors show a strong impact on the field reliability. Such information can help improving reliability cost-effectively on an organizational level.

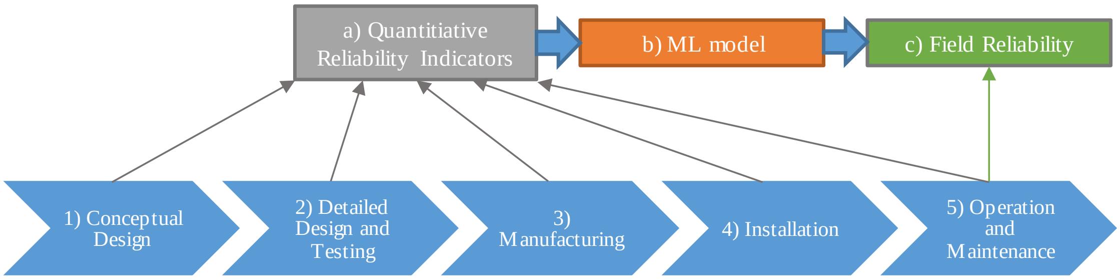


Figure 8.1: Illustration of the proposed approach. The achieved field-reliability (c) can be seen as the result of relevant processes during the whole product life cycle (1-5). It is not feasible to capture and model all of the relevant processes. Instead, it is proposed to learn a reduced-order statistical life cycle model (b) with machine-learning algorithms based on *quantitative reliability indicators* (a). [138]

8.2 Related Work and Methods Selection

Reliability predictions have evolved over recent decades [140]. IEEE standard 1413 and its successors give an overview over the numerous existing approaches towards reliability prediction [141, 142]. The standard proposes a classification by

- handbooks,
- stress and damage models (also called physics-of-failure-based), and
- field data.

These groups of methods are discussed in the following.

Handbook-Based Methods Handbook methods use catalogued reliability data for components to form a reliability estimate. A major criticism is that they do not consider interactions between components and their configuration but only fault frequencies of single components. Numerous studies have found that single-component failures only make up for a fraction of failures in the field [143, 140, 144, 145, 146]. This leads to deviations between predicted and actual field reliability by orders of magnitude [139]. Denson et al argue that handbook based methods should only be taken as preliminary estimates for early life cycle stage design choices.

Stress and Damage Model-Based Methods Stress and damage models yield more accurate reliability estimates than handbook based methods [147]. They are based on an understanding of the failure mechanisms in a system and how they are influenced by operational conditions. The method and use case from Chapter 7 are an example of a stress- and damage model. Although superior in the predictive performance, model generation requires much more modeling and data collection effort.

Field Data-Based Methods Methods based on field data estimate the reliability of a future system based on experience with similar systems and a similarity metric [148, 149]. Their performance mostly depends on the method to derive the similarity metric and the availability of reliability data of similar systems. The method proposed in this chapter could be seen as field data-based, whereas it derives the similarity metric automatically from the available data.

Miller et al [150] evaluate the likelihood of a system achieving a target reliability by reviewing its design process. They determine a score depending on the design steps carried out and show that it correlates with the probability of achieving the reliability target. This allows to include organizational aspects. The method proposed in this chapter takes a similar approach except that it directly estimates the expected reliability and identifies the most appropriate scoring method from the data.

Groen et al [151] developed a Bayesian framework to predict reliability of new systems based on similar systems and expert judgement. Their framework quantifies predictive uncertainty. However, it involves an iterative approach with manual data input which is dependent on expert judgement. The proposed method of this Chapter does not require manual input beyond data collection as it infers all relevant dependencies from the supplied data.

The studies of Miller et al [150] and Groen et al [151] demonstrate that reliability can be forecasted accurately with methods based on field data. The proposed approach extends these methods by automatically extracting significant factors that influence the reliability from the data, quantifying uncertainty of reliability predictions and its influencing factors, and being able to include all potentially relevant aspects during a system life cycle. It is described in detail in the following section.

8.3 Methodology - Statistical System Life Cycle Models

This section introduces relevant definitions, the mathematical formulation of the problem, the data collection, the model selection process, and the model learning algorithms.

8.3.1 Definitions

System Reliability Measure The method aims to predict a metric which expresses the reliability of the system. E.g., this can be a reliability function or the remaining useful life. For this scenario, repairable systems are considered. Therefore, we use the availability A as reliability measure as defined in Equation 3.7 based on the $MTBF$ and $MTTR$. The

$MTBF$ is calculated by

$$MTBF = \frac{t_{operation}}{n_{faults}}, \tag{8.1}$$

with $t_{operation}$ being the cumulative operational time of the considered repairable system and n_{faults} being the total number of faults within the operational time. The $MTTR$ is evaluated by

$$MTTR = \frac{t_{inrepair}}{n_{faults}}, \tag{8.2}$$

with $t_{inrepair}$ being the total time a system is in repair and n_{faults} the total number of faults during the operational time.

System Definition In this scenario, the considered systems are power converters of magnets in particle accelerators. The method is not restricted to certain types of systems as long as they can be assigned a reliability measure and a life cycle.

8.3.2 Approach

The method assumes that the achieved field reliability of a system is the result of all the processes in all phases of the system life cycle. Statistical models approximate the relation between all relevant processes and the achieved field reliability. The statistical models learn this approximation from observed reliability data and reliability indicators that describe the life cycle of a fleet of comparable systems. Inspection of the generated models allows to identify the most relevant factors that influence the field reliability of systems. Uncertainties, due to the approximation and the intrinsic stochasticity of the reliability prediction problem, can be quantified with appropriate modeling strategies, such as Bayesian regression methods.

Mathematical Formulation

One can hypothesize a deterministic model, $\mathbf{\Phi} : \mathcal{Z} \mapsto \mathcal{Y}$, that determines the actual field reliability of a system $\mathbf{Y} \in \mathcal{Y}$ based on data of all relevant processes during a system life cycle $\mathbf{Z} \in \mathcal{Z}$:

$$\mathbf{Y} = \mathbf{\Phi}(\mathbf{Z}). \tag{8.3}$$

Such a model is impossible to obtain since only a fraction of all relevant processes can be captured and modeled. Hence, a statistical model approximates the actual field reliability of a system,

$$\mathbf{Y} \approx \mathbf{y} = \phi(\mathbf{x}), \tag{8.4}$$

with $\mathbf{x} \in \mathcal{X}, dim(\mathcal{X}) \ll dim(\mathcal{Z})$ being the set of collected reliability indicators and $\phi : \mathbf{x} \mapsto \mathbf{y}, \mathbf{y} \in \mathcal{Y}$ being the statistical model. Supplying pairs of input and output data, $\mathcal{D} = \{(\mathbf{x_1}, \mathbf{Y_1}), ..., (\mathbf{x_N}, \mathbf{Y_N})\}$, a ML algorithm can identify such a statistical model through regression.

Three properties make a learning algorithm better suited for the reliability prediction and explanation purpose:

1. To identify the factors which influence the field reliability, the relevance of each of the input reliability indicators $x_{i,j}$ should be quantified with a relevance measure ρ_j, where i indicates the system and j the different reliability indicators.

2. Probabilistic methods allow to quantify the uncertainty of the reliability prediction and the relevance of the reliability indicators:

$$p(\mathbf{Y}|\mathbf{x}). \tag{8.5}$$

3. Methods that identify sparse models, which require fewer input reliability indicators are preferred. This reduces the data collection effort of reliability indicators. The preferred use of sparse models can also be generally justified by Occam's razor [152].

Data Collection

Some data collection guidelines are presented in the following. These refer to the selection of systems and reliability indicators.

Collection of Training Systems The method assumes that the reliability of a new system can be extrapolated from observed reliabilities of existing comparable systems. Hence, a set of comparable systems has to be chosen properly. Following guidelines can be given for this selection:

- The reliability metrics and reliability indicators have to be determined in a consistent and identical way for all systems.

- Systems must have been in use for a sufficient time so that their (statistical) reliability metrics have stabilized.

- The set of systems should be comparable in terms of system type and system usage.

Collection of Reliability Indicators A model can only capture the relevant factors that influence the reliability when the reliability indicators are chosen appropriately. Following recommendations can be given:

- Differences within the set of systems should be made explicit by the reliability indicators. E.g., if a fleet of cars consists of SUVs and sports cars, the car type, weight, dimension, power, etc. should be captured. Otherwise, the method will not be able to identify relevant dependencies.

- System experts and operators, as well as system life cycle managers and project coordinators, can provide relevant technical and organizational indicators based on their experience. In terms of technical indicators, e.g. different operational environments can affect reliability. In terms of organizational indicators, the choice of suppliers or the production volume may affect reliability.

- Data collection and validation can be a time intensive process, especially when reliability indicators are based on non-numeric engineering documentation. At the same time, the prediction of field reliability is a stochastic process with large uncertainties. Hence, it is recommended to start collecting those reliability indicators with the highest expected information content at lowest possible collection effort. Once adding more reliability indicators does not improve model predictions, data collection can be stopped.

- When few reliability indicators are missing for certain systems, appropriate methods for missing data, such as the expectation maximization algorithm, can be used.

- Reliability indicators are available at different life cycle phases. E.g., the specifications of a system are already available during the conception phase, whereas the chosen manufacturing technology might only be available towards the end of the design phase. Depending on the phases of the reliability indicators that the predictive model uses as input, the model can be used earlier or later for predicting the reliability of a new system.

8.3.3 Model Selection and Validation

The collected data can be arranged in a supervised data-set, $\mathcal{D} = \{(\mathbf{x_1}, Y_1), ..., (\mathbf{x_N}, Y_N)\}$, with $\mathbf{x_i}$ and Y_i being the collected reliability indicators and the field-reliability measure for each system, respectively. Based on the data set, a predictive model can be identified with appropriate algorithms. To select the best suited model and its (hyper-)parameters, a nested k-fold cross-validation approach is chosen, similar to the proposition of Hastie et al in Chapter 7 of [51]: The data set $\mathcal{D}$ is split in a training $\mathcal{D}_{train}$ and test set $\mathcal{D}_{test}$. The splitting is carried out so that the training set contains systems older than a threshold age a_s and the test set systems younger than the threshold age. This ensures that model testing mimics the prediction of the reliability of future systems and ensures that no bias in performance estimation is introduced.

The training set is used for model selection. In an outer five-fold cross-validation, different algorithms are compared. In an inner five-fold cross-validation, for each of the outer five folds, different (hyper-)parameters for each algorithm are compared and optimized [153]. The mean and variance of the cross-validation mean squared error, Err_{CV}, are reported and later cross checked with the mean squared error on the test set, Err_{test}.

By inspecting the relevance measure ρ and the model structure it can be inferred which reliability indicators have the strongest influence on the field reliability. Probabilistic learning algorithms can quantify the uncertainty of the prediction error and the relevance

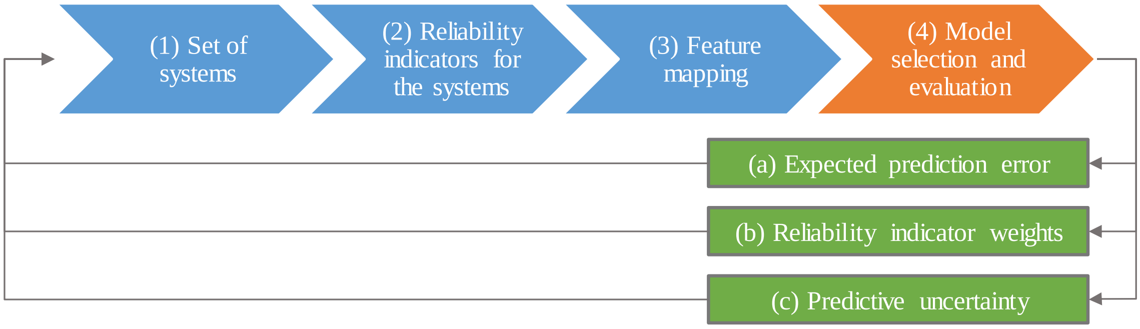


Figure 8.2: Illustration of the iterative data collection and reliability prediction process. The choice of (1) systems, (2) reliability indicators and (3) feature mappings influences the quality of the predictive model (4). The learning algorithm provides feedback in the form of an expected prediction error (a), relevance weights for the reliability indicators (b) and uncertainty bounds for the field-reliability predictions (c). [138]

measure. The results are discussed with system and project stakeholders. If the results are not sufficiently accurate or if no conclusions can be derived, the data collection can be refined as illustrated in Figure 8.2 [1]: The choice of (1) systems, (2) reliability indicators and (3) feature mappings influences the quality of the predictive model (4). The learning algorithm provides feedback in the form of an expected prediction error (a), relevance weights for the reliability indicators (b) and uncertainty bounds for the field-reliability predictions (c). Based on the feedback, it can be decided to refine the choice of systems, reliability indicators, or feature mappings.

Model Testing

The selected and validated models are tested on the full data set. 5-fold cross-validation is used to determine hyperparameters with the training data set. The test error is evaluated on the test set and compared to the cross-validation error from the model selection process. If the test error is within two standard deviations of errors as predicted by the cross-validation, the model is expected to predict the reliability of a future power converter with the same order of error.

The overall data collection, model selection and reliability prediction process is summarized in the pseudo-algorithm below. The use case in Section 8.4 follows the presented procedure closely.

Pseudoalgorithm illustrating the overall model selection and reliability prediction process:

1. $\mathcal{D} = \{(\mathbf{x_1}, \mathbf{Y_1}), ..., (\mathbf{x_N}, \mathbf{Y_N})\} \leftarrow$ `Initial data collection.`

[1] The number of data collection refinement iterations should be limited to avoid biasing of data collection and results.

2. Sort $\mathcal{D}$ by system age.
3. Split $\mathcal{D}$ in $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ with $a_{test} < a_s \leq a_{train}$.
4. Model Selection and Validation:
   (a) Shuffle $\mathcal{D}_{train}$ randomly.
   (b) Evaluate Err_{CV} by (nested) CV.
   (c) Evaluate parameter weights $\mathbf{w}$ and predictive uncertainty for one fold.
   (d) **If** Model has large Err_{CV} or predictive uncertainty **then**
      - Change set of systems, reliability indicators, or feature mapping based on expert discussions and jump to 4.
5. Model Testing:
   (a) Train predictive model with $\mathcal{D}_{train}$.
   (b) Evaluate Err_{test} and compare with Err_{CV}.
   (c) Evaluate parameter weights $\mathbf{w}$ and predictive distributions.

Algorithms

A vast range of algorithms for supervised regression problems exists. The requirements in terms of uncertainty-quantification, reliability indicator relevance, and sparsity narrow down the selection. A summary of the algorithms and their characteristics with respect to the requirements is presented in Table 8.1. The shown selection of algorithms is based on their popularity for different problem domains, their simplicity, and their expected suitability for stochastic problems. The characteristics are listed in the columns and are uncertainty quantification (UQ), provision of a relevance measure, producing sparse models, and global or local models. ARD and BAR algorithms fulfill all criteria.

Table 8.1: Summary of learning algorithms. Adapted from [138].

	UQ	**Relevance Measure**	**Sparsity**	**Global/Local**
ARD	yes	yes	yes	Global
BAR	yes	yes	yes	Global
GP	yes	no	no	Local
ENCV	no	yes	yes	Global
SVR	no	only for linear kernel	no	Local

By formulating the reliability prediction problem as a supervised ML problem, we can choose from a range of existing learning algorithms to generate the desired statistical model for predictive purposes. Since the uncertainty in the field-reliability predictions shall be quantified (i.e. finding a model as presented in equation 8.5), the choice of algorithms is narrowed down. Furthermore, sparse models are preferred since they potentially require fewer reliability indicators to be collected and - more importantly - since they allow an estimation of the relevance of the choice of reliability indicators and the generated features.

Table 8.1 summarizes the chosen algorithms. The implementation from sklearn have been used [154]. Detailed descriptions of the methods can be found in their user guide [106]. A summary of each algorithm is given below:

- ARD - Automatic Relevance Determination Regression: A sparse Bayesian regression method as introduced in [155] - Chapter 7.2.1. The implementation is described in [106] - Chapter 1.1.10.2.
- BAR - Bayesian Ridge Regression: A Bayesian regression method [156]. It resembles ARD Regression except for a simpler and less flexible parametrization of uncertainty which leads to fewer parameters that have to be learned from the data. It is described in [106] - Chapter 1.1.10.1.
- GP - Gaussian Process Regression. A Bayesian Regression technique using the kernel trick [157]. Its implementation is outlined in [158] - Algorithm 2.1 and was adapted from [106] - Chapter 1.7.1. A combination of a radial-basis-function kernel and a white-kernel is used. The kernel parameters are optimized during training.
- EN: Elastic Net Regression. A simple regression technique with regularization as described in [106] - Chapter 1.1.5. Hyperparameters are determined in a cross-validated grid-search.
- SVR - Support Vector Machine Regression: A regression method using the kernel trick as described in [106] - Chapter 1.4.2. Linear basis functions are chosen and a cross-validated grid-search is used to determine the hyperparameters.

8.4 Numerical Experiments

In this section, the proposed approach is tested for a real use case. The goal is to find a model to predict the reliability of magnet power converters used at CERN and to identify relevant factors influencing their reliability. The section is composed of an assessment of data requirements and actual data availability and quality, a set of model learning tasks, their results, and a discussion.

8.4.1 Data Requirements and Availability

Data Requirements

The data collection requirements are outlined in the Data Collection subsection of Section 8.3. Most of its requirements are optional but leading to a better performance of the

Table 8.2: Illustration of characteristic power converter attributes of the studied dataset. [138]

	Power [W]	**Current [A]**	**Voltage [U]**	**Age [yrs]**	**MTBF [hrs]**
Minimum	10^{-6}	10^{-4}	10^{-3}	2.2	10^{3}
Maximum	10^{8}	$4 \cdot 10^{4}$	10^{5}	49.7	$6 \cdot 10^{5}$

method and easier interpretation of its results. Compulsory requirements are that reliability indicators and metrics are available for a set of systems, obtained in a coherent and identical way, and based on sufficient operational experience so that they have stabilized.

In the following the collected data for the considered use case is described and whether it fulfills the stated requirements is discussed. These are the chosen set of system types, reliability indicators, and reliability metrics.

Set of Systems More than 6000 power converters of around 600 different types are used at CERN. A centralized computerized maintenance management system helps to track their field reliability. Of the 600 converter types, 300 have a cumulative operation time of more than ten years and consistent data recordings. The remaining 300 are removed. An overview of minimal and maximal attributes of the 300 different selected converter types are shown in Table 8.2 in terms of rated power, current, voltage, age and MTBF. It shows that the data set contains a wide range of different power converters.

Reliability Indicators The selection of reliability indicators is based on recommendations of CERN engineers and project managers in charge of systems for complete life cycles:[2]

1. Rated current of the converter (I). It influences the choice of technology for power conversion. Current causes heating in systems which can be handled by appropriate thermal management [8, 6].

2. Rated voltage of the converter (U). Arcing or corona discharge can be caused by high voltage. It requires the appropriate electrical insulation [8, 6].

3. Rated power of the converter (P). Power is the product of current and voltage. Hence, both above considerations can be valid for high powers.

4. Quantity of each power converter per converter type used at CERN (Quantity). When converters are produced in higher quantities, they tend to be handled differently during life cycle phases. It may impact the reliability even though it is not related to any physical failure mechanism.

[2]The terms in brackets correspond to the acronym later used in figures to express the relevance of each reliability indicator.

5. The average age of converters per type (Avg. Age). The age often has a strong influence on the reliability of systems due to wear-out mechanisms.

6. The cumulative age of converters per type (Cum. Age). Similar to the average age in term of wear-out mechanisms. However, the availability might also increase with cumulative age of systems as the organization learns to mitigate system deficiencies.

7. The polarity of the converter which indicates the operating modes, technology and complexity of the converter (Pol 0-9).[3]

8. The particle accelerator in which the converter is used (Acc. 1-9). Different accelerators have different operational environments in terms of radiation levels, operational schedules, and maintenance strategies.

9. The count of different particle accelerators in which each converter is used (in Acc.).

Reliability Metrics The availability of the systems are expressed in terms of their $MTBF$ and $MTTR$ which are calculated from the data of the Computerised Maintenance Management System (CMMS). To ensure validity of the data the raw fault and repair logs were checked for converter types with conspicuous data. Moreover, converters with incomplete data or too little operational experience were removed.

The full data set $\mathcal{D}$ contains 281 different converter types with nine reliability indicators (see above) and two reliability metrics each, their $MTBF$ and $MTTR$.

Data Availability and Quality Assessment

The data fulfills the above stated minimal requirements: reliability indicators and metrics are available for the set of systems, are obtained in a coherent and identical way, and are based on sufficient operational experience.

The data was selected based on discussions with experts. However, not all recommended indicators could be obtained for the several thousand converters. For example, operational temperatures might be relevant, but could not be collected. Nevertheless, the collected indicators capture the main differences between the converter types. The majority of indicators is available at early life cycle phases. Hence, any predictive model based on these indicators can be used to predict the field reliability of a new converter early in its system life cycle.

8.4.2 Model Selection and Validation

The full data set is split into a training set $\mathcal{D}_{train}$, containing 210 converter types older than fifteen years, and a test set $\mathcal{D}_{test}$, containing 71 types of less than fifteen years. Hence,

[3] The discrete set of polarities is given by: (1) Unipolar, (2) Bipolar Switch Mechanic, (3) Bipolar I - Unipolar U - 2 Quadrants, (4) Unipolar I Bipolar U 2 Quadrants, (5) Bipolar Pulse-Width-Modulation, (6) Bipolar Relay, (7) Bipolar Electronic I/U, (8) Bipolar Anti-Parallel 4 Quadrants, (9) Bipolar I-circulation 4 Quadrants and, (0) un-specified or other Polarity.[138]

the test setting assumes that the prediction was carried out fifteen years ago based on the operational experience up to that point in time.

As part of the model selection and validation process, the input data of the training folds was normalized to a mean of zero and unit variance, as some of the selected learning algorithms function better with scaled input data. The resulting scaling operator is later applied to the test folds. The logarithm of the reliability metrics, $\log(\mathbf{Y})$, is taken as output data for numerical purposes.

The following different configurations in terms of set of converter types, reliability indicators, and reliability indicator mappings are compared:

- Choice of converter types: The complete set of power converter types and a random sub-selection of only 42 converter types is compared. The goal is to assess whether the uncertainty of predictions and reliability indicator relevance increases for a smaller data set.

- Choice of reliability indicators: Models trained with the full set of reliability indicators are compared to those trained on a set without the quantity of converters per type. The goal is to see whether removing an important indicator can be compensated and whether it can lead to different explanations of relevant factors for field reliability.

- Choice of reliability indicator mappings: Following features are generated based on the reliability indicators:

  – Linear features and logarithmic features are generated from numeric indicators $\mathbf{x_{num}}$ (indicators 1-6 and 9) - $\Phi(\mathbf{x_{num}}) = [\mathbf{x_{num}}, \log(\mathbf{x_{num}})]^T$ (resulting in 7+7 values).
  – Categorical indicators, for the polarity of the converters and the particle accelerators in which the converter is used, $\mathbf{x_{cat}}$ (indicators 7 and 8) are encoded into binary features (resulting in 10+9 values).
  – An additional constant set to 1.

  This resulted in an input vector containing 34 values. Two mappings of this vector were chosen as input data:

  – First order mapping: The input vector without additional transformation (except for the scaling operation).
  – Second order mapping: A second order feature mapping to account for nonlinear interactions between the reliability indicators,

  $$\Phi(\mathbf{x_{num}}) = \left[\mathbf{x_{num}}, \log(\mathbf{x_{num}}), [\mathbf{x_{num}}, \log(\mathbf{x_{num}})] \cdot [\mathbf{x_{num}}, \log(\mathbf{x_{num}})]^T\right]^T.$$

  This results in 629 values for the input. The goal is to test whether a more accurate model can be generated when interactions between indicators are already modeled at the level of the input data and whether the relevant factors influencing the field reliability can still be retrieved.

The results of the model selection and validation are reported for each of the mentioned configuration in the following. The cross-validation mean squared error Err_{CV} and its variance on the hold-out sets is presented in tabular form and the predictions ($MTBF$ and $MTTR$) and the reliability indicator relevance ρ are plotted for the last cross-validation fold. Only the predictions of the BAR algorithm are plotted to reduce the number of overall plots. BAR performs well across prediction tasks, produces sparse models, and quantifies uncertainty of predictions and indicator relevance. The predictive performance of the algorithms can be assessed from the results Tables.

Reference Configuration The first configuration studied uses the complete set of converter types, all reliability indicators and the first order mapping. Applying the framework to the data set with the introduced model selection and validation scheme yielded the validation Meas-Squared-Error (MSE) shown in Table 8.3a for the $MTBF$ and in Table 8.4a for the $MTTR$. It is observed that all algorithms (shown in different columns of the table) yield similar errors.

The obtained reliability predictions and their 95% highest probability density interval on the hold-out set of the last validation fold[4] of the BAR algorithm are shown in Figure 8.3a for the $MTBF$ and Figure 8.3c for the $MTTR$. The converter types (denoted System on the x-axis) are ordered by increasing predicted $MTBF$ or $MTTF$, respectively. The orange line depicts the mean of the predictive distribution and the orange shaded area the 95% confidence intervals. The blue dots mark the actual observed field-reliabilites. It is observed that for the $MTBF$ the variation could be captured accurately. However, the $MTTR$ model does not manage to identify any relevant variations.

The obtained reliability indicator relevance (feature weight) of the last cross-validation fold is displayed in Figure 8.3b for the $MTBF$ model and in Figure 8.3d for the $MTTR$ model. The x-axis displays the different feature as obtained by the first order mapping of the collected reliability indicators. It is observed that similar indicators are relevant across different algorithms. The logarithm of the quantity of converters produced per type $\log(Qty)$ is the dominant factor.

It is very interesting to note that the logarithm of the quantity of converters per type is the most relevant factor in achieving an accurate reliability prediction. Obviously, the quantity is only a correlating and not a causal factor. Nevertheless, this hints at non-technical factors being relevant for the reliability of magnet power converters in this use case. A closer investigation might reveal causal factors, which are overlooked in technical analysis.

The fact that all models yield similar errors and assign similar relevance to similar factors is giving confidence in the results. It also suggests that due to the randomness of the prediction task, sophisticated algorithms do not lead to better results. Hence, it was not studied to use deep learning methods to improve the predictions.

As the $MTTR$ models did not identify any dependencies they are not discussed further.

[4]The size of the hold-out set correspond to one fifth of the training set of 210 converters. This is 42, which is by chance the same amount as the reduced set of training systems but should not be confused.

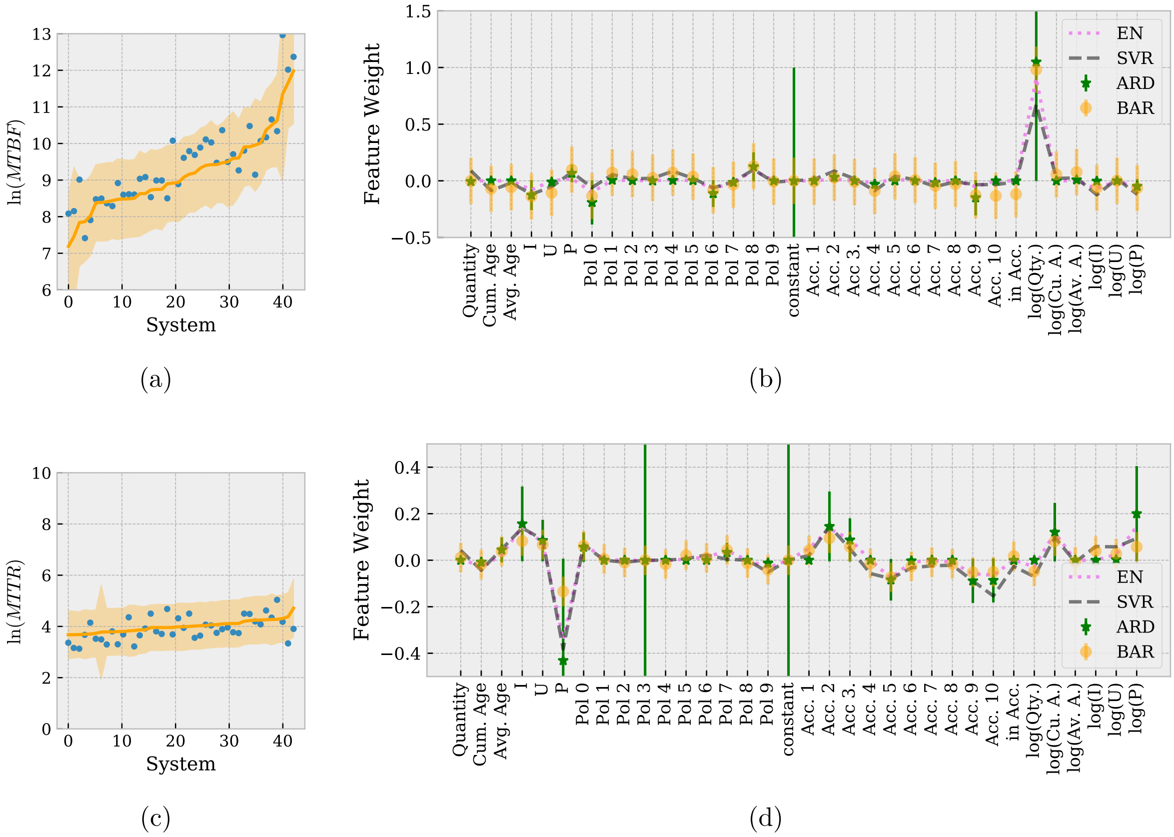


Figure 8.3: Results for the reference configuration. (a),(c): Prediction of the $\log(MTBF)$ and $\log(MTTR)$ by the BAR algorithm for the last fold of the cross-validation procedure. The orange line depicts the mean of the predictive distribution and the orange shaded area the 95% confidence intervals. The blue dots mark the actual observed field-reliabilites. Note that the converter types on the x axis of the last cross-validation fold were ordered by the mean of the predictions to recognize whether trends are properly captured. (b),(d): Estimated feature weights for the parametric models. [138]

Reduced Set of Training Systems In the second configuration of only 42 randomly sub-selected converter types, it is tested whether a reduction of the data set affects the identified models and their predictive and parametric certainty. The obtained validation MSEs Err_{CV} in Table 8.3b for the $MTBF$ and Table 8.4b for the $MTTR$ are larger than the values obtained with the reference data set. Nevertheless, the reliability indicator relevances (Figure 8.4b for $MTBF$ and Figure 8.4d for $MTTR$) are identified similarly to the reference configuration. Variations in the $MTBF$ are captured accurately again (see Figure 8.4a). Variations in the $MTTR$ are not properly captured (see Figure 8.4c), as was the case for the reference configuration.

The biggest difference appears in the certainty of the predictions (orange shaded area in Figures 8.4a and 8.4c) and relevance measures (uncertainty bars in Figures 8.4b and 8.4d), which are much larger than in the reference configuration. This confirms the expected behavior, as the algorithms have to learn from much fewer data.

Reduced Set of Reliability Indicators In the third configuration, it is tested whether omitting relevant reliability indicators can be compensated and whether it suggests conflicting conclusions in terms of relevant factors influencing reliability. To do so, the logarithm of the quantity of converters per type $\log(Qty)$ is removed.

The resulting MSEs for the $MTBF$ (Table 8.3c) are almost three times those of the reference configuration, whereas the MSEs of the $MTTR$ (Table 8.4c) increased slightly. The $MTBF$ reliability indicator relevance (Figure 8.5b) do not identify a single dominant factor anymore; the most relevant being the logarithm of the rated power $\log(P)$ with a negative influence on the reliability. It did also have a negative influence for the reference configuration. The $MTTR$ reliability indicator relevance (Figure 8.5d) are comparable to the ones identified for the reference configuration. For the predictions (Figure 8.4a for the $MTBF$ and Figure 8.4c for the $MTTR$) it is observed that in neither for the $MTBF$ nor for the $MTTR$ the variations are properly captured and that the uncertainties increase both for predictions and relevance estimates.

It is concluded that in this case the omission of a relevant indicator cannot be compensated. Furthermore, the relevant factors influencing reliability do not suggest conflicting conclusions in this scenario. However, a wider study is suggested to confirm these findings generally.

Second-Order Feature Mapping The fourth configuration tests whether the second-order feature mapping allows for more accurate modeling of reliability. The resulting MSEs Err_{CV} (Table 8.3d for the $MTBF$ and Table 8.4d for the $MTTR$) are comparable to those obtained with the reference configuration except for the model generated by ARD algorithm which performs significantly worse.

The second-order mapping yielded 629 features, which are not illustrated as their visual interpretation is not possible. The predictions are illustrated in Figure 8.5e for the $MTBF$ and in Figure 8.5f for the $MTTR$. They perform similarly to the predictions obtained with the first order mapping of the reference configuration.

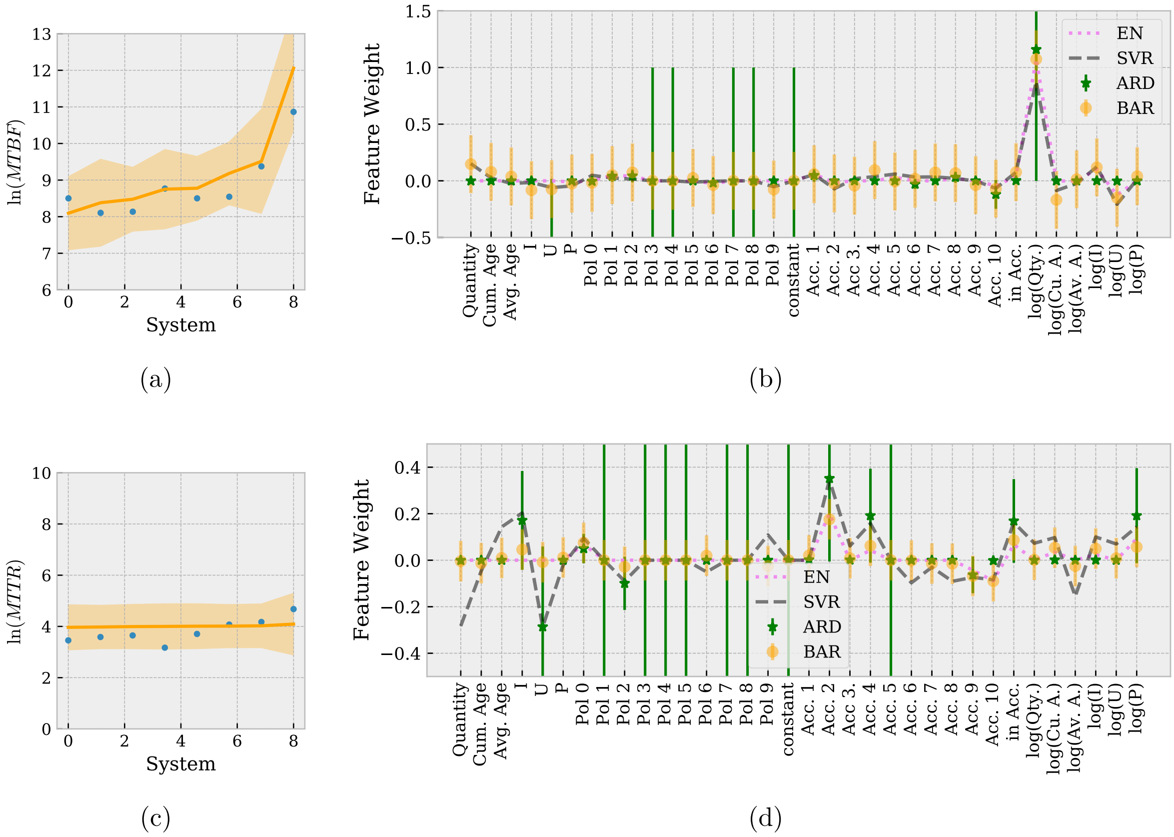


Figure 8.4: Results for a reduced set of data items in the learning data. (a),(c): Prediction of the log($MTBF$) and log($MTTR$) by the BAR algorithm for the last fold of the cross-validation procedure. The orange line depicts the mean of the predictive distribution and the orange shaded area the 95% confidence intervals. The blue dots mark the actual observed field-reliabilites. Note that the converter types on the x axis of the last cross-validation fold were ordered by the mean of the predictions to recognize whether trends are properly captured. (b),(d): Estimated feature weights for the parametric models. [138]

Table 8.3: Obtained mean-squared-errors for the $\log(MTBF)$ - a) Err_{CV} for the reference model, b) Err_{CV} for a reduced set of systems, c) Err_{CV} for a reduced set of reliability indicators, d) Err_{CV} for nonlinear numeric feature mappings, and e) Err_{test} for the predictions of the test data-set. Comparison of a) and e) indicates if the method can be extended to future converter types. [138]

	ARD	**BAR**	**GP**	**EN**	**SVR**
Err_{CV} **a)**	0.39±0.15	0.35±0.13	0.37±0.14	**0.34±0.12**	0.46±0.16
Err_{CV} **b)**	**0.90±0.79**	0.82±0.73	0.81±0.74	0.65±0.49	**0.64±0.50**
Err_{CV} **c)**	1.03±0.24	**1.00±0.19**	**1.00±0.19**	1.01±0.22	1.02±0.24
Err_{CV} **d)**	0.59±0.23	0.37±0.05	0.38±0.05	**0.32±0.05**	0.48±0.12
Err_{test} **e)**	**0.30**	0.33	0.32	**0.30**	0.38

It is concluded that the predictive performance is not improved by the second order feature mapping. This is expected, as it is similar to using more complex modeling approaches, which also do not lead to better predictions. Besides that, interpreting the 629 features can pose a problem.

8.4.3 Prediction

The reference configuration is identified as the most suitable modeling approach in the model selection and validation procedure. It is used to perform the reliability prediction and relevant factor identification on the whole data set. Models are learned using data of converters more than 15 years old and tested on data of converters of less than 15 years, as previously described. Hence, if the reliability of the more recent converters is accurately predicted based on older ones, it provides strong evidence that the reliability of future converters can be predicted accurately based on current ones. Only the $MTBF$ is predicted as no predictive $MTTR$ model could be identified during model selection.

The obtained test MSEs Err_{test} are shown in Table 8.3e. They are slightly better than the predicted generalization errors Err_{CV}, but also lie within their standard deviation, as estimated in the cross-validation procedure during model selection. This gives further confidence in the results. The reliability indicator relevance (Figure 8.6b) and the reliability predictions (Figure 8.6a) on the test set are consistent with the model validation results as well. Overall, it is demonstrated that the $MTBF$ can be predicted and most relevant factors identified accurately.

8.4.4 Discussion

For this use case, several points can be noted. Firstly, with the collected data it is possible to predict the reliability of future converters with an accuracy that is on par with existing methods [139]. However, the effort for data collection and modeling is greatly reduced and

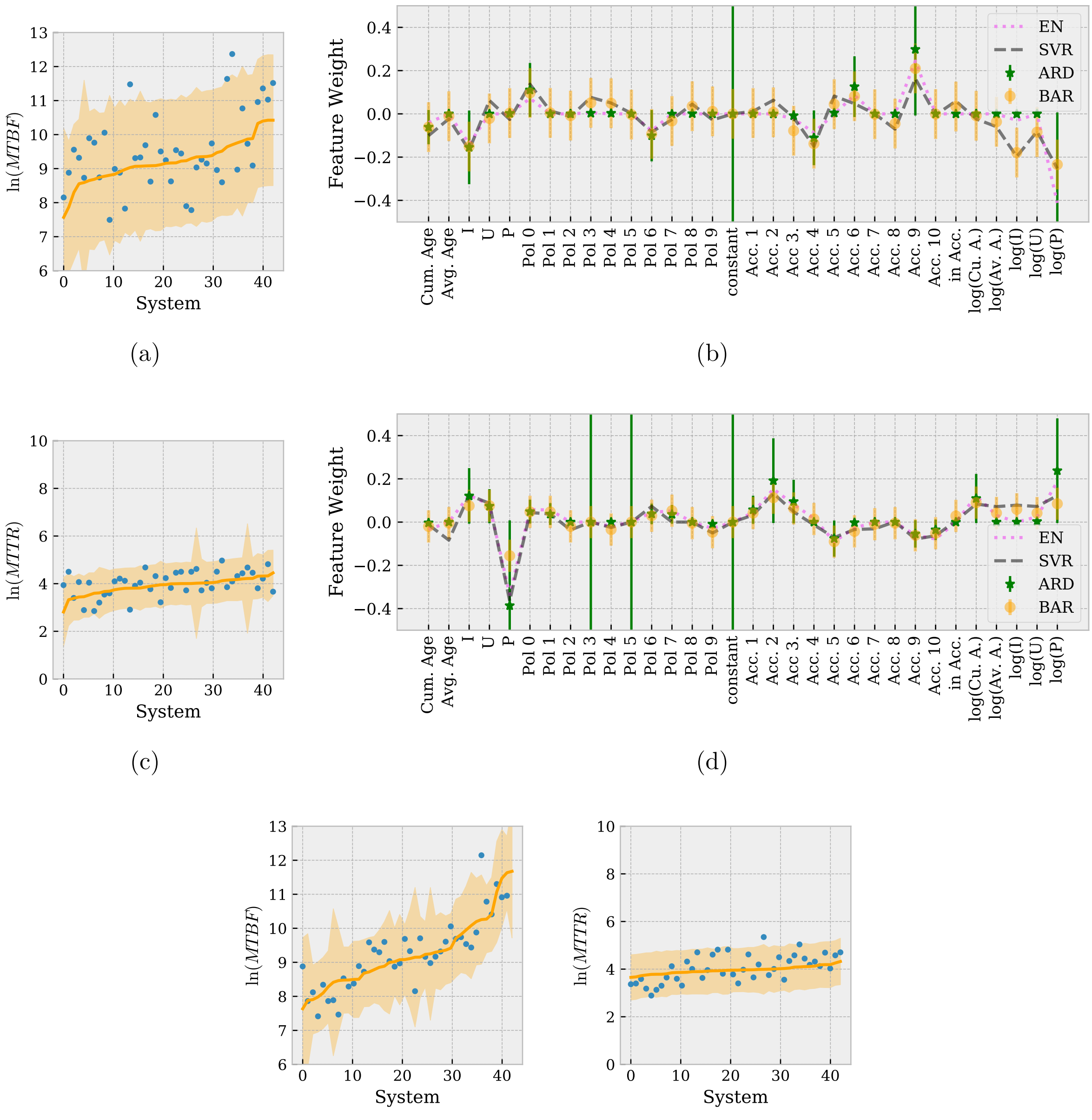
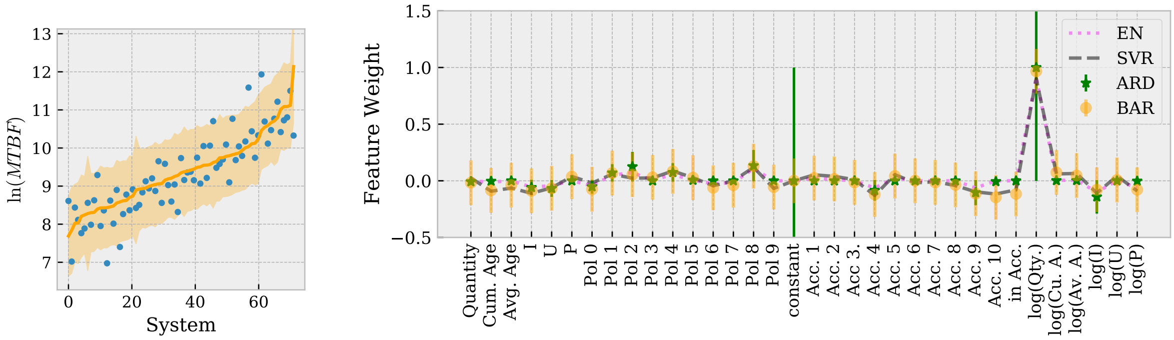


Figure 8.5: (a),(c),(e),(f): Prediction of the $\log(MTBF)$ and $\log(MTTR)$ by the BAR algorithm for the last fold of the cross-validation procedure. The orange line depicts the mean of the predictive distribution and the orange shaded area the 95% confidence intervals. The blue dots mark the actual observed field-reliabilites. Note that the converter types on the x axis of the last cross-validation fold were ordered by the mean of the predictions to recognize whether trends are properly captured. (b),(d): Estimated feature weights for the parametric models. Figures (a),(b),(c),(d) are for the configuration with a reduced set of reliability indicators and Figures (e),(f) for the second-order feature mapping. Note that the illustrations of the 629 second-order feature weights are omitted. [138]



(a) (b)

Figure 8.6: (a): Predictions of the $\log(MTBF)$ with the final models for the test data-set. The orange line depicts the mean and the orange shaded area the 95% confidence intervals. The blue dots mark the actual observed field-reliabilites. Note that the different converter types were ordered by the mean of the predictive distribution. (b):Estimated feature weights for the predictive models. [138]

Table 8.4: Obtained mean-squared-errors for the $\log(MTTR)$ - a) Err_{CV} for the reference model, b) Err_{CV} for a reduced set of systems, c) Err_{CV} for a reduced set of reliability indicators, d) Err_{CV} for nonlinear numeric feature mappings, and e) Err_{test} for the predictions of the test data-set. [138]

	ARD	**BAR**	**GP**	**EN**	**SVR**
Err_{CV} **a)**	0.23±0.05	0.22±0.0.04	0.22±0.04	0.22±0.04	0.23±0.05
Err_{CV} **b)**	0.32±0.17	0.24±0.11	0.24±0.12	0.23±0.09	0.25±0.17
Err_{CV} **c)**	0.30±0.16	0.23±0.06	0.23±0.06	.28±0.11	0.29±0.16
Err_{CV} **d)**	3.12±4.83	0.23±0.02	0.23±0.03	0.22±0.02	0.34±0.06
Err_{test} **e)**	0.38	0.35	0.35	0.35	0.36

independent of the complexity of the studied system, which can be critical for complex systems. The uncertainty of the overall process can be systematically captured, and relevant factors affecting the field reliability can be revealed.

Secondly, the most relevant indicator for an accurate prediction is the quantity of produced converters per type. This indicator is available very early during a system life cycle. Hence, a reliability prediction based on such an indicator is available at the beginning of the life cycle of a new converter when it is most valuable. It can be used as input to model the availability of a whole accelerator complex.

Thirdly, using complex modeling strategies with better modeling flexibility do not lead to better prediction results. Most likely, this is due to the stochasticity of the problem. The choice of the right reliability indicators has the strongest impact on the predictive performance of the learned models.

8.5 Chapter Summary, Conclusions and Outlook

An approach to predict the reliability of systems is presented. It allows to quantify the relevance of factors influencing reliability as well as the uncertainty of the overall modeling process. For a power converter use case it is demonstrated that the $MTBF$ can be predicted with state of the art performance at a reduced data collection and modeling effort.

The quantity of produced converters per type was the factor with the highest relevance for predicting reliability. Hence, the prediction can be carried out at the beginning of a system life cycle, as this indicator is already available then. The output of this study can be used as input for particle accelerator reliability optimization studies. The reduced data collection and modeling effort in comparison to traditional reliability prediction methods will make accurate reliability estimates available for more systems and hence improve the modeling of the overall particle accelerator reliability.

For future work, the causal factors behind the correlation of the produced quantity of converters per type with field reliability should be closer investigated. It hints at the relevance of factors not covered by traditional reliability analysis, which focuses on technical aspects. Moreover, the overall methodology should be validated for a wider range of systems.

Chapter Learning Summary

The findings indicate that reliability is influenced by all processes during a system life cycle. Instead of focusing on isolated aspects, such as the system design, all processes should be evaluated for their impact on reliability.

Chapter 9

Synthesis: Developing Robust Data-Driven Reliability Optimization Methods

In the previous three chapters, data-driven reliability optimization methods for different scenarios have been presented. They provide solutions to a wide range of representative challenges in industrial applications. This chapter combines the findings of the previous chapters to answer the Umbrella RQ. It is structured as follows:

- Section 9.1 is a critical assessment of the previous three constructive chapters to verify that the tailored CRISP-DM methodology helps to resolve practical limitations of data-driven reliability optimization methods. It consists of two parts: Firstly, Subsection 9.1.1 evaluates whether the practical limitations have been addressed successfully in each of the constructive methods of Chapters 6-8. Secondly, Subsection 9.1.2 studies the usefulness and potential improvements of the tailored CRISP-DM methodology.

  The critical assessment confirms that practical limitations are resolved and that the CRISP-DM methodology is useful. The assessment also reveals that two additional aspects, which are necessary to use the reliability optimization methods cost-effectively, are not covered in the previous chapters. These are the correct timing when applying reliability optimization methods within a system life cycle and the provision of high-quality data for the methods. These two aspects and their solutions are discussed in the two following sections.

- Section 9.2 discusses the optimal timing of each constructive method within a system life cycle to maximize their effectiveness.

- Section 9.3 lists the types of data required for reliability optimization methods. Suggestions are made for the effective collection and provision of these data in high quality.

- Finally, Section 9.4 combines all the previous findings and provides a cost-effective data-driven reliability optimization framework for complex engineered systems, which addresses the Umbrella RQ.

9.1 Critical Assessment of Constructive Chapters

9.1.1 Addressing Practical Limitations

The aim of this section is to assess whether the limitations of existing works, as introduced in Section 4.2, have been addressed appropriately by the methods developed in Chapters 6-8. All existing limitations are listed in Tables 9.1 and 9.2, respectively, and assessed for each of the developed methods. Each line in the table corresponds to an existing limitation. It is described in the first column and assessed for each of the methods from Chapters 6-8 in the columns marked Ch5-7. Additional remarks are given in the comment column.

The tables shows all stated limitations have been either fully or partially addressed by the presented data-driven reliability optimization methods. Nevertheless, some trade-offs remain, which are discussed in the following:

- The lack of fault data: This trade-off arises from fundamental conflicts of interest. As soon as a system has one or several faults, investigations will be started. It will not be waited until the system has accumulated a sufficient number of failures so that quantitative estimates can be made with high confidence. Hence, such methods will always be used at the limits of statistical validity.

  To alleviate associated risks, methods in this thesis provide explanations of their predictions and an uncertainty-quantification. Thereby, a human expert can assess whether a model learned from few data is technically sound or not.

  An alternative approach would be to use methods that only use the healthy system state as reference and detect deviations from it. Usually, there is sufficient data to characterize the healthy state of a system. However, such approaches cannot detect whether a deviation of system leads to a failure or not. Hence, replacing or stopping a system with a deviating behavior can prove highly uneconomical because it might not lead to failure. In complex systems, such as particle accelerators, where operational settings are frequently changed, deviating behavior is often expected not to lead to failure. Therefore, such methods have not been further investigated in this thesis.

- Life cycle phases without monitoring: Although tracing and monitoring of products during shipping or storage can be implemented (e.g. [159]), it is questionable whether complete monitoring and information exchange throughout all life cycle phases is feasible and cost-effective. Hence, in practice such solutions are rare for reliability purposes.

  In the fourth line of Table 9.1 it is written that the problem of un-monitored life cycle phases is 'Implicitly addressed' by the developed methods of Chapter 6-8. This

Table 9.1: Evaluation of proposed solutions against limitations of related work - part 1.

Limitations of related works.	**Ch5**	**Ch6**	**Ch7**	**Comment**	**Score**
Existing studies are mostly carried out on a component level, often considering only single failure modes. However, in reality systems are composed of many components and are affected by multiple, often overlapping failure modes and mechanisms.	Addressed	Addressed	Addressed	All Methods have been applied to realistic scenarios	4
Many studies do not account for the dependence of failure mechanisms on multiple factors.	Addressed	Can be included with Eyring model	Implicitly addressed	Used multivariate data-driven approach	3
Dynamic environments and unforeseen inputs can often not be robustly handled by data driven methods.	Addressed (but not tested systematically)	Addressed	Implicitly addressed	Data driven methods can be problematic. Handled by Appropriate model selection schemes	3
Degradation does not only happen during operation of systems. Systems are exposed to stresses during transport, storage and installation, which can remain unobserved by monitoring systems.	Implicitly addressed	Can be included	Addressed	Data acquisition challenging.	2-3
Of all 64 papers… only in 7 papers it was attempted to validate the developed methodologies with field data including any description of the data collection.	Addressed	Addressed	Addressed	Integral part of constructive methodology	4
An uncertainty quantification and propagation throughout all steps in reliability modeling is essential.	Not addressed	Addressed	Addressed	Transparency of methods allows ad hoc certainty assessment.	3
Methods proposed in the literature implicitly require a well functioning monitoring of equipment, computing infrastructures, and existing data sets of run to failure data. However, many organisations cannot meet these requirements. The methods developed in this thesis do not require additional hardware investments, but make use of sensing information whenever readily available.	Addressed	Addressed	Addressed	In some cases excellent monitoring infrastructure available	4
Sikorska et al [137] and An et al [8] criticise that existing review papers focus on mathematical aspects of different methods instead of the value of the methods in reliability optimization contexts.	Addressed	Addressed	Addressed	Integral part of constructive methodology.	4

Table 9.2: Evaluation of proposed solutions against limitations of related work - part 2.

Limitation	Ch5	Ch6	Ch7	Comment	Score
There is a lack of standardized approaches which would help practitioners navigate the vast choice of options.	Partially addressed	Partially addressed	Partially addressed	Method comparison and selection through literature review.	3
Tiddens et al [139] showed that practitioners choose the methods for reliability optimization based on the experience of project stakeholders or other companies and availability of ready-to-use implementations instead of systematically choosing an appropriate approach from the beginning.	Addressed	Addressed	Addressed	Methods chosen based on literature review. Projects only carried out after feasibility assessed.	4
Another frequently encountered challenge is the provision of data which can meet the requirements of the developed methods and subsequently support effective decision making.	Sufficient data	Sufficient Data	Sufficient Data	Data availability and quality assessed. Other projects with insufficient data rejected.	4
They propose a universal metric to measure data fitness for purpose and organisational incentives to improve data quality.	Addressed qualitatively	Addressed qualitatively	Addressed qualitatively	Data availability and quality addressed qualitatively. No metric used.	4
Other problems in data collection are the lack of standardized ways for data collection and the lack of knowledge of failure modes which should define the relevant data to collect.	Semi standardized. Failure modes known. Failure mechanisms unknown.	Standardized. Failure mechanisms known.	Standardized. Failure modes not applicable.	Standardized data collection not always observed to lead to improvements.	3
An et al [8] encourages the sharing of data sets as the existing benchmark data sets are limited in their usefulness.	Data partially on github. Can be requested from the author.	Can be requested from the author.	Can be requested from the author.		3
As pointed out by Sun et al [138] prognostics can provide many additional benefits across the system life cycle. Especially when it can be used to improve the next generation of systems at early life cycle stages, the expected cost benefit is largest.	Yielding a few insights useful across life cycles.	Yielding some insights useful across life cycles.	Yielding insights useful across life cycles.	Systematic way to derive insights across life cycles presented later in this chapter.	3

does not mean that they collect data from all life cycle phases, but that they can appropriately handle situations with missing life cycle data. The method from Chapter 6 would adapt its reliability model when the monitored system is affected from previous un-monitored damage. The models from Chapter 7 and Chapter 8 can include monitoring and damage data if they are available. However, if such data is not available, the additional uncertainty due to lack of data will be quantified.

- Complexity of methods: The complexity of realistic scenarios often demands the use of complex methods. Such methods require advanced mathematical skills from project stakeholders for the correct use of the methods and the interpretation of their results. For the core teams involved in the presented use cases, this could be ensured. However, for a wider adoption of the methods and effective use, special training would be necessary. It is noticed that with the emergence of ML as a discipline of its own and the associated education becoming more prevalent, the adoption of complex data-driven methods is facilitated.

Besides these trade-offs, two other aspects are critical for cost-effective reliability improvement, which have not been addressed in the constructive chapters:

- The optimal embedding of the presented methods in a system life cycle for cost-effective reliability optimization.
- The effective collection and provision of the data required for reliability optimization.

These two aspects and their solutions are presented in detail in Section 9.2 and Section 9.3, respectively. Before these two aspects are covered, the usefulness of the tailored CRISP-DM methodology is assessed in the following subsection.

9.1.2 Usefulness of the Tailored CRISP-DM Methodology

Having confirmed that practical limitations can be overcome in the previous section, the goal of this section is to assess whether the tailored CRISP-DM methodology for method development is appropriate.

This is examined by evaluating the usefulness of the recommended steps as stated in the methodology description (Section 5.2) for each of the use cases from Chapters 6-8. Tables 9.3 and 9.4 list the stated recommendations, the degree to which they have been adopted in each of the constructive chapters, and additional comments on the usefulness of each recommendation.

To summarize the tables, the CRISP-DM methodology with its minor adaptations has been very appropriate for the considered reliability optimization projects. The step of a modeling method evaluation in addition to establishing the decision objectives and assessing the data availability has proven useful to determine the correct modeling approaches. Moreover, the strict separation between project assessment and project implementation has been helpful to avoid investment of time and effort into projects, which are deemed to be unsuccessful.

Table 9.3: Evaluation of proposed CRISP-DM methodology - part 1.

Methodology Step (Recommendation)	RQ1	RQ2	RQ3	Comment
The first step is to identify the objective and the means [of the reliability optimization problem].	Performed	Performed	Performed	Very useful but often overlooked. Objective often imprecisely defined by project requestors.
The objective should be converted into a measurable quantity, such as cost, reliability or a combination thereof.	Partially Performed	Partially Performed	Partially Performed	Quality of predictions quantitatively measurable when objective is clear. Quality of explanations difficult to measure. Needs involvement of users of developed methods.
To further narrow down the choice of [potential methods] a literature review is carried out.	Performed	Performed	Performed	Time-consuming process. Few review papers focus on usefulness of methods in the context of project objectives and domain.
For each [potential methods] the minimum and optimal data and knowledge requirements are assessed. Then the actual data and knowledge availability is compared against its requirements.	Performed	Performed	Performed	Very important step to assess project feasibility at early stage. Data validity and consistency needs to be checked as well.
The compatibility of objectives, methods, available data and knowledge are confirmed. Sufficient software, hardware and time resources for implementing the modeling strategies need to be available.	Performed.	Performed	Performed	Slightly redundant if previous steps carried out appropriately. Hardware and software requirements could develop with project progress.
The missing data and knowledge is collected by running experiments, interviewing system experts, and literature research. Readily available data is cleaned, validated with system experts to ensure data quality meets its requirements, and stored in an accessible format.	Performed	Performed	Performed	Data validation of importance. Requires interaction with system experts. Time consuming. No additional experiments for data generation needed in use cases.
It is beneficial to employ several methods in parallel and compare their outputs.	Performed	Performed	Performed	Gives confidence in findings when different methods yield same results. Part of model selection procedure for machine learning.

Table 9.4: Evaluation of proposed CRISP-DM methodology - part 2.

Methodology Step (Recommendation)	RQ1	RQ2	RQ3	Comment
Whenever a novel method is implemented, it is first tested on a known problem to verify all functionalities are working as expected. When the novel method passes this test successfully, it is applied to the actual problem.	Performed	Performed	Partially Performed	Not strictly necessary when using trivial and tested methods. If known problem does not exist, synthetic data can be used.
Successful implementation of the optimization method allows to determine the decision parameters which lead to the best outcome. Project stakeholders are informed about the recommended decisions and its justification.	Partially Performed	Performed	Partially Performed	End-to-end uncertainty quantification very useful to give decision confidence. In data driven methods, the justification has to be verified with system experts.
To validate that the suggested decisions are actually leading to the desired outcome in the long term requires follow-up after implementation.	Partially Performed	Partially Performed	Partially Performed	Simulation based validation for use cases. Long term cost and reliability follow up challenging.
Implementations can often be reused for related projects due to the modular structure of data driven frameworks.	Performed	Performed	Performed	Modular structure of projects facilitates reuse. Often sufficient to adapt data loading interface.
[CRISP-DM methodology] steps can be repeated, the order does not have to be followed strictly, and the implementations can be updated continuously.	Performed	Performed	Performed	Steps were often repeated, but order not mixed. Important to carry out feasibility assessment before any implementation steps.
To avoid unplanned project failures, utmost importance is paid to initial phases of the CRISP-DM methodology of Business and Data Understanding. Moreover, an additional phase of Model Understanding is added in projects of this thesis.	Performed	Performed	Performed	Very useful. Other projects (not reported) without prior feasibility assessment had to be cancelled at later project stages. Systematic assessment would have prevented project investments.

Two minor improvements can be suggested: Firstly, the requirements definition for explanations provided by the implemented methods should receive more attention during the objectives definition stage. However, assigning appropriate goals and metrics for qualitative explanation outputs are more difficult than for quantitative predictive outputs in general and especially at such early project stages.

Secondly, the proposed long-term validation of any implemented method is difficult to realize. It would require follow-up over years and it is uncertain whether the effects from a certain method can be disentangled from effects due to other methods. To reduce the associated risks, the methodology recommends to test the methods extensively on benchmark problems or in simulation before they are used for decision support.

9.2 Embedding Data-Driven Reliability Optimization Methods in System Life Cycles

In this section, the goal is to embed the reliability optimization methods from Chapters 6-8 within a system life cycle so that they can be used effectively. As illustrated in Figure 1.1, reliability optimization methods are generally more cost-effective when they support decision making in early life cycle phases.

The phases of a system life cycle, the main engineering tasks at each phase, and popular established reliability methods and their role within different life cycle phases are presented at the end of Section 3.1. On this basis, the optimal integration of the optimization methods is discussed as described below.

In the following three subsections, each of the methods from Chapters 6-8 are characterized by their required (data) inputs and when they are available during a life cycle, as well as their outputs and how they can be used to improve system reliability. A distinction is made between systems with and without predecessors for which reliability expertise has already been accumulated, as this makes a difference for the usage of the methods. Differences to established reliability methods and how they can be complemented by the methods introduced in Chapters 6-8 are discussed.

In the fourth and final subsection, it is described how the methods can be used to support decision making in early life cycle phases, which leads to cost-effective reliability optimization.

9.2.1 Embedding Data-Driven Discovery of Failure Mechanisms

The method from Chapter 6, which addresses RQ1, can be used to predict and identify failure mechanisms in systems from their logging and monitoring data. The required input knowledge is limited to a selection of logging or monitoring signals, which signify the failures that one aims to predict, as well as a range of signals, which are expected to contain precursors of the relevant failures. The outputs of the method are predictions of faults in advance and a selection of the most relevant precursors of these faults, from which the failure mechanism can be inferred by experts.

This method has the advantage that it requires almost no a-priori system knowledge. However, it requires monitoring data of the studied system. Hence, for a system without predecessors, it can mainly be used as soon as a data logging environment has been set up. This may happen after prototyping or production. The choice of monitoring signals to consider can be guided by the results of an FMEA analysis. The output of the method can be used in an online and offline setting. In an online use, arising operational issues are predicted and mitigated by experts before they happen. In an offline use, failure mechanisms are detected and mitigation measures discussed. Again, the results of an FMEA analysis can guide the identification of failure mechanisms. The outputs may trigger further reliability investigations that use the inferred knowledge on failure mechanisms as input.

For systems with predecessors, the usage of the method is similar. The previously accumulated knowledge allows a more focused selection of relevant precursor and failure monitoring signals. However, more abstract knowledge, such as pre-existing failure behavior quantification, cannot be utilized by the method.

The classical equivalent of the data-driven method would be manual monitoring and failure analysis by machine operators and experts. Manual analysis is effective when (1) few or only a single failure have been observed, (2) the operators and experts are experienced and knowledgeable with the affected systems, and (3) the set of potential root-causes of the fault is small (e.g. because the affected systems are simple). Contrary to that, the introduced method is effective when (1) several faults have been observed, (2) the operators and experts are neither experienced nor knowledgeable about the affected systems, and (3) the set of potential root causes is large (because the affected systems are complex and interconnected).

In summary, the method is best suited for complex systems with sufficient monitoring data and limited expert knowledge. Its output helps to avoid arising operational failures and to build up expert knowledge on failure mechanisms within a complex system.

9.2.2 Embedding Data and Knowledge-Driven Parametric Model-Based Reliability Optimization

The method from Chapter 7, addressing RQ2, utilizes expert knowledge, failure data, and scientific literature to develop quantitative models of failure behavior. With a simulation engine, optimal operation and maintenance strategies are derived. The required inputs include quantification of failure behavior as a function of the operational conditions, as well as knowledge on the intended future system usage and costs associated with repairs and downtime events. The output is an expected system life cycle cost, which serves as decision metric for different operational, maintenance, and design strategies.

The advantage of this method is that it can effectively combine all available sources of knowledge. The required quantification of failure behavior can be obtained from operational experience from predecessor systems, established models and parameters from scientific literature, or reliability testing. For systems without predecessors, reliability testing can be carried out as soon as prototypes are available. To refine prediction models and to account for differences between the prototypes and the final systems, the reliability tests should be repeated after the production stage. Moreover, models should be continuously updated during operational phases. Bayesian parameter estimation techniques provide a systematic way for updating model parameters. Thereby, the quality of predictions of the models increase continuously and can be used to optimize operations of the existing system, as well as successor systems.

Successors, which reuse parts of the design, can also reuse the corresponding parts of the reliability models. Changed or newly introduced parts of the system have to be treated as systems without predecessors.

The introduced method is an extension of established reliability techniques. These include (accelerated) reliability testing, acceleration factor models, Weibull analysis, Monte Carlo simulation, and life cycle cost modeling. Whereas the established methods are used at certain life cycle stages only, the introduced method allows to combine data and knowledge across life cycle stages and reuse it for future derivations of systems. Overall, the method is best suited to combine various sources of data and knowledge across a system life cycle and use it to optimize the operation of existing and future systems.

9.2.3 Embedding Data-Driven Discovery of Organizational Reliability Aspects

The method from Chapter 8, addressing RQ3, predicts the expected field reliability of future systems as well as the most relevant factors influencing it based on the experience with existing comparable systems. The required input is a measured field reliability for the existing systems and a range of quantitative reliability indicators selected by experts for both the existing systems as well as the systems for which the reliability should be predicted. The method provides reliability predictions for new systems and the factors with a positive or negative impact on reliability.

The stage at which the method can be used to predict the reliability of a system depends on the set of relevant reliability indicators that have been identified. In the presented use case, the relevant indicators are already available at the concept stage of a life cycle. Hence, the reliability of a future system can be predicted at its concept stage already.

Comparable established reliability prediction methods have been outlined in Section 8.2. They are mostly characterized by focusing on technical aspects and certain life cycle phases. Hence, they can only be used when sufficient technical information about the system is available. Moreover, their modeling and data collection effort is coupled to the complexity of the studied system. In contrast, the introduced method integrates non-technical aspects seamlessly and considers all system life cycle phases. The effort for modeling and data collection is independent of the complexity of the investigated systems. Hence, it is recommended to use the introduced method when (1) non-technical aspects shall be considered, (2) the systems are complex, or (3) when predictions need to be available very early in the life cycle of a system.

9.2.4 Using the Methods for Cost-Effective Decision Making

To use the presented methods as tools for cost-effective reliability optimization, they need to be integrated in system life cycles in a way that they can support engineering decision making at early stages of the life cycle. This subsection assesses how to achieve this for each of the methods addressing RQ1-3, respectively.

The method addressing RQ3 satisfies above-mentioned requirement by providing reliability predictions and relevant factors at early life cycle stages. The methods addressing RQ1 and RQ2 do not satisfy this requirement up-front. Both provide useful outputs only after a system has been built and operated for some time.

However, the results of the RQ2 method can be reused for optimising future similar systems at early life cycle stages. Moreover, the RQ1 method can provide outputs that facilitate the implementation of the RQ2 method by indicating the relevant failure precursors and mechanisms. Hence, an iterative approach emerges: It starts with data-driven methods to explore fault patterns and mechanisms and inform system experts (RQ1 method). Then, it continues with model-driven methods (RQ2 method), which allow system experts to combine their built-up knowledge with the fault data to generate transparent quantification of failure behavior, which can be reused for future systems optimization.

This combination of data and model-driven methods across life-cycle allows to improve the reliability of future systems cost-effectively in early life cycle stages. Finally, the RQ3 method can help assessing the effectiveness of the employed reliability optimization methods in the long term.

9.3 Improving Data Quality through Automatic Reliability Data Collection

Throughout the use cases, provision of sufficient and high-quality reliability data was an issue. However, such data is crucial for a cost-effective implementation of the presented methods. This section presents potential solutions to address data provision issues based on evidence from the literature and experience from use cases in this thesis.

The scarcity of reliability data has been reported and studied in literature (see Section 4.2). The reasons for insufficient data are both of technical and organizational kind.

Technical reasons include that failures can be complex emergent phenomena that are hard to monitor and understand. Sensing and monitoring is expensive, increases the complexity of a system, and can introduce new failure modes.

Organizational reasons include that the responsibility of data taking is not assigned, teams responsible for data taking are not given enough context or incentives to carry it out properly, data needs to be stored and maintained over many years, and lastly that the process from data taking to effective decision for reliability optimization is complex and involves many steps and people.

To provide potential solutions to data provision issues, this section

- lists the types of data required for reliability optimization,

- provides recommendations for designing a system to automatically collect the required data, and
- provides recommendations for organising a project to facilitate manual collection the required data.

9.3.1 Types of Data Required for Reliability Optimization

For each of the reliability optimization scenarios in this thesis, a list of minimal data requirements and recommendations for optimal data provisioning has been compiled in the constructive chapters. Comparing the compiled data requirements reveals a significant overlap, although the scenarios cover a wide range of different reliability optimization scenarios, objectives, and methods. Hence, it is concluded that across many reliability optimization tasks, similar kind of data is required, which facilitates the provision of data collection guidelines.

The required data types for the reliability optimization scenarios of this thesis are listed below. These data are grouped by the requirements of different characteristic reliability modeling approaches. These are lifetime models (Weibull models, acceleration factor models, statistical reliability models), system condition models (Physics-of-Failure models, data-driven remaining useful life models), or qualitative evaluation models (Pareto analysis, design reviews; not presented in the thesis but dealt with during preparation of use cases):

1. System identifiers: system ID, assembly ID, component ID
2. Failure identifiers: fault timestamp, fault location (e.g. connector xyz), fault effect (e.g. open circuit), fault mechanism (e.g. corrosion), root cause (e.g. humidity because water leak)
3. Data for lifetime models:
   - System utilization time until failure
   - Suspension times: times/dates switched off, (Optional:) times in other usage before current use, (Optional:) times in storage and shipping before usage
   - Sample size: Number and ID of components, assemblies, or systems of the same type
   - (Optional:) Downtime due to failure
   - (Optional:) Downtime unrelated to failure
   - (Optional:) Operating condition history (mostly relevant if non-uniform across sample)
4. Data for system condition models:
   - Operational loads

- Environmental loads
- System condition indicators

5. Data for qualitative assessment methods:
    - Design documentation
    - Manufacturing documentation
    - Repair documentation

The first two items are required for all reliability investigations. The necessity of the third and fourth item depend on the type of modeling and decision objective. The fifth item is usually required to provide data on fault mechanisms and root causes.

The data types for system condition models (item 4) are only vaguely defined in the list. Methods to determine the precise kind of required data for system condition models can be found in the literature [147, 160, 44, 24, 45]. A few general guidelines are given in the following: The output of an FMEA analysis helps to identify failure precursors that are likely to occur and deserve monitoring. Indicators, which are cheap to measure and likely to cover several failure modes, should be prioritized as they are expected to improve the cost-effectiveness of monitoring. An example for electronic systems is the difference between input and output power, which indicates the dissipated power. An increase of power dissipation can be an indicator for a range of failure modes.

Having the listed data types available in high quality is expected to facilitate the achievement of a large variety of common reliability optimization objectives. Recommendations for the effective collection of these data items are provided in the following two subsections. The recommendations are not conclusive but should serve as starting points for future investigations.

9.3.2 Design for Automatic Reliability Collection

Designing automatic data collection mechanism into systems would be a promising approach for cost-effective provision of data. However, not all data collection procedures can be automated.

Criteria to assess whether automatic collection and storage of data through sensors, networks, and databases are feasible and economic are (0) the necessity and utility of the data for reliability optimization, (1) the feasibility of automatising, (2) band-width requirements, and (3) storage requirements for data logging. In the following, the data types listed in the previous subsection are grouped by the expected automation capability of their collection according to the criteria above and data collection considerations are discussed for each group.

- System identifiers, the fault timestamp, the sample size, and most data for system condition models are well suited for automated collection and storage. The implementation of an automatic collection system is not expected to be challenging.

- A range of data items might qualify for automatic collection. These are the fault location and effect, system utilization time until failure, times and dates a system has been switched off, times in storage and shipping, and downtime (un-/)related to failures.

  All these data, except for times in storage and shipping, are related to a fault or an unavailability of a system. Hence, the fault or unavailability can affect the monitoring system, which is supposed to collect the considered data automatically. Therefore, it is necessary to design automatic collection systems in a fault proof way so that a fault neither in the monitored system nor in the monitoring system causes inconsistencies in the data collection. Although such fault tolerant fault monitoring systems exist [161, 162], their cost-effectiveness has to be assessed for each scenario.

  The times in storage and shipping can principally be collected automatically with additional monitoring efforts. However, when a system is stored, handled and shipped properly, the aging and reliability effects should be limited [8] and monitoring is not justifiable.

- The remaining data items can partially be collected automatically. This includes the fault mechanism and root cause.

  The fault mechanism and root cause can be automatically detected for recurring and expected faults for which diagnostic systems are "designed-in". For new emergent fault behavior, it is unlikely that a reliable automatic mechanism and root cause diagnosis system, which works without additional expert input, can be developed.

  A human expert analysis, assisted by semi-automatized diagnostic systems, such as introduced in Chapter 6, is expected to be more cost- and time-effective for identifying novel fault mechanisms and root-causes. In the following subsection, organizational factors, which encourage system stakeholders to carry out such manual failure analysis and reporting in detail, are discussed.

9.3.3 Organization for Automatic Reliability Data Collection

Some data or information is most effectively collected by human experts. Especially failure mechanisms and root causes of novel and complex failures require manual analysis and validation by system experts. These analysis need to generate high-quality data to be useful for reliability optimization methods.

Based on the literature from Section 4.2 and experience from use cases in this thesis, the following factors improve the quality of manually collected reliability data:

- Data collectors need to be aware of the value of data collection for an organization. They need to understand the potential use of the data that they collect and have sufficient expertise on the systems they investigate.

- The responsibility for data collection has to be assigned. Appropriate means for reliability data collection have to be considered for a system at early life cycle stages.

- Across use cases it is observed that the quality of collected data improves when the data collection responsibility is assigned to experts, which were involved in the conception and design of a system.

Further recommendations for improving organizational data collection practices are provided by Unsworth et al [71].

9.4 A Data-Driven Framework for Cost-Effective Continuous Reliability Optimization

In this section, a proposal for a data-driven reliability optimization framework based on the previous findings is presented. Section 9.1 confirms that the developed reliability optimization methods resolve many practical limitations, Section 9.2 presents an embedding of these optimization methods within system life cycles to use them cost-effectively, and Section 9.3 outlines how sufficient data of high quality can be provided for the optimization methods.

The proposed framework combines the previous findings and leads to cost-effective data-driven system reliability optimization for complex engineered systems by overcoming many practical limitations:

1. Automatic Reliability Data Collection: Systems and processes need to be designed with data collection for reliability analysis and corresponding decision making in mind. A detailed listing of relevant data as well as guidelines for the implementation of effective data collection methods are presented in Section 9.3. These data should be available to stakeholders throughout the system life cycle.

2. Automatic data-driven failure pattern identification: A data-driven approach to obtain predictive models of system failures and supporting information on failure mechanisms from logging data as presented in Chapter 6. The predictive models can help to prevent unforeseen failures and the failure mechanism information simplifies failure analysis for complex systems.

3. Degradation quantification and generalization: A systematic approach to combine failure mechanism information, failure data and expert knowledge into a transparent quantification of system degradation as presented in Chapter 7. It can be generalized to systems with different operating conditions and reused for future generations of systems.

4. Reliability prediction and effectiveness evaluation: The recorded field reliability of a system can be correlated with factors characterising its system life as presented in Chapter 8. For a set of comparable systems, multivariate statistical models can be obtained. They allow to estimate future systems' field reliability at early life cycle stages and provide insight on the factors that impact reliability, which enables early strategic decision making.

The framework can be integrated into system life cycles and complement existing reliability efforts.

Chapter Learning Summary

Reliability data collection should be designed into systems at early life cycle phases. Effective use of data mining methods reveals insights that improves future systems' reliability.

Chapter 10

Conclusions and Future Research Directions

The complexity of modern engineered systems keeps increasing. At the same time, demands on reliability are growing. Traditional techniques, which are based on manual expert analysis, are reaching their limits in such situations and new approaches are required.

Modern data-science methods provide a possible solution. Such methods handle complex and dynamic data, which are common for modern systems but challenging for traditional approaches. They can extract valuable information from the increasing volumes of data, which are accumulated during operation of modern technical infrastructures, to help operators and experts improving the reliability of their infrastructures. As an example for one of the most complex technical infrastructures, this thesis focuses on particle accelerators, such as the LHC.

Despite the potential of modern data-science methods, they frequently fail in practical reliability optimization scenarios because they make too simplistic assumptions of the system behavior, do not consider organizational contexts for cost-effectiveness, and build on specific monitoring data, which are too expensive to record.

The goal of this thesis is to resolve these practical limitations and better leverage the capabilities of modern data-science methods. This has been achieved by the following contributions:

- A methodology for the development and implementation of practical data-driven reliability optimization methods, which address previous limitations and considers organizational contexts. It is based on the CRISP-DM methodology consisting of a project assessment and implementation phase.

  For three realistic use cases, it is demonstrated that applying the methodology led to the development of data-driven reliability optimization methods for addressing existing limitations and advance the state-of-the-art. The diversity of mathematical modeling approaches and reliability optimization objectives used across the scenarios is matched by few other studies in the field of reliability optimization. The contributions of the three methods to the state-of-the-art are detailed below.

1. Explainable deep learning methods predict failures in modern technical infrastructures and help to explain their failure mechanisms. This helps to increase system availability directly by predicting impeding failures and their precursors as well as indirectly by assisting experts to build up their expertise on how to improve the system.

   The presented method is among the first experimental studies of Explainable AI in time-series applications and demonstrates the advantages of deep learning techniques over existing approaches based on Support Vector Machine, Random Forest, or kNearestNeighbor methods for modeling the complexity of realistic phenomena. The generality of the method allows its application to other domains where the discovery of mechanisms in time series data is crucial.

2. Hierarchical parametric models allow to combine expert knowledge and operational data throughout the system life cycle for improved reliability model accuracy. Together with a Monte-Carlo simulation engine and an end-to-end uncertainty-quantification, the system operations and associated life cycle cost can be optimized. The transparent modeling structure allows generalizing results to new operating conditions as well as reusing models and parameters to optimize future generations of systems cost-effectively.

   The method is a realization of the digital twin concept and among the most matured examples in the reliability domain. As such, it serves as a valuable reference for the further development of digital twin solutions for reliability purposes.

3. The field-reliability of systems and its most influencing factors can be predicted with multivariate statistical models. They are trained on reliability data as well as on quantified life cycle descriptors for a group of comparable systems. In comparison to traditional reliability prediction methods, this allows to perform more accurate reliability predictions at earlier life cycle phases for a wider range of systems at reduced modeling efforts. Moreover, it can be used to disentangle the effects of various reliability improvement methods on the field-reliability.

- For the success of all three methods, the collection and provision of high-quality data is crucial. Section 9.3 contributes a list of data types that are required for a range of common reliability optimization objectives as well as recommendations for the effective collection of these data. These are derived with a systematic approach to identify data requirements across optimization objectives and with a translation of the obtained requirements into concrete suggestions for improving the data collection and provisioning.

- Finally, to increase the cost-effectiveness of the methods, their optimal embedding within the different phases of the system life cycle is derived in Section 9.2 with respect to the availability of data and the optimal timing of decision making. The cost-effective embedding and combined use of reliability optimization methods across system life cycles has not been covered elsewhere.

The contributions of this thesis push data-driven reliability optimization methods a substantial step closer to succeeding in realistic environments. It is demonstrated that these methods have the potential to improve the reliability of systems and reduce their cost at reduced additional capital and labor investment. For future generations of technical infrastructures, specifically particle accelerators, effective use of such methods might prove critical to ensure high reliability at reduced cost despite the continuously growing system complexity.

10.1 Future Research Directions

The findings of this thesis suggest a range of potential future research directions. These are grouped in three blocks:

1. Future data-science techniques and their impact on the presented methods.
2. Future research on the usage and application of the presented methods.
3. Future research on applying the findings of this thesis to other domains.

With respect to 1., future data-science techniques are expected to have a positive impact on the cost-effectiveness, user friendliness, and the range of applications of the presented methods:

- Few-Shot Learning (FSL) could extend the use of data-driven methods to scenarios with even less failure data to learn from.
- Bayesian parameter identification methods can combine data from prototype tests as well as operation in a systematic manner including quantification of uncertainty. User friendly, open source tools for such assessments are currently not available.
- Symbolic regression might provide more intuitive model interpretations than currently used input relevance measures.

With respect to 2., future research on the usage of the presented methods could lead to improved guidelines for practitioners:

- Objective criteria should be identified to judge whether traditional knowledge-driven or advanced data-driven reliability optimization methods are more suited for specific scenarios.
- The tailored CRISP-DM methodology should be validated and refined for reliability optimization projects not covered by the three considered representative scenarios.

With respect to 3., future research on applying the findings of this thesis to other domains could help to advance other areas of applied data-science:

- The method from Chapter 6 can be applied to domains where the discovery and understanding of mechanisms in time series data is critical.
- The method from Chapter 7 serves as a valuable reference for the further development of digital twin solutions in industrial environments.
- The systematic approach to improve the collection and provision of high-quality data from Section 9.3 could prove useful for other applied data-science domains, which are affected by insufficient data of low quality.
- The approach from Section 9.2 to embed data-driven optimization methods within a system life cycle to maximize cost-effectiveness can be generalized to other data-driven optimizations in organizational contexts.

Learning Summary

Using modern data-science methods delivers new approaches in many different fields. In this thesis, the practical applicability of such methods for reliability optimization of complex technical infrastructures is demonstrated in general and for the particle accelerators of CERN in particular.

List of Figures

List of Tables

Glossary

AI Artificial Intelligence. 5, 32, 36, 38, 55

CERN European Organization for Nuclear Research. 1, 7, 14–17, 30, 51, 67, 72, 77, 80, 108, 109, 141

CRISP-DM Cross-industry standard process for data mining. 47–49, 52, 125–127, 139, 142, 148

FCC Future Circular Collider. 1, 17

FMEA Failure Modes and Effects Analysis. 3, 30, 129, 133

IIoT Industrial Internet of Things. 5

LHC Large Hadron Collider. 1, 14–16, 18, 80, 92, 93

LINAC Linear Accelerator. 15, 16

ML Machine Learning. 5, 6, 8, 12, 25, 32, 33, 36, 38, 57, 59, 62, 63, 65, 75, 103, 107, 125, 142

Monte Carlo Broad class of computational methods to obtain numerical results relying on repeated random evaluations.. 9, 28, 83, 89, 96, 130

PSB Proton Synchrotron Booster. 15–17, 67, 68, 72, 73, 77, 141

Weibull Swedish mathematician who introduced the continuous Weibull distribution, which is particularly useful to model the lifetime of systems.. 25, 80, 82, 85–88, 91, 92, 97, 130

Bibliography

[1] Wikipedia: CERN — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=CERN&oldid=967135698 (2020), [Online; accessed 05-August-2020]

[2] Nakada, T.: The European Strategy for Particle Physics: Update 2013. Tech. Rep. CERN-ESU-003, Geneva (2013), https://cds.cern.ch/record/2690131, the European strategy for particle physics - Update 2013 was unanimously adopted by the CERN Council at the special Session held in Brussels on 30 May 2013

[3] 2020 Update of the European Strategy for Particle Physics (Brochure). Tech. Rep. CERN-ESU-015, Geneva (2020), http://cds.cern.ch/record/2721370

[4] Bordry, F., Aguglia, D.: Definition of Power Converters (arXiv:1607.01538), 29 p (2015). https://doi.org/10.5170/CERN-2015-003.15, https://cds.cern.ch/record/2038603, comments: 29 pages, contribution to the 2014 CAS - CERN Accelerator School: Power Converters, Baden, Switzerland, 7-14 May 2014

[5] Burnet, J.P.: Requirements on Power Converters. Requirements for Power Converters (arXiv:1607.01597. arXiv:1607.01597), 14 p (Jul 2016). https://doi.org/10.5170/CERN-2015-003.1, https://cds.cern.ch/record/2038602, 14 pages, contribution to the 2014 CAS - CERN Accelerator School: Power Converters, Baden, Switzerland, 7-14 May 2014

[6] Kapur, K.C., Pecht, M.: Reliability engineering. John Wiley & Sons (2014)

[7] Systems and software engineering — System life cycle processes. Standard, ISO/IEC JTC 1/SC 7 Software and systems engineering (May 2015)

[8] O'Connor, P., Kleyner, A.: Practical reliability engineering. John Wiley & Sons (2012)

[9] Shimizu, H., Otsuka, Y., Noguchi, H.: Design review based on failure mode to visualise reliability problems in the development stage of mechanical products. International journal of vehicle design **53**(3), 149–165 (2010)

[10] Kim, M., Lee, J., Jeong, J.: Open source based industrial iot platforms for smart factory: Concept, comparison and challenges. In: International Conference on Computational Science and Its Applications. pp. 105–120. Springer (2019)

[11] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)

[12] Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)

[13] Bayes, T.: Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. Philosophical transactions of the Royal Society of London (53), 370–418 (1763)

[14] Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)

[15] Manyika, J., Chui, M., Bisson, P., Woetzel, J., Dobbs, R., Bughin, J., Aharon, D.: Unlocking the potential of the internet of things. McKinsey Global Institute (2015)

[16] Hodkiewicz, M., Montgomery, N.: Data fitness for purpose: assessing the quality of industrial data for use in mathematical models. In: 8th International Conference on Modelling in Industrial Maintenance and Reliability, Institute of Mathematics and its Applications, Oxford. pp. 125–130 (2014)

[17] Sikorska, J., Hodkiewicz, M., Ma, L.: Prognostic modelling options for remaining useful life estimation by industry. Mechanical systems and signal processing **25**(5), 1803–1836 (2011)

[18] Tiddens, W.W., Braaksma, A.J.J., Tinga, T.: Towards informed maintenance decision making: identifying and mapping successful diagnostic and prognostic routes. In: 19th International Working Seminar on Production Economics. pp. 439–450 (2016)

[19] Javed, K., Gouriveau, R., Zerhouni, N.: 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 **94**, 214–236 (2017)

[20] Leahy, K., Gallagher, C., O'Donovan, P., O'Sullivan, D.T.: Issues with data quality for wind turbine condition monitoring and reliability analyses. Energies **12**(2), 201 (2019)

[21] Sun, B., Zeng, S., Kang, R., Pecht, M.G.: Benefits and challenges of system prognostics. IEEE Transactions on reliability **61**(2), 323–335 (2012)

[22] Tiddens, W.W., Braaksma, A.J.J., Tinga, T.: The adoption of prognostic technologies in maintenance decision making: a multiple case study. Procedia CIRP **38**, 171–176 (2015)

[23] Leahy, K., Hu, R.L., Konstantakopoulos, I.C., Spanos, C.J., Agogino, A.M., O'Sullivan, D.T.: Diagnosing and predicting wind turbine faults from scada data using support vector machines. International Journal of Prognostics and Health Management **9**(1), 1–11 (2018)

[24] Hecht, H.: Prognostics for electronic equipment: an economic perspective. In: RAMS'06. Annual Reliability and Maintainability Symposium, 2006. pp. 165–168. IEEE (2006)

[25] Goebel, R., Chander, A., Holzinger, K., Lecue, F., Akata, Z., Stumpf, S., Kieseberg, P., Holzinger, A.: Explainable ai: the new 42? In: International cross-domain conference for machine learning and knowledge extraction. pp. 295–303. Springer (2018)

[26] Holzinger, A., Kieseberg, P., Weippl, E., Tjoa, A.M.: Current advances, trends and challenges of machine learning and knowledge extraction: from machine learning to explainable ai. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. pp. 1–8. Springer (2018)

[27] Myers, S., Schopper, H.: Particle physics reference library: Volume 3: Accelerators and colliders (2020)

[28] Todd, B.: A beam interlock system for CERN high energy accelerators. Ph.D. thesis, Brunel U. (2006)

[29] Mobs, E.: The cern accelerator complex - 2019 (2019)

[30] Benedikt, M., Blas, A., Borburgh, J.: The ps complex as proton pre-injector for the lhc-design and implementation report. Tech. rep., European Organization for Nuclear Research (2000)

[31] Hanke, K.: Past and present operation of the cern ps booster. International Journal of Modern Physics A **28**(13), 1330019 (2013)

[32] Reich, K.: The cern proton synchrotron booster. Tech. rep. (1969)

[33] Psb machine (2020), `https://psb-machine.web.cern.ch/layout.htm7`

[34] Cho, A.: Higgs boson makes its debut after decades-long search (2012)

[35] Uznanski, S., Todd, B., Dinius, A., King, Q., Brugger, M.: Radiation hardness assurance methodology of radiation tolerant power converter controls for large hadron collider. IEEE Transactions on Nuclear Science **61**(6), 3694–3700 (2014)

[36] Kiran, D.: Total quality management: Key concepts and case studies. Butterworth-Heinemann (2016)

[37] Moeschberger, M.: Competing risks. In: Statistical Theory and Applications, pp. 279–292. Springer (1996)

[38] Weibull, W.: A statistical distribution function of wide applicability. journal of applied mechanics 18: 293-297. Statistical and Computational Analysis **291** (1951)

[39] Committee, F.T.S., et al.: Federal standard 1037c: Glossary of telecommunications terms (fed-std-1037c). National Communications System Technology Program Office, Arlington, Virginia (1996)

[40] White, R.V., Miles, F.M.: Principles of fault tolerance. In: Proceedings of Applied Power Electronics Conference. APEC'96. vol. 1, pp. 18–25. IEEE (1996)

[41] Brière, D., Traverse, P.: Airbus a320/a330/a340 electrical flight controls-a family of fault-tolerant systems. In: FTCS-23 The Twenty-Third International Symposium on Fault-Tolerant Computing. pp. 616–623. IEEE (1993)

[42] Yeh, Y.C.: Triple-triple redundant 777 primary flight computer. In: 1996 IEEE Aerospace Applications Conference. Proceedings. vol. 1, pp. 293–307. IEEE (1996)

[43] Kitano, H.: Biological robustness. Nature Reviews Genetics **5**(11), 826 (2004)

[44] Vachtsevanos, G.J., Lewis, F., Hess, A., Wu, B.: Intelligent fault diagnosis and prognosis for engineering systems, vol. 456. Wiley Hoboken (2006)

[45] Hecht, H.: Prognostics for electronic components. In: 2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS). pp. 1–4. IEEE (2013)

[46] Metropolis, N., Ulam, S.: The monte carlo method. Journal of the American statistical association **44**(247), 335–341 (1949)

[47] Rubinstein, R.Y., Kroese, D.P.: Simulation and the Monte Carlo method, vol. 10. John Wiley & Sons (2016)

[48] Levin, M.A., Kalal, T.T., Rodin, J.: Cover image for Improving Product Reliability and Software Quality, 2nd Edition Improving Product Reliability and Software Quality. Wiley, 2nd edn. (2019)

[49] McPherson, J.W.: Reliability physics and engineering. Springer (2010)

[50] Liew, A.: Understanding data, information, knowledge and their inter-relationships. Journal of knowledge management practice **8**(2), 1–16 (2007)

[51] Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media (2009)

[52] Géron, A.: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media (2019)

[53] Baer, T.: Understand, Manage, and Prevent Algorithmic Bias: A Guide for Business Users and Data Scientists. Apress (2019)

[54] Holzinger, A., Carrington, A., Müller, H.: Measuring the quality of explanations: the system causability scale (scs). KI-Künstliche Intelligenz pp. 1–6 (2020)

[55] Samek, W.: Explainable AI: interpreting, explaining and visualizing deep learning, vol. 11700. Springer Nature (2019)

[56] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)

[57] He, H., Ma, Y.: Imbalanced learning: foundations, algorithms, and applications. John Wiley & Sons (2013)

[58] Nelson, G.S.: The analytics lifecycle toolkit: A practical guide for an effective analytics capability. John Wiley & Sons (2018)

[59] Rose, R.: Defining analytics: A conceptual framework. Or/MS Today **43**(3) (2016)

[60] Coit, D.W., Zio, E.: The evolution of system reliability optimization. Reliability Engineering & System Safety **192**, 106259 (2019)

[61] Kuo, W., Prasad, V.R.: An annotated overview of system-reliability optimization. IEEE Transactions on reliability **49**(2), 176–187 (2000)

[62] Condition monitoring and diagnostics of machines — Prognostics — Part 1: General guidelines. Standard, ISO/TC 108/SC 5 Condition monitoring and diagnostics of machine systems (2015)

[63] Zio, E.: Reliability engineering: Old problems and new challenges. Reliability Engineering & System Safety **94**(2), 125–141 (2009)

[64] Atamuradov, V., Medjaher, K., Dersin, P., Lamoureux, B., Zerhouni, N.: Prognostics and health management for maintenance practitioners-review, implementation and tools evaluation. International Journal of Prognostics and Health Management **8**(060), 1–31 (2017)

[65] Nguyen, D., Kefalas, M., Yang, K., Apostolidis, A., Olhofer, M., Limmer, S., Bäck, T.: A review: Prognostics and health management in automotive and aerospace. International Journal of Prognostics and Health Management **10**(2), 35 (2019)

[66] Si, X.S., Wang, W., Hu, C.H., Zhou, D.H.: Remaining useful life estimation–a review on the statistical data driven approaches. European journal of operational research **213**(1), 1–14 (2011)

[67] Kim, N.H., An, D., Choi, J.H.: Prognostics and health management of engineering systems. Switzerland: Springer International Publishing (2017)

[68] An, D., Choi, J.H., Kim, N.H.: Options for prognostics methods: A review of data-driven and physics-based prognostics. In: 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. p. 1940 (2013)

[69] Elattar, H.M., Elminir, H.K., Riad, A.: Prognostics: a literature review. Complex & Intelligent Systems **2**(2), 125–154 (2016)

[70] Hodkiewicz, M., Kelly, P., Sikorska, J., Gouws, L.: A framework to assess data quality for reliability variables. In: Engineering Asset Management, pp. 137–147. Springer (2006)

[71] Unsworth, K., Adriasola, E., Johnston-Billings, A., Dmitrieva, A., Hodkiewicz, M.: Goal hierarchy: Improving asset data quality by improving motivation. Reliability Engineering & System Safety **96**(11), 1474–1481 (2011)

[72] Eker, Ö.F., Camci, F., Jennions, I.K.: Major challenges in prognostics: study on benchmarking prognostic datasets. In: First European Conference of the Prognostics and Health Management Society 2012 (2012)

[73] Wirth, R., Hipp, J.: Crisp-dm: Towards a standard process model for data mining. In: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining. pp. 29–39. Springer-Verlag London, UK (2000)

[74] Azevedo, A.I.R.L., Santos, M.F.: Kdd, semma and crisp-dm: a parallel overview. IADS-DM (2008)

[75] Felsberger, L., Apollonio, A., Cartier-Michaud, T., Müller, A., Todd, B., Kranzlmüller, D.: Explainable deep learning for fault prognostics in complex systems: A particle accelerator use-case. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. pp. 139–158. Springer (2020)

[76] Apollonio, A., Cartier-Michaud, T., Felsberger, L., Müller, A., Todd, B.: Machine learning for early fault detection in accelerator systems. Tech. rep., CERN (Jan 2020), `http://cds.cern.ch/record/2706483`

[77] Khan, S., Yairi, T.: A review on the application of deep learning in system health management. Mechanical Systems and Signal Processing **107**, 241–265 (2018)

[78] Guo, J., Li, Z., Li, M.: A review on prognostics methods for engineering systems. IEEE Transactions on Reliability (2019)

[79] Zhao, G., Zhang, G., Ge, Q., Liu, X.: Research advances in fault diagnosis and prognostic based on deep learning. In: 2016 Prognostics and System Health Management Conference (PHM-Chengdu). pp. 1–6. IEEE (2016)

[80] Salfner, F., Lenk, M., Malek, M.: A survey of online failure prediction methods. ACM Computing Surveys (CSUR) **42**(3), 1–42 (2010)

[81] Ran, Y., Zhou, X., Lin, P., Wen, Y., Deng, R.: A survey of predictive maintenance: Systems, purposes and approaches. arXiv preprint arXiv:1912.07383 (2019)

[82] Abdul, A., Vermeulen, J., Wang, D., Lim, B.Y., Kankanhalli, M.: Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In: Proceedings of the 2018 CHI conference on human factors in computing systems. pp. 1–18 (2018)

[83] Fulp, E.W., Fink, G.A., Haack, J.N.: Predicting computer system failures using support vector machines. WASL **8**, 5–5 (2008)

[84] Zhu, B., Wang, G., Liu, X., Hu, D., Lin, S., Ma, J.: Proactive drive failure prediction for large scale storage systems. In: 2013 IEEE 29th symposium on mass storage systems and technologies (MSST). pp. 1–5. IEEE (2013)

[85] Fronza, I., Sillitti, A., Succi, G., Terho, M., Vlasenko, J.: Failure prediction based on log files using random indexing and support vector machines. Journal of Systems and Software **86**(1), 2–11 (2013)

[86] Qiu, H., Liu, Y., Subrahmanya, N.A., Li, W.: Granger causality for time-series anomaly detection. In: 2012 IEEE 12th international conference on data mining. pp. 1074–1079. IEEE (2012)

[87] Vilalta, R., Ma, S.: Predicting rare events in temporal domains. In: 2002 IEEE International Conference on Data Mining, 2002. Proceedings. pp. 474–481. IEEE (2002)

[88] Serio, L., Antonello, F., Baraldi, P., Castellano, A., Gentile, U., Zio, E.: A smart framework for the availability and reliability assessment and management of accelerators technical facilities. In: Journal of Physics: Conference Series. vol. 1067, p. 072029. IOP Publishing (2018)

[89] Mori, J., Mahalec, V., Yu, J.: Identification of probabilistic graphical network model for root-cause diagnosis in industrial processes. Computers & chemical engineering **71**, 171–209 (2014)

[90] Liu, C., Lore, K.G., Sarkar, S.: Data-driven root-cause analysis for distributed system anomalies. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC). pp. 5745–5750. IEEE (2017)

[91] Saeki, M., Ogata, J., Murakawa, M., Ogawa, T.: Visual explanation of neural network based rotation machinery anomaly detection system. In: 2019 IEEE International Conference on Prognostics and Health Management (ICPHM). pp. 1–4. IEEE (2019)

[92] Amarasinghe, K., Kenney, K., Manic, M.: Toward explainable deep neural network based anomaly detection. In: 2018 11th International Conference on Human System Interaction (HSI). pp. 311–317. IEEE (2018)

[93] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7) (2015)

[94] Bach-Andersen, M., Rømer-Odgaard, B., Winther, O.: Deep learning for automated drivetrain fault detection. Wind Energy **21**(1), 29–41 (2018)

[95] Montavon, G.: Gradient-based vs. propagation-based explanations: an axiomatic comparison. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, pp. 253–265. Springer (2019)

[96] Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R.: Evaluating the visualization of what a deep neural network has learned. IEEE transactions on neural networks and learning systems **28**(11), 2660–2673 (2016)

[97] Zhou, B., Khosla, A., A., L., Oliva, A., Torralba, A.: Learning Deep Features for Discriminative Localization. CVPR (2016)

[98] Ribeiro, M.T., Singh, S., Guestrin, C.: "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, San Francisco, CA, USA, August 13-17, 2016. pp. 1135–1144 (2016)

[99] Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data Mining and Knowledge Discovery **33**(4), 917–963 (2019)

[100] Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 International joint conference on neural networks (IJCNN). pp. 1578–1585. IEEE (2017)

[101] Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural networks **4**(2), 251–257 (1991)

[102] Eichler, M.: Causal inference with multiple time series: principles and problems. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **371**(1997), 20110613 (2013)

[103] Bergmeir, C., Benítez, J.M.: On the use of cross-validation for time series predictor evaluation. Information Sciences **191**, 192–213 (2012)

[104] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)

[105] Zhao, B., Lu, H., Chen, S., Liu, J., Wu, D.: Convolutional neural networks for time series classification. Journal of Systems Engineering and Electronics **28**(1), 162–169 (2017)

[106] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

[107] Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K.T., Montavon, G., Samek, W., Müller, K.R., Dähne, S., Kindermans, P.J.: innvestigate neural networks! J. Mach. Learn. Res. **20**(93), 1–8 (2019)

[108] Niemi, A., Apollonio, A., Ponce, L., Todd, B., Walsh, D.J.: Cern injector complex availability 2018. Tech. rep., CERN (Feb 2019), `https://cds.cern.ch/record/2655447`

[109] Calderini, F., Stapley, N., Tyrell, M., Pawlowski, B.: Moving towards a common alarm service for the lhc era. Tech. rep., CERN (2003)

[110] Dua, D., Graff, C.: UCI machine learning repository (2017), `https://archive.ics.uci.edu/ml/datasets/PEMS-SF`

[111] Felsberger, L., Todd, B., Kranzlmüller, D.: Cost and availability improvements for fault-tolerant systems through optimal load-sharing policies. Procedia Computer Science **151**, 592–599 (2019)

[112] Felsberger, L., Todd, B., Kranzlmüller, D.: Power converter maintenance optimization using a model-based digital reliability twin paradigm. In: 2019 4th International Conference on System Reliability and Safety (ICSRS). pp. 213–217. IEEE (2019)

[113] Martin, C.: Cibd power supply overview. `https://indico.cern.ch/event/743988/` (2018, Presented at the Reliability and Availability Studies Working Group meeting, CERN, Geneva), [Online; accessed 20-August-2019]

[114] Martin, C., Todd, B., Thurel, Y.: Hccibd reliability analysis. `https://indico.cern.ch/event/743988/` (2018, Presented at the Reliability and Availability Studies Working Group meeting, CERN, Geneva), [Online; accessed 26-June-2019]

[115] Glaessgen, E., Stargel, D.: The digital twin paradigm for future nasa and us air force vehicles. In: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA. p. 1818 (2012)

[116] Tuegel, E.J., Ingraffea, A.R., Eason, T.G., Spottswood, S.M.: Reengineering aircraft structural life prediction using a digital twin. International Journal of Aerospace Engineering **2011** (2011)

[117] Cerrone, A., Hochhalter, J., Heber, G., Ingraffea, A.: On the effects of modeling as-manufactured geometry: toward digital twin. International Journal of Aerospace Engineering **2014** (2014)

[118] Reifsnider, K., Majumdar, P.: Multiphysics stimulated simulation digital twin methods for fleet management. In: 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. p. 1578 (2013)

[119] Gabor, T., Belzner, L., Kiermeier, M., Beck, M.T., Neitz, A.: A simulation-based architecture for smart cyber-physical systems. In: 2016 IEEE International Conference on Autonomic Computing (ICAC). pp. 374–379. IEEE (2016)

[120] Alaswad, S., Xiang, Y.: A review on condition-based maintenance optimization models for stochastically deteriorating system. Reliability Engineering & System Safety **157**, 54–63 (2017)

[121] Nelson, W.: Accelerated life testing-step-stress models and data analyses. IEEE transactions on reliability **29**(2), 103–108 (1980)

[122] Nelson, W.B.: A bibliography of accelerated test plans. IEEE Transactions on Reliability **54**(2), 194–197 (2005)

[123] Nelson, W.B.: Accelerated testing: statistical models, test plans, and data analysis, vol. 344. John Wiley & Sons (2009)

[124] Kapur, K., Lamberson, L.: Reliability in engineering design john wiley & sons. Inc., New York (1977)

[125] Iyer, R.K., Rossetti, D.J.: Effect of system workload on operating system reliability: A study on ibm 3081. IEEE Transactions on Software engineering (12), 1438–1448 (1985)

[126] Kvam, P.H., Pena, E.A.: Estimating load-sharing properties in a dynamic reliability system. Journal of the American Statistical Association **100**(469), 262–272 (2005)

[127] Kececioglu, D., Jiang, S.: Reliability of two load-sharing weibullian units. SAE Transactions pp. 1461–1469 (1986)

[128] Amari, S.V., Bergman, R.: Reliability analysis of k-out-of-n load-sharing systems. In: Reliability and Maintainability Symposium, 2008. RAMS 2008. Annual. pp. 440–445. IEEE (2008)

[129] Huang, W., Askin, R.G.: Reliability analysis of electronic devices with multiple competing failure modes involving performance aging degradation. Quality and Reliability Engineering International **19**(3), 241–254 (2003)

[130] Yang, K., Younis, H.: A semi-analytical monte carlo simulation method for system's reliability with load sharing and damage accumulation. Reliability Engineering & System Safety **87**(2), 191–200 (2005)

[131] Shekhar, C., Kumar, A., Varshney, S.: Load sharing redundant repairable systems with switching and reboot delay. Reliability Engineering & System Safety **193**, 106656 (2020)

[132] Pozsgai, P., Neher, W., Bertsche, B.: Models to consider load-sharing in reliability calculation and simulation of systems consisting of mechanical components. In: Reliability and Maintainability Symposium, 2003. Annual. pp. 493–499. IEEE (2003)

[133] Felsberger, L.: Combining reliability statistics with physics of failure to study and simulate the operational availability of redundant power converters. `https://indico.cern.ch/event/780256/` (2018, Presented at the Reliability and Availability Studies Working Group meeting, CERN, Geneva), [Online; accessed 26-June-2019]

[134] Meng, X., Sloot, J.: Reliability concept for electric fuses. IEEE Proceedings-Science, Measurement and Technology **144**(2), 87–92 (1997)

[135] Parler Jr, S.G., Dubilier, P.C.: Deriving life multipliers for electrolytic capacitors. IEEE Power Electronics Society Newsletter **16**(1), 11–12 (2004)

[136] Kulkarni, C., Biswas, G., Koutsoukos, X., Goebel, K., Celaya, J.: Physics of failure models for capacitor degradation in dc-dc converters. In: The Maintenance and Reliability Conference, MARCON. Citeseer (2010)

[137] Lahyani, A., Venet, P., Grellet, G., Viverge, P.J.: Failure prediction of electrolytic capacitors during operation of a switchmode power supply. IEEE Transactions on power electronics **13**(6), 1199–1207 (1998)

[138] Felsberger, L., Kranzlmüller, D., Todd, B.: Field-reliability predictions based on statistical system lifecycle models. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. pp. 98–117. Springer (2018)

[139] Jones, J., Hayes, J.: A comparison of electronic-reliability prediction models. IEEE Transactions on reliability **48**(2), 127–134 (1999)

[140] Denson, W.: The history of reliability prediction. IEEE Transactions on reliability **47**(3), SP321–SP328 (1998)

[141] Pecht, M.G., Das, D., Ramakrishnan, A.: The IEEE standards on reliability program and reliability prediction methods for electronic equipment. Microelectronics Reliability **42**(9-11), 1259–1266 (2002)

[142] Elerath, J.G., Pecht, M.: IEEE 1413: A standard for reliability predictions. IEEE Transactions on Reliability **61**(1), 125–129 (2012)

[143] Pandian, G.P., Diganta, D., Chuan, L., Enrico, Z., Pecht, M.: A critique of reliability prediction techniques for avionics applications. Chinese Journal of Aeronautics (2017)

[144] Foucher, B., Boullie, J., Meslet, B., Das, D.: A review of reliability prediction methods for electronic devices. Microelectronics reliability **42**(8), 1155–1162 (2002)

[145] Barnard, R.: What is wrong with reliability engineering? In: INCOSE International Symposium. vol. 18, pp. 357–365. Wiley Online Library (2008)

[146] Leonard, C.T., Pecht, M.: How failure prediction methodology affects electronic equipment design. Quality and Reliability Engineering International **6**(4), 243–249 (1990)

[147] Pecht, M., Gu, J.: Physics-of-failure-based prognostics for electronic products. Transactions of the Institute of Measurement and Control **31**(3-4), 309–322 (2009)

[148] Gullo, L.: In-service reliability assessment and top-down approach provides alternative reliability prediction method. In: Reliability and Maintainability Symposium, 1999. Proceedings. Annual. pp. 365–377. IEEE (1999)

[149] Johnson, B.G., Gullo, L.: Improvements in reliability assessment and prediction methodology. In: Reliability and Maintainability Symposium, 2000. Proceedings. Annual. pp. 181–187. IEEE (2000)

[150] Miller, R., Green, J., Herrmann, D., Heer, D.: Assess your program for probability of success using the reliability scorecard tool. In: Reliability and Maintainability, 2004 Annual Symposium-RAMS. pp. 641–646. IEEE (2004)

[151] Groen, G., Jiang, S., Mosleh, A., Droguett, E.: Reliability data collection and analysis system. In: Reliability and Maintainability, 2004 Annual Symposium-RAMS. pp. 43–48. IEEE (2004)

[152] Gauch, H.G.: Scientific method in practice. Cambridge University Press (2003)

[153] Cawley, G.C., Talbot, N.L.: On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research **11**(Jul), 2079–2107 (2010)

[154] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

[155] Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)

[156] MacKay, D.J.: Bayesian interpolation. Neural computation **4**(3), 415–447 (1992)

[157] Baudat, G., Anouar, F.: Kernel-based methods and function approximation. In: IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222). vol. 2, pp. 1244–1249. IEEE (2001)

[158] Williams, C.K., Rasmussen, C.E.: Gaussian processes for machine learning. The MIT Press **2**(3), 4 (2006)

[159] Mahlknecht, S., Madani, S.A.: On architecture of low power wireless sensor networks for container tracking and monitoring applications. In: 2007 5th IEEE International Conference on Industrial Informatics. vol. 1, pp. 353–358. IEEE (2007)

[160] Gouriveau, R., Medjaher, K., Zerhouni, N.: From prognostics and health systems management to predictive maintenance 1: Monitoring and prognostics. John Wiley & Sons (2016)

[161] Irimie, B.C., Petcu, D.: Scalable and fault tolerant monitoring of security parameters in the cloud. In: 2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). pp. 289–295. IEEE (2015)

[162] Goodloe, A., Pike, L.: Toward monitoring fault-tolerant embedded systems. SHM-2009 (2009)