FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Data-Driven Predictive Maintenance for Component Life-Cycle Extension

Margarida da Rocha Faísca Moreira

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Gil Manuel Magalhães de Andrade Gonçalves

September 16, 2024

Recognitions

Throughout the development of this dissertation, Professor Eliseu Pereira offered guidance, support, and supervision.

Margarida

Resumo

Esta dissertação aborda o desafio crítico de melhorar a saúde operacional e a longevidade de máquinas avançadas em sectores como a indústria, os cuidados de saúde, a defesa e os transportes. Com o nascimento da Indústria 4.0, os métodos de manutenção tradicionais provaram ser menos eficazes, levando à adoção de estratégias de manutenção preditiva. Esta mudança é crucial para melhorar a eficiência operacional e minimizar os tempos de paragem inesperados. A investigação centra-se no desenvolvimento e validação de uma ferramenta de manutenção preditiva baseada em dados, utilizando técnicas avançadas de aprendizagem automática e análise de dados para prever potenciais falhas do equipamento e otimizar os planos de manutenção.

O estudo utiliza uma abordagem de um conjunto de dados duplo, utilizando dados reais de um dataset da Gorenje e de um dataset de manutenção preditiva da Microsoft Azure presente no Kaggle, para validar os modelos preditivos propostos. Os modelos avaliados incluem Cox Proportional Hazards (CoxPH), Random Survival Forests (RSF), Gradient Boosting Survival Analysis (GBSA) e Survival Support Vetor Machines (FS-SVM). Os resultados demonstram níveis variáveis de eficácia do modelo, com valores de índice C nos dados de teste de 0,792 para CoxPH, 0,601 para RSF, 0,579 para GBSA e 0,514 para FS-SVM. O FS-SVM apresentou uma melhoria significativa no desempenho do teste em comparação com os dados de treino, indicando o seu potencial para um maior refinamento. É ainda realizada uma simulação na qual é possível prever quando os componentes vão falhar assim como avaliar os custos associadas às suas reparações.

Esta dissertação não só contribui para a área académica ao fornecer uma avaliação dos métodos de análise de sobrevivência no contexto da manutenção preditiva, como também introduz uma interface de painel de controlo de fácil utilização para a tomada de decisões de manutenção em tempo real. A integração destes modelos preditivos no âmbito da Indústria 4.0 permite operações industriais mais eficientes, fiáveis e sustentáveis. Recomenda-se trabalho futuro para melhorar a generalização do modelo, integrar dados em tempo real e expandir a aplicabilidade destas soluções em vários sectores industriais. Este trabalho tem como objetivo reduzir significativamente os custos de manutenção, minimizar o tempo de inatividade do equipamento e melhorar a eficiência operacional, apoiando assim o funcionamento sustentável e rentável das indústrias na Quarta Revolução Industrial.

Palavras-Chave: Manutenção Preditiva, Manutenção Baseada na Condição, Vida Útil Restante (RUL), Indústria 4.0, Métodos Baseados em Dados, Análise de Sobrevivência

Abstract

This dissertation addresses the critical challenge of improving advanced machinery's operational health and longevity in industries such as manufacturing, healthcare, defense, and transportation. With the birth of Industry 4.0, traditional maintenance methods have proven less effective, leading to the adoption of predictive maintenance strategies. This shift is crucial for enhancing operational efficiency and minimizing unexpected downtimes. This work focuses on the development and validation of a data-driven predictive maintenance tool, using advanced machine learning techniques and data analytics to forecast equipment failures and optimize maintenance schedules.

The document employs a dual-dataset approach, utilizing real-world data from a Gorenje dataset and another Microsoft Azure predictive maintenance dataset to validate the proposed predictive models. The evaluated models include Cox Proportional Hazards (CoxPH), Random Survival Forests (RSF), Gradient Boosting Survival Analysis (GBSA), and Survival Support Vector Machines (FS-SVM). The results demonstrate varying levels of model effectiveness, with C-index values on test data of 0.792 for CoxPH, 0.601 for RSF, 0.579 for GBSA, and 0.514 for FS-SVM. Notably, FS-SVM exhibited significant improvement in test performance compared to training data, indicating its potential for further refinement. A simulation is also carried out in which it is possible to predict when components will fail and to assess the costs associated with repairing them.

This dissertation not only contributes to the academic area by evaluating survival analysis methods in the context of predictive maintenance but also introduces a user-friendly dashboard interface for real-time maintenance decision-making. Integrating these predictive models within the Industry 4.0 framework enables more efficient, reliable, and sustainable industrial operations. Future work is recommended to improve model generalizability, integrate real-time data, and expand the applicability of these solutions across various industrial sectors. This work aims to significantly reduce maintenance costs, minimize equipment downtime, and enhance operational efficiency, supporting industries' sustainable and profitable functioning in the Fourth Industrial Revolution.

Keywords: Predictive Maintenance, Condition-Based Maintenance, Remaining Useful Life (RUL), Industry 4.0, Data-Driven Methods, Survival Analysis

United Nations' Sustainable Development Goals (SDGs)

Leaders worldwide gathered at the United Nations Headquarters to commemorate the organization's 70th anniversary from the 25th to the 27th of September 2015. During this event, they adopted the 2030 Agenda for Sustainable Development, a comprehensive framework to foster peace and prosperity for people and the planet. This agenda is built around 17 major Sustainable Development Goals (SDGs) and 169 targets, collectively balancing the three dimensions of sustainable development: social, economic, and environmental.

Integrating the SDGs into various sectors has become increasingly crucial, particularly in the era of Industry 4.0. This new industrial revolution is characterized by the fusion of physical and digital technologies, leading to significant advancements in efficiency and innovation. In this context, predictive maintenance emerges as a critical component for optimizing industrial operations, reducing waste, and conserving resources.

The present work, which focuses on data-driven predictive maintenance for component lifecycle extension, aligns with several key SDGs. By using advanced survival analysis models, this dissertation aims to improve the reliability and efficiency of industrial equipment. This leads to economic benefits by reducing downtime and maintenance costs and supports sustainable practices by minimizing resource consumption and waste.

In this chapter, we explore how the methodologies and findings of this dissertation contribute to the SDGs, specifically SDG 9 (Industry, Innovation, and Infrastructure) and SDG 12 (Responsible Consumption and Production). We identify the relevant targets under these goals, describe the contributions of our predictive maintenance approach, and outline the performance indicators used to measure its impact.

SDG	**Target**	**Contribution**	**Performance Indicators and Metrics**
9	9.4	Optimizing resource use and reducing waste in industrial processes through predictive maintenance.	- Downtime reduction (hours) - Resource efficiency improvement
	9.5	Encouraging innovation by integrating advanced predictive analytics and machine learning.	- Number of technologies integrated
12	12.4	Ensuring environmentally sound management of equipment and processes, reducing harmful waste.	- Hazardous waste reduction (tons) - Equipment lifespan extension (years)
	12.5	Minimizing waste through predictive maintenance, leading to fewer unnecessary replacements.	- Waste reduction (tons)

Agradecimentos

A elaboração da dissertação marca um fim de um capítulo, o qual fiz rodeado de diversas pessoas a quem devo o meu mais sincero obrigado.

Para começar gostava de agradecer às pessoas que me acompanharam na realização desta dissertação. Primeiro ao meu orientador, o professor Gil Gonçalves, que sempre se mostrou disponível para me prestar apoio sempre que precisei. De seguida ao meu coorientador, ao professor Eliseu Pereira, por todo o apoio e paciência que teve ao longo destes últimos meses, quando tantas vezes chateei para esclarecimento de dúvidas e que sempre me conseguiu ajudar.

Aos meus pais, que sempre foram a minha maior inspiração e que sempre estiveram presentes, nos bons e maus momentos. Obrigada por me apoiarem e nunca me terem deixado desistir. Obrigada por toda a liberdade que me deram nos últimos anos, por todo o esforço que colocaram para eu chegar aqui e por todos os ensinamentos que me deram e que levo para a vida. Obrigada por serem o meu porto seguro, sempre.

À Lisa, que desde pequenina me aturou e que tanto contribuiu para moldar a pessoa que sou hoje, vais ser sempre uma figura de extrema importância na minha vida.

Aos que entraram em 2019 comigo, Roberto Papelão, Gabriel SeiSer, João Azamat, João Bob, Viana Manafufa, Samuel Kano, Maganito Machachuxas, Tiago Xauxixas, João Morgado, ao Edu, à minha Rafinha, obrigada por viverem comigo estes últimos anos. Em especial àqueles que continuaram a fazer do jardim uma segunda casa, obrigada por tudo o que me ensinaram, por me fazerem a praxista que sou hoje e por tornarem tantas horas deste percurso académico tão bonitas. Obrigada por me mostrarem o que é ser estudante nesta casa.

Aos mais velhos, obrigada por me terem mostrado desde o início que é possível fazer mais do que apenas estudar e que realmente é desta casa que levamos amizades para a vida. Ao Pedro Moreira, por tornares-te das pessoas que sei que levarei para a vida. À Pocahontas, um agradecimento especial por toda a confiança que sempre colocaste em mim e por todas as palavras amigas que me fizeram acreditar que tudo o que eu quisesse alcançar, era possível.

Aos meus amigos do Centro de Eventos, ao Gabriel, Henrique, Bruno Francisco e Megui, obrigada por todas as gargalhadas e capacidade de desenrasque que me deram naquela régie. Aos meus amigos de Erasmus, Kat, Júlia, Maria, Mário, Laura, Margarida e Pedro, obrigada por fazerem parte de uma experiência da qual vos levarei sempre no meu coração.

Às meninas que sempre me acompanharam neste percurso, à minha Margarida, à minha Raquel, à minha Bea e à minha Bia, obrigada por todas as conversas. Conseguimos meninas.

À Inês, por desde pequenina ter sido a minha melhor amiga da vida, por todos os momentos partilhados, por todos os altos e baixos que passamos juntas e por ao fim de mais 5 anos, termos conseguido.

Ao Boruto Francisco, obrigada por toda a paciência e dedicação que tens comigo diariamente. Agradeço todos os dias pela faculdade e aquele jardim terem-nos colocado no mesmo caminho. És o meu refúgio todos os dias.

Quero agradecer à minha irmã, Catarina, por ser um exemplo que sempre tentei seguir e por toda a paciência que tem sempre comigo. És e serás sempre um dos maiores orgulhos que tenho. Ao meu cunhado, Zé, por me ter aturado e esclarecido muitas das dúvidas que tive ao longo destes 5 anos e por me ter dado a confiança que me faltava ao longo destes anos. Agradecer ainda ao Marley, Simba e Biggy por me meterem sempre um sorriso na cara nos dias mais complicados. Ao meu tio Paulo e ao meu primo Fernando, obrigada.

Quero acabar por agradecer à minha avozinha Inácia, obrigada por estar presente todos os dias da minha vida e por me colocar sempre um sorriso na cara. Saber que a paciência é algo que aprendemos a ter com aqueles que mais amamos foi dos maiores ensinamentos que tenho aprendido. Por fim, agradecer aos meus avós, ao meu tio Adão e à minha tia Micá, que apesar de já não estarem aqui para me ver, sempre acreditaram em mim. Sei que estão aí em cima todos reunidos a jogar uma suecada e a beber uns vinhinhos enquanto riem à volta de uma mesa e olham por mim.

Margarida

"My philosophy is:
It's none of my business what people say of me and think of me.
I am what I am and I do what I do. I expect nothing and accept everything.
And it makes life so much easier."

Sir Anthony Hopkins

Contents

List of Figures

List of Tables

Abbreviations

AE	Autoencoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
BDP	Bayesian deep learning
BPNN	Backpropagation Neural Network
CBM	Condition-Based Maintenance
CDF	Cumulative Distribution Function
CI	Concordance Index
CNN	Convolutional Neural Network
CSV	Comma-Separated Values
CoxPH	Cox Proportional Hazards
DBN	Deep Belief Network
DL	Deep Learning
DRL	Deep Reinforcement Learning
DT	Decision Tree
FS-SVM	Fast Survival Support Vector Machine
EDA	Exploratory Data Analysis
ExtraTrees	Extremely Randomized Trees
GAN	Generative Adversarial Networks
GBC	Gradient Boosting Classifier
GBSA	Gradient Boosting Survival Analysis
GNN	Graph neural network
GPR	Gaussian Process Regression
IoT	Internet of Things
KM	Kaplan-Meier estimator
k-NN	k-Nearest Neighbor
LR	Linear Regression
LSTM	Long Short-term Memory
ML	Machine Learning
MLPNN	Multilayer Perceptron Neural Network
MTTF	Mean Time to Failure
OSA-CBM	Open System Architecture for Condition Based Monitoring
PdM	Predictive Maintenance
PHM	Prognostics and Health Management
RF	Random Forest
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RSF	Random Survival Forest
RUL	Remaining Useful Life
RVM	Relevance Vector Machine
R2	Robot 2

SVM	Support Vector Machine
SVR	Support Vector Regression
TCN	Temporal Convolutional Network
TTF	Time to Failure
XGBoost	Extreme Gradient Boosting

Chapter 1

Introduction

In recent years, advanced machinery has become vital to various industries, including manufacturing, healthcare, defense, and transportation. This evolution has revolutionized our work, making machine reliability and efficiency critical to our daily operations. As a result, there's a growing need to enhance the operational health and longevity of these machines, directly impacting the performance and profitability of the companies.

The rise of machine learning (ML) and data analytics due to the development of Industry 4.0 is transforming how we approach machine maintenance. Reactive or scheduled maintenance are options that are no longer profitable. Predictive maintenance is a method that has been developed and used in the past few years. This innovative process utilizes the strength of data-driven models, applying machine learning to foresee potential failures and optimize maintenance times. It helps to reduce downtime, cut back on maintenance expenses, and extend the life of machine components, contributing to greater operational efficiency.

These advanced predictive maintenance models, however, are still being developed. These will be used to predict when a machine might break down and determine the optimal time for component replacement. This avoids wasting component life and reduces maintenance.

1.1 Context

The rapid advancement of machinery has become a cornerstone of modern industries, including manufacturing, healthcare, defense, and transportation. The integration of sophisticated machines has revolutionized these sectors, making the reliability and efficiency of machinery critical to operational success. In this context, maintaining the operational health and extending the machinery's life cycle is beneficial and essential. Industry 4.0, characterized by the convergence of physical and digital technologies, has introduced new challenges and opportunities in maintenance practices.

Traditional maintenance strategies, such as reactive maintenance (addressing issues after they occur) and preventive maintenance (scheduled maintenance), have shown limitations in handling the complexities of modern machinery [1]. These strategies often result in premature maintenance, leading to unnecessary costs or delayed maintenance, causing unexpected breakdowns. With the advent of advanced data analytics and machine learning, there is a paradigm shift towards predictive maintenance, which aims to optimize maintenance schedules and reduce downtime.

1.2 Motivation

The motivation for this dissertation arises from the pressing need to enhance maintenance practices in industries reliant on complex machinery. The traditional methods are increasingly inadequate, as they fail to leverage the vast data generated by modern equipment. Predictive maintenance offers a proactive solution, using data-driven insights to forecast potential failures and optimize maintenance activities. This minimizes costs and downtime and extends the lifespan of machinery components.

The adoption of predictive maintenance aligns with global trends towards sustainability and efficiency. By preventing unnecessary repairs and replacements, predictive maintenance reduces the environmental impact of industrial operations, contributing to sustainable practices. Furthermore, predictive maintenance is vital in ensuring safety and operational integrity in sectors where equipment reliability is critical, such as healthcare and defense.

1.3 Problem Definition

The primary problem addressed in this dissertation is the unpredictability of machinery component failures, which can lead to significant operational disruptions and financial losses. Traditional maintenance strategies are reactive and often fail to provide timely intervention, resulting in excessive maintenance costs or catastrophic equipment failures. The challenge lies in developing a predictive maintenance system to accurately forecast the components' Remaining Useful Life (RUL) or Mean Time to Failure (MTTF). This requires leveraging data from sensors and other sources to detect patterns and anomalies that precede failures.

The lack of effective predictive models in many industries results in a reliance on outdated maintenance practices. This gap presents an opportunity to develop and implement data-driven predictive maintenance models to significantly improve operational efficiency and reduce costs.

1.4 Dissertation Goals

This dissertation aims to develop and validate a data-driven predictive maintenance tool that extends the life-cycle of various industrial components, such as:

1. Development of a Predictive Maintenance Model.

2. Data Aggregation and Preparation.

3. Algorithm Selection and Optimization.

4. Validate the predictive maintenance tool using a real-world and already-used dataset to ensure its reliability and applicability across different industrial contexts.

5. Create a simulation to test the different models, evaluate the costs of the components repair, and create an intuitive user interface for the predictive maintenance tool, making it easy for technicians and maintenance managers without requiring deep technical knowledge.

Firstly, a general model should be developed to accurately predict the industrial components' RUL and ensure that it can process and analyze real-time data from sensors and historical maintenance records to make predictions.

Secondly, a systematic process for collecting, cleaning, and preparing data from diverse sources, including sensor outputs, will be established. This will occur using two datasets: one real and another already used. This last one will allow us to compare the results in the end.

Then, the most appropriate machine learning algorithms will be explored, evaluated, and selected to optimize them for the best performance in terms of accuracy and efficiency.

After that, the tool should be tested in various scenarios to ensure reliability and accuracy. This will be done by using the dataset previously chosen.

Lastly, an intuitive interface for the predictive maintenance tool will be designed to ensure it is accessible to technicians and maintenance managers without requiring deep technical knowledge of data analytics.

1.5 Structure

The document is composed of six distinct chapters:

- Chapter 2 presents a set of relevant background information related to the main topic of this dissertation and provides an in-depth review of existing literature on predictive maintenance, focusing on data-driven methods and relevant Industry 4.0 technologies.

- Chapter 3 Describes the architectural research implications and suggests future work directions components and their interactions.

- Chapter 4 discusses the implementation of the predictive maintenance tool, including data handling, model development, and user interface design.

- Chapter 5 presents the experimental setup, evaluation metrics, and results obtained from testing the predictive models.

- Chapter 6 summarizes the key findings, discusses the research implications, and suggests future work directions.

Chapter 2

Data-Driven Predictive Maintenance - A literature review

This chapter initially presents the background and some key terms related to this dissertation to offer a better understanding of predictive maintenance, survival analysis and remaining useful life in Industry 4.0. It then presents a comprehensive literature review on developing predictive maintenance algorithms, specifically focusing on data-driven methods. The review begins by detailing the methodology used for the literature search, identifying and defining all process phases. Following this, the selected articles are critically examined in critical areas essential for developing reliable models to predict the remaining useful life of industrial machinery components.

2.1 Background

2.1.1 Industry 4.0

During the last 240 years, an extensive revolution in the industry happened. Everything started with the first revolution, which was characterised by mechanisation and steam power. Then, almost 100 years later, the second industry began, characterised by mass production and electrical energy. Then came Industry 3.0, where the concepts of automation, computers and electronics appeared. After this, Industry 4.0 started to develop.

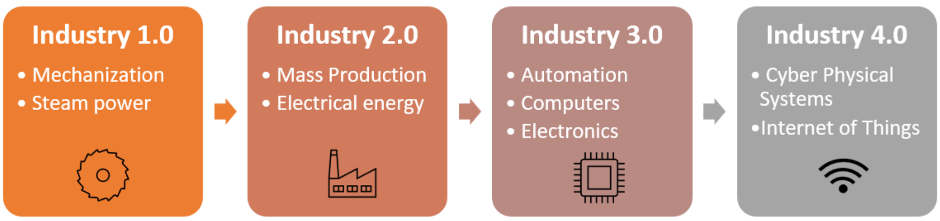


Figure 2.1: Stages of industrial development, based on [2].

"Industry 4.0", also referred to as the "Fourth Industrial Revolution", is considered the next phase of integration of digital technologies into manufacturing processes. The current era of connectivity, advanced analytics, automation, and manufacturing technology characterises this revolution [3]. Some of the technologies used in the Industry 4.0 are:

- **Big Data and Analytics**
- **Simulation**
- **Internet of Things (IoT)**
- **Artificial Intelligence (AI) and Machine Learning (ML)**
- **Digital Twins**

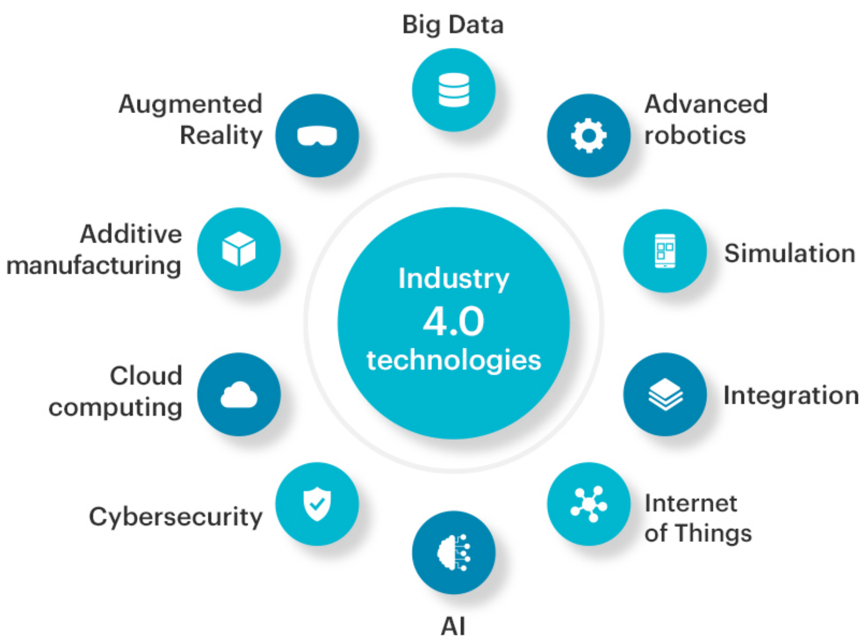


Figure 2.2: Industry 4.0 technologies, taken from [4].

These technologies transform industries into smart, connected, and data-driven sectors.

The technologies above are, more precisely, Big Data and Analytics, AI and ML. We can define, respectively, as the first being characterised by analysing large volumes of data generated by manufacturing processes that help in making informed decisions, predicting maintenance needs, and optimising production efficiency and the second as algorithms that are used for tasks such as predictive maintenance, quality control, and optimisation of production schedules, enabling machines to learn from data and make intelligent decisions.

2.1.2 Types of industrial maintenance

Predictive Maintenance (PdM) has been increasing in popularity in the last few years, mainly in Industry 4.0. It is estimated that "The impact of maintenance represents a total of 15 to 60% of the total costs of operating of all manufacturing" [5].

Nowadays, the three main types of industrial maintenance are reactive maintenance, preventive maintenance, and predictive maintenance.

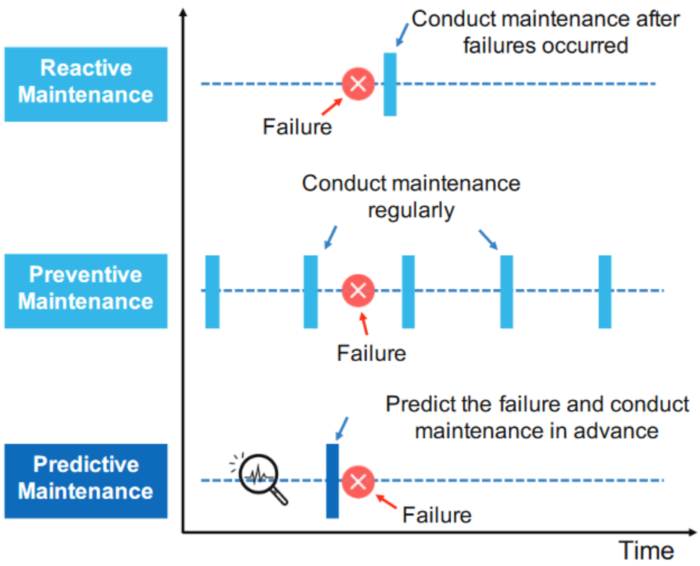


Figure 2.3: Types of industrial maintenance, taken from [6].

Reactive maintenance was one of the first being used in the industry. It is characterised by responding to equipment failures, in other words, without prior planning. This approach often leads to unexpected downtimes and can be costly regarding repairs and lost production time. It is still used mainly in less critical equipment, where breakdowns do not significantly impact overall operations.

Preventive maintenance refers to activities performed on a fixed schedule, generally in regular intervals. This technique reduces the likelihood of equipment failure. However, it may lead to over-maintenance, where components are often fixed than necessary.

As we can imagine, those are little efficient and cost-effective, leading to predictive maintenance.

Predictive maintenance relies on the real-time monitoring of equipment conditions to predict when maintenance should be performed. It uses a combination of data analysis, ML, and sensor technologies to indicate potential failures before they happen. The main goal is to optimise maintenance frequency, reduce costs, and increase equipment uptime. It predicts and prevents component failures. It has been widely adopted due to its superior results to other techniques, such as reactive maintenance and preventive maintenance [7].

In the last years, a new kind of maintenance has been developed. This type, called proactive maintenance, is considered the previous type in this evolution process. It consists of identifying the source of the fault and remediating conditions that can fail [8].

2.1.3 Predictive maintenance models

There are three main models, as presented below [9]:

- **Condition-Based Maintenance (CBM)**: CBM is a maintenance strategy where the repair or replacement decisions are taken based on the current or future condition of the equipment. This should only be performed when specific indicators show decreasing performance or upcoming failure.

- **Prognostics and Health Management (PHM)**: Consists of the dynamic monitoring of the system state for a predictable type of maintenance. It can predict and evaluate the progression of potential deterioration or faults in a system.

- **Remaining Useful Life (RUL)**: It refers to the estimated lifespan of a device before it requires maintenance or replacement. It is also the remaining time until the equipment's health conditions reach the failure threshold [10].

The main focus of this dissertation will be on the Remaining Useful Life (RUL) strategy. The accuracy of the RUL estimation depends on the quality and quantity of available health monitoring data and the methods used for such analysis [11].

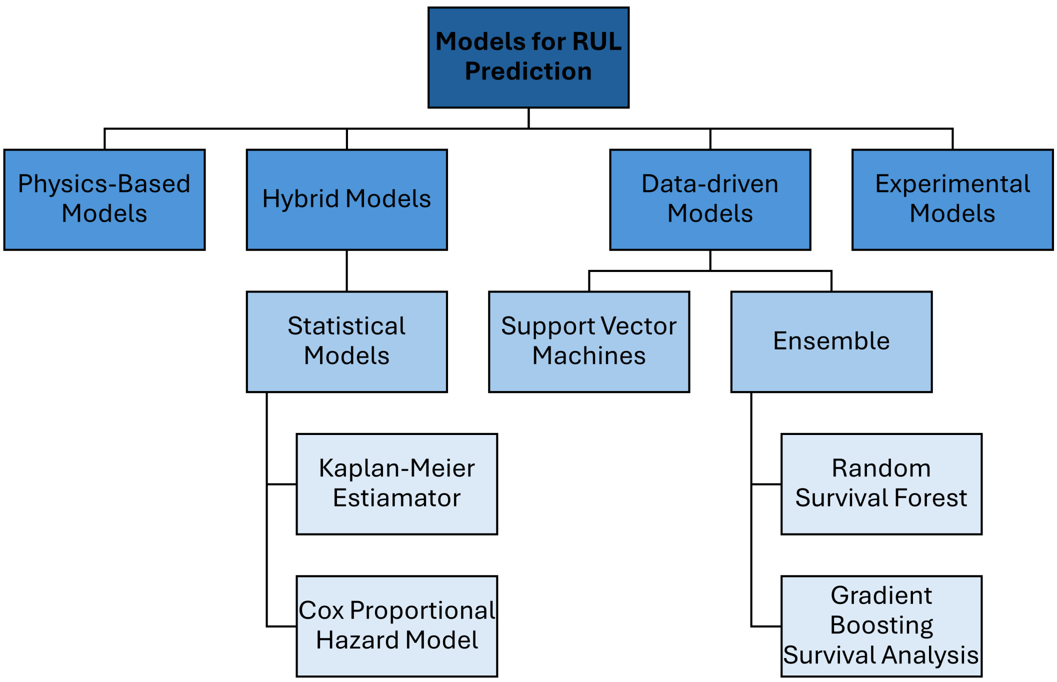


Figure 2.4: Models for RUL prediction, based on [9].

2.1.4 Survival Analysis

Survival analysis encompasses various methods of analysing data related to time-to-event outcomes. It is also comprehended as a reliability theory and an event history analysis. The term 'survival' is equivalent to probabilities in survival analysis, implying that survival can be 'measured'. The survival function can be expressed as [7]:

$$S(t) = P(T > t) = 1 - F(t) \tag{2.1}$$

$S(t)$ is the survival function representing the probability that a system or component survives beyond a particular time t. The random variable T, a non-negative continuous variable, represents the time until the component fails or an event occurs. The whole function describes the probability that the component is not worn out by t cycles. The $F(t)$ is the cumulative distribution function (CDF) of the random variable T, which gives the probability that an event (such as failure or death) has occurred by time t.

The survival function present on 2.1 can be estimated using the Kaplan-Meier Estimator (KM). It is a non-parametric estimator used to measure the fraction of living subjects after a certain amount of time concerning one or more specific events [12]. It is defined as:

$$\hat{S}(t) = \prod_{t_i \le t} \left(1 - \frac{d_i}{n_i}\right) \tag{2.2}$$

In the equation 2.2 [13], the t_i is the time when at least one event (worn-out) occurred, the d_i is the total number of events (worn-out) at t number of shots and n_i is the number of components at risk at t number of shots.

The Kaplan-Meier survival curve starts from 1 when all components work correctly and decreases to 0 as events (worn-outs) occur.

In addition to non-parametric methods like the Kaplan-Meier estimator, parametric and semi-parametric models are crucial in survival analysis. One of the most widely used semi-parametric models is the Cox proportional hazard model (CoxPH). The Cox model is used to investigate the effect of several variables on the time a specified event takes to happen, and it has the following hazard function [7]:

$$h(t,X) = h_0(t)e^{\beta X} \tag{2.3}$$

In the equation 2.3, $h_0(t)$ is the baseline hazard function, β is the coefficient, and X is the covariate. The CoxPH model assumes that survival times t are independent, the hazard is proportional, i.e., the hazard ratio is proportional, the hazard function is the linear function of the numerical covariates, and the values of X's do not change over time.

To evaluate and compare the performance of survival analysis models, the scikit-survival module from Python can be used, which offers comprehensive tools and metrics tailored for survival data. The bigger difference between survival analysis and traditional ML is that parts of the training data can only be partially observed. [14]. Consequently, the performance metrics used are the

Concordance index (C-index), Brier Score, integrated Brier score and time-Dependent receiver operating characteristic (ROC).

2.1.5 Performance Metrics

Performance metrics are used to evaluate and estimate the survival models. Some of the most common include the Concordance Index (C-index), Brier Score and the integrated Brier score (IBS) [15].

The C-index, also called Harrell's Index [13], is one of the most common evaluation metrics for measuring survival models' quality and efficiency. It ranges from 0 to 1, where results closer to 1 indicate a better performance of the model [16].

The Brier score measures the difference between the predicted probability and the actual outcome, with higher scores [17]. The Brier score ranges from 0 to 1, and contrarily to C-index, 0 indicates a perfect prediction while 1 indicates poorer prediction accuracy and calibration.

The IBS extends the Brier score to the context of time-to-event data, such as survival analysis. It reflects calibration over all time points, with a smaller value indicating greater accuracy. Like the Brier score, a smaller value indicates a greater accuracy [18].

2.1.6 Data Preparation and Feature Engineering

The dataset is usually chosen based on the data collected by the sensors, more precisely, temperature, voltage, pressure, rotation, and vibration, among others.

To compare the results, two different datasets are used for testing, one real and another already used in other works.

The dataset that would be chosen to be compared with the real one posteriorly should contain a large number of points to have more accurate results.

As shown below, the collected data should be filtered using a real dataset to obtain better results.

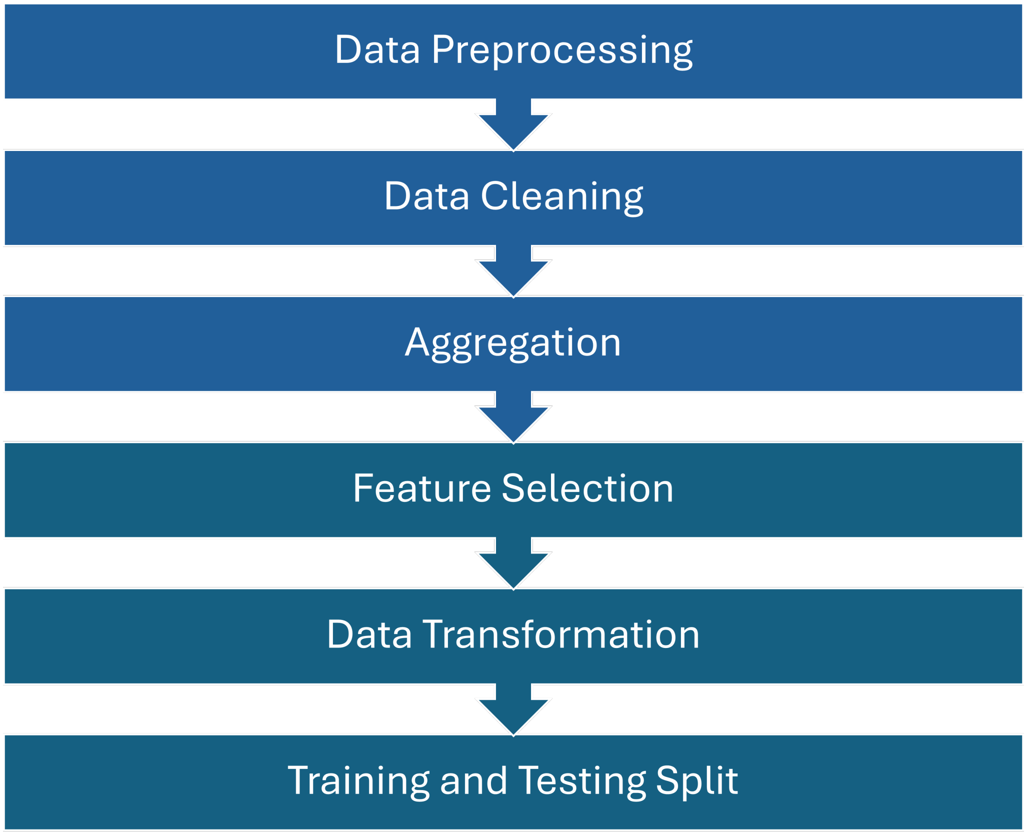


Figure 2.5: Dataset creation process, based on [19].

The first step usually consists of the removal of low-coverage parameters, replacing missing values with median to maintain data integrity, remove non-numerical parameters, and eliminate non-numeric features that are not useful for analysis.

After that, the aggregation process begins by summarising and consolidating the data. After that, the feature selection is realised, which starts by first summarising and consolidating the data and then selecting relevant features and removing redundant ones.

The use of multiple datasets allows the validation of the models that are being tested, allowing the generalizability of the results. It also permits us to compare the efficacy of predictive maintenance tools against established benchmarks. Finally, the comparative dataset can help interpret results from the real dataset and vice versa.

After these processes, some data transformation is accomplished, where the normalisation of the features occurs.

To end this procedure, the data is usually split into training and testing to posteriorly evaluate the model's performance using supervised learning algorithms.

Feature engineering is a technique that creates new features based on the existing features of the dataset [20]. Some of the most common techniques are feature creation and feature extraction. Feature creation involves generating new features by creating new variables used in the machine learning models [21]. Feature extraction is based on extracting the most relevant features from a raw data sample used as input to the models. The most common domains to be used to extract these variables are time, frequency and time-frequency [10].

2.1.7 Data-driven methods

To predict the behaviour of the machines and determine when an intervention should be conducted to restore some components, it is necessary to evaluate and analyse the data available from the sensors.

There are three main types of data-driven methods, as shown below [22]:

1. **Statistical approaches**;
2. **Stochastic approaches**;
3. **Machine Learning Techniques**.

The first two methodologies are best suited for examining complex systems whose evolutionary processes are not straightforward to anticipate.

The third one is the approach that will be considered in this work.

This can be summarised in the following figure 2.6.

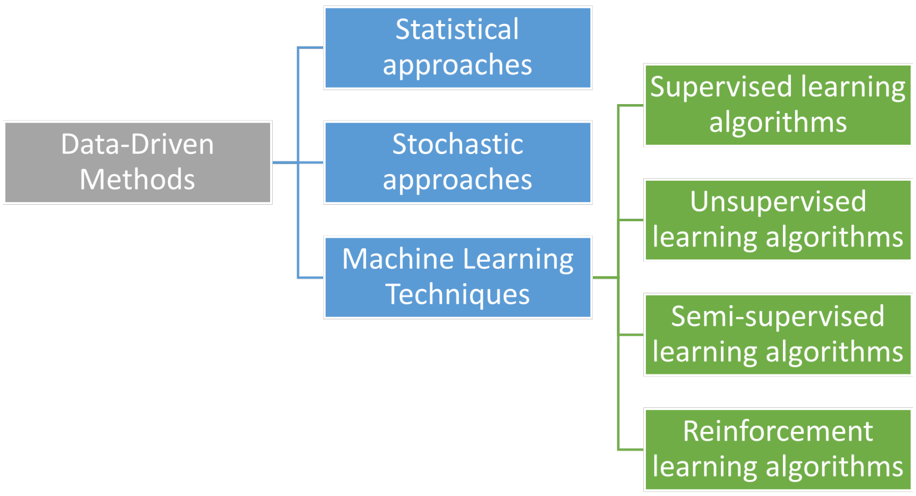


Figure 2.6: Types of data-driven methods and the fourth main methodologies of machine learning

The landscape of machine learning features four principal methodologies. The difference between them is the way that each other learns the data to make predictions:

- **Supervised learning algorithms**: it uses labelled training data, which means "input variables and the corresponding output are supplied to the machine to learn a mapping from the inputs to the outputs, often adjusting the model iteratively" [22]. After repeating the process multiple times, the model attains the required level of accuracy, successfully predicting outputs for new data. Regarding the four types, this one is probably the most used;

- **Unsupervised learning algorithms**: the data is not labelled, and the model self-learns to extract patterns and structures from the data without external guidance;
- **Semi-supervised learning algorithms**: it's a combination of labelled and unlabeled data regarding the inputs of the model;
- **Reinforcement learning algorithms**: The system can learn through rules, trial runs, and error correction to uncover the most advantageous actions.

Regarding the ml algorithms, we can divide them into three main categories, as shown in figure 2.7:

- **Traditional machine learning approaches**
  - Linear Regression (LR)
  - Gaussian Process Regression (GPR)
  - Artificial Neural Network (ANN)
  - Support Vector Machine (SVM)
  - k-Nearest Neighbors (k-NN)
  - Decision Tree (DT)
- **Deep Learning Approaches**
  - Long short-term memory (LSTM)
  - Autoencoder (AE)
  - Convolutional neural network (CNN)
  - Recurrent neural network (RNN)
  - Deep beliefs networks (DBN)
  - Generative adversarial network (GAN)
  - Transfer learning
  - Deep reinforcement learning (DRL)
- **Ensemble**
  - Bagging
    - Random forest (RF)
    - Extra Trees
  - Boosting
    - Gradient Boosting Machine (GBM)
    - Extreme Gradient Boosting (XGBoost)

        * Adaptive Boosting (AdaBoost)
        * Light Gradient Boosting Machine (LightGBM)
        * Categorical Boosting (CatBoost)
    - Stacking

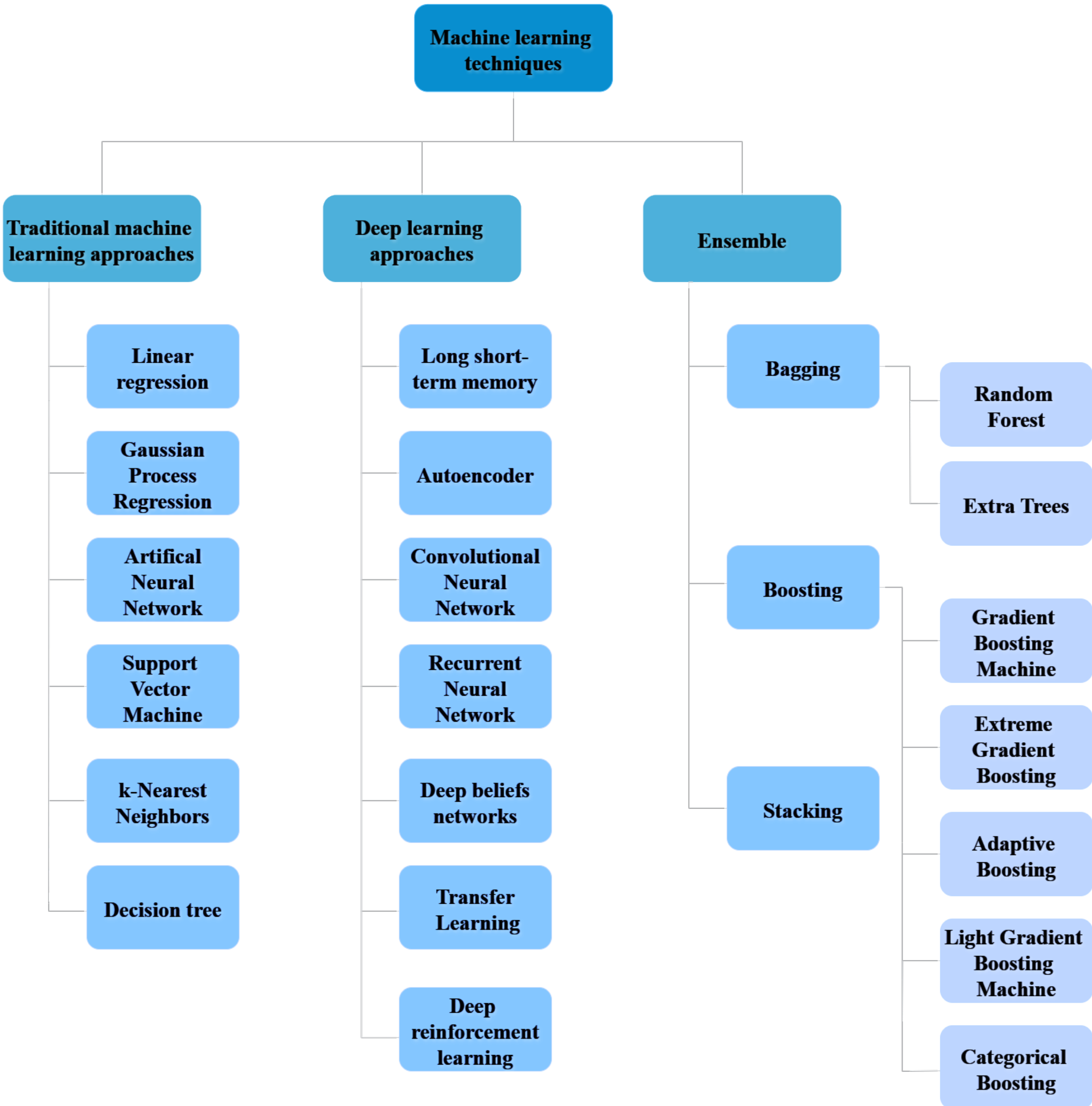


Figure 2.7: Machine learning algorithms divided into the three main categories.

The first technique involves using simpler algorithms that are more easily interpreted.

The second one uses multilayered artificial neural networks to create a more suitable mapping function between given inputs and outputs. It usually requires a considerable amount of data to achieve better accuracy.

Ensemble learning methods involve combining two or more ML algorithms to achieve better performance than using individual algorithms separately. Instead of depending on a single model, the predictions from the individual algorithms are combined using a specific rule to produce a more accurate single prediction. Ensemble methods are typically categorised into parallel (bagging) and sequential ensembles (boosting). Furthermore, parallel ensembles can be classified as homogeneous or heterogeneous, depending on the similarity of the base learners. Homogeneous ensembles comprise models constructed using the same ML algorithm, while heterogeneous ensembles consist of models from different algorithms. The effectiveness of ensemble learning techniques primarily depends on the accuracy and variety of the base learners. Bagging is an ensemble learning approach where techniques like random forest and Extra Trees build multiple decision trees on different subsets of the data and combine their predictions to enhance performance and reduce overfitting. Boosting is a machine learning technique that can convert weak learners into a robust classifier. It is an ensemble meta-algorithm to decrease bias and variance [23].

2.1.8 Data-driven methods and survival analysis

Incorporating machine learning techniques into traditional survival analysis improves the capacity to work with complex, high-dimensional datasets. While survival analysis helps to analyse time-to-event data, such as the Kaplan-Meier estimator and the Cox proportional hazards model, these methods may have limitations when dealing with high-dimensional data or complex, non-linear relationships among variables. This is where machine learning models can complement traditional survival analysis.

Some of the data-driven methods, such as support vector machines, random forests, and gradient boosting machines, can be adapted using the *scikit-survival* package to calculate survival functions.

RF, as seen above, is an ensemble learning method that operates by constructing a multitude of decision trees during training and outputting the mode of the classes or mean prediction of the individual trees. When combined with survival analysis, this method becomes known as the Random Survival Forest (RSF), which extends the capabilities of traditional Random Forests by allowing the modelling of time-to-event data. The RSF, as said in [24], is a recursive process of a growing tree, where from the top to the bottom of a tree, each node is split into two daughter nodes following the criteria to maximize the survival difference between the resulting daughter nodes. This mechanism helps separate dissimilar observations at each node. As a result, it achieves a fully grown tree with terminal (or leaf) nodes containing only cases with similar survival profiles.

SVM, when adapted for survival analysis, can censor data, creating a model that predicts the time until an event occurs.

When applied to survival analysis, gradient boosting can be adapted to handle censored data and predict survival outcomes. This adaptation is referred to as Gradient Boosting for Survival Analysis or Survival Gradient Boosting.

2.2 State of the art

2.2.1 Research Methodology

2.2.1.1 Systematic Review Objectives

This literature review presents the results related to Data-Driven Methods and Predictive maintenance. This research aims to answer the following questions:

1. How can predictive maintenance by utilising data-driven methods and Industry 4.0 technologies improve equipment failure prediction accuracy and timeliness while considering the remaining useful life?

2. What are the best data-driven degradation models to predict the machines' failure and evaluate the equipment's performance, reduce maintenance costs and downtime, and increase production efficiency?

3. How can implementing a comprehensive data-driven maintenance framework combined with effective decision support systems facilitate predictive maintenance planning and resource allocation in an Industry 4.0 environment, ensuring the optimisation of maintenance activities while minimising the risk of unexpected failures and production interruptions?

As identified by the questions above, this review aims to understand the concepts of Industry 4.0 better and learn about the data-driven methods already used in predictive maintenance. Finally, the topic of optimisation will also be searched.

Based on this, some keywords were chosen to search for some papers. The keywords used were predictive maintenance, condition-based maintenance, industry 4.0, survival analysis, degradation modelling, remaining useful, and data-driven methods.

2.2.1.2 Search Strategy and Results

In this step, the strings and database for the search were defined. The database chosen as the starting point was Scopus [25] since it would have a more focused approach for our research.

By using the combination of the following strings within the field of "Article title, Abstract, Keywords":

- (("Predictive maintenance") OR ("Condition-based maintenance")) AND "Industry 4.0" AND (("Survival analysis") OR ("Remaining useful life" OR "RUL"))

- (("Predictive maintenance") OR ("Condition-based maintenance")) AND "Industry 4.0" AND (("Survival analysis") OR ("Remaining useful life" OR "RUL") OR ("Data-Driven Methods"))

On the database, Scopus [25], it was achieved with the first string a total of 107 results with the distribution of the papers relative to the year, shown in figure 2.8.

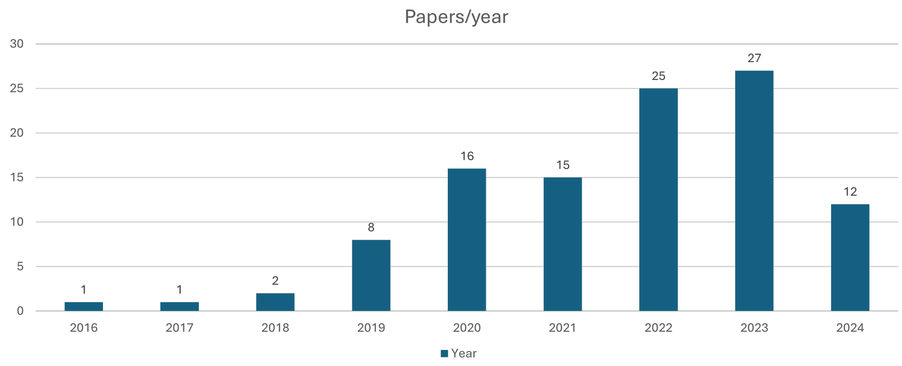


Figure 2.8: Evolution of the number of publications per year for the first string.

With the second string, where the keyword "Data-Driven Methods" was added, the search increased to 116 documents, with the distribution of the papers relative to the year shown in figure 2.9.

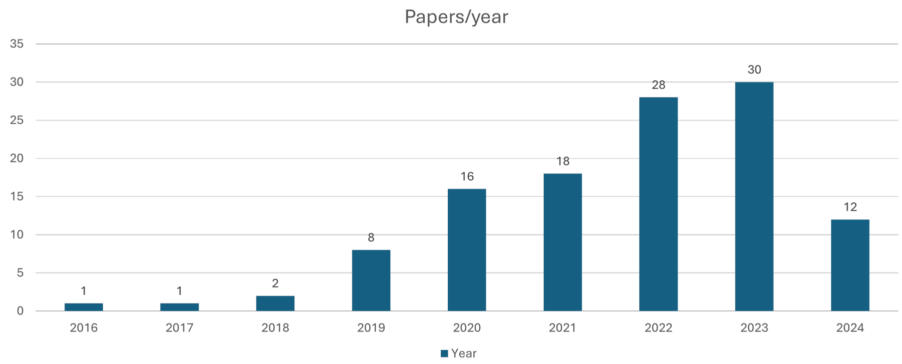


Figure 2.9: Evolution of the number of publications per year for the second string.

When trying to specify the string more by adding the keyword "Data-Driven Methods" to have the keyword necessarily, it was obtained a total of 6 papers in the same database, with only one paper in the years 2017, 2019, 2021, 2022, 2023 and 2024. The string used was (("Predictive maintenance") OR ("Condition-based maintenance")) AND "Industry 4.0" AND (("Survival analysis") OR ("Remaining useful life" OR "RUL")) AND ("Data-Driven Methods").

2.2.1.3 Article Selection

To select the final articles for this review, it was considered the following elements:

- The publication is in English;
- There's open access to the publication;
- The publication does refer to predictive maintenance or data-driven methods or Industry 4.0;
- Document has more than four pages.

After this first filtering, the documents were read and evaluated using the three parameters below.

- Yes (1.0)
- More a less (0.5)
- No (0.0)

These parameters would evaluate the subsequent quality questions formulated to choose the papers to consider in the literature search.

- Does the paper refer to different data-driven methods?
- Does the paper refer to the methodologies of machine learning models?
- Does the paper fits into the industry 4.0?
- Does the paper indicate different types of industrial maintenance?
- Does the paper address survival analysis?
- Does the paper refer to RUL?
- Does the paper explain how to process the data?
- Does the paper explain the different machine-learning algorithms?

In the end, they were eliminated if the papers didn't sum a total score of at least 2.0.

2.2.2 Review of Search Results

After choosing the papers, the table 2.1 was created to identify the most relevant information about each article.

Table 2.1: Papers and relevant search topics.

Reference	Industry 4.0	Dataset Processing	Types of industrial maintenance	Machine learning algorithms	RUL	Survival Analysis
[6]	✓	✓	✓	✓	✓	
[7]	✓	✓	✓		✓	✓
[8]	✓		✓	✓	✓	
[9]	✓	✓	✓	✓		
[10]	✓	✓	✓	✓	✓	
[12]		✓				✓
[13]	✓	✓	✓		✓	✓
[19]		✓		✓	✓	
[22]	✓	✓	✓	✓	✓	
[26]		✓	✓	✓	✓	✓
[27]	✓	✓		✓	✓	

Starting with the first paper, Ran et al. in [6] refer to the different types of maintenance and write about the different ML algorithms in an Industry 4.0 context. The ML algorithms written about are the artificial neural network, decision tree, support vector machine, k-nearest neighbours, autoencoder, convolutional neural network, recurrent neural network, deep belief networks, generative adversarial network, transfer learning and deep reinforcement learning. These algorithms are mentioned when predicting the RUL of the components. The Open System Architecture for Condition Based Monitoring (OSA-CBM) is the framework used to design and implement the PdM system. This system has different layers where data acquisition, data manipulation, and prognostics of the current health state of the system are done. On page 9 of the paper, a system architecture for the PdM in the context of 4.0 is presented, where a data preprocessing module is presented. The document thoroughly reviews the existing approaches, including knowledge-based, traditional ML-based, and DL-based approaches. However, it doesn't mention any survival approaches.

Moat and Coleman in [7] start by explaining the different types of maintenance that exist and proceed to describe the data cleaning process. It frames predictive maintenance as an essential maintenance in Industry 4.0. It then proceeds to write about survival analysis, which details the Kaplan-Meier Methods and the Cox Proportional Hazard Model. It mentions the data cleaning and lightly mentions the RUL. It uses Python's library 'lifelines' to perform the KM method and model the Cox PH model.

In [8], Ramezani et al. start to describe Industry 4.0 and the different types of industrial maintenance. After that, they talk about ML algorithms used for predictive maintenance, such as recurrent neural networks, long short-term memory, convolutional neural networks, temporal convolutional networks, generative adversarial networks, autoencoder, support vector machines, relevance vector machines, and Gaussian process regression. These models' performance is used for RUL prediction. Other challenges that the authors mention in this paper are the problems with data capturing, precision in fault detection, the creation of prototypes based on large datasets and the scalability and efficiency of a model.

In [9], Achouch et al. write about Industry 4.0 and explore the different types of industrial maintenance and Predictive Maintenance Models. It presents some challenges that predictive maintenance still encounters, such as data source limits and machine repair activity limits. After this, it explains the predictive maintenance workflow and describes data collection, understanding, and preparation as essential steps. After this, it enumerates the different PdM models, including a full explanation of RUL and the models to predict it. It ends with a case study example used to estimate the remaining life of a centrifugal compressor. In this paper, no survival approaches or methods are mentioned.

Gupta et al. present a complete critical review in [10], with a paper published in 2024. It starts with framing the importance of PdM in Industry 4.0 and later explains some data-driven methods for applying PdM. It presents the different maintenance strategies that are being used in industries nowadays. It is written about the process of data acquisition, data preprocessing and feature extraction and selection, providing a deeper explanation of those processes. Then, proceed to explain in more detail the RUL prediction and then present some data-driven techniques for PdM, such as decision tree, support vector machine, K-Nearest Neighbor, random forest, ANN and K-Means, where it presents the different applications, advantages, limitations and references for each model. The analysis concludes by addressing the challenges and future work that needs to be done. It discusses the effect of noise on the quality of data captured by sensors and how it can decrease the performance of predictive models. However, it suggests that future studies should investigate the impact of noise in more detail to enhance the robustness of predictive models.

In [12], it explains some methods from survival analysis to estimate a survival function and a hazard function, describing some parametric and non-parametric estimators (KM estimator).

Frumosu et al. present a case study on the plastic injection moulding industry in [13], which frames it into the Industry 4.0 context. It briefly introduces the different types of industrial maintenance that are being used, as well as the injection moulding process. Then, a short description of survival analysis is explained, and the Random Survival Forest model and its advantages are presented. After that, it describes the data processing process and shows the survival curves using the Kaplan-Meier estimator and random survival forest.

Matuszczak et al. work with linear regression, random forest and extremely randomised trees, support vector machines, XGBoost, neural networks – multilayer perceptrons and ensemble models in [19]. Those are also used to predict RUL. The dataset creation, aggregation, feature selection and data transformation are also mentioned. This paper discusses the different Python packages used to develop the different models. There is also a section that explains the validation process. There is also no reference to survival models.

Arena et al. in [22] makes a literature review about the different data-driven methods and their approaches. They also indicate different ML algorithms, such as linear regression, gaussian process regression, artificial neural network, decision tree, support vector machine, and k-nearest neighbours, related to the traditional ones and long short-term memory, autoencoder, convolutional neural network, recurrent neural network, deep beliefs networks, generative adversarial network, transfer learning, deep reinforcement learning, and random forest related to the deep learning approaches.

On the other hand, Ouda et al. in [26] starts by mentioning the preprocessing method and the uses the logistic regression, random forest and gradient boosting classifier by using Scikit-learn library as ML algorithms to predict the failures. Later in the article, they propose a mathematical optimisation model to reduce maintenance costs.

Kumar Sharma in [27] approaches linear regression, support vector machine, decision tree, random forest, XGBoost and multilayer perceptron algorithms to predict the tool wear and RUL. It also mentions the data pre-processing and feature engineering.

Table 2.2: ML algorithms and DL techniques implemented in the different papers.

ML Algorithm	Reference
AE	[6, 8, 10, 22]
ANN	[10]
BDP	[6, 10, 22]
CNN	[6, 8, 10, 22]
DBN	[6, 10, 22]
DRL	[6, 22]
DT	[6, 10, 27]
ExtraTrees	[19]
GAN	[6, 8, 22]
GNN	[10]
GBC	[26]
GPR	[8, 22]
k-NN	[6, 10, 22]
Logistic Regression	[26]

ML Algorithm	Reference
LR	[19, 22, 27]
LSTM	[8, 10, 22]
MLPNN	[19, 27]
RF	[10, 19, 22, 26, 27]
RNN	[6, 8, 22]
RVM	[8]
SVM	[6, 8, 10, 19, 22]
SVR	[27]
Transfer Learning	[6, 10]
TCN	[8]
XGBoost	[19, 27]

Based on this table, two algorithms were mentioned in five different papers. Based on Gupta et al. in [10], the following ML techniques are described below.

- **Random Forest**: The random forest (RF) algorithm is an ensemble learning approach widely used for classification and regression tasks. It operates by creating multiple decision trees, as illustrated in Figure 2. These trees are generated from various samples, each incorporating a degree of randomness. The term "random" in RF refers to using unique and randomly selected subsets of predicted values for each tree. Instead of relying on a single decision tree, RF combines the predictions from multiple trees, using majority voting for classification and averaging for regression. Despite being composed of decision trees, random forests and individual decision trees differ significantly. Random forests build decision trees using random subsets of features, resulting in varied and smaller trees. This approach helps mitigate overfitting, a common issue with deep decision trees, and generally yields more accurate predictions by leveraging the diversity of the ensemble. The schematic of the technique can be found in figure 2.10.

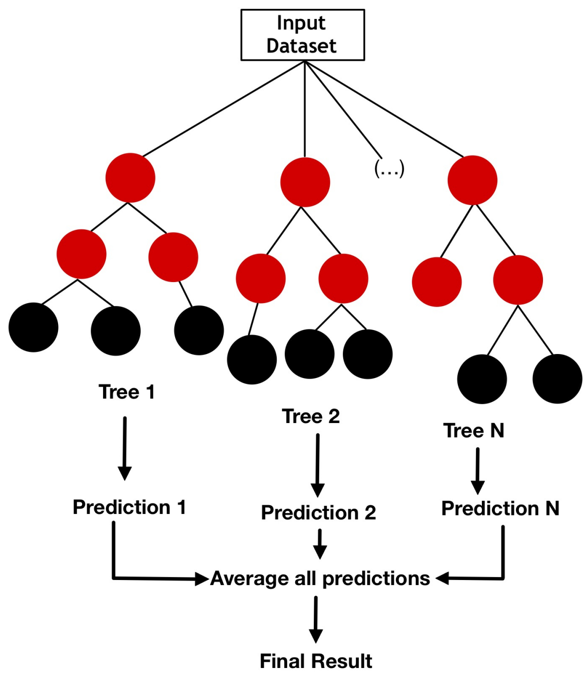


Figure 2.10: Random Forest Algorithm Schematic.

- **Support Vector Machine**: Support Vector Machine (SVM) is a prominent supervised machine learning technique widely used for regression and classification tasks. The algorithm aims to identify the optimal decision boundary, the decision surface or hyperplane, to classify data points in an m-dimensional space into distinct groups or classes. The hyperplane is chosen because it provides the most significant margin width, ensuring the closest possible spacing between the positive and negative data points. SVM identifies the extreme points that help form this hyperplane, known as support vectors, which give the algorithm its name. SVM is capable of handling a large number of features, provided there is sufficient computational power available. SVM divides two distinct classes with a decision surface or hyperplane. This model is vastly used in machine fault diagnosis and prognosis due to its effectiveness and precision. In figure 2.11 can be found the schematic of the technique of SVM.

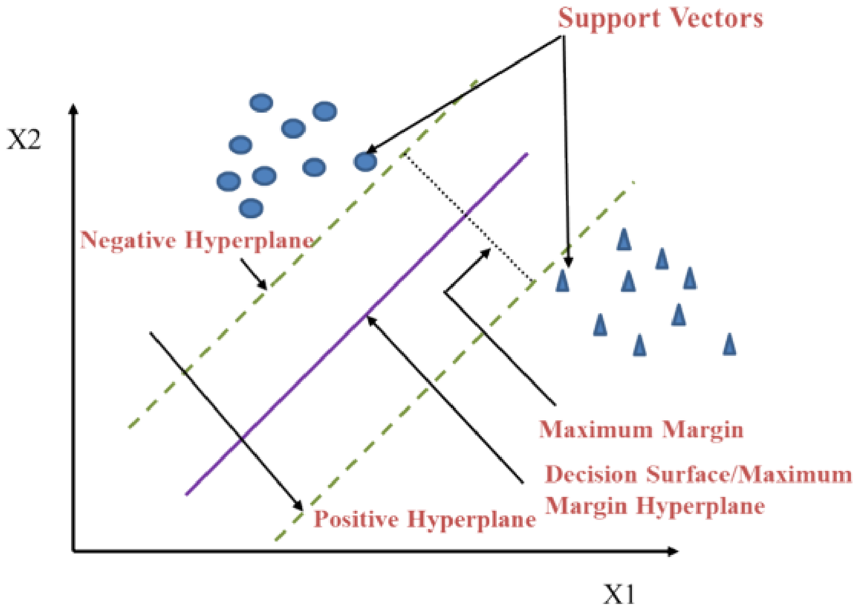


Figure 2.11: Support vector machine Algorithm Schematic, taken from [10].

Both algorithms are commonly used in RUL prediction.

2.3 Conclusions

To conclude, it is observed that the results of this research are practically only available from 2016. Emphasises the fact that prediction maintenance allied with data-driven methods is a technique that is recent and still being developed. This work presents a general tool that can be used in the equipment of different sectors, from health to security to automotive. When searching for papers related to the scikit-survival module and the implementation of its models and performance measures, most papers are still related to medicine and survival trials in these areas using these models. Therefore, in the context of Industry 4.0 and predictive maintenance, there is not much scientific documentation specifically related to all three of these topics. From January to July, searching on the same database with the exact first string mentioned on 2.2.1.2 verified a growth of 18 papers, which shows an increased interest in the topic. Also, the papers analysed from 2024 present a full literature with almost all the issues used as keywords in the search as opposed to the papers available from the past years.

Chapter 3

Systems' Architecture

In section 3.1, the architecture of the proposed data-driven predictive maintenance system is presented, with the demonstration of the system's design in section 3.2 and, finally presents how the validation of the system can be realised on section 3.3, with the comparison of the predictions using the CoxPH, RSF, GBSA, and FS-SVM models.

3.1 Problem definition

This project focuses on developing a Data-Driven Predictive Maintenance system to extend the life cycle of industrial components. The primary problem addressed is the unpredictable nature of component failures, which can lead to costly downtime and maintenance. By analysing historical data and implementing survival models, the system seeks to predict when different components are likely to fail, allowing for proactive maintenance and reducing unexpected downtimes.

The scope to solve this problem includes:

- **Data Preprocessing:** Handling data from multiple components and preparing it for analysis.
- **Predictive Modeling:** Utilising machine learning models to forecast component degradation and potential failures.
- **Maintenance Planning:** Translating model predictions into actionable maintenance plans to extend component life cycles.

To solve and organise the project, a systems architecture was created in order to obtain a better visualisation.

3.2 System's Design

The primary goal of this dissertation is to develop a data-driven predictive maintenance tool that extends the life-cycle of diverse industrial components by analysing different data and creating different survival models that will predict when the various components fail.

A primary figure was created to demonstrate how the system was developed to achieve the final result.

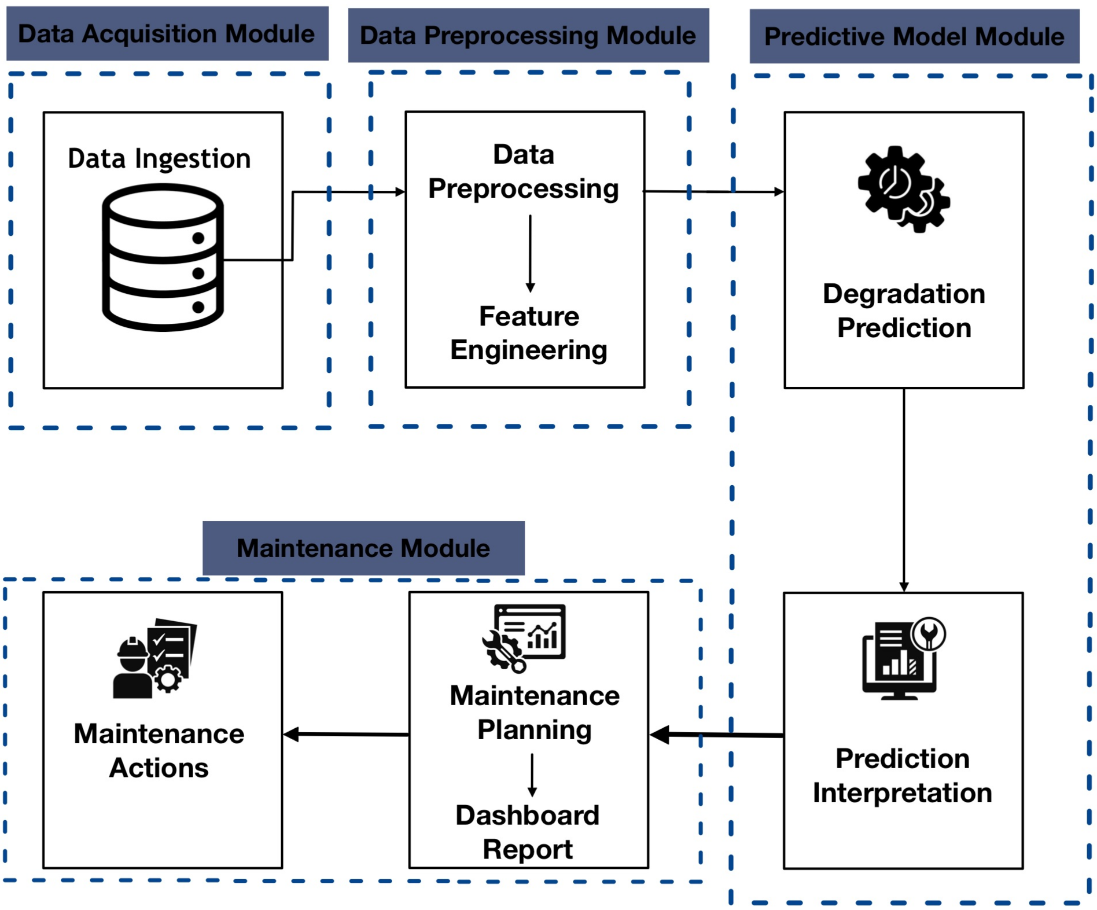


Figure 3.1: Systems' Architecture Overview.

As illustrated in figure 3.1, the systems' architecture for our predictive maintenance solution is organised into four main modules, each contributing to the overall process of predicting and managing maintenance needs. This modular design improves the system's flexibility.

Since the data was already included in the datasets, an exploratory data analysis (EDA) was conducted to analyse raw data from various sensors and sources. It serves as the foundation for the entire system, providing the essential inputs needed for further analysis and modelling. In this project, the data is ingested since the files that contain it are already ready to be preprocessed.

Afterwards, the data preprocessing module ensures that it undergoes a preprocessing phase where it is cleaned, normalised, and transformed into a suitable format for analysis. This step includes feature engineering, which extracts valuable insights from the raw data to enhance the accuracy and efficiency of the predictive models.

The predictive model module employs machine learning algorithms to predict components' degradation and potential failures. The insights gained from these predictions are crucial for planning maintenance activities, thus preventing unexpected downtimes and extending the lifespan of the components.

The final maintenance module interprets the predictions generated by the models and provides actionable insights for maintenance planning. This includes developing reports and dashboards that aid decision-making and ensuring timely and effective maintenance actions based on data-driven insights.

Overall, this architecture illustrates the flow from data collection to actionable maintenance decisions, ensuring the system's effectiveness in managing and predicting maintenance needs.

3.3 Validation of the system

To validate the effectiveness of the proposed system, two datasets were employed:

1. **Gorenje Industrial Dataset:** This dataset contains data from a spot welding robot, including metrics like current measurements and machine cycles. It provides a real-world industrial environment to test the applicability of the models in a real context. The data was preprocessed, including merging sensor and counter data, calculating statistics, and labelling cycles based on failure history. In this dataset, there were two robots. However, only the R2 was used for the project. This one was related to spot welding and had a file which contained the failures and stops related to different machines. Related to the robot R2, we could see a change in the Electrode, which had a time of failure of 5 minutes and the error code 9. This failure occurred twice in August, October, November of 2022, and February of 2023. It occurred once in September of 2022 and three times in January of 2023.

2. **Microsoft Azure Predictive Maintenance Dataset, from Kaggle:** This dataset contains operational metrics and sensor readings, such as voltage, rotation, pressure, and vibration. It is used to evaluate system performance and benchmark against predictive maintenance models. This system contains a file with five errors encountered by the machines while in operating condition. However, these errors don't shut down the machine and are not considered failures. It also contains a file with the information on when a component is replaced when it occurred, and in each component. This replacement can be done during a regular technician visit (using proactive maintenance) or when a component breaks down, and the technician needs to do unscheduled maintenance to replace the component (using reactive maintenance). When this failure occurs, it is recorded when the replacement due to failure occurred. The unexpected failures in machine 1 occurred seven times during the 2015 year: 1 time in component 1, 2 times in component 2 and 4 times in component 4. The development of this work considered all the replacements as inputs to calculate the RUL of the different components. The proactive maintenance was usually done twice monthly, at the beginning and then in the middle of the month.

Predicting failures in both datasets will help prevent unnecessary technician visits through reactive or proactive maintenance. This allows for the maximum use of components without downtime and helps minimize technician visit expenses and component failures.

The validation process involved comparing the predictions made by different survival analysis models, such as Cox Proportional Hazards, Random Survival Forest, Gradient Boosting Survival Analysis, and Survival Support Vector Machines. These models were assessed using the concordance index to measure their accuracy and reliability. The metric was chosen based on [13] since it is the most frequently used evaluation metric of survival models [28].

The Gorenje dataset's applicability in a real industrial setting demonstrates the system's practical relevance. By proving the system's effectiveness in predicting maintenance needs in this context, we establish its potential for broader applications in various industrial environments.

3.4 Conclusions

The comprehensive system design and validation ensure that the predictive maintenance tool is applicable in real-world scenarios. Using dual datasets and a comparative analysis provides confidence in the system's ability to predict maintenance needs, potentially saving significant costs and extending the life cycle of critical components.

Chapter 4

Implementation

In this chapter, we will explore the details of implementing the data-driven predictive maintenance system designed to prolong the life cycle of critical components. The implementation is organised into separate modules mentioned on 3.1, each handling a specific aspect of the data processing and maintenance prediction workflow. This modular approach guarantees scalability, flexibility, and ease of maintenance. The section 4.1 demonstrates the EDA that was performed in both datasets and the data that each one contained. Then, in section 4.2, it is shown how the data is preprocessed for both datasets, using similar approaches of merging and cleaning data, feature engineering techniques and similar processes to save data, using similar approaches for both datasets, the real dataset scenario and one dataset that was taken from Kaggle [29]. Subsequently, in section 4.3, the different predictive models based on survival analysis implemented are described. To conclude, section 4.4 presents the implementation of the simulation created related to the second dataset.

4.1 Exploratory Data Analysis

The Data Acquisition Module involves ingesting pre-collected raw data from various sources. This module ensures data is appropriately structured and available for subsequent processing stages. The data includes sensor readings and operational metrics necessary for predictive maintenance. First, a dataset from Gorenje [30], related to component welding, was used to develop this dissertation.

The second posteriorly used dataset was taken from Kaggle [29], a Microsoft Azure predictive maintenance dataset [31], created for predictive maintenance model building.

4.1.1 Gorenje dataset

The real dataset consisted of data produced with two robots, spot welding R2 and seam welding R6. However, only the spot welding R2 was considered for this work. These datasets were presented on diverse CSV files. The *R2.csv* presented minimum, maximum and mean current of the welding component from 2022-09-30 10:43:27+00:00 till 2023-03-08 14:45:25+00:00, with a total of 629444 samples. Then, the *rotary_table.csv* presented a value from 2023-03-08 12:15:47+00:00 till 2023-03-08 12:14:32+00:00, with 96869 samples. Then there was a file called *join_failures.csv* that presented when the failure occurred, the time of failure and the error code number.

- *R2.csv*: This file contains the minimum, maximum and mean welding current values.
- *rotary_table.csv*: This file contains counter values over time, represented by timestamps (*present_time*) and counter values (*value*). These counter values are crucial for determining machine cycles, indicating the machine's operational status. Finally
- *join_failures.csv*: It contains the data about the machine failures, including the dates and reasons for failures.

4.1.2 Microsoft Azure Predictive maintenance dataset

The second dataset used for comparison was taken from Kaggle, a Microsoft Azure predictive maintenance dataset. This dataset was created to build predictive maintenance models and included various operational metrics and sensor readings necessary for predictive analysis. It includes diverse CSV files with different metrics. All the telemetry is collected at an hourly rate.

The first file, *PdM_telemetry.csv*, consists of hourly average measures of voltage, rotation, pressure and vibration collected from 100 machines for 2015. It has a total of 876101 samples.

The file *PdM_errors* contains 3920 samples, each including the datetime, the machineID (ranging from 1 to 100), and the specific error that occurred at each machine at a particular datetime. These errors are encountered while the machines are in operation, but they do not cause them to shut down, so they are not considered failures. Since the telemetry data is collected hourly, the error date and times are rounded to the closest hour.

PdM_maint.csv contains 3287 samples and containers, the datetime, the machineID (ranging from 1 to 100) and the replaced component. Components can be replaced under two situations. The first can be during the regularly scheduled visit when the technician replaces it (Proactive Maintenance). The second occurs when a component breaks down, and the technician does an unscheduled maintenance to replace the component (Reactive Maintenance). The file *PdM_failures* shows the replacement of a component due to failure. This is a subset of Maintenance data.

The last file, *PdM_machines.csv* presents each machine's model type and age.

Since there was only data related to the telemetry from 2015, the preprocessing module was only considered the data from this year.

4.2 Data Preprocessing

Firstly, the data was preprocessed, and some feature engineering techniques were applied to develop the predictive models. For both datasets, a similar approach was used to preprocess the data, which was divided into the merging and cleaning data, the feature engineering and the saving processed data.

For the first dataset, Gorenje's dataset, this was done by creating two scrips, *cycles_calc.py* and the *surv_model.py*. The initial step involved loading raw data from various CSV files, such as *rotary_table.csv*, *R2.csv* and *join_failures.csv* mentioned on 4.1. After the load process, the robot failures and sensor data were selected to be used posteriorly.

A similar preprocessing and feature engineering approach was applied to the Microsoft Azure Predictive Maintenance dataset. The *join_telemetry_failure.py* and *join_tudo.py* scripts were created for this dataset.

4.2.1 Merging and Cleaning Data

In Gorenje's dataset, the next step was merging the sensor data with the counter data to align timestamps correctly, using the *pd.concat* function. This provided a comprehensive view of the machine's operational status, incorporating sensor readings and counter value. This helped to determine the machine's operational status and identify cycles posteriorly.

For the Microsoft Azure Predictive Maintenance dataset, after using the *join_telemetry_failure.py* script, the telemetry data was aligned with the failure data to match the timestamps accurately, ensuring a cohesive dataset for analysis. Then, the maintenance and error features were added with the *join_tudo.py* script.

4.2.2 Feature Engineering

Feature engineering was performed to extract meaningful metrics from the raw data calculate the machine's cycles, and to prepare the data for predictive maintenance modelling. The approach was very similar for both datasets but had some specifications for each one.

- **Determining Machine Cycles**: For the Gorenje dataset, the counter values in *rotary_table.csv* were used to determine machine cycles. By calculating the difference between successive counter values, the script identified the duration of each cycle (*cycle_time*). Each cycle represents a complete operational period during which the machine performs welding. When the component is replaced, the cycle count restarts, indicating the beginning of a new operational period. Additionally, an inverse sequence of the cycles was created to track the time to failure (TTF) for the component, helping to predict when a failure might occur based on past performance.

  For the Microsoft Azure Predictive Maintenance dataset, before calculating the cycles, the dataset was filtered to only include data from Machine 1. After that, for each component (comp1, comp2, comp3, comp4), cycles were calculated by counting the number of operations since the last replacement event. This was done by grouping the data based on replacement events and using cumulative counts (*cumcount*) to generate a cycle number for each operational period. After that, each component's Time to Failure (TTF) was calculated as an inverse sequence of the cycles.

- **Computing Sensor Data Statistics**: On the Gorenje dataset, statistics such as the mean and standard deviation of current sensor readings were computed for each cycle. This was done by segmenting the data into individual cycles and calculating the statistics for each segment. These statistical measures allow us to understand the machine's behaviour during each cycle and could indicate potential issues such as instability or irregular performance. For instance, a high standard deviation in current readings might suggest fluctuations in power usage that could lead to the failure of the components

- **Labeling Cycles**: In the Gorenje dataset, each cycle was labelled as either in normal operation (censored) or resulting in a failure (dead) based on historical failure data when a component was replaced due to it. This labelling is essential for training predictive models as it can differentiate between healthy operations and those that are likely to result in component failure by accurately labelling the cycles.

- **Input Variables for Modeling**: For the first dataset were used five input variables to train the models to predict the RUL of the components. For the Gorenge, the following input variables were then used:

  - *cycle_time*: duration of each operational cycle;
  - *current_mean*: average current during each cycle;
  - *current_std*: variance in current during each cycle;
  - *cum_hour*: cumulative hours of operation;
  - *cum_cycles*: cumulative cycles completed;

  For the Microsoft Azure Predictive Maintenance dataset, the following input variables were used to train the models:

  - *cycle_comp1, cycle_comp2, cycle_comp3, cycle_comp4*: number of cycles since the last replacement for each component;
  - *ttf_comp1, ttf_comp2, ttf_comp3, ttf_comp4*: Time to Failure for each component, calculated as the inverse of the cycle count;
  - *Sensor Data (Voltage, Rotation, Pressure, Vibration)*: These features were used to understand the operational conditions under which the machine components are operating;

Like the process done on the Gorenje dataset, a similar approach was applied to the Microsoft Azure Predictive Maintenance dataset and was then performed on *cycles_calc_newdata.py* script the filtration of the DataFrame only to include the data from Machine 1. After that, each cycle for the four components was calculated based on whether the component had been replaced. When a component was replaced, the cycles would restart the count. Then, each component's time to failure (TTF) was calculated using an inverse string of the cycles. After this was created, the label with the cycles for each component, a boolean variable whether the component was replaced or not and the TTF (in hours) for each component.

4.2.3 Saving Processed Data

On the Gorenje Dataset, the processed data, which included engineered features and labels, was then saved to new CSV files, such as *work_cycles_raw_R2.csv*, *work_cycles_R2.csv* and *hist_replace_R2.csv*.

The second file was created after the first one, with the specification of eliminating lines where the work feature was considered false, replacing the empty values of *current_std* with 0 and creating the cumulative values and time to failure of the component. The last file was created to show when the value replacement occurred and the associated cycle.

In the Microsoft Azure Predictive Maintenance dataset, the processed data, which included engineered features and labels, was saved to a new final CSV file, such as *machine1_data_final.csv*, in order to be used in the survival prediction for each component.

4.2.3.1 Data Preprocessing and Splitting

The datasets are preprocessed by standardising the features. To address class imbalance, the dataset is downsampled by taking a fraction of the majority class and combining it with the minority class.

Subsequently, the data is divided into training and test sets using a custom function, `dataset_split`. This function splits the data, guided by the parameter `sample2test`, which defines the specific range for the test set. This method ensures the inclusion of particular samples in the test set for targeted evaluation.

- **Training Set:** Contains data points used for model training, excluding those allocated to the test set.

- **Test Set:** Comprises data used for evaluating model performance, defined by indexes from `idx_start` to `idx_end`, focusing on specific instances of interest.

The `dataset_split` function works as follows:

1. **Identify Test Set Indexes :** The test set indexes are identified from `idx_start` to `idx_end`, determined by the first occurrence of the target event as per the `sample2test` parameter.

2. **Extract Features and Labels:** Features (`x_features`) and labels (status and time-to-failure) for both training and test sets are extracted based on these indexes.

3. **Return Split Data:** The function returns the split data for training and testing: `x_train`, `y_train`, `x_test`, `y_test`.

This controlled and reproducible splitting method is crucial for evaluating the model's generalisation capability, particularly in predicting critical events.

4.3 Predictive Models

The Predictive Model Module is a critical component of the architecture designed to predict the degradation and failure of machinery. This module utilises advanced survival analysis techniques and machine learning models to estimate the time-to-failure (TTF) for the equipment.

For the Gorenje Dataset, the preprocessed data obtained in the previous module was used to train various survival analysis models and the steps involved were:

1. Loading the preprocessed data from *work_cycles_R2.csv*.

2. Select relevant features for the models, such as cycle time, current mean, standard deviation, and cumulative metrics.

3. Training and evaluating different survival models, including Cox Proportional Hazards, Random Survival Forest, Gradient Boosting Survival Analysis, and Fast Survival Support Vector Machine.

4. Visualising the survival functions and predicted times to failure.

For the Microsoft Azure Predictive Maintenance dataset, the degradation prediction and prediction interpretation were calculated similarly to the Gorenje dataset. Still, the models for each of the four components were recreated.

1. Loading the preprocessed data from *machine1_data_final.csv*.

2. Select relevant features for the models, such as voltage, rotation, pressure, vibration and cycles of each component.

3. Training and evaluating different survival models, including Cox Proportional Hazards, Random Survival Forest, Gradient Boosting Survival Analysis, and Fast Survival Support Vector Machine.

4. Visualising the survival functions and predicted times to failure.

The models were implemented similarly to both datasets and were then evaluated using the concordance index, and survival functions were plotted to visualise the estimated survival probabilities over time. In the end, each trained model was saved to be used posteriorly on the maintenance module.

After this step, on the Microsoft Azure Predictive Maintenance, the models were stored on the disk using the dump function to be used in the maintenance module.

4.3.1 Cox proportional-hazards model

The Cox proportional hazard model (PH) was the first model to be implemented. It is a method in survival analysis that models the relationship between the time of an event and one or more predictor variables (covariates). It is a semi-parametric model, meaning it does not require the specification of a particular distribution for the survival times. Instead, it assumes that the hazard function, which represents the event rate at a given time, is the product of a baseline hazard function and a linear combination of the covariates. The algorithm was implemented as follows:

Algorithm 1: Cox Proportional Hazards Model Training.

Data: Training data $x_{\text{train}}, y_{\text{train}}$, Test data $x_{\text{test}}, y_{\text{test}}$
Result: Trained Cox model, Feature importance, Survival functions
Initialize: Cox Proportional Hazards model *cox_model*;
cox_model ← CoxPHSurvivalAnalysis();
Split data: The dataset is split into training and test sets, with x_{train} and y_{train} used for model training, and x_{test} and y_{test} for evaluation;
Call: TrainModel(*cox_model*);
Print: "Feature importance:";
Print: pd.Series(*cox_model.coef_*, index=*x_features*);

The code above starts by initializing the Cox proportional hazards model [32]. After it fits the model, it estimates the coefficients for the covariates, quantifying each predictor's effect on the hazard function. After training the model, the coefficients of the covariates are extracted and printed. After that, by using the *plot_survival_function* that receives the model, x_test and y_test plots the survival function.

4.3.2 Random Survival Forest

The Random Survival Forest is a collection of tree-based models that ensures each tree is built on a different bootstrap sample of the original training data, consequently removing correlations between the trees. At each node, only a randomly selected subset of features and thresholds are used to evaluate the split criterion. The final predictions are made by combining the predictions of each tree in the ensemble [16].

Algorithm 2: Random Survival Forest Model Training.

Data: Training data $x_{\text{train}}, y_{\text{train}}$, Test data $x_{\text{test}}, y_{\text{test}}$
Result: Trained RSF model, Concordance index
Initialize: Random Survival Forest model *rsf*;
rsf ← RandomSurvivalForest(n_estimators=100, min_samples_split=10, min_samples_leaf=15, max_features="sqrt", n_jobs=-1, random_state=20);
Fit the model: rsf.fit($x_{\text{train}}, y_{\text{train}}$);
Evaluate the model using concordance index;
cindex_train ← rsf.score($x_{\text{train}}, y_{\text{train}}$);
cindex_test ← rsf.score($x_{\text{test}}, y_{\text{test}}$);
Print: " RSF Model ';
Print: "cindex train: ", round(*cindex_train*, 3);
Print: "cindex test: ", round(*cindex_test*, 3);

The RSF model above is initialized with a total of 100 trees (n_estimators), the minimum number as 10 of samples required to split a node (min_samples_split), and the number of features considered at each split (max_features). Then, the model is trained by fitting it to the training data, where each tree learns the relationship between the covariates and survival times. Ultimately, the model's performance is evaluated using the concordance index (c-index), which measures how well the predicted survival times rank compared to actual survival times. The c-index is calculated for both the training and test datasets to assess the model's accuracy and generalization.

4.3.3 Gradient Boosting Survival Analysis

Gradient Boosting leverages the principle of strength in numbers by combining the predictions of multiple base learners to create a robust overall model.

A gradient boosted model shares similarities with a Random Survival Forest: both rely on multiple base learners to generate an overall prediction. However, they differ in their combination methods. A Random Survival Forest fits several Survival Trees independently and averages their predictions, whereas a gradient boosted model is built sequentially in a greedy stagewise manner.

Algorithm 3: Gradient Boosting Survival Analysis Model Training.

Data: Training data $x_{\text{train}}, y_{\text{train}}$, Test data $x_{\text{test}}, y_{\text{test}}$
Result: Trained Gradient Boosting model, Concordance index
Initialize: Gradient Boosting Survival Analysis model *gbsa*;
gbsa $\leftarrow$ GradientBoostingSurvivalAnalysis(n_estimators=100, learning_rate=0.01);
Fit the model: gbsa.fit($x_{\text{train}}, y_{\text{train}}$);
Evaluate the model using concordance index;
$cindex\_train \leftarrow$ gbsa.score($x_{\text{train}}, y_{\text{train}}$);
$cindex\_test \leftarrow$ gbsa.score($x_{\text{test}}, y_{\text{test}}$);
Print: "Gradient Boosting Survival Analysis Model";
Print: "cindex train: ", round($cindex\_train$, 3);
Print: "cindex test: ", round($cindex\_test$, 3);

The algorithm begins by initializing the Gradient Boosting Survival Analysis model using the GradientBoostingSurvivalAnalysis class [33]. It defines the number of n_estimators as 100, which specifies the number of boosting stages in each attempt to correct the errors of the previous ones. The learning_rate is defined as 0.01, which controls the contribution of each tree to the final model. After the initialization of the model, it is trained by fitting it to the training data (xtrain, ytrain). Each new tree added to the model aims to minimize the error from the previous trees, gradually improving the model's accuracy. Then, the model's performance is evaluated for both the training and test sets.

4.3.4 Fast Survival Support Vector Machines

Fast Survival Support Vector Machines are an extension of traditional Support Vector Machines adapted for survival analysis. The FS-SVM method integrates the principles of SVM into survival analysis, aiming to find a hyperplane that maximises the margin between different survival times. Unlike traditional SVM, which is used for classification or regression, Survival SVM deals with censored data, where the event of interest has not occurred for all subjects during the observation period.

Algorithm 4: FS-SVM Model Training.

Data: Training data x_{train}, y_{train}, Test data x_{test}, y_{test}
Result: Trained FS-SVM model, Concordance index
Procedure: TrainSVM();
Initialize: Fast Survival SVM model *svm_model*;
svm_model ← FastSurvivalSVM(rank_ratio=0.0, max_iter=1000, tol=1e-5, random_state=42);
Print: "FS-SVM Model";
Fit the model: svm_model.fit(x_{train}, y_{train});
Evaluate the model using concordance index;
cindex_train ← concordance_index_censored($y_{\text{train}}['status']$, $y_{\text{train}}['ttf\_comp1']$, $-svm\_model.predict(x_{\text{train}})$);
cindex_test ← concordance_index_censored($y_{\text{test}}['status']$, $y_{\text{test}}['ttf\_comp1']$, $-svm\_model.predict(x_{\text{test}})$);
Print: "cindex train: ", round($cindex\_train[0]$, 3);
Print: "cindex test: ", round($cindex\_test[0]$, 3);
Return: *svm_model*, *cindex_train*, *cindex_test*;

The algorithm begins by initializing the Fast Survival SVM model using the FastSurvivalSVM class. It then initializes the rank_ratio at 0 since it only has the objective of regression and is focused on survival time prediction without any ranking component [34]. The maximum interactions are then defined as 1000 so the model can converge to an optimal solution. It is also defined as the tolerance level for stopping the optimization. The model is then trained and fitted with the training data, where it attempts to find the hyperplane that best separates the survival times while maximizing the margin between different survival times. Finally, the model's performance is measured for the training and test data.

4.4 Maintenance Simulation

Firstly, a simulation was created where the different models produced in each component would be called using the load() function. In this simulation, a machine class was created to simulate a machine with multiple components that may fail and need replacement. Inside this class, diverse methods were created, such as:

- log_event: that logs significant events, such as component failures and repairs, for debugging and record-keeping, including timestamps and costs associated with the maintenance of the components;

- predict_time_to_failure: where is specified the model to predict the remaining useful life of a component based on its features;

- run_machine: It simulates the machine operation and decrements the remaining life of components, checks for failures, and starts repairs when necessary. It also checks component lifetimes at the end of each simulated day;

- log_remaining_times: It logs the remaining times for all components at regular intervals for monitoring purposes;

- update_remaining_times: Regularly updates the remaining times of components every 8 hours;

- start_repair: Manages the repair process of a failed component, including scheduling repairs within working hours and handling repair costs;

- complete_repair: Completes the repair of a component and updates its predicted remaining life.

The simulation of machine operation and the handling of component failures and repairs are conducted as described below.

Algorithm 5: Machine Operation and Repair.

```
Data: Environment env, Machine machine
Result: Simulated operation and repair events
Initialize: Set initial component statuses and predicted times;
while Simulation time not finished do
    foreach component in machine do
        if component is not broken and not under repair then
            Decrement remaining time;
            if remaining time ≤ 0 then
                Log failure event;
                Set component as broken;
                Start repair process;
    Check remaining life at the end of each day;
```

In this simulation, a cost of 40€ per component repair was also added, and 40€ was referred to as the displacement of the maintenance specialists. It was also added some restrictions, such as the repayment only does the maintenance of the components from Monday to Friday, from 08:00 to 17:00. Also, the first component takes four hours to be repaired, and the subsequent on that day takes only one hour.

For the simulation, the following algorithm was performed.

Algorithm 6: Simulation Setup and Parameter Initialization.

Data: Selected date and time *start_date*, Simulation weeks *simulation_weeks*

Result: Simulation parameters initialized

if *Start button clicked* **then**

  if *Start date and time valid* **then**

    Set simulation parameters:

    - REPAIR_TIME $\leftarrow$ 4 hours
    - SIM_TIME $\leftarrow$ *simulation_weeks* $\times 7 \times 24$ hours
    - START_DATE $\leftarrow$ *start_date*
    - REPAIR_COST $\leftarrow$ 40
    - DISPLACEMENT_COST $\leftarrow$ 40

  else

    Display error message: "Please select a start date and time between the allowed range."

In the simulation, the following algorithm can predict the time to failure based on the selected model chosen by the person and machine features.

Algorithm 7: Predict Time to Failure for a Component.

Data: Machine features *machine_features*, Model choice *model_choice*, Mean Time to Failure *mttf*, Component *component*

Result: Predicted time to failure *predicted_time*

Initialize: Load the appropriate model for the *component*;

Switch on *model_choice*;

- **Case** 'coxph' or 'rsf' or 'gbsa':
  - Predict survival function *survival_function* $\leftarrow$ model.predict_survival_function(*features*);
  - Calculate *median_failure_time* $\leftarrow \arg\max(survival\_function[0].y < 0.5)$;
- **Case** 'svm':
  - Calculate risk scores *risk_scores* $\leftarrow$ model.predict(*features*);
  - Convert risk scores to time *predicted_time* $\leftarrow$ convert_risk_to_time(*risk_scores*, *training_data*, *component*);
  - Take median *median_failure_time* $\leftarrow$ median(*predicted_time*);
- **Default:** *median_failure_time* $\leftarrow$ *mttf*;

Return *median_failure_time* as *predicted_time*;

The survival function is directly utilized in the algorithm to estimate the median time to failure. The median time to failure is defined as the time at which the survival probability drops below 50%. By monitoring the survival function over time, we identify the first instance when the probability of survival becomes less than or equal to 50%. This specific time point is used to estimate the median time to failure [35].

After the simulation, a dashboard was created with the support of Streamlit. This Dashboard first opens a webpage where the people can select a date and time between 01/01/2015 06:00 and 01/01/2016 06:00. The person can also choose the time of simulation he wants to produce. This time is referred to as the weeks. After the person presses the simulation button, a report with the total cost associated with the repairs and displacement is demonstrated, as well as the start and end date of the simulation.

Algorithm 8: Logging and Analysis.

Data: Simulation data

Result: Logs and analyses machine performance

foreach *failure or repair event* **do**

  Log event details;

  Update component status and remaining time;

Analyse and report:

- Total cost
- Number of failures per component
- Remaining life data

Output results and save logs;

It then shows a table with the repair log of each component, the time it started being repaired and the model that was used to predict the failure. It ends with showing graphs, one with the remaining time of each component during the weeks simulated and another with the failures of each component.

This module provides a robust framework for simulating machine maintenance, offering insights into the reliability and cost-effectiveness of various maintenance strategies. The use of multiple predictive models allows for a detailed analysis of each component's performance, aiding in informed decision-making for maintenance planning.

4.5 Conclusions

In this chapter, a data-driven predictive maintenance system was implemented by using a modular approach. It detailed the data acquisition and preprocessing steps from Gorenje and Microsoft Azure Predictive Maintenance datasets, focusing on extracting relevant features for analysis. Different survival analysis models were trained to predict time-to-failure for machine components, including Cox Proportional Hazards, Random Survival Forest, Gradient Boosting Survival Analysis, and Survival Support Vector Machines.

The Maintenance Module was then introduced, which simulates machine operations and manages repairs based on predictive models. A Streamlit dashboard was developed to provide an interactive interface for simulation setup and results visualisation, including repair logs and cost analysis. This chapter demonstrates the comprehensive implementation of a system that aids in informed maintenance decision-making, ensuring the reliability and cost-effectiveness of machinery.

For the development of the implementation code, the open-source scikit-learn package used version 1.3.2 [36], the scikit-survival package used version 0.22.2 [37], the scipy package version used version 1.10.1 [38], and the Python version 3.9.0 [39].[1]

[1] The resulting code from this dissertation can be found on the following link https://github.com/margaridarfaiscam/Dissertation/tree/master.

Chapter 5

Experiments and results

The results can be divided between the performance of each model for each dataset and the dashboard created with the help of the Streamlit application. In the end, based on this report, some conclusions related to the combination of maintenance and the actions that can be performed can be concluded.

5.1 Predictive Models Performance

The first result to be obtained was the curve of the survival function using the Kaplan-Meier estimator for the Gorenge dataset. As seen in section 2.1.4, the Kaplan-Meier estimator estimates the survival function from lifetime data. It provides the probability of surviving beyond a certain point in time, or in this case, beyond a certain number of cycles. The vertical axis represents the estimated probability of survival $S(t)$, which ranges from 0 to 1, and the horizontal axis represents the cycles c. The graphics are shown below:

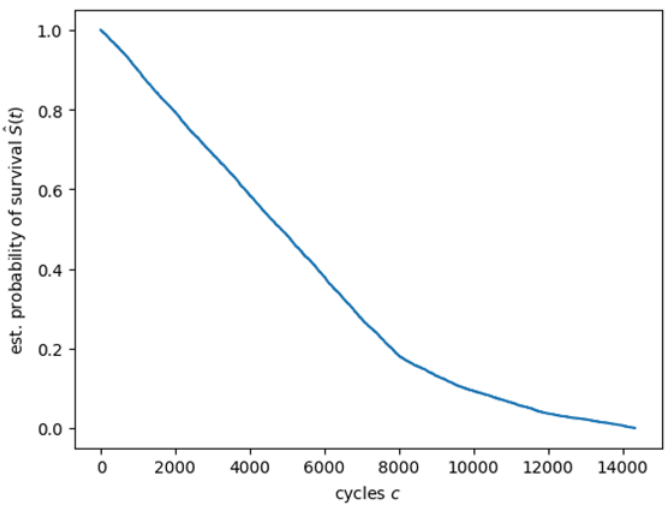


Figure 5.1: Predicted survival function using Kaplan-Meier estimator for Gorenge dataset.

In this first graph, the curve demonstrates a consistent decrease in survival probability as the number of cycles increases. The curve's slope is smooth, indicating gradual component degradation or system measurement. The survival probability decreases to nearly 0 around 14,000 cycles, indicating that the component has failed by this point.

After analysing the Kaplan-Meier estimator curve, the predicted survival functions using different models for the Gorenje dataset were calculated below.

The evaluation of the different models trained using C-index performance measures on Gorenje's dataset can be observed in the table 5.1.

Table 5.1: Evaluation of the different models trained using C-index performance measure for the Gorenje's dataset.

	CoxPH		RSF		GBSA		FS-SVM	
	C-index train	C-index test	C-index train	C-index test	C-index train	C-index test	C-index train	C-index test
Spot Welding Comp	0.763	0.974	0.897	0.791	0.789	0.789	0.722	0.865

The following curves can be found in figure 5.2, where the top-left graphic represents the Cox model, the top-right the RSF model, the bottom-left the GBSA and the bottom-right the FS-SVM.

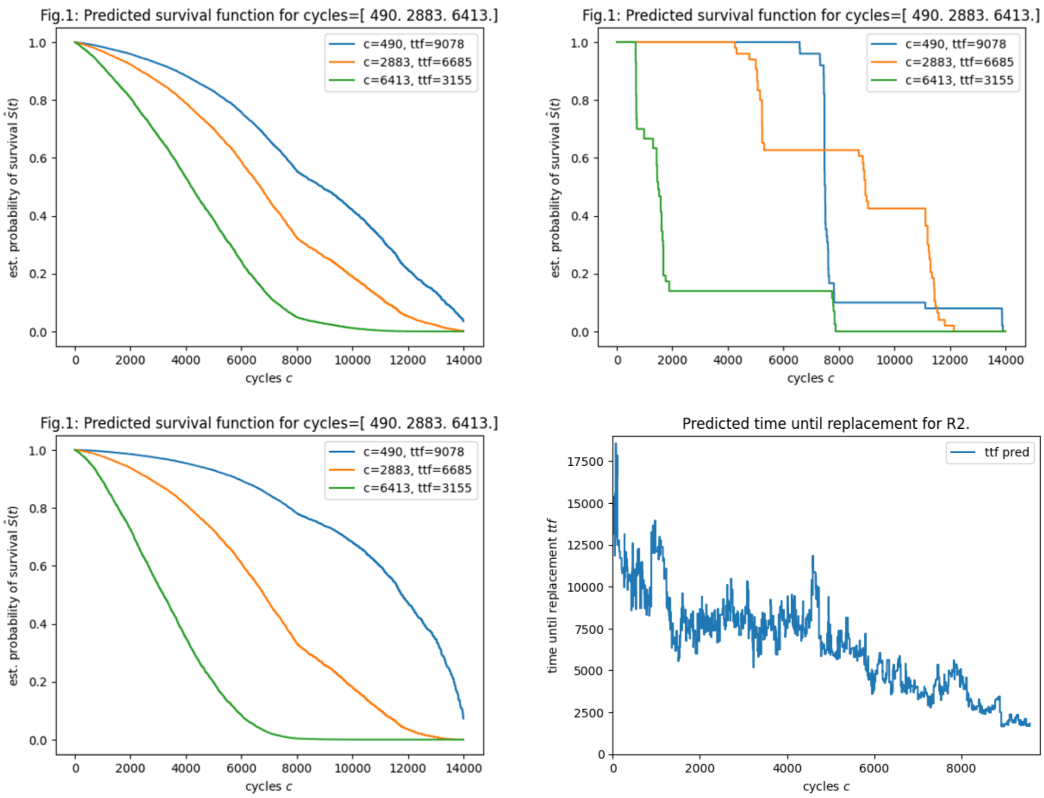


Figure 5.2: Predicted survival functions using different models for the Gorenje Dataset.

The first graphic presents the predicted survival function using the Cox Model for the Gorenje Dataset. It can be observed that it presents a proper performance on the training data (0.763), but it performs exceptionally well on the test data (0.974). This unusually high test c-index compared to the training c-index might indicate potential overfitting. Generally, the test performance is expected to be slightly lower or comparable to the training performance.

RSF shows high performance on the training data (0.897), suggesting it has learned it well. However, its performance drops on the test data (0.791), maintaining a good level of predictive accuracy on the test data. The sharp drops in the survival curves indicate that certain failures occur suddenly at specific cycles.

The GBSA model consistently performs on training and test datasets with identical c-index values (0.789). This consistency suggests that the model generalises well and has neither overfitted nor underfitted the data.

FS-SVM presents lower performance measures on the training data (0.722) but conducts significantly better on the test data (0.865). This improvement could imply that the model is better suited to the distribution of the test data. It directly predicts the time until failure (TTF) for the component.

In the first three graphics, the models calculate the survival functions because they are specifically designed to handle survival data. The FS-SVM mainly aims to predict the survival times instead of directly estimating a survival probability function. As a result, FS-SVM is commonly utilized for predicting a continuous outcome such as time to failure (TTF) rather than a survival curve [34].

When testing the results on the Microsoft Azure Predictive Maintenance dataset, the following predicted survival functions using the Kaplan-Meier estimator were obtained, where the top-left graphic represents Component 1, the top-right represents Component 2, the bottom-left represents Component 3, and the bottom-right represents Component 4.

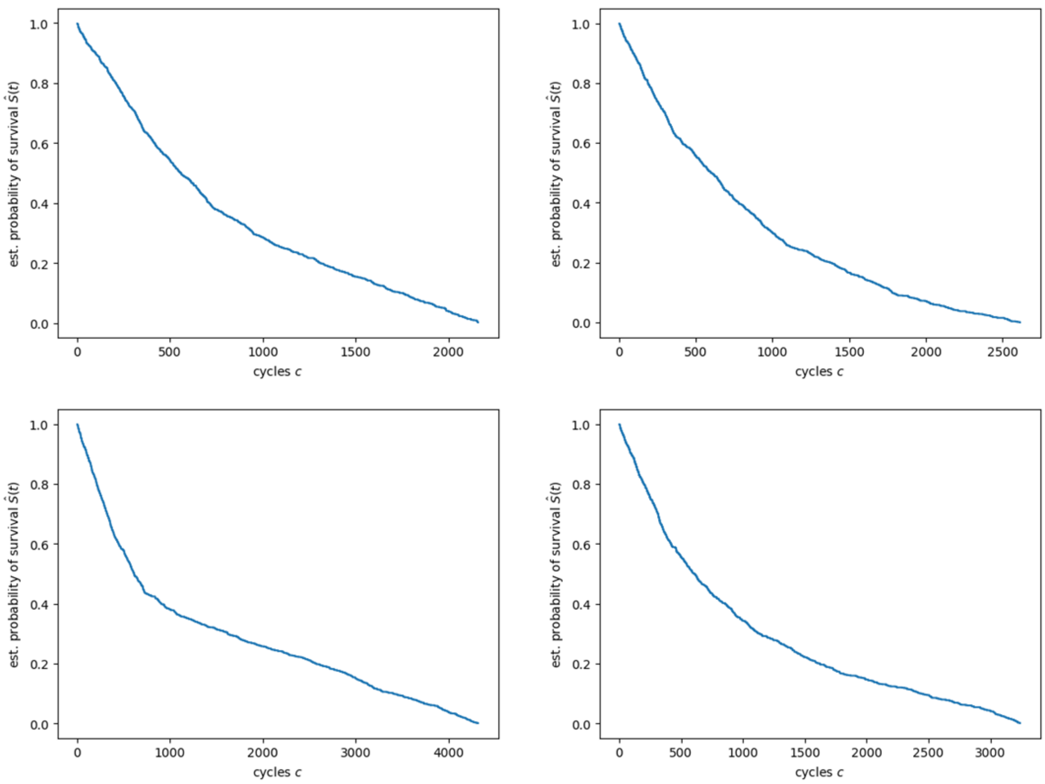


Figure 5.3: Predicted survival function using Kaplan-Meier estimator for Microsoft Azure Predictive Maintenance for the four components.

The Kaplan-Meier estimator curves for the four components in Figure 5.3 reveal different patterns in the survival probabilities over time, indicating differences in the expected lifespans of each component. Component 1 displays a steady decline in survival probability, reaching near zero by around 2,000 cycles.

The survival probability for Component 2 approaches zero just beyond 2,500 cycles, indicating a marginally longer lifespan than Component 1.

In contrast, Components 3 and 4 demonstrate a more gradual decline in survival probability, extending to around 4,000 cycles for Component 3 and 3,500 cycles for Component 4. This suggests that these components have a longer lifespan and a lower failure rate over time than Components 1 and 2. Compared to the first dataset, it can be shown that all of the componets present a smaller lifespan.

After analysing the different Survival Curves using the Kaplan-Meier estimator, the predicted survival models were tested for the different components, and the performance was evaluated for each component of machine 1 using the c-index performance measure, present in table 5.2.

Table 5.2: Evaluation of the different models trained using C-index performance measure for the Microsoft Azure Predictive Maintenance dataset.

	CoxPH		RSF		GBSA		FS-SVM	
	C-index train	C-index test	C-index train	C-index test	C-index train	C-index test	C-index train	C-index test
Comp1	0.531	0.926	0.709	0.689	0.627	0.645	0.510	0.553
Comp2	0.518	0.858	0.702	0.387	0.627	0.509	0.516	0.544
Comp3	0.480	0.649	0.751	0.458	0.722	0.543	0.511	0.496
Comp4	0.518	0.792	0.687	0.601	0.602	0.579	0.507	0.514

The curves of the model performance can be found below, with figure 5.4, 5.5 and 5.6 presenting the different survival functions for the four different components for the CoxPH model, the RSF model and GBSA model respectively and the figure 5.7 the predicted time until replacement for each component. In all figures, the top-left graphic represents component 1, the top-right the Component 2, the bottom-left the Component 3 and the bottom-right the Component 4.

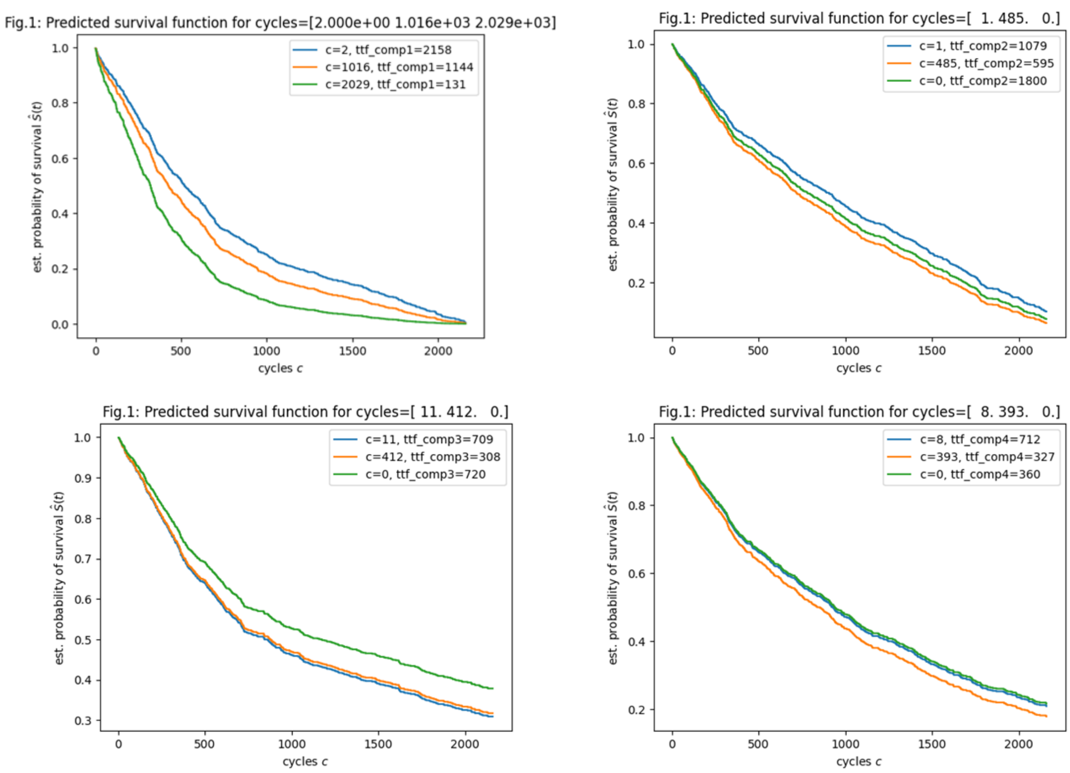


Figure 5.4: Predicted survival functions for the four components using the CoxPh model on Microsoft Azure Predictive Maintenance dataset.

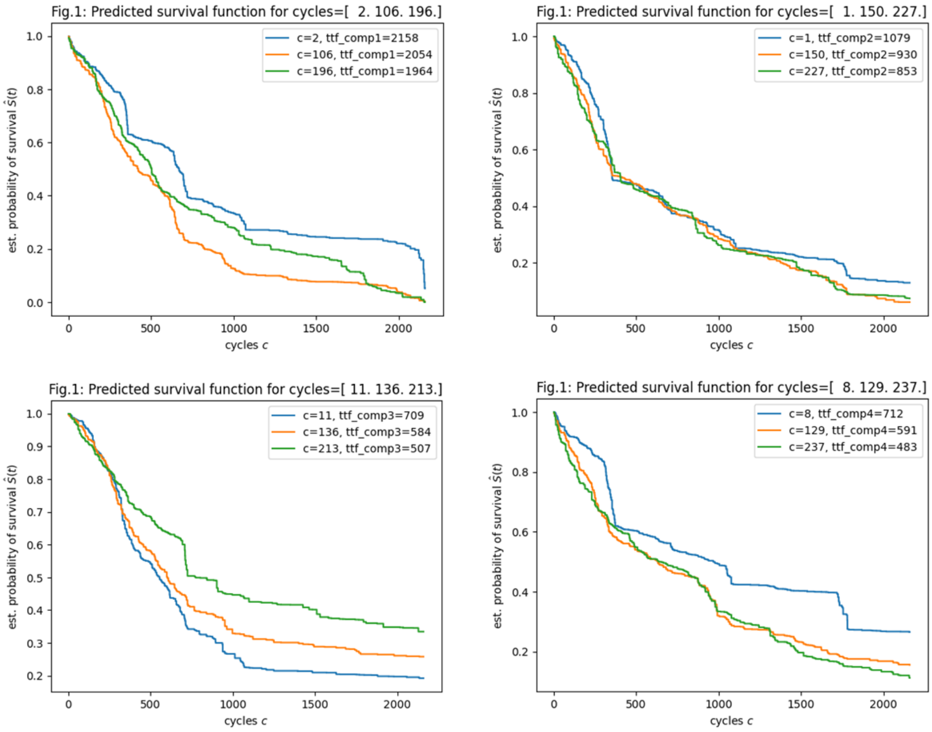


Figure 5.5: Predicted survival functions for the four components using the RSF model on Microsoft Azure Predictive Maintenance dataset.

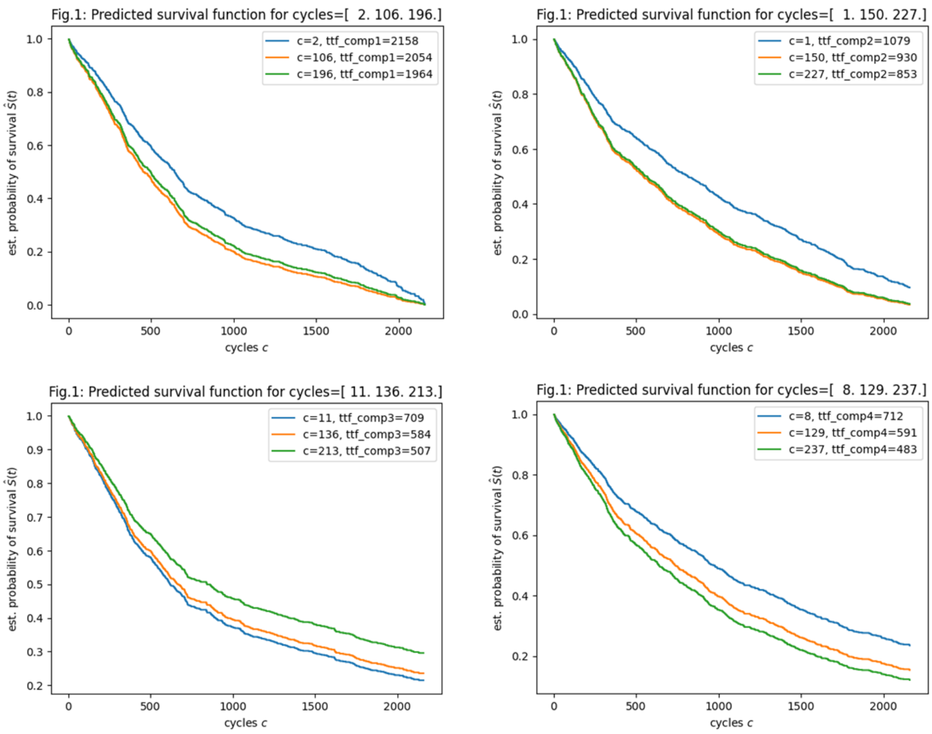


Figure 5.6: Predicted survival functions for the four components using the GBSA model on Microsoft Azure Predictive Maintenance dataset.

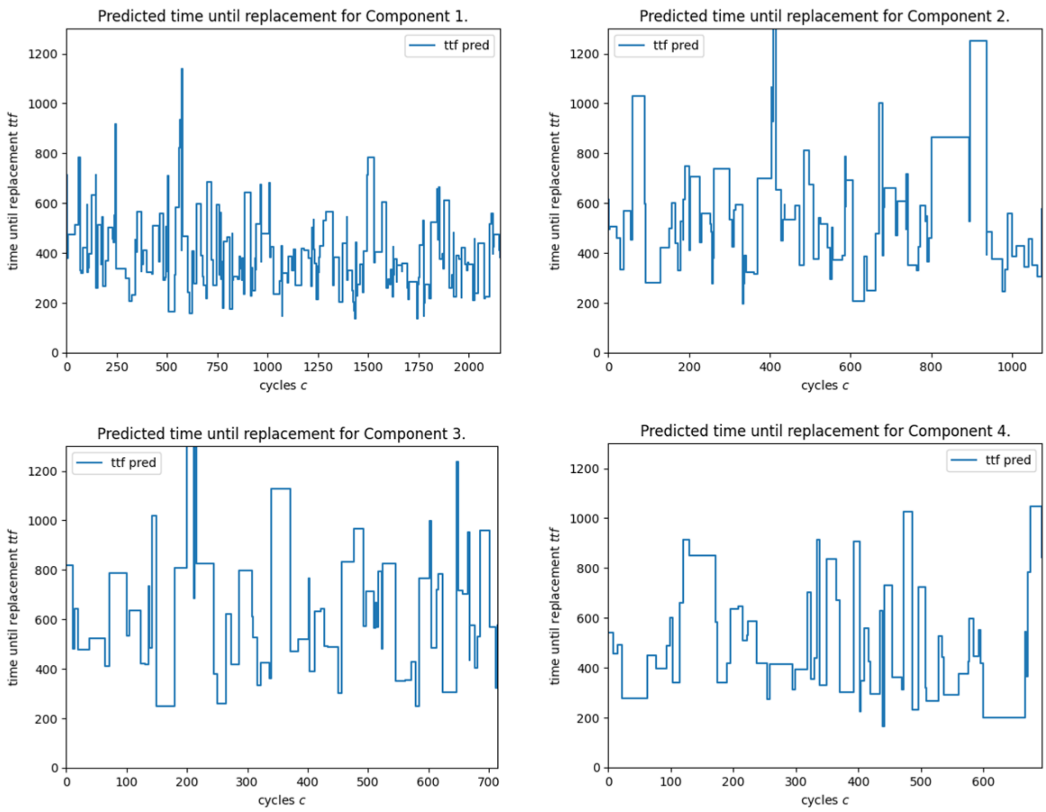


Figure 5.7: Predicted survival functions for the four components using the FS-SVM model on Microsoft Azure Predictive Maintenance dataset.

As we can see based on the results, we can confirm that the CoxPH Model presents a pattern of overfitting across all components, with test c-index values significantly higher than training c-index values.

On the other hand, the RSF model represents a good generalisation on components one and four, but components two and three present a significant drop between the train and test c-index, which may suggest an overfitting [40].

The GBSA model presents a consistent performance for components one, two and four and component three shows a moderate drop in test performance, suggesting some overfitting.

In the end, FS-SVM does present a consistent performance for components three and four, although the results are not that high. The time-to-failure predictions presented in figure 5.7 are more fluctuating and less stable compared to since they focus on the TTF prediction instead of the survival analysis.

5.2 Maintenance Module

5.2.1 Simulation and Dashboard report

When simulating, the models for each component were chosen based on the results of the c-index values. The models chosen were:

- model_choice_comp1 = 'rsf'
- model_choice_comp2 = 'gbsa'
- model_choice_comp3 = 'svm'
- model_choice_comp4 = 'svm'

These models could be changed in the simulation script. When the script was run, a CSV document was generated with information regarding when the components failed, when less than 55 hours were remaining, when the components started to be repaired, and when it was finished.

We could obtain the following UI when the simulation was accessed, present in figure 5.8.

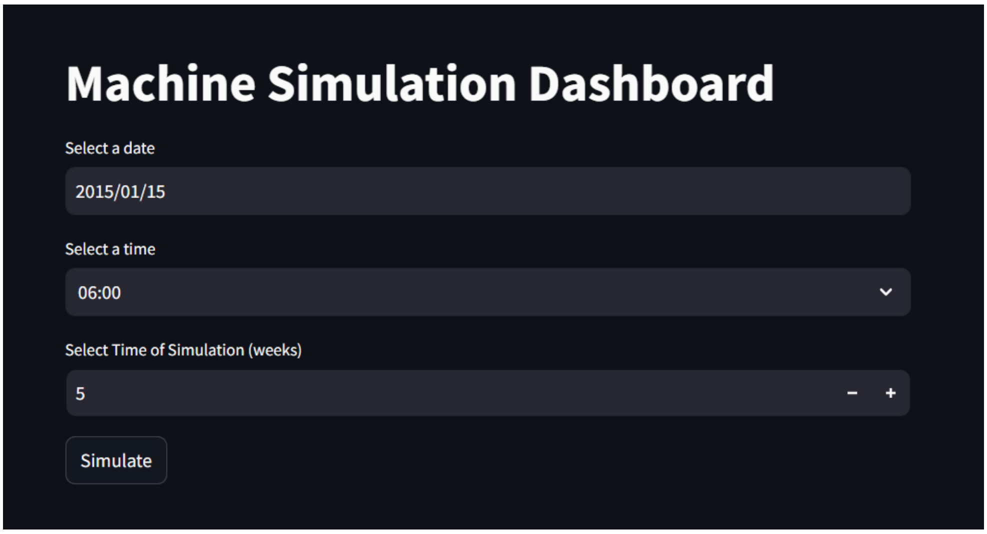
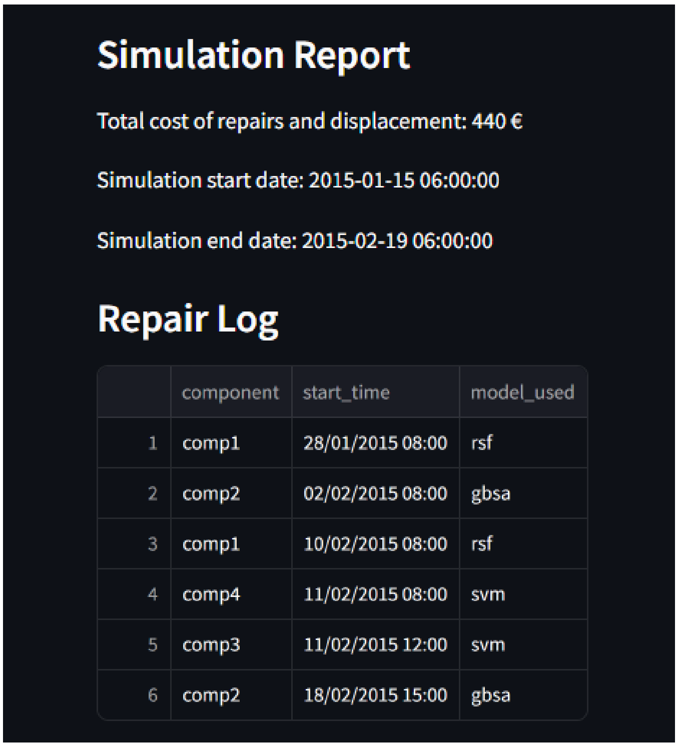


Figure 5.8: Simulation UI.

With this interface, the person can choose the date and the weeks they want to simulate, according to the rules mentioned in section 4.4. For the following results, the data chosen was 2015/01/05 06:00, and the time for simulation was five weeks. The CSV file related to this simulation can be found on A. The first ones to acknowledge are the simulation report and the repair log, present in the figure 5.9.



Simulation Report

Total cost of repairs and displacement: 440 €

Simulation start date: 2015-01-15 06:00:00

Simulation end date: 2015-02-19 06:00:00

Repair Log

	component	start_time	model_used
1	comp1	28/01/2015 08:00	rsf
2	comp2	02/02/2015 08:00	gbsa
3	comp1	10/02/2015 08:00	rsf
4	comp4	11/02/2015 08:00	svm
5	comp3	11/02/2015 12:00	svm
6	comp2	18/02/2015 15:00	gbsa

Figure 5.9: Simulation Report and Repair Log for the simulated dates.

After the report, the following graphics were presented, which were related to the remaining time of each component and the times that a component failed during the simulated time.

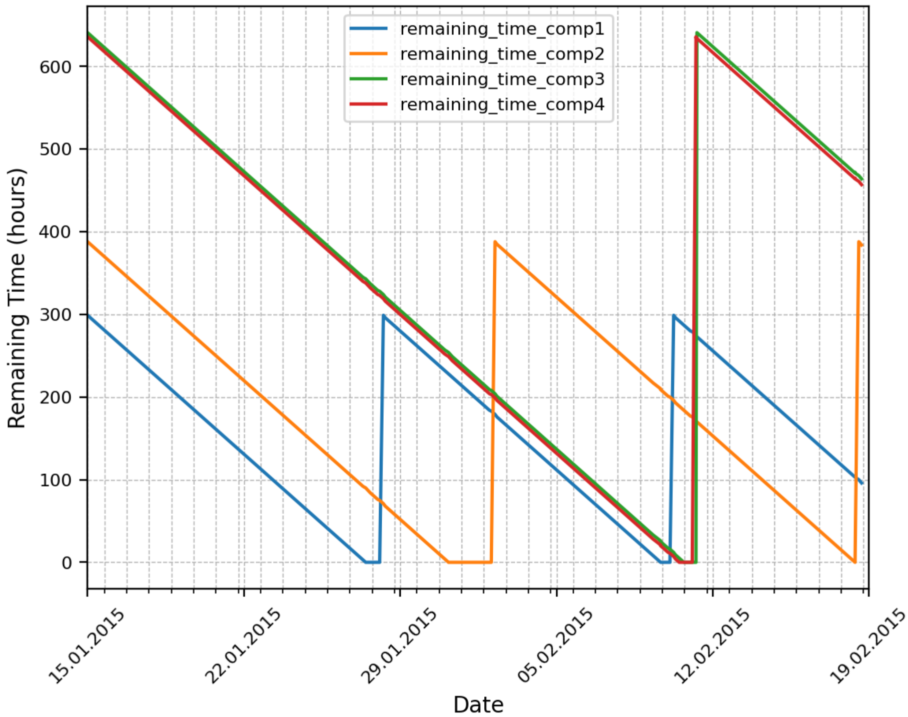


Figure 5.10: Remaining time for each component.

In figure 5.10, the predictions were used to calculate the RUL in hours at the beginning of the simulation, and then the hours were decremented linearly.

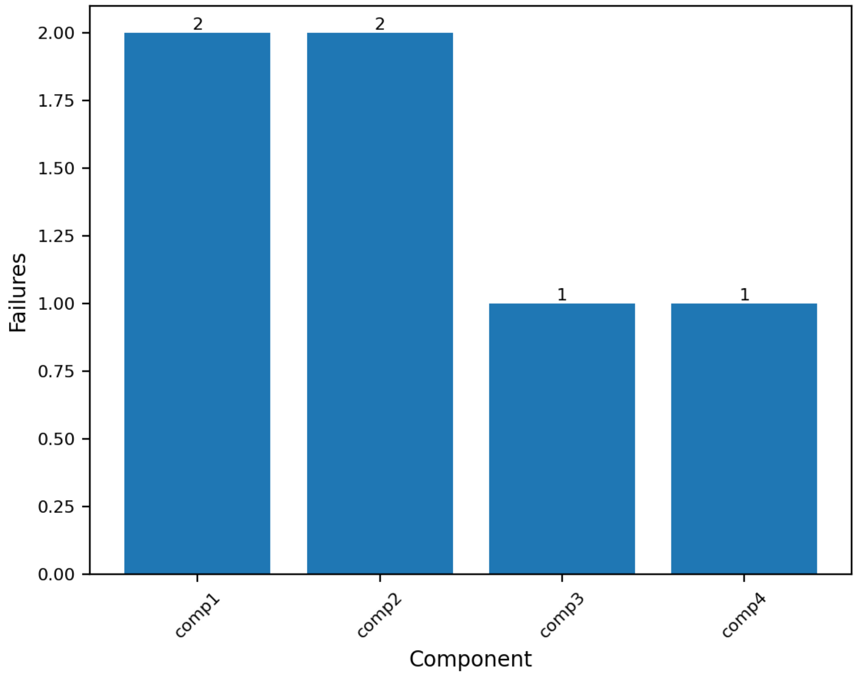


Figure 5.11: Failures of each component during the simulated time.

In order to obtain a better and quicker visualization of the components that failed and how many times, the graphic presented in figure 5.11 was created.

5.2.2 Maintenance Actions

Regarding the simulation report in figure 5.9, we can see that six maintenance was performed and 440€ was spent. However, based on this information and the graphic presented in figure 5.10, we can see that we can still optimise the process to combine even more maintenance. The information on A can help decide when to perform the maintenance.

As we may see, the component's prediction to fail occurs on days:

- Component 1 - 27/01/2015 17:00
- Component 2 - 31/01/2015 10:00
- Component 1 - 09/02/2015 22:00
- Component 4 - 10/02/2015 18:00
- Component 3 - 10/02/2015 23:00
- Component 2 - 18/02/2015 15:00

Table 5.3: January 2015.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Table 5.4: February 2015.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Component 2 presents less than 75 hours to fail when the first component fails. As we know, the repairman only works from 8h to 17h. Therefore, components 1 and 2 can be repaired on day 27 at 08:00 so the machine can be stopped by the 28th.

With this will have a total cost of 120€

After this, we can see that the following components will fail on the 09th and 10th February. This way, we can combine the three repairs for the element on the 09th of February. This way we will have a total cost of 160€.

Combining the repairs on the 09th of February was based on minimising machine downtime and efficiently utilising the available repair window. By aligning the repairs with the availability of maintenance specialists, we ensure that all necessary maintenance is completed within a single shift, preventing additional disruption to production.

The last component, since it had more than 200 hours of remaining life when the other components were replaced, can be waited till the 18th of February, as predicted repair, adding a cost of 80€. However, we can schedule the repair for 8 in the morning.

This way, the components will be replaced on the following days and hours:

- Component 1 - 27/01/2015 08:00
- Component 2 - 27/01/2015 12:00
- Component 1 - 09/02/2015 08:00
- Component 4 - 09/02/2015 12:00
- Component 3 - 09/02/2015 13:00
- Component 2 - 18/02/2015 08:00
- **Total Cost: 360€**

5.3 Conclusions

Analysing the predictive models' performance across different datasets highlights critical insights. The CoxPH model, despite its strong performance on test data, exhibited signs of overfitting, necessitating further validation and regularisation techniques. The RSF model demonstrated good generalisation in some components but struggled with overfitting in others, indicating the need for potential adjustments or alternative strategies for specific data subsets. GBSA consistently delivered reliable results across most components, confirming its robustness and generalizability. Finally, FS-SVM showed varied performance, excelling in some test scenarios while underperforming in others, suggesting it might benefit from further refinement or hybrid approaches.

In the simulation phase, the models selected for each component, based on their c-index values, provided a solid foundation for generating accurate predictions and facilitating the creation of a user-friendly interface for real-time decision-making. This comprehensive approach validated the chosen models and highlighted areas for further enhancement. The analysis demonstrates that predictive maintenance scheduling based on model predictions can significantly reduce costs and prevent unexpected downtime.

Chapter 6

Conclusions and future work

This chapter concludes the analysis of the application of survival analysis methods for predictive maintenance in the context of Industry 4.0. By using advanced predictive models, this dissertation explored the potential of accurately predicting component failures, optimizing maintenance schedules, and reducing costs in industrial environments. This chapter will summarize the key findings, discuss the contributions and limitations of the research, and outline potential future work directions to enhance the effectiveness and applicability of predictive maintenance strategies.

6.1 Overview

The dissertation investigates applying advanced survival analysis techniques for predictive maintenance within the context of Industry 4.0. Effective predictive maintenance strategies become essential as industries increasingly adopt digital technologies and data-driven approaches. Predictive maintenance involves predicting equipment failures before they occur, allowing for timely interventions that minimize downtime and reduce maintenance costs.

As mentioned at the beginning of this dissertation, it can contribute to the United Nations' SDGs by optimizing resources and reducing waste in industrial processes.

This study specifically focused on evaluating the effectiveness of several survival analysis models, including the Cox Proportional Hazards (CoxPH) model, Random Survival Forests (RSF), Gradient Boosting Survival Analysis (GBSA), and Survival Support Vector Machines (FS-SVM). These models were applied to Gorenje and Microsoft Azure Predictive Maintenance datasets, representing different industrial settings and equipment types.

Despite the promising results, several limitations were identified, including data constraints and the need for more extensive validation across diverse industrial environments. The dissertation concludes with recommendations for future work, emphasizing the integration of real-time data, enhanced model refinement, and scalability testing further to solidify the role of predictive maintenance in Industry 4.0.

6.2 Contributions

This dissertation explored the application of survival analysis models for predictive maintenance in industrial settings. Previous studies in this area were mainly related to the medical field. The study demonstrated the effectiveness of models such as the Cox Proportional Hazards model, Random Survival Forests (RSF), Gradient Boosting Survival Analysis (GBSA), and Survival Support Vector Machines (FS-SVM) in predicting maintenance needs. These models were validated across different datasets, including those from Gorenje and the Microsoft Azure Predictive Maintenance dataset. This work provides a valuable reference for future research, highlighting the adaptability of these models to diverse industrial scenarios.

A thorough comparison of multiple survival analysis models was conducted, offering insights into the strengths and limitations of each model. This benchmarking against established datasets and methodologies identified the most suitable models for specific applications, guiding future research and development. The study underscored the importance of using diverse datasets to test model performance, ensuring generalizability and reliability.

Another contribution that can be added to the development of this dissertation is the development of a dashboard interface using Streamlit, which made it much easier for predictive maintenance. This tool allowed us to see the Remaining Useful Life (RUL) estimates and simulate different maintenance scenarios. By providing a user-friendly interface, the tool helps maintenance planners schedule more efficiently and take proactive measures to prevent predicted failures, ultimately improving operational efficiency. This will allow the technicians in the factories to get a better visualization of the processes.

Another practical contribution, more significant to the industrial area, was implementing predictive maintenance models that significantly reduce downtime and maintenance costs. Potential savings in operational costs and improved efficiency were demonstrated by accurately predicting when maintenance should be performed. The ability to schedule maintenance proactively, based on robust model predictions, helps avoid unplanned downtimes and extend the lifespan of critical industrial components. Furthermore, the tool developed in this study can be adapted for various industries, offering a scalable and flexible solution for predictive maintenance, as it was explained in section 1.4 as one of the goals of this dissertation.

Finally, it is important to highlight that derived from the development of this dissertation, a paper with the title "Data-Driven Predictive Maintenance for Component Life-Cycle Extension" was written and submitted to the ICINCO 2024 conference. The title and abstract of the article can be found in the appendix B for consultation.

6.3 Limitations

During the realization of this work, some limitations could be addressed, such as:

- Data Limitations: For example, the Gorenje dataset focused on a specific set of equipment, limiting the generalizability of the findings. Additional datasets with diverse types of machinery and failure modes are needed to validate the broader applicability of the models.

- Model Limitations: Some models, particularly the CoxPH and RSF, exhibited signs of overfitting, suggesting that they may not generalize well to unseen data. This limitation highlights the need for further validation using different datasets.

- Validation Limitations: While useful, the Kaggle dataset (Microsoft Azure Predictive Maintenance dataset) used for benchmarking may not fully capture the complexities of real-world industrial environments. Operational conditions, maintenance practices, and data quality vary widely and can affect model performance.

- Advanced Model Refinement: At the end of the realization of this work, it was concluded that some could have been done further refinement of models, particularly those showing signs of overfitting or underperformance. This could involve hybrid approaches integrating additional data sources to improve prediction accuracy. Using other performance measures, such as the Brier Score mentioned in section 2.1.5, can help to compare the results obtained in the future. This will provide a more comprehensive assessment of the models' predictive reliability and accuracy. Due to time limitations, this measure couldn't be applied.

Integrating these elements into the discussion on limitations provides a thorough analysis of the constraints faced during the study. It sets the stage for future research directions, demonstrated in section 6.4.

6.4 Future Work

Despite the promising results, there are several areas for future work to enhance the application of predictive maintenance, which can be highlighted:

- Real-Time Data Integration: Implementing models that can handle real-time data streams will improve the timeliness and accuracy of maintenance predictions. This integration will also allow for dynamic updating of RUL estimates and more responsive maintenance planning.

- Enhanced User Interface: Developing a more complex and detailed user interface could provide additional insights and functionality, such as more control over simulation parameters and detailed visualizations of component health trends over time.

- Scalability Testing: Testing the models and simulation framework in larger, more diverse industrial settings will ensure the scalability and robustness of the solutions proposed. This testing will ensure that the solutions can handle different types and volumes of data effectively.

- Non-linear degradation simulation: Introduce a more realistic degradation model using the survival function to simulate non-linear RUL decaying. This approach would multiply the survival function probabilities by the predicted RUL, allowing for a non-linear decrease over time and reflecting a more realistic component degradation.

- Optimization function for cost-efficiency tuning: In this dissertation, the prediction of component failures and necessary repairs were addressed, but the optimization of maintenance schedules to minimize factory costs was performed manually. Developing an automated function that optimizes this process based on the degradation patterns of the components, thereby reducing costs more efficiently, would contribute significantly to the optimization of the process.

To conclude, this dissertation underscores the potential of survival analysis methods for predictive maintenance within Industry 4.0 environments. Industries can achieve more efficient, cost-effective, and reliable maintenance strategies by adopting and integrating these advanced models into a user-friendly interface. The findings offer a foundation for future research, emphasizing the need for real-world validation, advanced model refinement, and broader applications. Continued development in this field promises to enhance industrial maintenance practices, leading to greater operational efficiency and reduced costs.

Appendix A

Appendix A

A.1 CSV Results related to the simulation from 15/01/2015 till 19/02/2015.

datetime	remaining_	remaining_	remaining_	remaining_	event	cost
15/01/2015 06:00	299	388	641	636		
15/01/2015 14:00	291	380	633	628		
15/01/2015 22:00	283	372	625	620		
16/01/2015 06:00	275	364	617	612		
16/01/2015 14:00	267	356	609	604		
16/01/2015 22:00	259	348	601	596		
17/01/2015 06:00	251	340	593	588		
17/01/2015 14:00	243	332	585	580		
17/01/2015 22:00	235	324	577	572		
18/01/2015 06:00	227	316	569	564		
18/01/2015 14:00	219	308	561	556		
18/01/2015 22:00	211	300	553	548		
19/01/2015 06:00	203	292	545	540		
19/01/2015 14:00	195	284	537	532		
19/01/2015 22:00	187	276	529	524		
20/01/2015 06:00	179	268	521	516		
20/01/2015 14:00	171	260	513	508		
20/01/2015 22:00	163	252	505	500		
21/01/2015 06:00	155	244	497	492		
21/01/2015 14:00	147	236	489	484		
21/01/2015 22:00	139	228	481	476		
22/01/2015 06:00	131	220	473	468		
22/01/2015 14:00	123	212	465	460		
22/01/2015 22:00	115	204	457	452		
23/01/2015 06:00	107	196	449	444		
23/01/2015 14:00	99	188	441	436		
23/01/2015 22:00	91	180	433	428		
24/01/2015 06:00	83	172	425	420		
24/01/2015 14:00	75	164	417	412		
24/01/2015 22:00	67	156	409	404		

Figure A.1: Data from 15/01/2015 till 24/01/2015.

25/01/2015 06:00	59	148	401	396		
25/01/2015 14:00	51	140	393	388		
25/01/2015 22:00	43	132	385	380		
26/01/2015 06:00	35	124	377	372	comp1 has less than 55 remaining life (35 hours)	0
26/01/2015 14:00	27	116	369	364		
26/01/2015 22:00	19	108	361	356		
27/01/2015 06:00	11	100	353	348	comp1 has less than 55 remaining life (11 hours)	0
27/01/2015 14:00	3	92	345	340		
27/01/2015 17:00	0	90	343	338	comp1 failed	0
27/01/2015 22:00	0	84	337	332		
28/01/2015 06:00	0	76	329	324	comp1 has less than 55 remaining life (0 hours)	0
28/01/2015 06:00	0	76	329	324		
28/01/2015 08:00	0	74	329	324	comp1 displacement cost added	40
28/01/2015 08:00	0	74	329	324	comp1 repair started	40
28/01/2015 12:00	299	70	325	320	comp1 repaired	0
28/01/2015 14:00	296	68	321	316		
28/01/2015 22:00	288	60	313	308		
29/01/2015 06:00	280	52	305	300	comp2 has less than 55 remaining life (52 hours)	0
29/01/2015 06:00	280	52	305	300		
29/01/2015 14:00	272	44	297	292		
29/01/2015 22:00	264	36	289	284		
30/01/2015 06:00	256	28	281	276	comp2 has less than 55 remaining life (28 hours)	0
30/01/2015 06:00	256	28	281	276		
30/01/2015 14:00	248	20	273	268		
30/01/2015 22:00	240	12	265	260		
31/01/2015 06:00	232	4	257	252	comp2 has less than 55 remaining life (4 hours)	0
31/01/2015 06:00	232	4	257	252		
31/01/2015 10:00	228	0	254	249	comp2 failed	0
31/01/2015 14:00	224	0	249	244		
31/01/2015 22:00	216	0	241	236		
01/02/2015 06:00	208	0	233	228	comp2 has less than 55 remaining life (0 hours)	0

Figure A.2: Data from 25/01/2015 till 01/02/2015.

01/02/2015 06:00	208	0	233	228		
01/02/2015 14:00	200	0	225	220		
01/02/2015 22:00	192	0	217	212		
02/02/2015 06:00	184	0	209	204	comp2 has less than 55 remaining life (0 hours)	0
02/02/2015 06:00	184	0	209	204		
02/02/2015 08:00	182	0	207	203	comp2 displacement cost added	40
02/02/2015 08:00	182	0	207	203	comp2 repair started	40
02/02/2015 12:00	178	387	203	199	comp2 repaired	0
02/02/2015 14:00	176	385	201	196		
02/02/2015 22:00	168	377	193	188		
03/02/2015 06:00	160	369	185	180		
03/02/2015 14:00	152	361	177	172		
03/02/2015 22:00	144	353	169	164		
04/02/2015 06:00	136	345	161	156		
04/02/2015 14:00	128	337	153	148		
04/02/2015 22:00	120	329	145	140		
05/02/2015 06:00	112	321	137	132		
05/02/2015 14:00	104	313	129	124		
05/02/2015 22:00	96	305	121	116		
06/02/2015 06:00	88	297	113	108		
06/02/2015 14:00	80	289	105	100		
06/02/2015 22:00	72	281	97	92		
07/02/2015 06:00	64	273	89	84		
07/02/2015 14:00	56	265	81	76		
07/02/2015 22:00	48	257	73	68		
08/02/2015 06:00	40	249	65	60	comp1 has less than 55 remaining life (40 hours)	0
08/02/2015 06:00	40	249	65	60		
08/02/2015 14:00	32	241	57	52		
08/02/2015 22:00	24	233	49	44		
09/02/2015 06:00	16	225	41	36	comp1 has less than 55 remaining life (16 hours)	0
09/02/2015 06:00	16	225	41	36	comp3 has less than 55 remaining life (41.0 hours)	0

Figure A.3: Data from 01/02/2015 till 09/02/2015.

09/02/2015 06:00	16	225	41	36	comp4 has less than 55 remaining life (36.0 hours)	0
09/02/2015 06:00	16	225	41	36		
09/02/2015 14:00	8	217	33	28		
09/02/2015 22:00	0	209	25	20	comp1 failed	0
09/02/2015 22:00	0	209	25	20		
10/02/2015 06:00	0	201	17	12	comp1 has less than 55 remaining life (0 hours)	0
10/02/2015 06:00	0	201	17	12	comp3 has less than 55 remaining life (17.0 hours)	0
10/02/2015 06:00	0	201	17	12	comp4 has less than 55 remaining life (12.0 hours)	0
10/02/2015 06:00	0	201	17	12		
10/02/2015 08:00	0	199	15	10	comp1 displacement cost added	40
10/02/2015 08:00	0	199	15	10	comp1 repair started	40
10/02/2015 12:00	298	195	11	6	comp1 repaired	0
10/02/2015 14:00	296	193	9	4		
10/02/2015 18:00	292	189	5	0	comp4 failed	0
10/02/2015 22:00	288	185	1	0		
10/02/2015 23:00	287	184	0	0	comp3 failed	0
11/02/2015 06:00	280	177	0	0	comp3 has less than 55 remaining life (0 hours)	0
11/02/2015 06:00	280	177	0	0	comp4 has less than 55 remaining life (0 hours)	0
11/02/2015 06:00	280	177	0	0		
11/02/2015 08:00	278	175	0	0	comp4 displacement cost added	40
11/02/2015 08:00	278	175	0	0	comp4 repair started	40
11/02/2015 12:00	274	171	0	635	comp4 repaired	0
11/02/2015 12:00	274	171	0	635	comp3 repair started	40
11/02/2015 13:00	273	170	641	634	comp3 repaired	0
11/02/2015 14:00	272	169	640	633		
11/02/2015 22:00	264	161	632	625		
12/02/2015 06:00	256	153	624	617		
12/02/2015 14:00	248	145	616	609		
12/02/2015 22:00	240	137	608	601		
13/02/2015 06:00	232	129	600	593		
13/02/2015 14:00	224	121	592	585		

Figure A.4: Data from 09/02/2015 till 13/02/2015.

13/02/2015 22:00	216	113	584	577		
14/02/2015 06:00	208	105	576	569		
14/02/2015 14:00	200	97	568	561		
14/02/2015 22:00	192	89	560	553		
15/02/2015 06:00	184	81	552	545		
15/02/2015 14:00	176	73	544	537		
15/02/2015 22:00	168	65	536	529		
16/02/2015 06:00	160	57	528	521		
16/02/2015 14:00	152	49	520	513		
16/02/2015 22:00	144	41	512	505		
17/02/2015 06:00	136	33	504	497	comp2 has less than 55 remaining life (33 hours)	0
17/02/2015 06:00	136	33	504	497		
17/02/2015 14:00	128	25	496	489		
17/02/2015 22:00	120	17	488	481		
18/02/2015 06:00	112	9	480	473	comp2 has less than 55 remaining life (9 hours)	0
18/02/2015 14:00	104	1	472	465		
18/02/2015 15:00	103	0	471	464	comp2 failed	0
18/02/2015 15:00	103	0	471	464	comp2 displacement cost added	40
18/02/2015 15:00	103	0	471	464	comp2 repair started	40
18/02/2015 19:00	99	387	467	460	comp2 repaired	0
18/02/2015 22:00	96	384	464	457		

Figure A.5: Data from 13/02/2015 till 19/02/2015.

Appendix B

Appendix B

Data-Driven Predictive Maintenance for Component Life-Cycle Extension

Margarida Moreira[1][a], Eliseu Pereira[1,2][b] and Gil Gonçalves [1,2][c]
[1]*Faculty of Engineering, University of Porto, Portugal*
[2]*SYSTEC-ARISE, Faculty of Engineering, University of Porto, Portugal*
up201905217@edu.fe.up.pt, eliseu@fe.up.pt, gil@fe.up.pt

Keywords: Predictive Maintenance, Industry 4.0, Remaining Useful Life (RUL), Data-Driven Methods, Survival Analysis

Abstract: In the era of Industry 4.0, predictive maintenance is crucial for optimizing operational efficiency and reducing downtime. Traditional maintenance strategies often cost more and are less reliable, making advanced predictive models appealing. This paper assesses the effectiveness of different survival analysis models, such as Cox Proportional Hazards, Random Survival Forests (RSF), Gradient Boosting Survival Analysis (GBSA), and Survival Support Vector Machines (FS-SVM), in predicting equipment failures. The models were tested on datasets from Gorenje and Microsoft Azure, achieving C-index values on test data such as 0.792 on the Cox Model, 0.601 using RSF, 0.579 using the GBSA model and 0.514 when using the FS-SVM model. These results demonstrate the models' potential for accurate failure prediction, with FS-SVM showing significant improvement in test data compared to its training performance. This study provides a comprehensive evaluation of survival analysis methods in an industrial context and develops a user-friendly dashboard for real-time maintenance decision-making. The integration of survival analysis into Industry 4.0 frameworks can significantly enhance predictive maintenance strategies, paving the way for more efficient and reliable industrial operations.

Figure B.1: Title and abstract from the paper submitted to the ICINCO 2024 conference.

References

[1] R. Rossini, G. Prato, D. Conzon, C. Pastrone, E. Pereira, J. Reis, G. Goncalves, D. Henriques, A. R. Santiago, and A. Ferreira, “AI environment for predictive maintenance in a manufacturing scenario,” in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Vasteras, Sweden: IEEE, Sep. 2021, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/9613359/

[2] “Industry 4.0: A sneak peek into industrial iot in manufacturing,” https://www.aberdeen.com/blogposts/industry-4-0-industrial-iot-manufacturing-sneak-peek/, [Online; accessed January 2, 2024].

[3] McKinsey & Company, “What are industry 4.0, the fourth industrial revolution, and 4ir?” [Online]. Available: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-are-industry-4-0-the-fourth-industrial-revolution-and-4ir. [Accessed: Jan. 2, 2024].

[4] AuraQuantic, “Technologies for intelligent industry,” https://www.auraquantic.com/technologies-intelligent-industry/, [Online; accessed January 2, 2024].

[5] T. Zonta, C. da Costa, R. da Rosa Righi, M. de Lima, E. da Trindade, and G. Li, “Predictive maintenance in the Industry 4.0: A systematic literature review,” *Computers and Industrial Engineering*, vol. 150, 2020. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85092176538&doi=10.1016%2fj.cie.2020.106889&partnerID=40&md5=58758677c79a213ebc608e0a67715eb0

[6] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, “A survey of predictive maintenance: Systems, purposes and approaches,” [Online]. Available: http://arxiv.org/abs/1912.07383, [Accessed: Jan. 8, 2024].

[7] G. Moat and S. Coleman, “Survival analysis and predictive maintenance models for non-sensored assets in facilities management,” in *2021 IEEE International Conference on Big Data (Big Data)*. Orlando, FL, USA: IEEE, 2021, pp. 4026–4034, accessed: 2023-11-28. [Online]. Available: https://ieeexplore.ieee.org/document/9671625/

[8] S. B. Ramezani, L. Cummins, B. Killen, R. Carley, A. Amirlatifi, S. Rahimi, M. Seale, and L. Bian, “Scalability, Explainability and Performance of Data-Driven Algorithms in Predicting the Remaining Useful Life: A Comprehensive Review,” *IEEE Access*, vol. 11, pp. 41 741–41 769, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10103701/

[9] M. Achouch, M. Dimitrova, K. Ziane, S. Sattarpanah Karganroudi, R. Dhouib, H. Ibrahim, and M. Adda, “On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges,” *Applied Sciences*, vol. 12, no. 16, p. 8081, Aug. 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/16/8081

[10] S. Gupta, A. Kumar, and J. Maiti, "A critical review on system architecture, techniques, trends and challenges in intelligent predictive maintenance," *Safety Science*, vol. 177, p. 106590, Sep. 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925753524001802

[11] C. Ferreira and G. Gonçalves, "Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods," *Journal of Manufacturing Systems*, vol. 63, pp. 550–562, Apr. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0278612522000796

[12] Z. Yang, J. Kanniainen, T. Krogerus, and F. Emmert-Streib, "Prognostic modeling of predictive maintenance with survival analysis for mobile work equipment," *Scientific Reports*, vol. 12, no. 1, p. 8529, May 2022. [Online]. Available: https://www.nature.com/articles/s41598-022-12572-z

[13] F. D. Frumosu, G. Ø. Rønsch, and M. Kulahci, "Mould wear-out prediction in the plastic injection moulding industry: a case study," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 12, pp. 1245–1258, Dec. 2020. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/0951192X.2020.1829062

[14] S. Pölsterl, *scikit-survival: A Python package for time-to-event analysis*, 2024, accessed: 2024-07-18. [Online]. Available: https://scikit-survival.readthedocs.io/en/latest/index.html

[15] C. Zeng, J. Huang, H. Wang, J. Xie, and Y. Zhang, "Deep Bayesian survival analysis of rail useful lifetime," *Engineering Structures*, vol. 295, p. 116822, Nov. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0141029623012373

[16] R. B. De Santis, T. S. Gontijo, and M. A. Costa, "A Data-Driven Framework for Small Hydroelectric Plant Prognosis Using Tsfresh and Machine Learning Survival Models," *Sensors*, vol. 23, no. 1, p. 12, Dec. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/23/1/12

[17] J. Shen, D. Yang, Y. Zhou, J. Pei, Z. Wu, X. Wang, K. Zhao, and Y. Ding, "Development of machine learning models for patients in the high intrahepatic cholangiocarcinoma incidence age group," *BMC Geriatrics*, vol. 24, no. 1, p. 553, Jun. 2024. [Online]. Available: https://bmcgeriatr.biomedcentral.com/articles/10.1186/s12877-024-05154-3

[18] T. T. Tran, J. Lee, M. Gunathilake, J. Kim, S.-Y. Kim, H. Cho, and J. Kim, "A comparison of machine learning models and Cox proportional hazards models regarding their ability to predict the risk of gastrointestinal cancer based on metabolic syndrome and its components," *Frontiers in Oncology*, vol. 13, p. 1049787, Mar. 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fonc.2023.1049787/full

[19] M. Matuszczak, M. Zbikowski, and A. Teodorczyk, "Predictive modelling of turbofan engine components condition using machine and deep learning methods," *Eksploatacja i Niezawodnosc*, vol. 23, no. 2, pp. 359–370, 2021. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85104001159&doi=10.17531%2fEIN.2021.2.16&partnerID=40&md5=9208d81663cc9a4dc6c55c47bf477ae9

[20] B. Zhou, T. Pychynski, M. Reischl, E. Kharlamov, and R. Mikut, "Machine learning with domain knowledge for predictive quality monitoring in resistance spot welding," *Journal of Intelligent Manufacturing*, vol. 33, no. 4, pp. 1139–1163, Apr. 2022. [Online]. Available: https://link.springer.com/10.1007/s10845-021-01892-y

[21] S. Madasamy, B. P. Shankar, R. K. Yadav, and J. K. P, "A Machine Learning Approach in Predictive Maintenance in the IoT Enabled Industry 4.0," in *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*. Trichy, India: IEEE, Sep. 2023, pp. 418–423. [Online]. Available: https://ieeexplore.ieee.org/document/10276226/

[22] F. Arena, M. Collotta, L. Luca, M. Ruggieri, and F. G. Termine, "Predictive Maintenance in the Automotive Sector: A Literature Review," *Mathematical and Computational Applications*, vol. 27, no. 1, p. 2, Dec. 2021. [Online]. Available: https://www.mdpi.com/2297-8747/27/1/2

[23] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," *IEEE Access*, vol. 10, pp. 99 129–99 149, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9893798/

[24] H. L. Wang, X. Ma, and P. O. Arnäs, "A data-driven survival modelling approach for predictive maintenance of battery electric trucks," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5999–6004, 2023, 22nd IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896323010169

[25] Scopus, "Scopus database," [Online]. Available: https://www.scopus.com, [Accessed: Jan. 2, 2024].

[26] E. Ouda, M. Maalouf, and A. Sleptchenko, "Machine Learning and Optimization for Predictive Maintenance based on Predicting Failure in the Next Five Days:," in *Proceedings of the 10th International Conference on Operations Research and Enterprise Systems*. Online Streaming, — Select a Country —: SCITEPRESS - Science and Technology Publications, 2021, pp. 192–199. [Online]. Available: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010247401920199

[27] R. Kumar Sharma, "Improving quality of predictive maintenance through machine learning algorithms in industry 4.0 environment," *Proceedings on Engineering Sciences*, vol. 5, no. 1, pp. 63–72, Mar. 2023. [Online]. Available: http://pesjournal.net/journal/v5-n1/6.pdf

[28] S. Pölsterl, "Evaluating survival models," [Online]. Available: https://scikit-survival.readthedocs.io/en/stable/user_guide/evaluating-survival-models.html, 2023, [Accessed: Aug. 28, 2024].

[29] "Kaggle datasets," [Online]. Available: https://www.kaggle.com/datasets, [Accessed: Jan. 2, 2024].

[30] G. Portugal, "Gorenje portugal," https://www.gorenje.com.pt/, 2024.

[31] A. Biswas, "Microsoft azure predictive maintenance," https://www.kaggle.com/datasets/arnabbiswas1/microsoft-azure-predictive-maintenance/data, 2024.

[32] S. Pölsterl, "Introduction to survival analysis," [Online]. Available: https://scikit-survival.readthedocs.io/en/stable/user_guide/00-introduction.html, 2023, [Accessed: Aug. 28, 2024].

[33] ——, "Boosting survival models," [Online]. Available: https://scikit-survival.readthedocs.io/en/stable/user_guide/boosting.html, 2023, [Accessed: Aug. 28, 2024].

[34] ——, *Support Vector Machines for Survival Analysis*, [Online]. Available: https://scikit-survival.readthedocs.io/en/stable/user_guide/survival-svm.html, 2023, [Accessed: Aug. 24, 2024].

[35] P. Schober and T. R. Vetter, “Survival analysis and interpretation of time-to-event data: The tortoise and the hare,” *Anesthesia & Analgesia*, vol. 127, no. 3, pp. 792–798, 2018, [Accessed: Jun. 14, 2024]. [Online]. Available: https://journals.lww.com/00000539-201809000-00032

[36] scikit-learn developers, “Release notes for scikit-learn 1.3.2,” [Online]. Available: https://scikit-learn.org/stable/whats_new/v1.3.html#version-1-3-2, [Accessed: Aug. 28, 2024].

[37] S. Pölsterl, “Installation guide for scikit-survival 0.22.2,” [Online]. Available: https://scikit-survival.readthedocs.io/en/v0.22.2/install.html, [Accessed: Aug. 28, 2024].

[38] SciPy community, “Scipy general tutorial (version 1.10.1),” [Online]. Available: https://docs.scipy.org/doc/scipy-1.10.1/tutorial/general.html, [Accessed: Aug. 28, 2024].

[39] Python Software Foundation, “Python 3.9.0 release,” [Online]. Available: https://www.python.org/downloads/release/python-390/, [Accessed: Aug. 28, 2024].

[40] T. Minami *et al.*, “Machine learning for individualized prediction of hepatocellular carcinoma development after the eradication of hepatitis C virus with antivirals,” *Journal of Hepatology*, vol. 79, no. 4, pp. 1006–1014, Oct. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0168827823004245