INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL

Remaining Useful Life Prediction on the NASA CMAPSS Dataset Comparing LSTM and Transformer models

DAVID NUNO VILAS BOAS GUILHERME

Outubro de 2025

Remaining Useful Life Prediction on the NASA CMAPSS Dataset Comparing LSTM and Transformer models

David Nuno Vilas Boas Guilherme

Student n.: 1160684

Dissertation for obaining the degree of
Master in Artificial Intelligence Engineering

Supervisor: Professor Carlos Fernando da Silva Ramos

Jury:

President: Maria Goreti Carvalho Marreiros, Full Professor, Institute of Engineering - Polytechnic of Porto (ISEP/IPP)

Members:

Ana Maria Neves Almeida Baptista Figueiredo, Associate Professor, Institute of Engineering - Polytechnic of Porto (ISEP/IPP)

Carlos Fernando da Silva Ramos, Full Professor, Institute of Engineering - Polytechnic of Porto (ISEP/IPP)

Porto, September 2025

Resumo

A manutenção preditiva tem vindo a ganhar uma importância crescente na indústria, sobretudo em sistemas complexos e críticos, como os motores *turbofan* utilizados na aviação. O objetivo central desta dissertação consiste na previsão do tempo de vida útil (RUL) de motores a jato, recorrendo ao conjunto de dados disponibilizados pela NASA, denominado de *Commercial Modular Aero-Propulsion System Simulation* (CMAPSS). A correta estimativa do RUL permite reduzir custos de manutenção, prevenir falhas inesperadas e aumentar a segurança operacional.

Esta pesquisa iniciou-se com uma análise detalhada do *dataset,* explorando as diferentes variantes, que representam condições de operação e modos de falha distintos. Realizou-se o pré-tratamento dos dados, incluindo normalização, seleção de sensores relevantes e construção das sequências temporais. Aplicaram-se técnicas de seleção de características, como filtros de baixa variância e elevada correlação e ainda o método *Boruta*. Desta forma é possível assegurar que apenas as variáveis com impacto real sobre o RUL fossem utilizadas no treino dos modelos.

Posteriormente, avançou-se para a fase de implementação de dois modelos com base em arquiteturas presentes na literatura. O primeiro modelo, baseado em redes *Long Short-term memory* (LSTM), explora a capacidade de capturar dependências temporais de longo prazo. O segundo modelo foi um *Transformer*, cuja principal inovação reside no mecanismo de atenção.

Os resultados experimentais foram avaliados através das métricas *Root Mean Squared Error* (RMSE) e *Mean Absolute Error* (MAE). O modelo LSTM obteve desempenhos competitivos no *subset* FD001, confirmando estudos prévios que o apontam como uma base robusta para cenários simples. O *Transformer* revelou vantagem no *subset* FD002. Contudo, em *subsets* mais complexos como o FD004, o desempenho de ambos aproximou-se, refletindo os desafios ainda existentes na generalização destes métodos.

A comparação entre LSTM e *Transformer* revelou que as LSTM são mais consistentes em cenários controlados com condições operacionais simples e bem definidas. Já o *Transformer* apresentou algum potencial em conjuntos com maior variabilidade, como o FD002, embora os resultados não tenham sido consistentes em todos os casos. Estes contrastes reforçam a idea de que, as LSTM continuam a ser uma escolha fiável, enquanto os *Transformers* ainda enfrentam desafios de generalização. Apesar disso, a literatura aponta para futuras melhorias, mais concretamente implementação de arquiteturas híbridas ou variantes especializadas, que podem permitir ultrapassar estas limitações.

Em suma, a presente dissertação contribui para o avanço do conhecimento na área da manutenção preditiva, fornecendo uma análise comparativa entre as duas arquiteturas mais relevantes. Os resultados obtidos reforçam a necessidade de continuar a explorar combinações inovadoras entre modelos e metodologias, de forma a desenvolver sistemas de prognostico cada vez mais precisos, interpretáveis e aplicáveis em contextos industriais reais.

Palavras-chave: Manutenção Preditiva, *Remaining Useful Life*, *Commercial Modular Aero-Propulsion System Simulation* (CMAPSS), *Long Short-Term Memory* (LSTM), *Transformer*

Abstract

Predictive maintenance has been gaining importance in industry, especially in complex and critical systems, such as turbofan engines used in aviation. The main objective on this dissertation is the prediction of the Remaining Useful Life (RUL) of jet engines, using the dataset provided by NASA, known as Commercial Modular Aero-Propulsion System Simulation (CMAPSS). Accurate RUL estimation reduces maintenance costs, prevents unexpected failures and improves operational safety.

This research began with a detailed dataset analysis, exploring its different subsets, each representing distinct operating conditions and fault modes. Data preprocessing was then performed, including normalization, feature selection, and construction of temporal sequences. Feature selection techniques were also applied, such as low variance and high correlation filters as well as Boruta method, to reduce the number of features used. Thus, only selecting variables with real impact on RUL were employed in model training.

Subsequently, two models were implemented based on architectures studied in the literature. The first model, based on Long Short-Term Memory (LSTM) networks, leverages their ability to capture long term temporal dependencies. The second model was a Transformer, whose main innovation lies in the attention mechanism.

Experimental results were evaluated using the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics. The LSTM model achieved competitive performance on FD001, confirming prior studies that highlight it as a robust baseline for simple scenarios. The Transformer showed an advantage on the FD002 subset. However, in more complex subsets such as FD004, the performance of both models converged, reflecting the remaining challenges in generalizing these models.

The comparison between LSTM and Transformer revealed that LSTMs are more consistent in controlled scenarios with simple, well defined operating conditions. The transformer demonstrated potential in datasets with greater variability, such as FD002, although its results were not consistent across all subsets. These contrasts reinforce the idea that, in their current state, LSTMs remain a dependable choice, while Transformers still face generalization challenges. Nevertheless, the literature points to future improvements, particularly through the implementation of hybrid architectures or specialized variants, which may overcome these limitations.

In summary, this dissertation contributes to the advancement of knowledge in predictive maintenance by providing a comparative analysis between two of the most relevant architectures. The results reinforce the need to continue exploring innovative model and methodology combinations to develop prognostic systems that are increasingly accurate, interpretable and applicable in real industrial scenarios.

Keywords: Predictive Maintenance, Remaining Useful Life, Commercial Modular Aero-Propulsion System Simulation (CMAPSS), Long Short-Term Memory (LSTM), Transformer

Contents

Figure list

Tables list

Acronyms

Acronyms List

AI	*Artificial Intelligence*
LSTM	*Long-Short Term Memory*
CM	*Corrective Maintenance*
RUL	*Remaining Useful Life*
PM	*Predictive Maintenance*
ML	*Machine Learning*
PHM	*Prognostics and Health Management*
CMAPSS	*Commercial Modular Aero Propulsion System Simulation*

1 Introduction

1.1 Introduction

In modern industrial environment, companies face an increasing pressure to ensure equipment reliability while reducing both downtime and operation expenses[1], [2], [3]. Traditional maintenance strategies, that consists on the application of reactive repairs, also known as corrective maintenance (CM), are often inefficient and may cause production interruptions [4]. However, with technologies advancement and data analytics, predictive maintenance has (PM) has gained relevance as a proactive strategy, offering a way to overcome the limitations of reactive approaches [4]. Recent studies reinforced this trend by demonstrating the effectiveness of predictive maintenance in real scenarios. Fernandes et al. [5] provides a systematic review of industrial case studies, showing consistent benefits in terms of cost reduction, equipment availability, and safety improvements.

Predictive maintenance uses data collected from sensors, in combination with machine learning algorithms, and predictive analytics, to forecast equipment failures before they happen. This allows teams to react and execute maintenance interventions to prevent costly downtime and prolong asset lifespan [6], [7], [8]. Unlike conventional strategies that depend on rigid schedules or react only after breakdowns, this method shifts the focus toward proactive, data driven decision making.

In the manufacturing industry, where unexpected downtime can often result in substantial financial losses and production bottlenecks, the implementation of effective predictive maintenance strategies is essential, as it accounts for a notable share of overall production related expenses [9], [10], [11]. However, despite its potential benefits, deploying predictive maintenance in manufacturing introduces unique challenges that must be addressed to ensure it is properly integrated into daily operations.

This chapter offers a look at predictive maintenance within the manufacturing sector. It begins by outlining how maintenance practices have evolved from reactive and preventive approaches to predictive methodologies. It also highlights the obstacles companies encounter when trying to sustain reliable operations and assets. Finaly, it reflects on the significance of predictive maintenance and its influence on improving manufacturing performance.

1.2 Evolution

Over time, maintenance practices have evolved considerably. What began as purely reactive measures has gradually shifted toward predictive approaches, aiming to optimize costs, efficiency, and quality [12], [13]. The literature usually distinguishes four key stages: reactive, preventive, proactive and finally predictive maintenance.

- Reactive maintenance is performed only after a machine breaks down. It does not involve upfront expenses or extensive planning, but it carries the risk of unexpected failures.[11], [13]
- Preventive maintenance, which gained traction during the second industrial revolution with electrification and assembly lines production, follows a predetermined schedule designed to slow down or prevent deterioration of components [1], [11]. Literature also refers to two types of preventive maintenance:
    - Periodic, which refers to a maintenance executed in periodic cycles.
    - Interventions triggered by the actual condition of the equipment.
- Proactive maintenance, aligned with the third industrial revolution and the spread of automation, combines reactive and preventive elements while incorporating data analysis to identify issues that could cause failures[13].
- Predictive maintenance focus on preventing breakdowns by studying production data to uncover patterns and forecast potential problems. This method relies on real-time asset monitoring combined with external data sources[7], [11].

1.3 Principles of Predictive Maintenance

Most studies highlight a set of fundamental activities that characterize predictive maintenance practices. The first is data acquisition, which involves the continuous monitoring of the equipment condition, using sensors readings and other collection devices. Reliable acquisition is crucial ([11], [14] since sensor readings are often affected by noise or errors caused by harsh operational environments [14].

Once the data is collected, it must be carefully processed. This analysis stage is often referred to as preprocessing. This includes statistical treatment to transform raw signals into usable information. Typical tasks are cleaning the dataset and managing missing values. Skipping or poorly executing this step can leave the data too noisy or complicated to be used for predictive modelling [3], [15], [16].

Following preprocessing, the next essential state is the feature extraction and selection. At this point, the goal is to identify the most informative variables from the large pool of collected data, since not all raw information can directly contribute to fault detection or remaining useful life (RUL) estimation. This step involves feature selection methods that can be applied to reduce dataset dimensionality by discarding redundant or irrelevant data. By focusing only on the most relevant features, maintenance systems can achieve higher accuracy and robustness in subsequent analysis [10].

Once the main features are identified, they are passed into the modelling and prognostics stage. Here, different techniques that go from statistical methods to advanced machine learning and deep learning algorithms are applied to detect early signs of failure. These models can capture both linear and nonlinear behaviours, adapt to complex operating environments, and learn from historical data to provide increasingly reliable predictions. In the context of predictive maintenance, this step marks the transition from descriptive monitoring to future prognostics, offering organizations the ability to anticipate breakdowns and plan interventions in advance [1], [7].

The final stage is the decision making, where the outputs of predictive models are translated into actionable maintenance strategies. This phase ensures that prognostic information is not only accurate but also practical in guiding operational choices. Maintenance planners can use RUL estimates and fault predictions to prioritize repairs, schedule interventions, and optimize the resources allocation. By integrating predictive insights into structured planning, organizations can significantly reduce unplanned downtime, avoid cascading failures, and maintain production continuity [8], [12].

1.4 Predictive Maintenance benefits

It is worth mentioning the benefits provided by the predictive maintenance at both operational and strategic levels. One of the primary advantages is the reduction of unplanned downtime. As highlighted in [1], inappropriate maintenance strategies can reduce overall production capacity by 5 to 20%, while digital predictive approaches are expected to increase asset availability by 5 to 15% and reduce maintenance costs by 18 to 25 %. This demonstrates how predictive strategies not only mitigate unexpected failures but also enhance system availability.

Predictive Maintenance also leads to significant cost savings. According to [7] traditional corrective methods often result in catastrophic failures that increase downtime and cost, whereas predictive approaches allow failures to be anticipated and addressed in advance. Similarly, [10] stresses that predictive maintenance is often the most economical maintenance strategy as it maximizes component lifetime without excessive interventions. By integrating condition monitoring and prediction, companies can avoid both the high costs of unexpected and the inefficiencies of premature replacements.

Another important benefit is the extension of equipment life and reliability. As noted in [11], the main objective of preventive and predictive approaches is to reduce the failure rate or failure frequency of the equipment minimizing production losses and improving product quality. By focusing on the actual degradation state of assets rather than fixed schedules, predictive maintenance enables machinery to operate safely and effectively for longer periods of time.

Predictive strategies also enhance safety and environmental protection. Failures in critical infrastructures such as water networks or power grids can cause resource waste, service disruption, and even ecological consequences. As shown in [10], predictive approaches help operators act in advance to avoid such disruptions, while [1] underlines their role in improving safety, health and environmental performance. In industries where failures carry severe risks, predictive maintenance provides a crucial safeguard.

Moreover, predictive maintenance improves planning and resource allocations. The review presented in [7] where the authors emphasize that predictive strategies not only forecast failures but also integrate these predictions into optimized maintenance plans. This allows operators to schedule interventions, manage spare parts more efficiently, and reduce the overhead costs associated with emergency repairs. Similarly, [12] the work highlights how advancements in predictive maintenance positively affect production quality, reinforcing its role as strategic enabler in manufacturing.

Finaly, predictive maintenance supports long-term competitiveness and sustainability. The literature in [12] shows that modern maintenance approaches contribute not only to productivity but also to production quality, sustainability and safety. This reflects how predictive strategies align with broader industrial goals, such as waste reduction, energy optimization and supporting more resilient production systems.

In summary, predictive maintenance reduces, downtime [1], minimizes costs [7], [10], extends asset life and reliability [11], improves safety and environmental protection [1], [10], enables more efficient planning [7], [12] and enhances sustainability [12]. These benefits combined explain why predictive maintenance has become important to modern industrial operations and a fundamental component of the Industry 4.0 paradigm [1]. This conclusion is consistent with empirical findings reported in industrial reviews, where predictive maintenance initiatives have delivered measurable gains in efficiency and reliability across manufacturing systems [5].

1.5 Structure of dissertation

Regarding the structure of this dissertation, the contents are distributed through 5 Chapters. This first chapter being the Introduction where it is provided a contextualization and the definition of the problem being addressed. It also introduces the principles and benefits of predictive maintenance.

The second chapter - State of the art reviews the existing literature on predictive maintenance. It provides the theoretical background, surveys and discussion on advancements in deep learning for Remaining Useful Life (RUL) prediction. It also encompasses the benchmark datasets with particular emphasis on NASA CMAPSS dataset.

Chapter three Implementation describes the methodology applied on this thesis. It details the dataset overview, exploratory data analysis, preprocessing methods and the design of the predictive models. Two architectures are implemented, a Long Short Term Memory (LSTM) network and a Transformer model.

Moving to Chapter four, Results, that presents the experimental results obtained from LSTM and Transformer implementations. The models are evaluated using established error metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). It is also provided a comparison of the two architectures between them and a comparison with the state-of-the-art implementations as well.

The last chapter, Conclusion, summarizes the main contributions of this work, presents a reflection on its limitations and provides directions for future work.

2 State of the art

2.1 Background of Predictive Maintenance

Predictive maintenance, as it was presented on the previous chapter, represents a proactive approach to equipment maintenance that aims to anticipate and prevent failures before they occur, thereby minimizing downtime, reducing maintenance costs, and optimizing asset performance[15], [17], [18] With this chapter it is intended to provide an in-depth exploration of the concept, evolution, and principles underlying predictive maintenance in the context of manufacturing industries.

In the literature it is most often referred two key concepts. The first one is traditional maintenance approaches such as preventive or reactive maintenance, which rely on fixed schedules or responses to failures, predictive maintenance leverages real-time data from sensors, historical maintenance records, and advanced analytics to identify potential issues before they cause disruptions[19], [20]. And the second one and more recent concept that is predictive maintenance, also known as condition-based maintenance or prognostics, is a maintenance strategy that relies on data analysis and machine learning algorithms to predict equipment failures based on the condition of the asset.[21]

Advancements in sensor technology that allow capturing and transfer data[22], data analytics, and computing power have provided a way for predictive maintenance to become a viable strategy for modern manufacturing industries[11], [23], [24]. By using the power of big data and machine learning, predictive maintenance enables organizations to move beyond fixed schedules and take a proactive approach to equipment maintenance[25]. The ability to predict failures in advance improves maintenance teams abilities to prioritize and plan maintenance activities more effectively, leading to improved asset reliability and operational efficiency[15].

The literature of this field of study often points to some key principles to effectively predict and prevent equipment failures. The first one being Data Acquisition that encompass the continuous equipment monitoring health through sensors and other data collection devices[15]. This activity used to be performed by time-based and hands on equipment maintenance, but these techniques have increasingly been seen as flawed and unreliable in recent years thus the increased usage of predictive maintenance, or as referred in other studies online maintenance[26].

Equipment monitoring improves two key factors on a production line, repair downtime and cost of equipment failures. The machines used have high technology hidden, so if one of these machines is idling due to repair operations it will lead to less productivity consequently leading to out of schedule deliveries and dissatisfied customers[27], [28]

2.2 Predictive maintenance state of the art

The advancement on the Internet of Things technology was a key enabler of predictive maintenance strategies by allowing the record of continuous and real time data from physical assets through a distributed sensors network [21], [29], [30]. The spread of these connected devices allows the collection of high amounts of data, which is then analysed supporting early fault detection. The ultimate result is the optimization of the maintenance Schedule and the operational cost reduction. However, all these benefits come with several challenges that need to tackle. The first one is the unreliable data or noise on the data collected [22], [23], [24], then there is also the need of low latency data processing [18]. Lastly, real world scenarios are not simple or similar environments, each has its own characteristics[23], [31]. Literature reveals research to address these challenges, exploring the integration of artificial intelligence and machine learning algorithms alongside with edge computing to process data closer to the source and thus improving the response times[18], [31]. The works from[22], [23], [24] propose the application of anomaly detection techniques and evaluation of the data credibility. For instance, [23] introduces a credibility detection method based on an area shared context, leveraging spatial proximity between devices to enhance anomaly identification using a probabilistic detector paired with a sliding window mechanism. Furthermore, the authors on [23] present an extended version of this work present the ability to detect contextual anomalies achieving detection performance above 97% highlighting its compatibility with real time edge computing environments. Other research presents a method to improve data quality by filtering noise whilst preserving meaningful anomalies [24]. This research method is a noise scoring technique based on deviation and rate of change. On the same topic similar research focuses on distinguish true events from noise data using exponential moving averages and sensor correlation patterns. Concluding a proved effectiveness in high volume IoT scenarios such as traffic monitoring[22]. All these papers show the need of a good data preprocessing mechanism when using IoT pipelines, to ensure accuracy and trustworthy data streams to feed the predictive models. Comparable concerns are visible in the monitoring of hydraulic systems, where faulty or inconsistent sensors can seriously compromise condition assessment. Helwig et al. [32] presented a systematic approach in which hundreds of features were automatically extracted from pressure and flow measurements, and the most relevant ones were selected using correlation analysis. By combining these features with linear analysis, their framework could classify fault types and severity levels, even under variable load cycles. At the same time, studies from the same authors [33]showed how the compensation for defective sensor readings before feeding them into diagnostic algorithms can improve robustness in practice. This shows that in addition to filtering noise, reliable prognostics often depend on actively handling sensor degradations to preserve the prediction's integrity. As a result of the decentralization of IoT deployments several studies focused on edge-based solutions to detect anomalies in sensor data streams[18], [21], [31] . Starting with[18] where the authors present their approach, IRESE, using unsupervised machine learning directly on IoT gateways to identify rare events such as gunshots, alarms, achieving high precision and recall rates above 90% despite the absence of labelled data. Other study describes a distributed anomaly detection model based on one class principal component classifier (OCPCC). This method takes advantage of spatial correlations within wireless sensor networks, decentralizing the detection process across clustered nodes, outperforming both local and existing distributed methods[21]. In high voltage systems, where failures like partial discharges can lead to significant damage, embedded unsupervised models have been proposed to monitor insulation conditions in real time[31]. Lourenço et al. [34] propose an unsupervised anomaly detection framework for railway wheel out-of-roundness diagnosis that segments strain gauge signals with Hidden Markov Model (HMM), extracts time frequency features with Short Time Fourier Transform (STFT) and using the Principal Feature Analysis (PFA) for feature selection. The authors compute a single severity indicator using an isolation Forest anomaly score, enabling detection of defective wheels under several operating conditions.

Literature also has explored how IoT technology can be integrated into broader predictive maintenance platforms. One example is the SIMAP, an intelligent system designed for real time diagnosis using data from diverse sensors[29]. This system also provides continuous monitoring of equipment conditions, enabling early detection of anomalies and supporting the dynamic adjustment of the maintenance schedules. The authors apply this system in a wind farm to demonstrate its adaptability to both technical and economic constraints, more precisely on the health monitoring of wind turbine gearboxes. Another study focuses on facility management in the construction and building sectors, where preventive maintenance approaches often fail. This work proposes a data driven predictive maintenance framework that combines IoT data with Building Information Modelling (BIM)[30]. This framework applies techniques such as artificial neural networks (ANN) and support vector machines (SVM) that predict future condition of mechanical, electrical and plumbing (MEP) components. In this research the authors also apply a case study to validate the effectiveness of this integration. A different application domain where predictive maintenance has made rapid progress is rotating machinery. Ghanbari et al. [35] combined convolutional neural networks with long short-term memory layers, allowing the model to capture both local vibration patterns and longer temporal dynamics. Their approach achieved high accuracy in distinguishing between subtle gear faults under varying loads. Nurudeen et al. [36] evaluated bagging, boosting, stacking and voting strategies applied to vibration data, showing that ensembles consistently outperform individual learners by improving stability and reducing variance. Similar research by Arshad Mayo et al. [37] reported high precision rates, above 98%, when deep learning was applied directly to raw vibration signals. Collectively, these contributions indicate that gearboxes serve as a representative test case where the hybrid models and ensemble diversity offer improvements in diagnostic reliability.

Moving on to machine learning techniques where they play a central role in predictive maintenance enabling the early fault detection, feature selection and the estimation of Remaining useful life (RUL) as it will be discussed further on this chapter. Several studies have demonstrated their value in enhancing maintenance strategies across industrial domains[38], [39], [40], [41]. For instance, the authors in[38] present an approach that integrates anomaly detection, time segmentation and machine learning classification algorithms. This junction allows the prediction of downtime in stamping presses, achieving notable improvements in macro F1-score and ROC AUC. Similarly,[39] addresses the challenge of high dimensional data, in other words, the classification objects often are characterized by numerous attributes, many of them revealing to be irrelevant attributes. So, the authors present a method to identify significant features using an enhanced random forest algorithm. The algorithm' ability to differentiate relevant features form noise provides useful insights towards a more reliable classification. The work on[40]expands on optimization techniques by reviewing twelve bio inspired algorithms, such as artificial bee colony and particle swarm optimization. This work emphasizes their relevance in handling large scale, dynamic problems that are often encountered un predictive maintenance. Parallelly, the work on[41] proposes a wavelet-based signal processing technique in order to improve fault diagnosis in machinery, as an alternative to traditional Fourier based methods. The contribution of these works allows to understand how machine learning is adapting to try to meet the complex demands of predictive maintenance systems. Literature also presents a large body of work exploring predictive maintenance through RUL estimation.[42] presents a review of condition-based maintenance (CBM) models, where decisions are made based on real time equipment health assessments. This review categorizes prognostic models into four main types:

- Physical;
- Knowledge based;
- Data driven;
- Hybrid;

To all these model types, the authors present the distinct techniques and algorithms best suited to each. This review highlights ongoing trends in machine prognostics and suggests direction for further development within CBM frameworks. To address challenges in RUL prediction arising from hard failures and system differences,[43] proposes a Weibull proportional hazard model. This model considers degradation as time-varying covariate and integrates failure time data to derive RUL distributions, demonstrating effective application in the context of lead-acid battery maintenance. Complementary to statistical modelling,[44] introduces a two-module approach for RUL estimation. The first model constructs a health indicator named weighted minimum quantisation error that fuses mutual information from multiple features to represent machinery degradation. The second module uses a maximum likelihood estimation algorithm and particle filtering to estimate RUL, applied to vibration signals obtained from accelerated degradation tests of rolling element bearings. Variability between systems, known as unit-to-unit variability (UtUV), present another complication for accurate RUL prediction. To address this aspect,[45] proposes a Wiener process model (WPM) that incorporates both age as well as state dependence to model diverse degradation paths. This methodology includes a unit maximum likelihood estimation (UMLE) algorithm for parameter estimation without requiring predefined distributions, and a particle filtering framework that uses a fuzzy resampling algorithm to mitigate sample impoverishment. This approach is validates using a turbofan engine degradation dataset. Finally,[46] presents a similarity-based RUL prediction system that leverages data from multiple runs to failure units. By constructing a degradation pattern library, the method estimates the RUL of a test unit based on similarity to previously observed paths. Its effectiveness is demonstrated in the 2008 PHM Data Challenge Competition, showcasing the potential of data driven pattern matching strategies in industrial prognostics. Although these studies highlight the role of data driven approaches, it is equally important to consider how benchmark datasets are produced. For example, Saxena et al. [47], [48] introduced a damage propagation model that is affected by flow and efficiency modifiers. This simulation continues until a predefined health index reaches zero, which is considered a failure. The result is a run to failure sensor trajectories that can be used in PHM competitions. Similar work in other engineering domains also stresses the value of simulation as a test environment. To exemplify, Altosole et al. [48] employed a real-time, hardware simulations for naval propulsion systems, demonstrating how such setups can reduce commissioning costs and speed up the tuning of control strategies.

Fuzzy logic and clustering techniques have increasingly been applied in predictive maintenance and condition monitoring tasks, particularly for handling uncertainty, noisy sensor data, and ambiguous diagnostic signals[25], [49], [50], [51]. A notable example is the integration of fuzzy clustering with probabilistic anomaly detection methods for the filtering of wind data. This method addresses the challenges posed by the integration of wind generators into electrical grids. With the increasing penetration of wind power, system operators are compelled to enhance operational functions to mitigate the effects of intermittent and non-programmable generation profiles. By leveraging wind forecasting and reliability models based on experimental data, this approach offers promise in improving the accuracy and reliability of wind power forecasting and grid operation planning. Through the fusion of innovative techniques, this methodology aims to facilitate the effective integration of wind energy into existing electrical grids, despite the challenges posed by outlier and incoherent data acquired from Supervisory Control and Data Acquisition systems[25]. In a similar case, fuzzy logic has been leveraged to develop early warning systems for condition-based maintenance (CBM). The primary goal is to furnish early warnings regarding potential breakdowns or hazardous scenarios in production lines, empowering stakeholders to make more informed decisions on maintenance strategies. Through an illustrative example focused on a car production line, the study illustrates the practical utility of this approach in mitigating operational risks and optimizing maintenance efforts[52]. Extending this capability further, [53] presents an ontology-based method that combines fuzzy clustering with semantic technologies. Since data mining techniques have been employed for predicting the timing of machinery failures, existing approaches often overlook the criticality of these failures, resulting in suboptimal maintenance strategies. By leveraging machine historical data, fuzzy clustering techniques ascertain the criticality of failures, while semantic technologies utilize this information to predict both the timing and criticality of anomalies. The approach culminates in the development of a domain ontology for predictive maintenance knowledge modelling, accompanied by a set of Semantic Web Rule Language (SWRL) predictive rules. Through a real-world industrial case study, the proposed methodology is rigorously evaluated, affirming its utility and effectiveness in predictive maintenance applications. Another application focuses on grinding wheels in automated production lines. Using fuzzy logic to develop a maintenance policy that accounts for diagnostic uncertainty. This approach aims to enhance decision-making by incorporating fuzzy logic's ability to handle vagueness and uncertainty, ultimately optimizing maintenance strategies, and prolonging the lifespan of grinding wheels[54]. Another work explores the value of fuzzy systems in diagnostics highlighted in bearing fault analysis, offering a novel approach through the application of a fuzzy expert system (FES). Through the utilization of the Similarity partition method, fuzzy rules are automatically induced from numerical data, facilitating a more efficient diagnosis process. Given the inherent challenge of high noise levels in faulty bearing data, the paper introduces the Improved Range Overlaps method (IRO) to select input feature vectors by assigning the validity degrees. Validation and efficacy of the proposed model are demonstrated through experimental data sourced from the Case Western Reserve University database and the NSF I/UCR Center on Intelligent Maintenance Systems (IMS) database, affirming its practical utility in bearing fault diagnosis within induction machines[49].

Another technique presented on the literature is the Random Forests, that represent a powerful ensemble learning technique that integrates a multitude of decision trees where each tree's decision relies on a random vector sampled independently and uniformly across the forest[55]. With this structure as the number of trees in the forest grows, the generalization error asymptotically converges to a limit. The work from[55] apply this concept and using random subsets of features for node splitting, Random Forests achieve performance levels comparable to Adaboost while maintaining greater robustness to noise. Additionally, the model offers internal monitoring of error and variable importance, making it suitable for both classification and regression tasks. Despite their strengths, a gap remains in the application of ensemble methods to RUL Prediction[56]. Addressing this, a novel ensemble strategy was proposed, tailored to consider degradation's impact on RUL prediction. By assigning optimized, degradation-dependent weights to each learner, this method synthesizes prediction results from multiple algorithms, enhancing RUL prediction precision for engineered systems. Demonstrated through case studies on aircraft bearings and engines, the approach's efficacy surpasses conventional ensemble learning methods, yielding superior predictive models capable of optimizing maintenance scheduling and resource allocation[56]. In parallel, another team demonstrated the integration of Random Forests leveraging real-time data from IoT sensors. Through machine learning techniques, the system proactively detects signals indicative of potential failures, enabling early intervention by alerting operators before production disruptions occur. Real-world manufacturing IoT data was utilized to evaluate the system's efficacy, demonstrating its capability to identify impending failures and mitigate production stoppages. Comparative assessments revealed Random Forest and XGBoost as the topperforming algorithms, surpassing standalone approaches. Consequently, these superior models have been seamlessly integrated into the factory's production system, promising enhanced operational reliability and efficiency.

On the topic of RUL prediction there has also been documented work that applies Bayesian modelling techniques. The studies employ various Bayesian approaches, from rule-based models and filtering to joint degradation and failure time analysis, to improve decision making in uncertain and dynamic environments. Starting with the project from[57] focused on the improvement of the training of Bayesian Rule-Based (BRB) models by using probabilistic methods rather than fixed optimization. It is proposed a novel method based on Bayesian estimation that treats model parameters as random variables. Unlike traditional optimization techniques that treat BRB parameters as fixed but unknown values, Bayesian estimation treats them as random variables. Rather than seeking singular optimal parameter values, this method aims to estimate the posterior distribution of BRB parameters, considering the entire spectrum of possible parameter values. However, due to the inherent nonlinearity of BRB models, calculating the posterior distribution analytically becomes impractical. To work around this limitation, the Sequential Monte Carlo (SMC) sampling technique is employed for online approximation of the posterior distribution of BRB parameters. The efficacy of this approach is demonstrated through both numerical experiments and a real-world application in pipeline leak detection, showcasing its superior performance compared to traditional methods. Similarly focused on real time system adaptation,[58] proposes a Bayesian filter-based model for predicting drum degradation in hot rolling within heating coilers of Steckel mills. By integrating expert knowledge and operational data from over 118000 processes, the model enhances decision making for maintenance in industrial environments. In a related topic the study[59] develops a Bayesian updating methods to adjust the stochastic parameters of exponential degradation models based on real-time condition monitoring data. By effectively modelling the degradation signal, it becomes feasible to compute a residual-life distribution for the monitored device, facilitating informed decision-making. Utilizing these models, a closed-form residual-life distribution for the monitored device is derived, thereby enhancing the precision of decision models. The application of these methodologies to degradation signals obtained from accelerated testing of bearings further demonstrates their practical utility and relevance in industrial contexts. Further advancing on work related with Bayesian degradation modelling,[60] combines Wiener process with drift and proportional hazards (PH) model within a joint framework that integrates degradation signals and time to event data. This approach benefits from the Markovian property of the Wiener process and demonstrates strong predictive accuracy through simulations and case studies. Complementing this,[61] proposes the Sensory Updated Degradation Based Maintenance (SUDM) policy, which continuously refines the residual life distribution using both real time signals and population level reliability data. This real-time updating mechanism ensures a more precise estimation of remaining component life. Utilizing a stopping rule, maintenance activities are scheduled based on the latest residual life distribution (RLD) updates. Evaluation against benchmark policies, including reliability-based and conventional degradation-based approaches, is conducted using a simulation model of a manufacturing cell, analysing metrics such as unexpected failure frequency and overall maintenance costs. The SUDM policy demonstrates its efficacy in enhancing maintenance decision-making, potentially reducing operational disruptions and costs in industrial settings. Finaly,[62] presents the statistically planned and individual improved predictive maintenance (SPII PdM) policy. Merging traditional statistical lifetime distribution-based maintenance with predictive models to enhance system availability. This integration shows that predictive maintenance techniques can be successfully embedded into established preventive frameworks for practical industrial use.

Moving on to neural network approaches where literature presents some implementations for predictive maintenance and anomaly detection. A common theme among recent works is the use of unsupervised neural architectures that allow the automatic extraction of meaningful features without prior knowledge of system behaviour or data labels. One example is the work in[63] , which proposes an enhanced Restricted Boltzmann Machine (RBM) architecture with a regularization term, aimed at automatically generating features optimized for remaining useful life prediction. This approach maximizes feature trends to better capture system degradation patterns. Benchmarking against traditional methods like principal component analysis, our approach demonstrates promising results when applied to run-to-failure datasets from rotating systems. While RBM focus on feature extraction for RUL prediction, other works tackle a different challenge. For example, to address the problem of catastrophic forgetting in continual learning,[64] presents a dual model system called Deep Generative Replay. This system simulates the behaviour of the hippocampus which serves as a short-term memory system. This framework features a cooperative dual model architecture comprising a deep generative model called “generator” and a task solving model “solver”. Through experimentation across various sequential learning settings involving image classification tasks, the efficacy of this approach is assessed, offering promising prospects for mitigating catastrophic forgetting in AI systems. Neural networks have also been applied to anomaly detection at the sensor node level using a lightweight One Class Principal Component Classifier. By focusing on detecting misbehaviour in sensor measurements before data transmission, the model reduces energy consumption and false alarms. Experiments with real world datasets show its efficacy and potential for scalable deployment in large sensor networks[65].

Research also shows some work done regarding the extraction of informative features from noisy signals. In the context of mechanical systems,[66] addresses the difficulty of identifying weak defect signals within noisy measurement environments, proposing a wavelet filter combined with Self Organizing Maps (SOM) for improved bearing defect detection at early stages. Similarly,[67] explores fatigue behaviour in copper conductors, where geometrical irregularities affect stress signal interpretation. This helps highlighting the importance of accurate measurement and analysis in structural signal evaluation. Shifting to the audio domain, [68] and[69] tackle the problem of signal representation for speech and non-speech audio, respectively. While[68]enhances Automatic speech Recognition (ASR) performance in noisy settings using Mel Frequency Cepstral Coefficient (MFCC) extraction and its shifted variant, alongside Vector Quantization and fuzzy modelling. This technique has proven instrumental in boosting ASR performance, especially in noisy environments. By incorporating spectral domain information from discourse signals, this approach exhibits robustness to noise. The research in[69] extends this approach with Gammatone Cepstral Coefficients (GTCC), showing improved classification accuracy in diverse acoustic scenes. These studies collectively emphasize that, whether in mechanical, structural, or audio systems, effective signal processing and feature extraction techniques are critical to ensure robust performance in real world environments. Going further on the context of turbofan engines, the authors[70] propose a deep learning method that tackles aspects such as data noise, diverse data types and complex degradation behaviour in the CMAPSS dataset. Their model combines an improved stacked sparse autoencoder (imSSAE) to extract deep features and built health indicator (HI) curves. The results show higher accuracy and better overall performance compared to other models.

Case Based Reasoning (CBR) is another approach documented in the literature that presents some contributions leveraging solutions to past problems to address new ones, meaning, this methodology uses historical cases that can provide insight into current challenges[71], [72]. On an industrial context, [71] this work presents a CBR framework aimed to detect and manage faults in large scale machinery, more precisely gearboxes in steel processing plants. By incorporating user friendly graphical interfaces like Microsoft Visual Basic, the framework facilitates fault diagnosis for process engineers without requiring deep expertise in artificial intelligence. The authors emphasize CBR's advantages over more complex techniques such as neural networks, highlighting its practicality in harsh and data rich environments. Complementing this perspective,[72] delves into the foundational aspects of CBR, detailing its core mechanisms solution reuse, adaptation and evaluation.

Cloud and edge computing offer complementary capabilities for distributed data processing, particularly in environments where real time responsiveness and resource optimization are critical [73], [74]. In this context,[73] presents a method for automated anomaly detection in heterogeneous wireless sensor networks. This system gathers environmental data such as temperature, humidity and light and through a hybrid approach that combines unsupervised neural network processing at the edge with multi parameters edit distance algorithm in the cloud. This system is validated using real world and synthetically distorted data, demonstrating its ability to adapt autonomously to the environment variations. In the industrial field,[74] proposes a predictive maintenance solution based on cloud computing using mobile agents and low-cost cloud nodes to enhance fault diagnosis and scheduling in manufacturing settings. This method improves fault diagnosis, service life prediction, and maintenance scheduling by enabling timely information sharing and utilization. Mobile agents distribute analysis algorithms to these nodes, facilitating local data processing and result sharing. Compared to traditional models, this approach enhances system flexibility, reduces data transmission, and swiftly adapts to changes. Validation on a motor testing system confirms its effectiveness, promising improved manufacturing efficiency and reliability. Lastly,[17] focuses on underground mining safety, where traditional data processing methods often struggle due to poor wireless connection and complex sensor data. To solve this, the authors propose a hierarchical edge computing model that processes data at both the sensor and base station levels. They also introduce smart anomaly detection techniques based on fuzzy logic and multi-source data analysis, which improve accuracy and reduce delays.

Ensuring reliability and fault tolerance is fundamental in both critical computing systems and modern manufacturing environments[75], [76]. In the context of computer systems,[75] presents a comprehensive approach to reliability analysis through dynamic fault-tree modelling, applied to fault tolerant architectures such as parallel processors, avionics systems and hypercube configurations. These systems leverage redundancy and sophisticated error recovery and are analysed using tools like Hybrid Automated Reliability Predictor (HARP), a software solution jointly developed by Duke University and NASA Langley Research Center. This is used to predict and manage faults effectively. On the manufacturing side,[76] explores how emerging cloud computing infrastructures can support reliability by enabling advanced prognostics techniques. The paper emphasizes the evolution of predictive maintenance within a cloud framework, highlighting their role in improving scheduling, resource usage and system longevity across distributed environments.

Clustering techniques have become a cornerstone in anomaly detection frameworks, particularly in IoT, industrial systems, and large-scale sensor environments[15], [77], [78]. One study, [15] proposes an anomaly detection method that relies on correlation to sustain the physical integrity of the data. Contrary to traditional methods that fail to capture the actual relationship between the variables when devising new dimensions, this paper advocates for a novel approach rooted in clustering highly correlated datasets. By eschewing dimension reduction and instead integrating clustering models, this method illuminates the relationships within the data, especially evident when equipment operates abnormally. Through clustering highly correlated datasets along a straight line, deviations from this pattern signal anomalies. By calculating stochastic distances and abnormality detection indices, this approach effectively discerns faults. Application to a hydraulic system with 17 sensors validated its efficacy, yielding comparable results in detecting reduced cooler efficiency and superior precision in detecting failures. Notably, it identified anomalies such as internal pump leakage operations, elucidating the physical meaning obscured by conventional methods. Thus, this method holds promise not only for engineering anomaly detection but also for diverse data analysis challenges. Another study[79] compares several well-known outlier detection algorithms, Elliptic Envelope, Isolation Forest, and Local Outlier Factor. These algorithms are then combined into an ensemble method, Ensemble Outlier Detector (EOD). Results show the EOD performance to be superior when compared to previous findings. In the context of Big Data, where detecting problems in real time is difficult due to the size and speed of the data[78] on team presents a method that looks not just at the content of the data but also at its context. The content detector tries to detect anomalies in real time while the context detector, considering content and contextual anomalies, refines the results. This process of refinement is achieved by using profiles, that are basically clusters of similar data points generated by a multivariate clustering algorithm. The effectiveness of the proposed framework has been demonstrated through evaluations using real-world sensor datasets provided by a local company in Brampton, Canada, as well as against the open-source Dodgers dataset from the UCI machine learning repository and the R statistical toolbox. This framework offers a promising approach to enhancing anomaly detection in Big Data environments by integrating both content and contextual information, thus mitigating false positives and improving overall accuracy. Finally,[77] resents a more advanced method that combines clustering with feature selection and optimization algorithms to improve how Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is used. This method works especially well in smart environments like industrial IoT and connected vehicles.

Cyber-physical systems (CPS) represent a groundbreaking fusion of physical and computational realms, revolutionizing human engagement with the tangible world. As the internet reshaped interpersonal dynamics, CPS promise to redefine our relationship with the physical environment. Within pivotal sectors such as transportation, healthcare, manufacturing, agriculture, energy, defence, aerospace, and buildings, CPS present a myriad of challenges and opportunities. The intricate design, construction, and validation of CPS necessitate a collaborative effort across diverse fields of expertise, calling for a unified front of researchers and educators to tackle these technical hurdles head-on[80].

Convolutional neural networks (CNN) have shown strong potential in predictive maintenance applications by automatically extracting features from raw sensor data and effective modelling complex time series, as demonstrated by these studies[16], [81]. First[16], the authors explore the use of raw three-axis accelerometer signals represented as high definition 1D images, which were fed directly into deep learning layers. The CNN architecture was able to automatically extract meaningful features and achieved high classification accuracy by effectively recognizing patterns in the raw input matrix. This approach proved successful in classifying various operational states of rotary machinery without relying on handcrafted features. In a similar direction, another study proposed a temporal convolutional network (TCN) enhanced with adaptive shrinkage to predict the remaining useful life (RUL) of equipment. This method specifically addressed the limitations of recurrent models in capturing long-term dependencies by leveraging the temporal capabilities of the TCN. Additionally, the inclusion of a subnetwork to reduce noise-related information contributed to a notable improvement in prediction accuracy, as reflected in the reduction of both mean absolute error and root mean square error metrics [81].

Recurrent Neural Networks (RNN), concretely Long Short Term Memory (LSTM) models and their variants, have gained popularity in RUL prediction tasks due to their ability to model temporal dependencies in sensor data. These techniques can capture degradation trends in complex systems such as turbofan engines[82], [83]. On study,[82] applied a Bidirectional LSTM model using the NASA CMAPSS dataset, highlighting its potential for accurate and reliable RUL prediction in aerospace applications. Another study using the same dataset emphasized the advantage of models based on neural networks over traditional techniques like regression and support vector regression. Also, in this study is mentioned the importance of data preprocessing and visualization to enhance model performance[83]. Other work tries to respond to the fact that the RNN and CNN implementations struggle with sequence information[84]. The authors of this paper present an implementation of LSTM for RUL prediction which uses the sensor sequence data to expose hidden patterns within sensor information on multiple operation conditions and degradation models. This approach is then applied on three Prognostics and Health Management data sets. The results show that LSTM for RUL prediction outperforms traditional approaches and Convolutional Neural Networks for RUL estimation. Zhang et al. [85]

introduced an attention mechanism that directs the model's focus toward the most informative segments of long input sequences, reducing information loss and improving accuracy in remaining useful life estimation for rotating machines. Beyond sequence modelling, Li et al. [86] argued that knowledge of the physical system can also be embedded into deep models. The author proposed a framework in which relationships between sensors and engine components were explicitly encoded, guiding the construction of a CNN-LSTM architecture. This method was tested on the CMAPSS dataset, and the results show an improvement on the accuracy and interpretability, since the model design choices correspond to physical dependencies. The work on [87] present an encoder-decoder architecture based on long short-term memory (LSTM-ED) to obtain an unsupervised health index (HI), since in many scenarios the degradation may not follow a pattern. This method is applied on a multi sensor time series data system where it is trained to trained to reconstruct the time series corresponding to a healthy state. The reconstruction error is used to compute HI leading to the RUL prediction. This approach is then validated using the Turbofan engine and Milling machine datasets. Beyond aerospace, generalization across varying operation conditions is a key concern in industrial environments[3]. To address this, research proposed a method for detecting bearing failures and predict RUL in electric motors under different motor conditions. It is also worth mention that this method does not need retraining nor fine-tuning. This was achieved by combining time and frequency signal analysis with a BI LSTM. The result is translated in a robust and accurate prediction across diverse conditions[3]. Another contribution, for the same challenge, presented an approach that integrates physics insights and ISO 10816 standards into a two-stage ensemble LSTM framework. This model used a predictor-corrector structure with Gaussian layers to quantify both accuracy and efficiency. Even in settings constrained by the resources, with reduced data transmission and extended battery life[88]. More recent works contribute to reinforce the importance of LSTM based models in RUL prognostics. Tippareddy et al. [89] evaluated different recurrent architectures on CMAPSS, showing that BiLSTMs can consistently achieve stable results across subsets, but further tuning and hyperparameter optimization are required to improve performance. Noot et al. [90] explicitly considered the presence of right censored data, where some engines are replaced before reaching failure and the true RUL is not observed. They adapt an ordinal regression LSTM (LSTM-OR) comparing it with a transformer model. By introducing artificially different levels of censoring on the CMAPSS dataset, they show that when most trajectories were complete (low censoring) into the CMAPSS dataset, both models achieved a similar performance, but when many trajectories ended prematurely (high censoring), the LSTM-OR provided more accurate RUL predictions. Syuhada [91] provided a detailed performance study on FD001, where carefully tuned LSTM achieved a good baseline results concluding that a well configured LSTM architecture can provide a baseline for engine prognostic tasks.

Architectures based on transformers have recently gained attention in predictive maintenance applications due to their capacity to model sequential sensor data and capture long range dependencies, which is essential for tasks such as RUL prediction[92], [93], [94]. The authors of [95] proposed the FORMULA framework, which applies a Transformer encoder to industrial alarm logs. Their approach specifically targeted the prediction of rare alarms, a scenario where class imbalance usually hinders standard methods. In the context of turbofan engines, Babaei et al. [96] introduced a Convolutional Transformer that merges convolutional layers for local feature extraction with self-attention for global context. Evaluated on the CMAPSS dataset, the model reached an accuracy close to 99.6% in classifying fault types. Vaswani et al.[97] presented a new simple network architecture, the Transformer, a mechanism based only on attention mechanism. Firstly, developed for machine translation, this architecture demonstrated not only superior performance but also greater efficiency in training. Building on this method,[98]proposed a transformer encoder architecture enhanced with a gated convolutional unit to predict RUL of industrial components. The gated convolutional unit allows the model to extract local temporal features before applying the self-attention mechanism, effectively combining the strengths of convolutional structures and transformer encoders. This hybrid approach demonstrated improved performance on the CMAPSS dataset and highlights the ongoing innovation in adapting transformer-based models to time series degradation patterns. Continuing the adaptation of the transformer the authors on [99] introduced a transformer framework for multivariate time series for RUL estimation. The transformer encoder structure was evaluated on all four subsets of the CMAPSS benchmark dataset. The authors conducted extensive experiments comparing layer normalization and batch normalization, fixed and learnable positional encodings. Additionally, they proposed a clustering normalization method and a novel expanding window. Their model outperformed 13 other state-of-the-art methods, showing an average performance increase of 137.65% across datasets. One study proposed a dual scale transformer model (DSFormer) with an encoder-decoder architecture that combines a dual attention mechanism with a temporal convolutional network. The encoder-decoder is responsible for capturing both sensor and temporal features and the temporal convolutional network retains positional information. The decoder further incorporates a feature decomposition module to capture trend characteristics, achieving superior results on the CMAPSS dataset for predicting RUL. The authors also point toward optimizing this architecture for edge deployment to enhance Industrial IoT reliability and efficiency[92]. Another study introduced a pretraining and fine-tuned framework for transformer models applied to bearing fault classification. This approach explored several strategies for tokenization and data augmentation and utilized masked pretraining to improve performance in low data scenarios. It allowed the model to adapt to unseen fault classes and new datasets, outperforming traditional end to end training while supporting faster deployment in low data settings [93]. Complementing this line of research, in [94] a framework is proposed that combines signal processing techniques and a transformer model, to predict lubricant degradation. By leveraging tools such as the temporal variation transfer function and harmonic sideband matrix (H-S Matrix) for feature extraction. The authors also applied Explainable AI, enabling the model to provide transparent insights into the critical frequency of components being affected by degradation. Although the study was limited to specific experimental conditions, the results show potential for broader predictive maintenance applications. Another recent study [100] presented an Adaptive Graph Convolutional Transformer Encoder (AGCTE) that combines graph neural networks (GNN) with an Encoder structure of the transformer architecture. Building on the limitations of traditional deep learning models extracting spatial features through convolution or fully connected layers when trying to capture complex inter-sensor relationships. The GNN component of AGCTE models the relationships between sensors with a graph structure. This approach was after evaluated using the CMAPSS dataset achieving high accuracy in RUL estimation.

To conclude this chapter, it is possible to verify the importance of predictive maintenance and its impact on modern industries be it on cost reduction or efficiency and safety increase[21], [29], [30]. Literature also reveals the shift from reactive maintenance, where only after the error occurrence there is an intervention to fix the machines, to a preventive maintenance model. This shift was enabled by many factors one of which the advancements on IoT technology affecting the way on how the data from the machines was collected. Of course, this also brought some challenges as presented on this document as well, namely noisy and unreliable data[22], [23], [24], need of real time processing[18] and system variability in real world scenarios[23], [31]. Many studies show the contributions towards a response for these challenges. Starting with noise filtering and credibility scoring as well as the importance of preprocessing before applying machine learning models to improve data quality[22], [24]. There are also techniques used for feature selection, for example on[39]with Random Forest variants, early fault detection as presented on[38] , and classification as shown on[41] using a wavelet-based signal processing for fault diagnosis, as an example. One aspect that is clear across the considered papers is the evolution towards hybrid and context aware systems that not only predict failures more accurately but also adapt to different conditions and deployment scenarios. The CMAPSS dataset appears frequently as a benchmark, underlining its role in standardizing RUL evaluation across studies. Recent trends focus on combining the strengths of various models such as attention mechanism of Transformers[97], [98], [99], [100] with the local feature extraction of CNN or temporal modelling of LSTM. These implementations also need to consider deployment constraints like computation limits on edge devices[17], [31], [73], [74].

2.3 Available datasets

Nasa Turbofan dataset

The NASA Turbofan Engine Degradation Simulation Dataset is widely used in the predictive maintenance domain[47], [85], [86], [96], [101], [102], especially for Remaining Useful Life (RUL) prediction tasks. The dataset is generated using the C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) model, which simulates the behaviour of a turbofan engine under varying operating conditions and fault scenarios.

The dataset contains multiple subsets, each representing engines operating under different conditions and degradation patterns. Key characteristics of the dataset include:

- **Engine ID**: Unique identifier for each engine unit.

- **Cycle Index**: The time step or cycle for a given engine unit, the cycle being a single operational run of an engine from start to stop.
- **Sensor Measurements**: A series of 21 sensor readings capturing operational data.
- **Operating Conditions**: Three operational settings that impact engine performance.
- **RUL Labels**: Available for training datasets to facilitate supervised learning.

ALPI https://ieee-dataport.org/open-access/alarm-logs-packaging-industry-alpi

This dataset comprises alarm sequences logged by packaging equipment in industrial environments, collected from 20 machines across various global plants between February 21 2019 and June 17 2020. It includes 154 distinct alarm codes with a highly unbalanced distribution. The csv available is a comma separated file where each row is a logged alarm event providing the alarm code timestamp of the occurrence and the machine id [95]. Each record on this dataset includes:

- The timestamp of the alarm.
- The alarm code.
- The machine ID involved in the alarm event.

There is already 3 pre-processed files (all_alarms.pickle, all_alarms.json, all_alarms.npz) containing data suitable for machine learning tasks formatted for predicting which alarms might occur in a future time window based on past alarm sequences.

Naval Propulsion CBM

This dataset was generated using a simulator of a naval frigate powered by a Gas Turbine (GT) propulsion system. This simulator includes key blocks of the vessel such as the propeller, hull, gearbox, turbine and control systems based on several similar real propulsion plants [48]. The data captures system dynamics under gradual performance degradation, offering a realistic scenario for studying condition-based maintenance (CBM). Variables include operational and physical parameters, for example ship speed, compressor efficiency, turbine decay.

Gearbox Fault Diagnosis (https://www.kaggle.com/datasets/brjapon/gearbox-fault-diagnosis)

Gearbox Fault Diagnosis dataset comprises vibration information collected using SpectraQuest gearbox diagnosis simulator. The data includes:

- Readings of vibration data under several conditions.
- Different fault conditions, such as misalignment, broken gear teeth.
- Readings of healthy and fault scenarios from gearbox operations.

The data was collected from different load and speed settings to try to simulate real-world operational environments [35], [36], [37].

Hydraulic systems dataset

This data was originated from a series of experiments conducted on a test rig developed to simulate real-world operating conditions found in industrial machinery [32], [33]. The system includes components such as colling circuit, accumulator, valves, pump, and motor. This dataset includes sensor measurements such as:

- Pressure
- Flow state
- Temperature
- Valve positions
- Motor conditions

2.4 Dataset Selection

Several datasets were reviewed, including turbofan engine simulations, industrial alarm logs, naval propulsion simulations, gearbox fault diagnosis and hydraulic systems. These alternatives provide valuable insights for tasks such as fault detection, anomaly detection, anomaly diagnosis, and condition monitoring. However, most of them lack labelled RUL information, present limited temporal resolution, or are less established as benchmarks in the predictive maintenance field.

The selected dataset for this thesis had to fulfil specific criteria. First public availability, the dataset should be of open access and well documented, this way it is easier to compare results. Another topic is the acceptance in the literature as a benchmark dataset. The selected dataset should be adopted by the research community to enable comparison with state-of-the-art methods. Sensor diversity is another characteristic of the dataset, as it should provide sufficient variables and temporal depth. Lately, labelled information, the presence of well-defined target values is important for training and evaluation.

After analysis of the available datasets, for this work the CMAPSS dataset was chosen. The aspects that mainly influenced this decision were its comprehensive structure and wide acceptance in the literature as a benchmark for evaluating RUL prediction models [84], [85], [86], [87], [103]. It supports supervised learning through labelled lifecycle trajectories and has been used as a foundation of many recent works applying LSTM, Transformer, and hybrid deep learning architectures [70], [84], [85], [86], [87]. It was also designed to have a balance between simulation control and operational realism, enabling fair comparison among algorithms while still being representative of real-world engine degradation.

3 Implementation

In this section it is presented the dataset used as well as the full solution implementation for the RUL prediction models. It is also highlighted the decisions made one each phase of development based on literature review and in the context of predictive maintenance.

3.1 Dataset overview

The dataset used was the Nasa Turbofan dataset described on section 2.3 which is generated using the CMAPSS (Commercial Modular Aero-Propulsion System Simulation) model. This model simulates the behaviour of a turbofan engine under varying operating conditions and fault scenarios. This was developed by NASA for examining engine degradation[101], [102], [103] and contains four subsets: FD001, FD002, FD003, FD004, each containing different fault modes and operational conditions.

Table 1 – Subset overview

Subset	Operation Conditions	Fault modes	Use case
FD001	1	1	Baseline single condition case
FD002	6	1	Varying conditions but same failure type
FD003	1	2	Same conditions with multiple failure types
FD004	6	2	Multiple conditions and failure types

Table 1 provides an overview of the four subsets available on the CMAPSS dataset and in this context operating conditions refer to variations on external and internal flight parameters, such as altitude, Mach number and ambient temperature. These operating conditions have an impact on engine performance and determine the baseline operating environment under which degradation occurs [47]. Another characteristic of the subsets mentioned on Table 1 is the fault mode, that represents the physical degradation mechanisms that lead to an engine failure. These physical mechanisms can be the efficiency loss in the high-pressure compressor (HPC) or reduction on flow capacity in the low-pressure turbine (LPT). The engine in Figure 1 presents the main elements of the engine model used in the dataset.

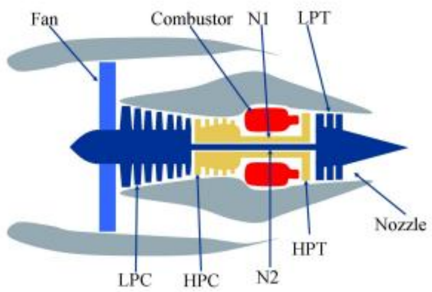

Figure 1 – Simplified engine diagram [47]

Key characteristics of the dataset include:

- **Engine ID**: Unique identifier for each engine unit.
- **Cycle Index**: The time step or cycle for a given engine unit, the cycle being a single operational run of an engine from start to stop.
- **Sensor Measurements**: A series of 21 sensor readings capturing operational data.
- **Operating Conditions**: Three operational settings that impact engine performance.
- **RUL Labels**: Available for training datasets to facilitate supervised learning.

These sensor readings include measurements like temperature, pressure, vibration, and flow values. According to [100], these variables are designed to reflect real world engine performance that suffers degradation and can show changes over time, depending on the issue and operation condition [98], [99], [100].

Unlike other datasets that are labelled, the CMAPSS does not provide RUL values explicitly for each time step. A time step, in the context of this dataset represents a single operational cycle of an engine.

All subsets contain a Train file that is a full run until failure, a Test file with early life engine data, meaning that the last cycle of each machine is not the failure cycle. Lastly the RUL file that are the true RUL values for the test engines. Each row on these datasets is a time cycle of an engine.

Literature reveals that even though this dataset can be used for prognostics and health management, there is also some challenges. The first one, and maybe the most obvious is the sensor noise that can be introduced, some sensors exhibit noise and redundancy [22], [24], [81]. Not all sensors contribute equally to degradation prediction, and some may introduce irrelevant information. Secondly normalization and windowing must adapt to the subset that is being used, FD001 and FD003 have fixed operation conditions whereas subsets FD002 and FD004 have variable operation conditions. Also, methods such as LSTMs and Transformers required effective sliding window approaches [84].

3.2 Exploratory Data Analysis

On this section it will be described the exploratory data analysis to have a better understanding of the dataset that we will use to work with. Starting with a simple plot on Figure 2 regarding the number of engines available on each dataset.

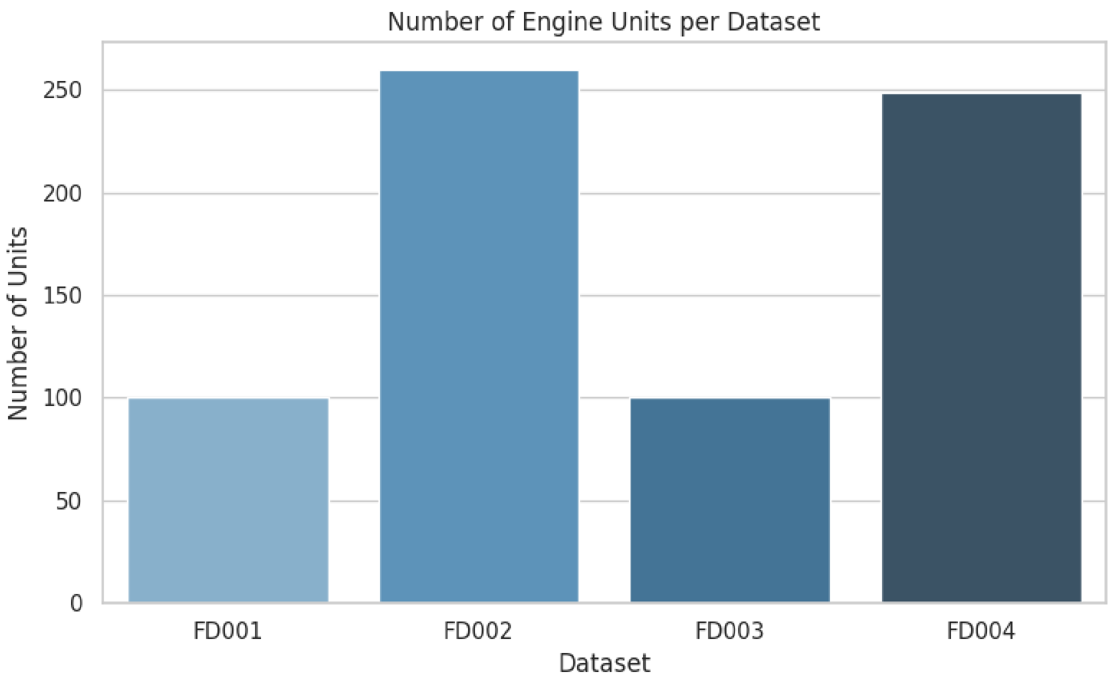

Figure 2 – Number of Engine units per dataset

The length of cycles of each subset is also worth mentioning as presented in Figure 3 where the growth on the cycle's length is visible after each subset.

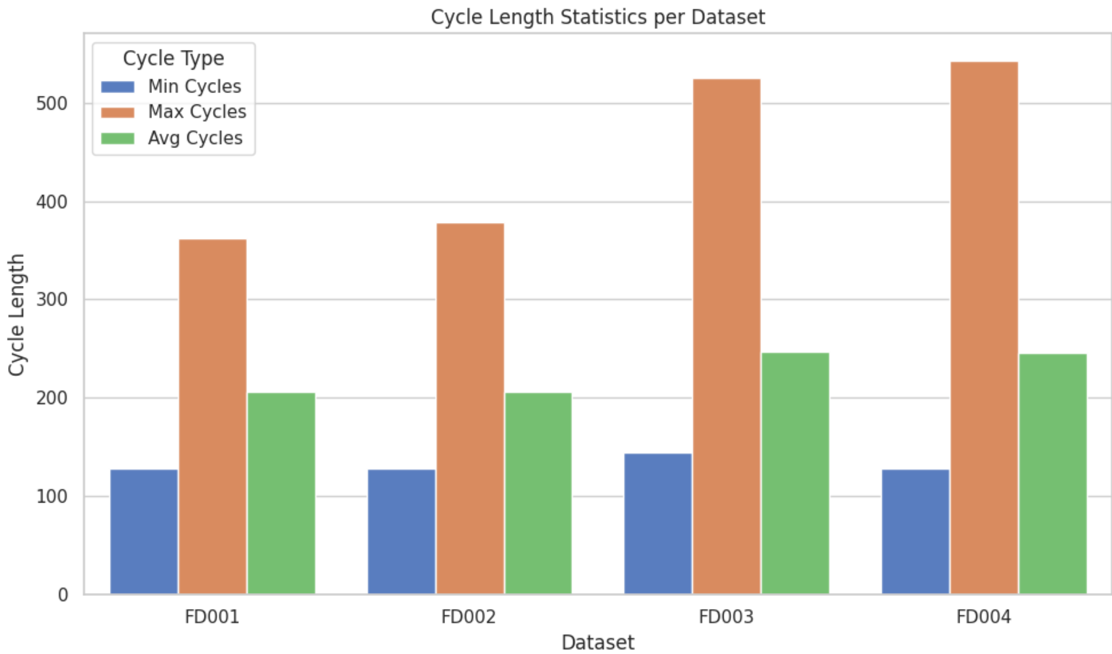

Figure 3 – Cycle Length Statistics per Dataset

It is possible to also check the Cycle distribution chart presented on Figure 4 where on X-axis: *"Cycle"* is the cycle number and on the Y-axis: "Frequency" is the frequency (count) of those cycles in the dataset. The purpose of this chart is to observe how frequently each cycle appears and so having a sense of the typical length of an engine's run before failure.

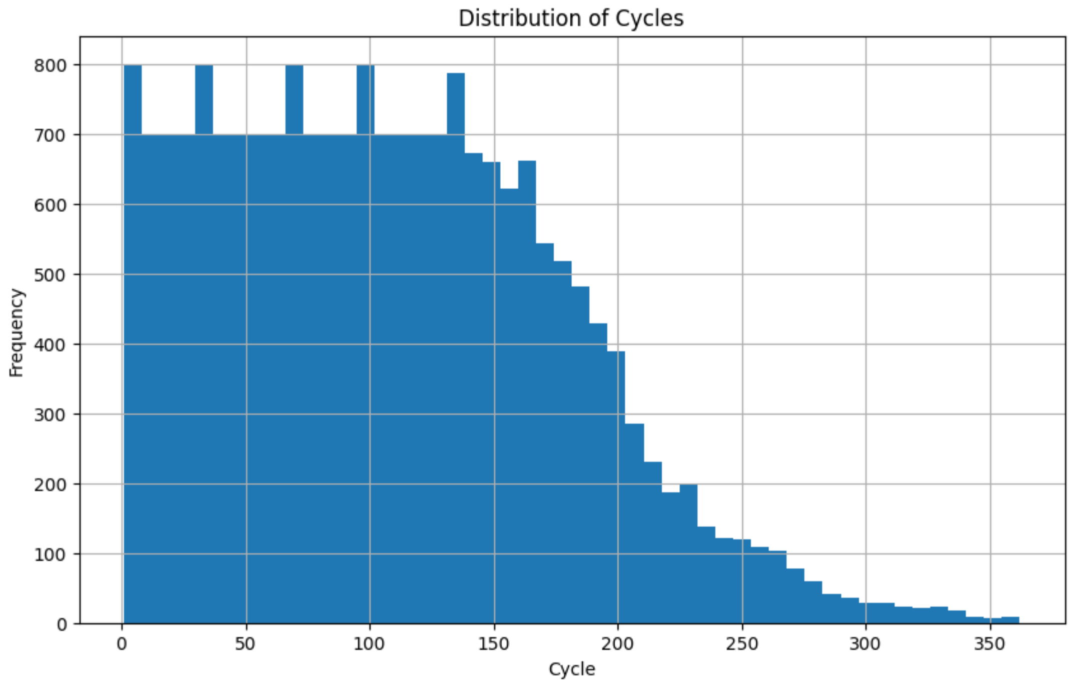

Figure 4 – Distribution of Cycles for FD001

Some observations made regarding the previous Figure 4 regarding cycle frequency is that there is a tendency for short cycles, meaning that the chart shows that cycles are heavily concentrated in the lower range (e.g., below 150), with the frequency gradually decreasing as the cycle count increases. This could indicate that many engines have shorter lifespans. This also could mean that most of the engines fail earlier, having fewer engines sustain a longer operational cycle.

To improve the dataset analysis a sensor plot was also used to present the sensor data available for each machine, since there are a lot of machines, Figure 5, shows the data only for machine id 1.

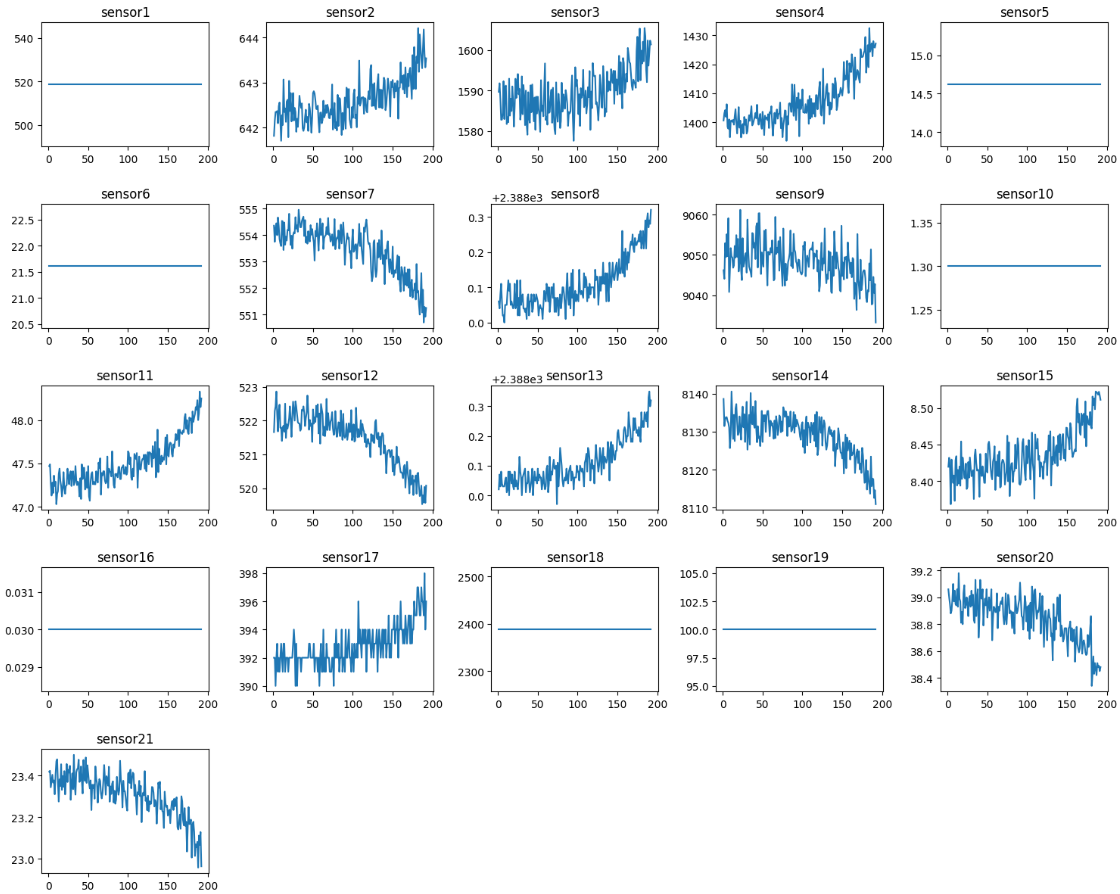

Figure 5 – Sensor data for machine 1 for FD001

After some consideration of the plots some sensors appear constant over all cycles, for example sensor 1 and 6, that could indicate that are less sensitive to the degradation process or that might not contribute much information for predicting Remaining Useful Life (RUL) and could be our first indication for removal during preprocessing phase. Contrary to the sensors 1 and 6 others clearly show trends, gradual increases and decreases, which can indicate potential relevance for modelling engine degradation. Examples of this are sensors like 3 and 12. This chart also can give information related to noise on some of the sensors as represented on sensor 9.

Having examined the individual behaviours of sensors over time, we now delve deeper into the interrelationships between sensors and operational settings, represented on Figure 6. By analysing correlations, we can identify potential redundancies, dependencies, or clusters among the variables, which can inform feature selection and model optimization.

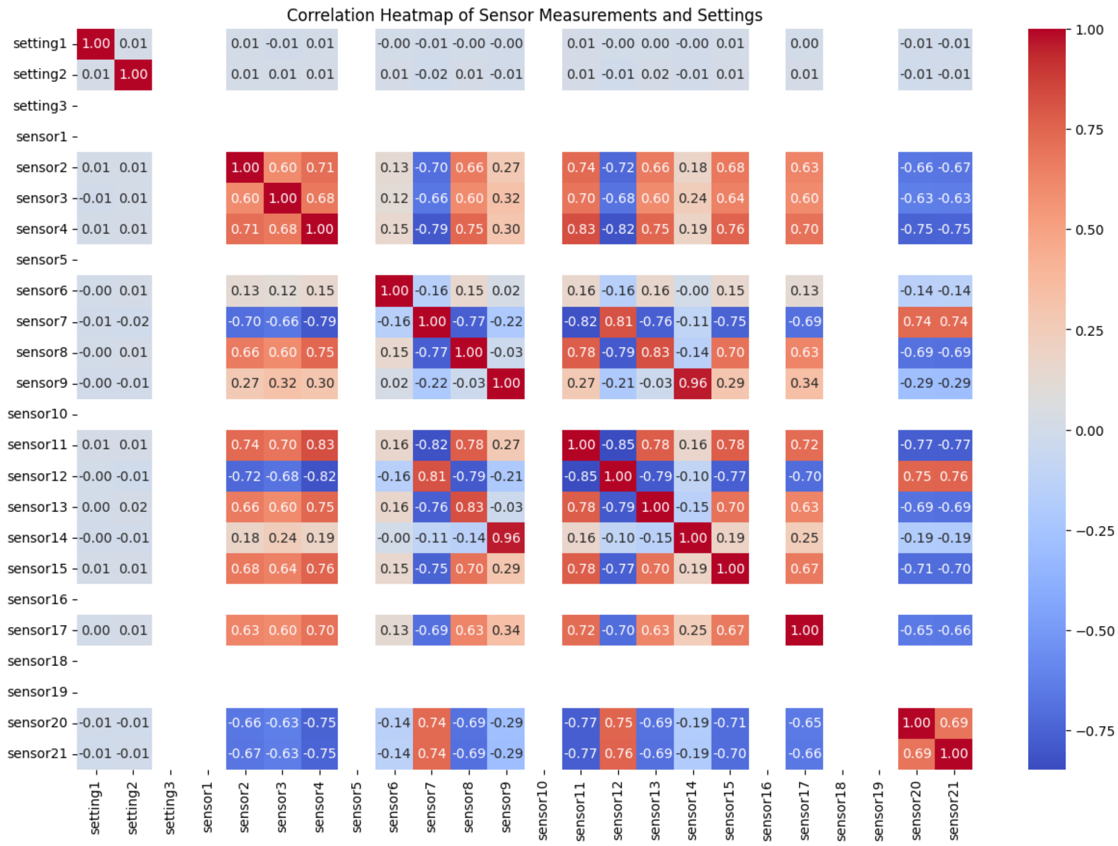

Figure 6 – Heatmap correlation for FD001

The correlation heatmap on Figure 6 highlights the relationships between operational settings and sensor measurements in the dataset. Several sensors exhibit strong positive correlations, such as Sensor 2, Sensor 3, and Sensor 4, suggesting potential redundancy. In contrast, the map also shows strongly negative correlations, such as between Sensor 7 and Sensor 11, point to inverse relationships that could also be relevant for predicting Remaining Useful Life (RUL). Operational settings show minimal correlation with sensor data, which could indicate a potential independence from system degradation trends.

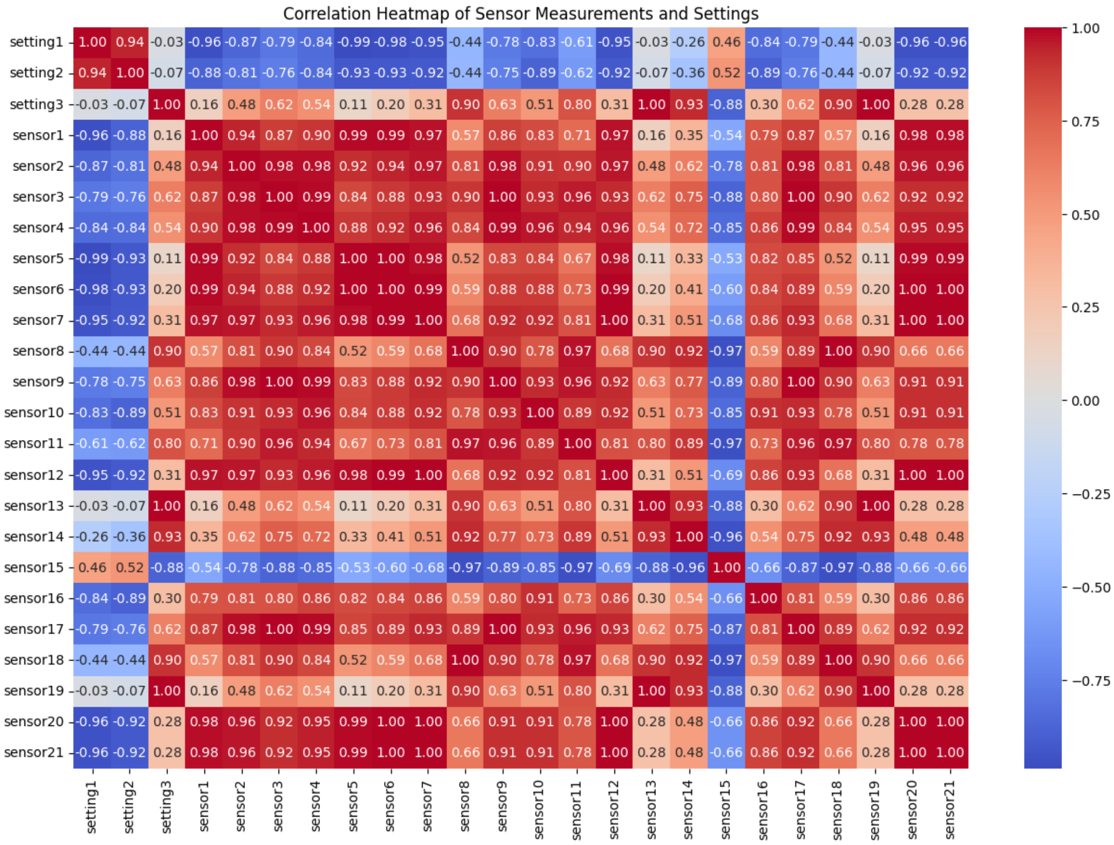

Figure 7 – Heatmap correlation for FD002

The correlation heatmap presented on Figure 7 reveals a denser structure of interdependencies when compared to FD001, with several variables showing almost perfect correlations. This suggests that this subset contains redundant information among certain sensors, which must be carefully considered during the feature selection process. Unlike FD001, the operational settings are not independent in FD002. Settings 1 and 2 show a strong positive correlation and present significant associations with several sensors. In contrast setting 3 displays weak correlations with most features, thus may provide complementary information.

It is also worth noting that FD003 resembles the correlation structure of FD001, while FD004 follows the same behaviour observed in FD002, with dense interdependencies across variables.

Understanding the distribution of RUL across the dataset is critical to assess its variability, and so the Figure 8 presents the distribution of Remaining Useful Life for this dataset.

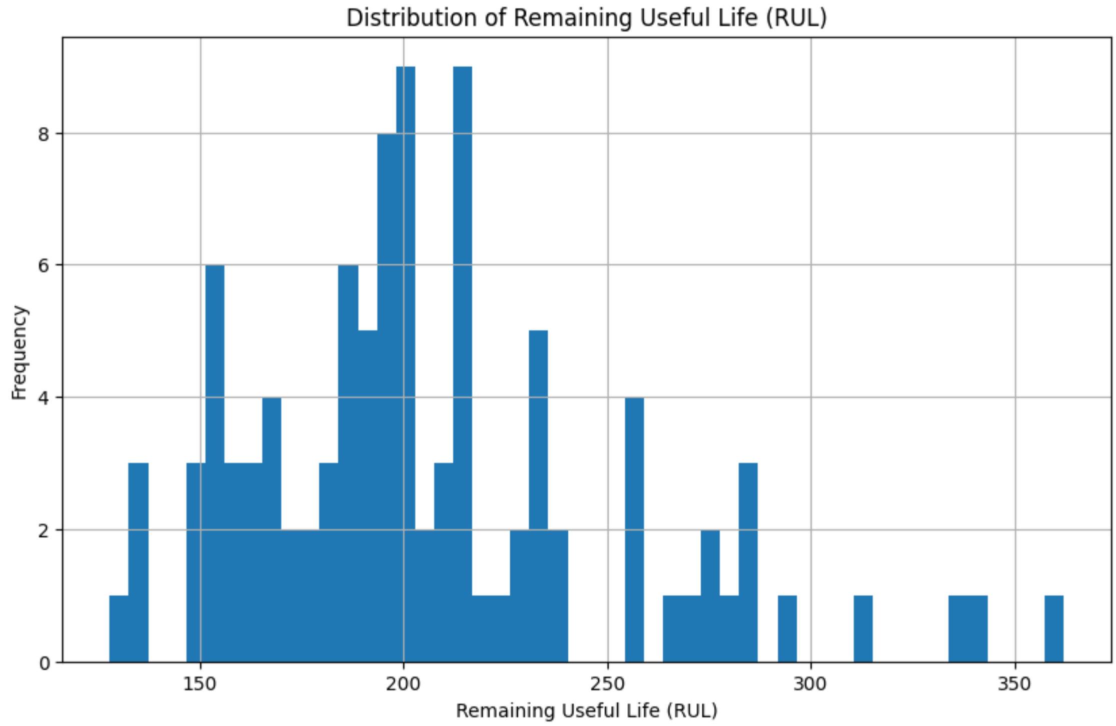

Figure 8 – RUL distribution for FD001

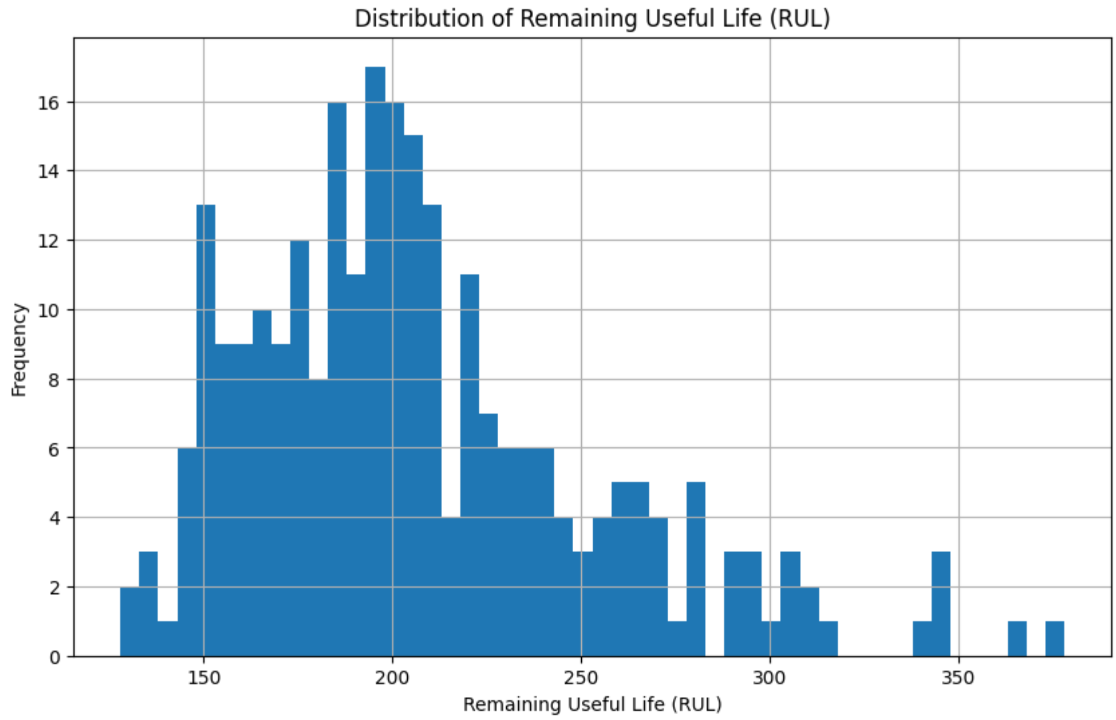

Figure 9 – RUL distribution for FD002

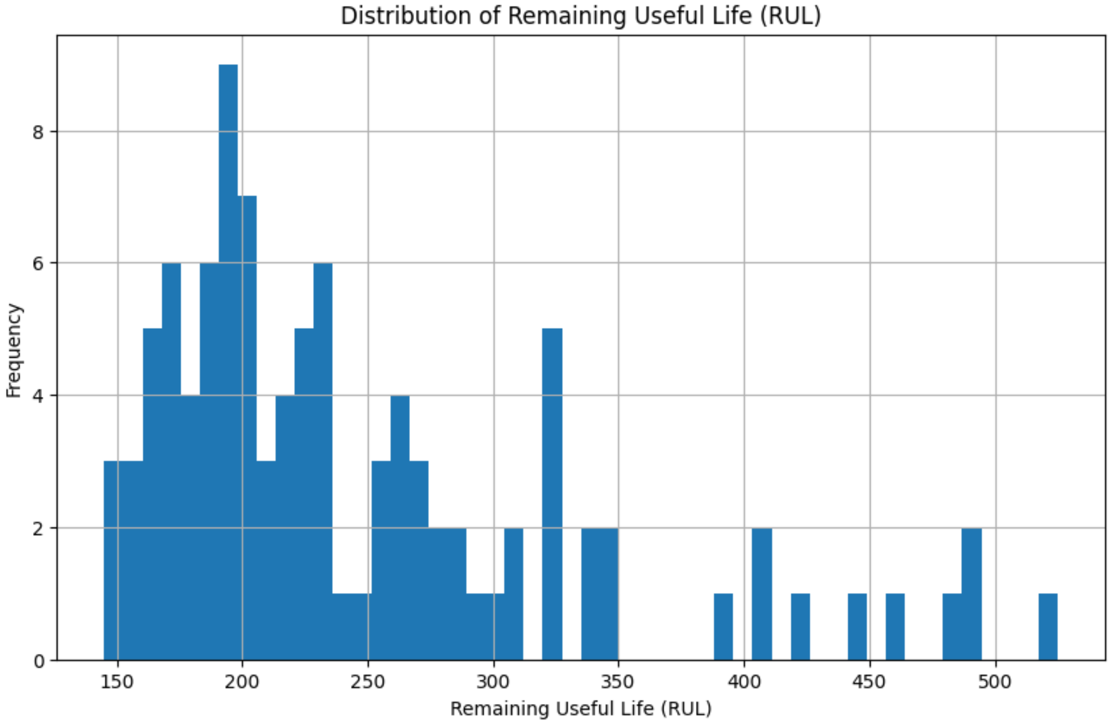

Figure 10 – RUL distribution for FD003

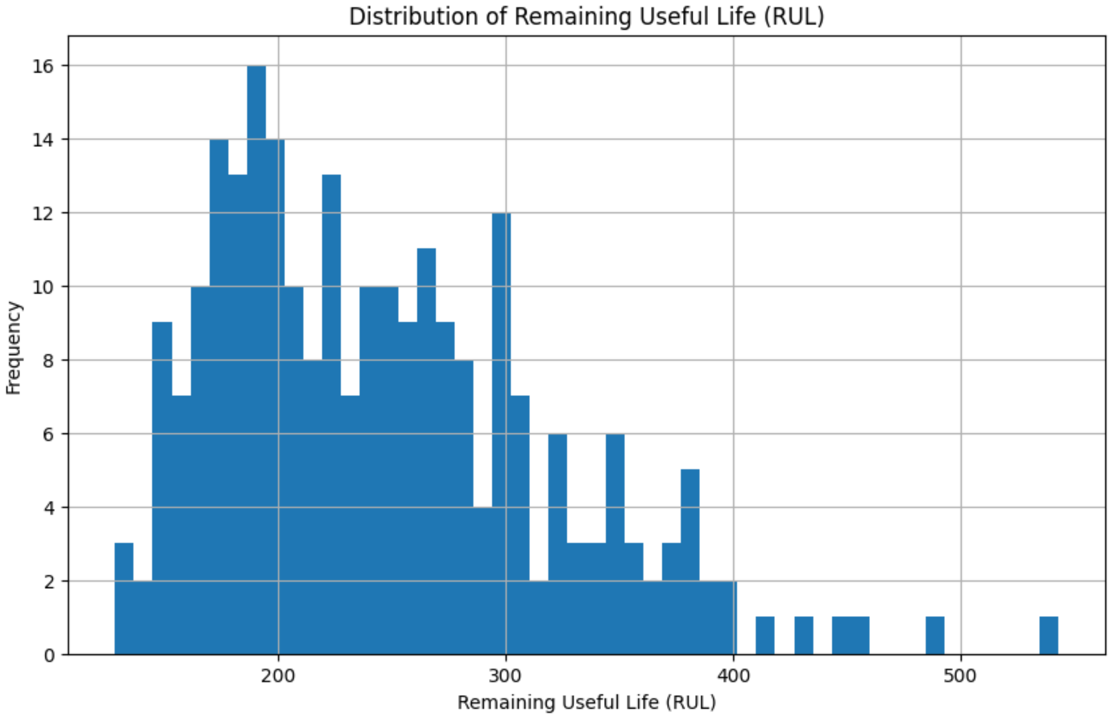

Figure 11 – RUL distribution for FD004

Regarding Figure 8, most engines have a RUL between 150 and 250 cycles, outliers such as the ones that go beyond 300 could represent a portion of the engines that operate with less wear.

It is possible to consider other speculations after reading the chart regarding class imbalance, meaning that, the higher frequency RUL values could dominate the model training, making it more difficult to predict low frequency values for cycles above 300. In FD002 presented on Figure 9, while the bulk of observations still lies below 300 cycles, the distribution is more heterogeneous that FD001, this could be an effect of the multiple operating conditions. Regarding FD003 on Figure 10 which the majority of RUL values remain between 150-300 cycles, presents multiple peaks that could be an indicator of distinct failure behaviours. This can complicate the learning process, as models must separate fault specific patterns within the same operational environment. Lastly, in Figure 11, referent to FD004 shows the broadest and most irregular of all distributions with values ranging from 150 to 500 cycles. This dispersion makes this subset the most challenging, highlighting both class imbalance and variability.

3.3 Data preprocessing

To advance this project, it was necessary to convert these raw CMAPSS sensor datasets to be used in the training phase. This section is intended to detail the steps taken to prepare all four subsets (FD001 to FD004) ready to be used on models such as LSTM and Transformer networks.

Starting with the dataset structure and predefined splits that characterize this CMAPSS dataset, since it already comes divided into predefined training and testing sets. As mentioned previously, each subset contains a training file, a testing file and a RUL file that is associated with the test file. This structure tries to reflect a realistic industrial scenario where a predictive model must estimate the RUL of machines that are still operating. It is a standard setup found in literature for training and evaluating supervised learning models for prognostics[51], [101].

```
    train_data = pd.read_csv(f'/content/drive/MyDrive/Python/predictive-
maintenance-
main/datasets/cmapss/train_{dataset_name}.txt/train_{dataset_name}.txt',
delim_whitespace=True, header=None)
    test_data = pd.read_csv(f'/content/drive/MyDrive/Python/predictive-
maintenance-
main/datasets/cmapss/test_{dataset_name}.txt/test_{dataset_name}.txt',
delim_whitespace=True, header=None)
    rul_data = pd.read_csv(f'/content/drive/MyDrive/Python/predictive-
maintenance-
main/datasets/cmapss/RUL_{dataset_name}.txt/RUL_{dataset_name}.txt',
delim_whitespace=True, header=None)
```

One thing this dataset does not have is an included validation split, so in this study a part of the training data was manually set aside for validation purposes during model development. This will be later used for applying early stopping.

```
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train,
test_size=0.2, random_state=42)
```

This dataset, as mentioned before, has 21 sensors and 3 operational settings but as verified on the 3.2 chapter not all features of this dataset are helpful, some of them remain constant or display noisy behavior [39], [70], [82]. In this work two approaches were considered to understand the impact that the features selected have on the result. First, the implementation of the feature selection by low variance and high correlation.

```
features = globals()[train_data_var].drop(columns=['id', 'cycle'])

selector = VarianceThreshold(threshold=0.001)
selector.fit(features)
```

This way sensor data that does not present any changes throughout the sequence of cycles is considered not important for our study. On the same concept it was also analyzed the correlation between the sensors.

```
corr = filtered_df.corr().abs()
upper = corr.where(np.triu(np.ones(corr.shape), k=1).astype(bool))
high_corr_features = [column for column in upper.columns if
any(upper[column] > 0.95)]
```

The result of this code for all subsets of this dataset is presented on the next Table 2.

Table 2 – Feature selection for low variance and high correlation method

Subset	Selected Features	Dropped due to low variance	Dropped due to high correlation
FD001	['sensor2', 'sensor3', 'sensor4', 'sensor7', 'sensor8', 'sensor9', 'sensor11', 'sensor12', 'sensor13', 'sensor15', 'sensor17', 'sensor20', 'sensor21']	['setting1', 'setting2', 'setting3', 'sensor1', 'sensor5', 'sensor6', 'sensor10', 'sensor16', 'sensor18', 'sensor19']	sensor14
FD002	['setting1', 'setting2', 'setting3', 'sensor2', 'sensor8', 'sensor14']	['sensor16']	sensor1, sensor3, sensor4, sensor5, sensor6, sensor7, sensor9, sensor10, sensor11, sensor12, sensor13, sensor15, sensor17, sensor18, sensor19, sensor20, sensor21
FD003	['sensor2', 'sensor3', 'sensor4', 'sensor7', 'sensor8', 'sensor9', 'sensor11', 'sensor15', 'sensor17', 'sensor20', 'sensor21']	['setting1', 'setting2', 'setting3', 'sensor1', 'sensor5', 'sensor6', 'sensor10', 'sensor16', 'sensor18', 'sensor19']	sensor12, sensor13, sensor14

FD004	['setting1', 'setting2', 'setting3', 'sensor2', 'sensor8', 'sensor14']	['sensor16']	sensor1, sensor3, sensor4, sensor5, sensor6, sensor7, sensor9, sensor10, sensor11, sensor12, sensor13, sensor15, sensor17, sensor18, sensor19, sensor20, sensor21

Table 2 contains the results of applying the low variance and high correlation filter. The outcome show that many sensors were discarded because they either had little variation along the dataset or carried redundant information already contained in other sensors. Keeping such features would only add noise and increase the risk of overfitting without bringing real value to the prediction task[70], [88], [92]. By focusing on focusing on the subset of sensors that present meaningful changes relative to the engine condition. The reduced input space also improves training efficiency and helps the model to converge more easily, while at the same time making the interpretation of the results more straightforward [83], [88].

The other method to select features is one that was found during the state-of-the-art research, the Boruta method [39]. The output of this method is the group of features, in this case the sensors and settings that through comparison with other random group of features from the dataset have more importance. The implementation for this method used a random forest to then pass the Boruta method.

```
forest = RandomForestRegressor(n_estimators=100, random_state=42, n_jobs=-
1)
  boruta_selector = BorutaPy(forest, n_estimators='auto', random_state=42)
```

The result of this implementation is presented on the next Table 3

Table 3 – Feature selection for Boruta method

Subset	Selected Features
FD001	['sensor2', 'sensor3', 'sensor4', 'sensor7', 'sensor9', 'sensor11', 'sensor12', 'sensor14', 'sensor15', 'sensor20', 'sensor21']
FD002	['sensor2', 'sensor3', 'sensor4', 'sensor7', 'sensor8', 'sensor9', 'sensor11', 'sensor12', 'sensor13', 'sensor14', 'sensor15', 'sensor21']
FD003	['sensor2', 'sensor3', 'sensor4', 'sensor6', 'sensor7', 'sensor8', 'sensor9', 'sensor11', 'sensor12', 'sensor13', 'sensor14', 'sensor15', 'sensor21']
FD004	['sensor2', 'sensor3', 'sensor4', 'sensor6', 'sensor7', 'sensor8', 'sensor9', 'sensor11', 'sensor12', 'sensor13', 'sensor14', 'sensor15', 'sensor21']

The Table 3 shows the set of features selected by the Boruta algorithm. Unlike the correlation method, Boruta uses the importance score derived from a random forest to evaluate each sensor contribution. This allows it to capture non-linear and more complex relationships between sensors and RUL target, which are not visible through simple correlation analysis [39]. The resulting feature set is larger, preserving more sensors that may hold relevant information in subtle ways. While this means the models will have to process more data, it also increases the chance of retaining signals that reflect hidden degradation patterns. In practice, this makes Boruta particularly useful for subsets of the dataset with multiple operating conditions, where the interactions between sensors are harder to separate. The broader selection obtained here highlights its role as a more conservative yet also more flexible method for feature selection, complementing the stricter filtering done by the correlation method.

In the end of this step the results of these two approaches are different so for this study both contributions will be tested to see their impact on the model training.

For this data preprocessing stage, it is still missing the RUL labelling. To accomplish this a new feature was added to the dataset, the Remaining Useful Life (RUL). The state-of-the-art research showed two different approaches for this calculation. The most used on is considering that the engine degradation starts on the first cycle and it is linear, on the other end another paper proposed that the degradation only starts after some cycles [84], [85], [98], [99]. The first option, the one that considers the degradation starts right on the first cycle, was calculated for each engine by subtracting the current cycle from the maximum cycle for that engine. This transformed the dataset into a supervised learning format with RUL as the target variable. The second option, also called Piece-Wise was also implemented on this work to compare the results in the end.

```
if use_piecewise_rul:
        train_data['RUL'] = train_data['RUL'].clip(upper=max_rul_cap)
        test_data['RUL'] = test_data['RUL'].clip(upper=max_rul_cap)
```

It was used a 125 cycle RUL cap mainly because of two reasons. The first being that the data distribution related to engine lifetime is not well balanced. There are few engines with long survival times, having a small contribution on the learning process [84], [86]. This information is visible on Figure 4 where it is visible a clear drop on the quantity of engines that can surpass 120 cycles.

To ensure all sensor readings are on a comparable scale, min-max normalization was applied to scale the features to a range of [0, 1]. This prevents sensors with large numerical ranges from dominating the training process.[70], [82]

To prepare the dataset for sequenced based models like LSTMs and Transformers it was necessary to transform the time series data into overlapping sequences of fixed length[85], [86]. Given that each engines follows a different degradation trajectory, the data was structured in a way such that each sequence represents a sliding window of sensor readings up to a predefined time step. Specifically, for each engine instance, a sequence of length T where T represents the number of past time steps considered for predicting the Remaining useful life. The target value for each sequence is the RUL at the last time step of that sequence[85], [86]. For this study the length used was 50 and 60 so that it is possible to check the impact of sequence length on the results since not all studies, from the research conducted, agree on a length.

3.4 Implementation of LSTM

This chapter details the implementation of the deep learning models used to predict the remaining useful life of aircraft engines in the NASA CMAPSS dataset. The implementation covers two approaches: Long short-term memory networks and Transformer based architecture, discussing their architectures, training procedures and evaluation metrics.

3.4.1 Model architecture

The LSTM architecture was selected due to its effectiveness in modelling sequential and time-dependent data, such as those found in condition monitoring and prognostics tasks [82], [84], [87]

The architecture of this model consists of:

- First LSTM Layer: With 100 units returning the sequences used on the following LSTM layer. This layer processes the time series input allowing the model to capture the temporal dynamics of the sensors throughout the multiple cycles in the sequence.
- Second LSTM layer: With 50 units returning a single vector representation of the sequence.
- Dropout layer: With a dropout rate of 0.2 after each LSTM layer to reduce the risk of overfitting [104]. This also helps to improve the model generalization.
- Dense layer: With a single unit to output the predicted RUL, considering this task as a regression problem.

Model implementation on Colab.

```
 model = Sequential()
    model.add(LSTM(lstm_units[0], return_sequences=True,
input_shape=(seq_length, len(useful_sensor_cols))))
    model.add(Dropout(dropout_rate))
    model.add(LSTM(lstm_units[1]))
    model.add(Dropout(dropout_rate))
    model.add(Dense(1))
```

Following the experiments conducted by previous work mentioned on the literature this was the general structure of the model used, the quantity of units of each layer was subject of research also to understand what values to use. During model development the number of units was gradually increased [84], [87], [105], [106].

Furthermore, the model was implemented using the TensorFlow/Keras using the sequential API allowing to stack layers linearly according to the proposed design.

3.4.2 Training Evaluation

The training process was carried out using Google Colab free version, which provides a practical cloud-based development environment. While Colab has this practicality and ease of use, it has some limitations regarding RAM, only having 12GB of RAM available, restricted storage and idle time disconnections. So, it was necessary to adapt the training pipeline to work around these limitations. This model was compiled using the Adam optimizer and the mean squared error (MSE) loss function[107].

The model was trained using the following hyperparameters:

- **Optimizer:** Adam with a learning rate of 0.001.
- **Loss Function:** Mean Squared Error (MSE).
- **Batch Size:** 64.
- **Epochs:** 50.

Model fit in colab

```
history = model.fit(X_train,
                    y_train,
                    validation_data=(X_val, y_val),
                    batch_size=batch_size,
                    epochs=epochs,
                    verbose=0)
```

Since there was a lot of variables to test and update on the code it was created an array of configuration to more easily execute and get the results for the different scenarios. Here is an example of one configuration iteration.

```
{
        "dataset_name": "FD001",
        "scaler_type": "minmax",
        "seq_length": 50,
        "use_piecewise_rul": True,
        "max_rul_cap": 125,
        "feature_set": FD001_usefulfeatures_correlation,
        "batch_size": 64,
        "epochs": 50,
        "lstm_units": [100, 50],
        "dropout_rate": 0.2,
        "loss_function": "mse",
        "optimizer": "adam",
        "learning_rate": 0.001
}
```

3.5 Transformer Implementation

3.5.1 Model Architecture

The Transformer-based model follows the architecture presented by Vaswani [97]. It is a sequence-to-sequence architecture that consists of an encoder and a decoder. The encoder takes the input sequence and maps it into a higher dimensional vector, which is then fed into the decoder to generate an output sequence[97], [98]. For the implementation of this project, it is used the contributions of [98], [99], [100], [108] where the transformer architecture presented did not use the decoder part, so it is a transformer using only an encoder layer, as represented on Figure 12.

The implementation was carried out using the TensorFlow Keras Functional API, used to define custom sublayers such as multi-head attention blocks and layer normalization operations.

Key components of the model include:

- Input embeddings basically are the input sequences configured as a sliding window of the sensor readings, that are fed into the model.
- Multi head Attention, the core of this transformer is the self-attention mechanism, which enables to attend to different time steps in the input sequence, capturing temporal dependencies.

```
attn_output = layers.MultiHeadAttention(key_dim = head_size, num_heads =
num_heads, dropout = dropout)(inputs, inputs)
```

- Add and Norm to follow the sublayer in the original transformer to help with model stability during training.

```
x = layers.LayerNormalization(epsilon=1e-6)(inputs + attn_output)
```

- Feed forward network, was implemented using the Conv1D layers increasing the model representational power while preserving the sequence structure.

```
ff_output = layers.Conv1D(filters=ff_dim, kernel_size=1,
activation="relu")(x)
ff_output = layers.Conv1D(filters=inputs.shape[-1],
kernel_size=1)(ff_output)
```

- Second Add and Norm

```
x = layers.LayerNormalization(epsilon=1e-6)(x + ff_output)
```

- Global Pooling and output layer, will collapse the sequence dimention into a single vector representation, which is then passed to a dense layer for RUL prediction.

```
x = layers.GlobalAveragePooling1D()(x)
```

This encoder structure is then repeated creating a model with two encoder layers. This number is configurable too. Both the number of encoder blocks and the internal dimensions (head_size, ff_dim) were defined as configurable parameters, allowing experimentation without modifying the model structure manually.

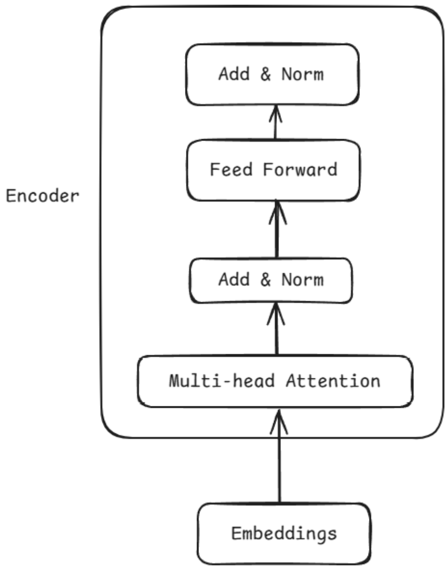

Figure 12 – Encoder architecture

3.5.2 Training Evaluation

The Transformer model was also trained on the Google Colab free version. The model was compiled using the Keras API with the following configuration:

```
model.compile(loss="mse",optimizer=keras.optimizers.Adam(learning_rate=1e-
4), metrics=["mae"])
```

This model was compiled using the Adam optimizer and the mean squared error (MSE) loss function.

The model was trained using the following hyperparameters:

- **Optimizer:** Adam with a learning rate of 0.001.
- **Loss Function:** Mean Squared Error (MSE).
- **Batch Size:** 64.
- **Epochs:** 100.

4 Results

In this chapter it is intended to explore and present the results of the experiments conducted for different models for the RUL prediction using the NASA CMAPSS FD001 dataset. The models implemented include a Long short term memory network (LSTM), and a Transformer. Their effectiveness is analysed based on key evaluation metrics.

4.1 Evaluation metrics

Following the same metrics used on literature, for this work were used the root mean square error (RMSE) and mean absolute error (MAE).

- Root Mean Square Error – Smaller values for the RMSE indicate that model predictions are closer to true values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

  Where y_i is the actual value and $\hat{y}_i$ is the predicted value.

- Mean Absolute Error – Measures the average of the absolute difference between the predictions and true values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

4.2 Model performance comparison

In this section it will be presented the results of each implementation, LSTM and Transformer, using the metrics presented previously and comparing the results between both approaches along the four subsets as well as making a comparison with state of the art results.

4.2.1 LSTM Results

Using what was previously described the trained model was evaluated using **Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE),** and the results are presented on the following Table 4.

Table 4 – Training results LSTM

Subset	Sequence Length	Piecewise Rul	Feature Set	MAE	RMSE
FD001	**50**	**True**	**Corr**	**9.14**	**12.57**
FD001	50	False	Corr	22.25	31.01
FD001	60	True	Corr	9.59	12.89
FD001	60	False	Corr	20.57	30.80
FD001	50	True	Boruta	10.45	13.63
FD001	50	False	Boruta	22.42	32.00
FD001	60	True	Boruta	11.33	14.46
FD001	60	False	Boruta	21.16	30.19
FD002	50	True	Corr	35.97	39.60
FD002	50	False	Corr	51.41	65.50
FD002	60	True	Corr	36.15	40.20
FD002	60	False	Corr	50.23	64.29
FD002	50	True	Boruta	36.53	40.23
FD002	50	False	Boruta	37.64	48.90
FD002	**60**	**True**	**Boruta**	**23.81**	**27.18**
FD002	60	False	Boruta	51.56	66.01
FD003	50	True	Corr	7.53	11.70
FD003	50	False	Corr	35.18	50.77
FD003	60	True	Corr	7.24	11.24
FD003	60	False	Corr	32.47	46.64
FD003	50	True	Boruta	6.92	10.98
FD003	50	False	Boruta	33.44	47.60
FD003	**60**	**True**	**Boruta**	**6.63**	**10.63**
FD003	60	False	Boruta	32.14	46.03
FD004	50	True	Corr	34.00	36.21
FD004	50	False	Corr	75.94	101.31
FD004	60	True	Corr	26.69	29.48
FD004	60	False	Corr	75.79	101.03
FD004	50	True	Boruta	12.03	17.19
FD004	50	False	Boruta	49.12	68.41

FD004	**60**	**True**	**Boruta**	**11.40**	**15.83**
FD004	60	False	Boruta	57.55	79.26

In this Table 4 it is possible to verify the different results on the experiments conducted to check the impact of each scenario. This projected includes the variation of sequence length of 50 and 60, meaning that the sliding window which was used to create the sequences was tested with length of 50 and 60 respectively. Along with this sequence length variation it was also used the max cap RUL value of 125 [84], [86], if the piece wise RUL is true the max RUL cap was considered otherwise the normal RUL value was considered. Lastly, the feature set, as mentioned before both methods of feature selection would be tested.

For FD001, the results clearly show that applying the piecewise RUL improves prediction accuracy. With correlation features and sequence length of 50, the model reached an MAE of about 9.1 and RMSE of 12.6, which is the best result for this subset. When the linear RUL was used, so no maximum RUL value was used, the error values increased significantly with MAE above 20 and RMSE passing 30. This shows that the model tends to overestimate in the early cycles if the degradation phase is not highlighted. Differences between the sequence length of 50 and 60 were small, meaning that for this simple operation condition, the model is not very sensitive regarding window size. However, the increase of the window size translated in better performance metrics as well. The performance of the Boruta feature selection for this subset had a similar performance, so the sequence length of 50 and the piecewise RUL had the best values for MAE and RMSE, 10.45 13.63 respectively, although, when compared with the correlation selection had a worse performance.

Regarding FD002 subset, the scenario presented himself more challenging. It is possible to identify that when it is not applied the piecewise RUL both feature sets gave high errors, with MAE values above 50 and RMSE above 64 for the correlation feature selection and MAE above 51 and RMSE above 66. However, the combination of piecewise RUL with the Boruta feature selection and the sequence length of 60 led to a clear improvement, reducing the MAE to around 23.8 and the RMSE to 27.2. This shows that for more complex scenarios with multiple operating conditions, both feature selection and longer windows are necessary to capture the variability of the degradation patterns.

For FD003, the LSTM model achieved a good performance. With piecewise RUL calculation and Boruta Features at 60 cycles, the model reached MAE of 6.63 and RMSE of 10.63, overall, the best of all 4 Subsets. Correlation features also produced solid results, with RMSE values just above 11. The difference between piecewise and non-piecewise setups is again clear, as removing the cap on RUL increases the RMSE to over 46. These results suggest that the LSTM can capture degradation trends effectively even when multiple fault modes are present, proving the operation condition stable.

Finaly, FD004 is the most complex scenario, and the results reflect the difficulty. Correlation features gave RMSE around 29.48 when using piecewise RUL calculation and window size of 60 while Boruta features reduced it to RMSE of 15.83. However, without piecewise RUL, the errors increased, with RMSE values above 100. It is also worth mentioning that Boruta features proved much more effective than correlation features in this subset, showing that richer features selection plays a bigger role when both conditions and fault modes vary simultaneously.

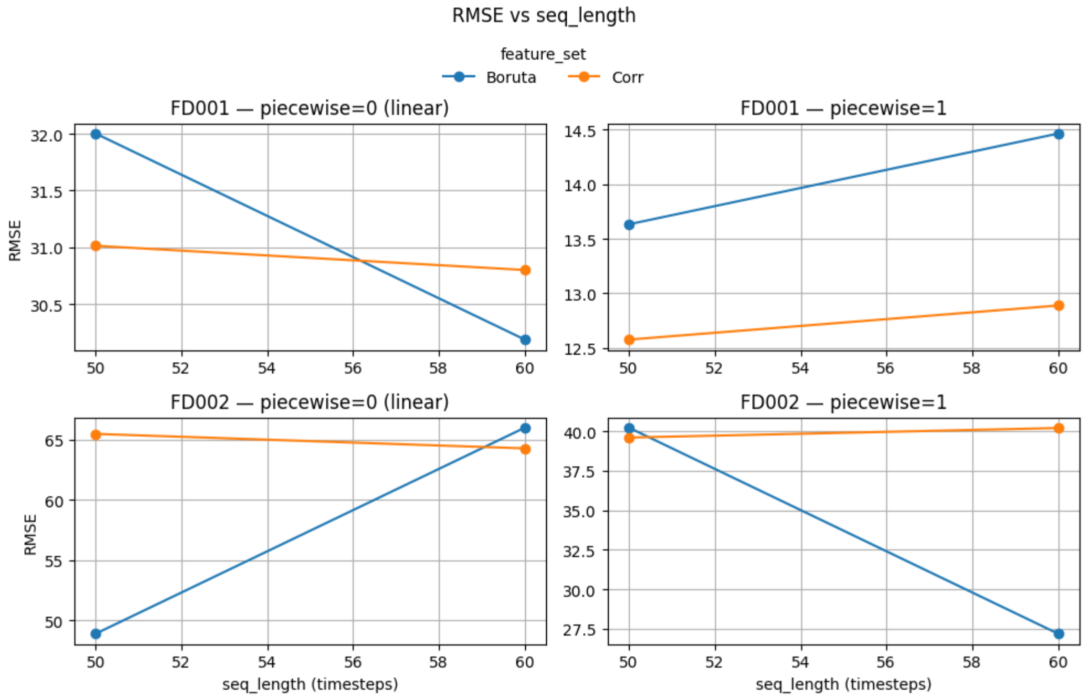

Figure 13 – RMSE Variation

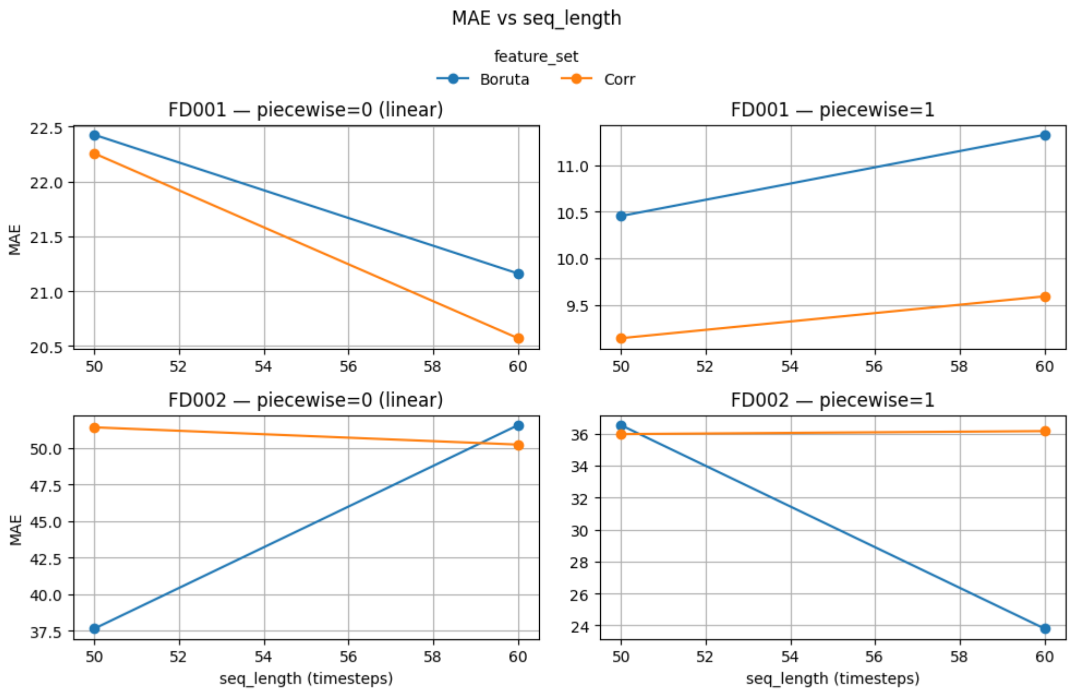

Figure 14 – MAE Variation

The information collected provided the two plots presented on Figure 13 and Figure 14. Starting with FD001, the plots show that when a linear RUL is used, both feature sets lead to noticeably higher errors. The RMSE values remain above 30 and 20 for MAE regardless of the window size. It is possible to verify that there is a slight downward trend when the window is increased from 50 to 60, but it is not much of an improvement. Once the piecewise RUL is applied, the errors drop. The correlation feature selection outperforms the Boruta selection, keeping MAE close to 10 and RMSE between 12 and 13. This stability across window sizes suggests that, under single operation condition, the model can already capture the degradation dynamics without needing longer input sequences. This can be evidence that the biggest factor for the FD001 scenario is the RUL formulation.

For FD002, as expected since it is a different scenario the plots show a different result. Under linear RUL setup the values for both MAE and RMSE remain high with Boruta showing more sensitivity to sequence length. In fact, the results for the Boruta have a delta of 26 for MAE and 13.92 for RMSE. However, when it is used the piecewise RUL calculation the Boruta line drops when the sequence length increases. The correlation selection does not present much change with the sequence length variation. This scenario faced with FD002, with multiple operation conditions, benefits from both longer historical context and a more refined feature set. The plots show that Boruta selection with a window size of 60 with piecewise formulation is the only configuration where the curve clearly breaks away from the higher error region.

Table 5 – RMSE Results

Model/Reference	FD001	FD002	FD003	FD004
This work (Best configs)	12.57	27.18	10.63	15.83
LSTM[91]	15.22	--	--	--
LSTM-MLP-OR[90]	14.24	12.00	17.27	14.94
LSTM-MLP-ORE[90]	13.20	12.77	13.84	13.64
LSTM[89]	17.60	29.67	17.62	31.84

Table 5 contains the best configuration results from this work and the other mentioned on the literature review chapter. Other works will be mentioned, however the authors do not specify in which subset the results are achieved so, for that reason they are not presented on the table.

The results observed in this work are consistent with what has been reported in the literature. In the Deep LSTM work [84], the authors already noted that performance is noticeably stronger in FD001, which is a scenario that contains only one operating condition and one fault mode, while the performance decreases in FD002 due to the added variability from multiple operating regimes. This same pattern is reported in this work where our model reaches very low values in FD001 but has difficulties reaching comparable levels in FD002. The authors from [87] with their LSTM encoder approach, highlighted how the choice of RUL formulation and preprocessing strategies can make or break the results. It was reported on this work that using capped RUL stabilizes the model's predictions across different scenarios. This verification is also aligned with the strong improvement observed when applying the piecewise RUL, which reduced error in both FD001 and FD002 when compared to the linear alternative. Other works such as [82], [83] have looked more closely at model hyperparameters. The first, which reports the application of a Bi-directional LSTM to the same dataset and underlined the importance of capturing longer temporal dependencies. The second work compared a range of machine learning models and pointed out that longer input sequences and careful choice of features improved the results under variable operating conditions. This information goes along with our findings regarding the improvement on FD002 when the window size is increased combined with the feature selection from Boruta. In addition, Tippareddy [89] applied a BiLSTM model reporting a RMSE values of 17.60 on FD001, 29.67 on FD002, 17.63 on FD003 and 31.84 on FD004. Compared to our results, their FD001 and FD003 values and the ones observed on this work are very close, indicating that LSTMs handle single operation condition subsets very effectively. However, the results for FD002 and FD004 show that a careful feature selection and preprocessing methods, can lead to noticeable improvements under multi condition scenarios. Noot et al. [90] investigated the impact of censored data which our work didn't, considering that their LSTM maintained stability under different scenarios that could be and improvement to consider on this work. Finally, Syuhada [91] focused only on FD001 confirming that the application of piecewise RUL calculation is a good strategy.

In short, the LSTM model proved to be effective for FD001, reaching errors in line with state of the art results. The experiments also showed the importance of piecewise RUL calculation and feature selection mechanism to control prediction errors. For FD002 the best configuration, which was piecewise RUL plus Boruta and window size of 60, gave a much stronger performance compared to baseline setups. In FD003 the model reached its lowest error values overall, this fact helped reinforce its ability to capture degradation trends under single condition settings with multiple fault modes. Finally, FD004, results were less competitive, reflecting the difficulty of modelling multiple conditions and multiple fault modes with a standard LSTM. These findings highlight the strengths of LSTM for stable operation conditions and their limitations in more complex environment.

4.2.2 Transformer Results

For the analysis of the transformers results it is also used the MAE and RMSE metric values, that are presented on the Table 6. Like the LSTM training and results four factors were varied. The sequence length of 50 and 60, the use or not of piecewise RUL calculation and two different feature selection strategies.

For the subset FD001, the best results were consistently obtained when using the piecewise RUL calculation strategy. This paired with a sequence length of 60 and correlation features, the transformer achieved a MAE value of 10.34 and RMSE of 14.35. In contrast, when the same setup was applied without piecewise calculation, the errors increased to MAE of 23.02 and RMSE of 33.21. This pattern was repeated along all configurations, regardless of the feature selection method, the absence of piecewise RUL calculation resulted in at least a twofold increase in error values. Boruta features produced similar results with the lowest MAE of 10.92 and RMSE of 14.4.

Table 6 – Training results Transformer

Subset	**Sequence Length**	**Piecewise Rul**	**Feature Set**	**MAE**	**RMSE**
FD001	50	True	Corr	10.86	14.56
FD001	50	False	Corr	27.21	37.93
FD001	**60**	**True**	**Corr**	**10.34**	**14.35**
FD001	60	False	Corr	23.02	33.21
FD001	50	True	Boruta	11.13	14.72
FD001	50	False	Boruta	26.60	35.80
FD001	60	True	Boruta	10.92	14.40
FD001	60	False	Boruta	23.91	33.03
FD002	50	True	Corr	21.73	26.64
FD002	50	False	Corr	40.27	52.79
FD002	60	True	Corr	23.04	28.16
FD002	60	False	Corr	35.37	45.29
FD002	50	True	Boruta	20.64	25.02
FD002	50	False	Boruta	35.53	45.94
FD002	**60**	**True**	**Boruta**	**15.61**	**20.02**
FD002	60	False	Boruta	31.25	40.64
FD003	**50**	**True**	**Corr**	**7.43**	**12.42**
FD003	50	False	Corr	44.69	59.8
FD003	60	True	Corr	7.83	12.69
FD003	60	False	Corr	42.83	58.52
FD003	50	True	Boruta	8.13	13.37
FD003	50	False	Boruta	43.46	58.57
FD003	60	True	Boruta	8.60	12.86
FD003	60	False	Boruta	41.36	57.45
FD004	50	True	Corr	15.83	23.03
FD004	50	False	Corr	57.65	77.07
FD004	60	True	Corr	14.36	21.46
FD004	60	False	Corr	50.99	69.47
FD004	**50**	**True**	**Boruta**	**12.03**	**19.08**
FD004	50	False	Boruta	56.36	76.39
FD004	60	True	Boruta	12.20	19.34
FD004	60	False	Boruta	49.73	68.24

For FD002, the same effect was observed, though the performance gap was even more noticeable. The lowest error values were obtained with Boruta features and a sequence of 60 where the transformer reached MAE of 15.61 and RMSE 20.02. Without piecewise, the performance of the same configuration decreased significantly, with MAE rising to 31.25 and RMSE to 40.64. The results regarding the correlation feature selection followed this same pattern, MAE and RMSE values increased when piecewise was not used. To help our analysis of the results the plots presented on Figure 15 and Figure 16 show the impact of the different test parameters used on RMSE and MAE for FD001 and FD002 subsets.

Analysing the FD003, the transformer here also achieved good results, with the lowest error observed using piecewise RUL and correlation features with a window size of 50 achieving MAE of 7.43 and RMSE of 12.42. Boruta features performed slightly worse in this case with RMSE value of 12.86. In contrast, when the linear RUL was applied, the RMSE surged to nearly to 60. These findings confirm the importance of piecewise formulation in capturing degradation trends, even when multiple fault modes are present in a stable operation environment.

Finaly looking for the results on the most complex scenario, that highlighted the model dependency on preprocessing choices. The best configuration was piecewise RUL calculation with Boruta feature selection with a window size of 60 timesteps. This configuration led to MAE value of 12.20 and RMSE of 19.34. Correlation feature selection method, while still effective, produced higher errors with RMSE values between 21 and 23. Without piecewise RUL, the model's accuracy got worse. This is visible due to the RMSE values between 68 and 77. The strong advantage of Boruta in this subset indicates that robust features selection is necessary for dealing with both multiple fault modes and operating conditions.

For FD001, the plots confirm a strong impact of the piecewise RUL calculation. In fact, both RMSE and MAE values present lower values when comparing with the linear RUL calculation method. Using only piecewise calculation the difference between the two methods for feature selection is small, however the correlation selection achieved slightly better results at 60 timesteps, with RMSE of 14,35 and MAE of 10,34. Without piecewise, the error values remained higher even when increasing the sequence length. For FD002, the analysis of the plots reveals the importance of both piecewise and Boruta methods. When piecewise was applied was applied, Boruta combined with longer sequences led to best overall performance with RMSE of 20,02 and MAE of 15,61. Analysing the linear RUL calculation, both Boruta and correlation methods showed a strong decline with sequence length, but errors remained high with the best scenario with RMSE above 40 and MAE above 30.

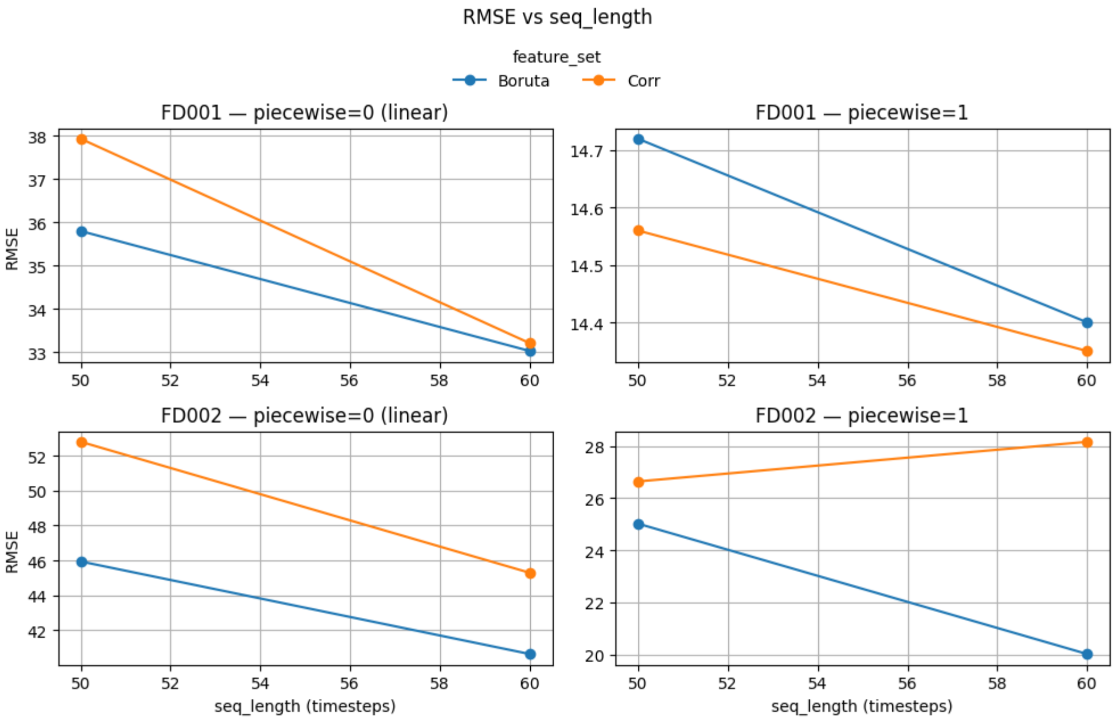

Figure 15 – RMSE Variation

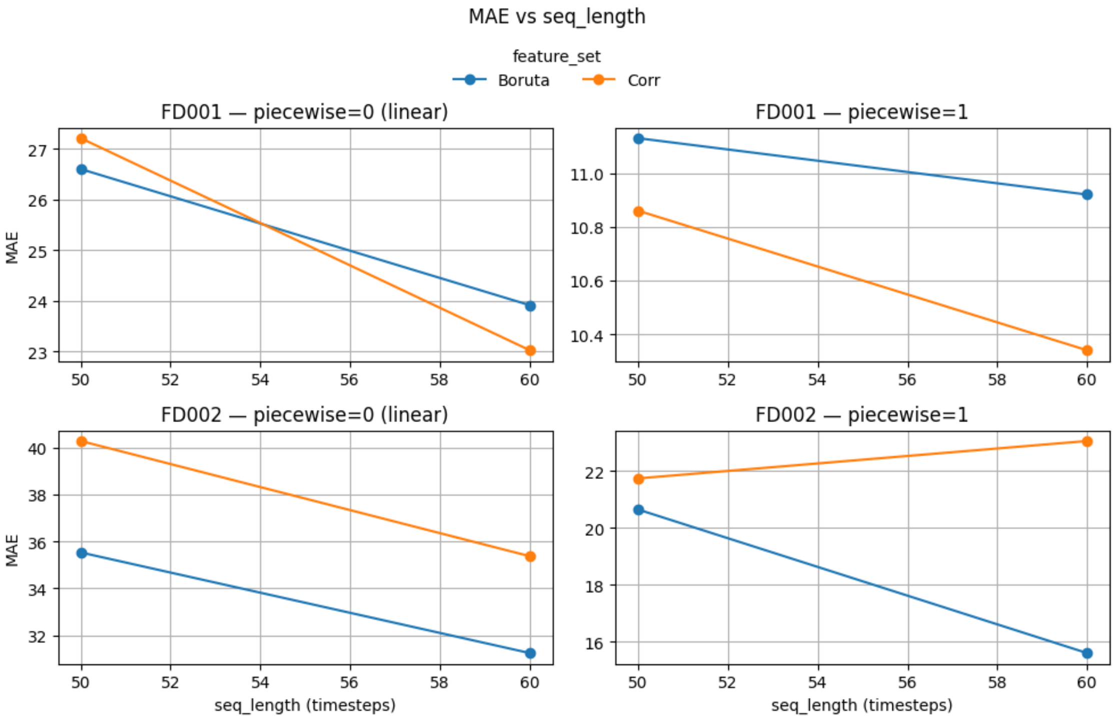

Figure 16 – MAE Variation

These results show that the piecewise RUL calculation strategy is decisive for achieving stable and accurate training. The choice of feature selection method also had an impact, mainly on FD002. Boruta feature selection, when combined with longer input sequences, provided the best overall performance, suggesting that this method better captured the variables relevant for modelling degradation under multiple operating conditions.

The trends observed in this experiment are consistent with the studies presented that have investigated Transformer based architectures for RUL prediction using the CMAPSS dataset. Table 7 compares the best result overall of the best run of this work with other Transformer-based approaches mentioned of section 2.2 that also use this same dataset.

Table 7 – RMSE comparison on FD001 and FD002

Model	FD001	FD002	FD003	FD004
This work model	14.35	20.02	12.42	19.08
GCU-Transformer[98]	11.27	22.81	11.42	22.15
AGCTE[100]	12.46	13.70	12.95	15.83
DSFormer[92]	10.77	14.82	10.12	18.17

Regarding FD001, the best configuration in this work resulting in RMSE of 14.45 performs worse than more advanced Transformer variants. Mo et al. [98] reported an RMSE of 11.27 when using their implementation, GCU-Transformer, which adds a gated convolutional unit to capture local context. Li et al. [92] went further with DSFormer, introducing a dual scale attention and temporal convolutional modules, achieving 10.77 RMSE on FD001. This can indicate that while the baseline transformer can already learn temporal dependencies, additional modules designed to enhance local feature extraction, or multi scale temporal representation may present noticeable improvements.

For FD002, there is a different scenario. This work configuration of Boruta feature selection and piecewise RUL calculation method improves upon the GCU-Transformer performance. This can indicate that preprocessing choices can, in some cases, overcome architectural limitations. However, more recent methods, still achieve lower errors, Ma et al. [92], [100] AGCTE transformer that combines an encoder with and adaptive graph convolutional network, reached 13.70 RMSE, while Li et al. DSFormer [92] obtained 14.82.

Looking for the FD003 results the baseline transformer of this work reaches 12.42 for RMSE, which is close to AGCTE 12.95 and bit higher than GCU-Transformer with 11.42 and DSFormer 10.12. This shows that the model can already handle single condition data with multiple fault modes reasonably well but specialized architectures that combine convolution or dual scale attention achieve a further reduction in error values.

Finally, for FD004, this work obtains 19.08 of RMSE. Comparing to other works this value is lower than the GCU-Transformer with 22.15 and close to DSFormer 18.17, however outperformed by the AGCTE model with 15.83. The results suggest that, in the most complex scenario, incorporating additional modules such as adaptive graph convolution, as proposed by Ma et al. [100], can provide a clearer advantage by better modelling sensor interdependencies.

Some conclusions can be made from the Transformer experiment. Starting with the mechanism to calculate the RUL. Across both FD001 and FD002, removing the piecewise calculation consistently led to superior error values. This confirms that stabilizing the target function is a prerequisite for training deep learning models effectively on CMAPSS dataset. Also, for more complex scenarios feature selection as an impact on the results. While FD001 showed similar performance between Correlation and Boruta methods, FD002 benefited from Boruta selection. This can reflect that for more complex scenarios selecting the most informative features becomes more important. Looking for FD003 the transformer reached values close to the ones presented on the state-of-the-art papers indicating that the model can capture degradation patterns under single condition settings with multiple fault modes. Even though our model outperformed state of the art models on the FD004 it still got behind to architectures that incorporate additional components such as adaptive graph convolution. Lastly, this baseline transformer can be competitive but not considered state of the art. The encoder only transformer achieved strong results, particularly when combined with piecewise RUL calculation and Boruta feature selection but cannot reach the performance levels of hybrid approaches. Advanced models such as DSFormer [92], AGCTE [100] and GCU-Transfomer [98] demonstrate that incorporating convolutional, multi-scale or graph mechanisms provides additional gains on CMAPSS.

4.2.3 Transformer versus LSTM

To understand better the performance of both architectures the Table 8 contains the best configuration results obtained for each subset using both LSTM and Transformer architectures.

Table 8 – Best RMSE values per subset

Subset	Best LSTM	Best Transformer
FD001	12.57	14.35
FD002	27.18	20.02
FD003	10.63	12.42
FD004	15.83	19.34

The comparison reveals that the behaviour of both architecture is similar, having the same variation while the scenario complexity grows. For FD001 and FD003, which represent single operating conditions, the LSTM achieved lower errors that the Transformer. In FD001, the best LSTM configuration reached RMSE 12.57 compared to 14.35 for the Transformer, while in FD003 the gap was similar, with the LSTM reaching 10.63 RMSE versus the 12.42. For FD002, the transformer outperformed the LSTM. This can suggest that in this specific multi condition scenario, the transformer was able to capture the variability more effectively. In FD004, the most complex subset, the LSTM again proved more accurate, reaching RMSE value of 15.83 compared to 19.34 for the transformer. Although, recent literature shows that enhanced Transformer architectures can surpass traditional recurrent models in this subset, the baseline encoder only tested in this work could not match the performance of the LSTM.

Overall, the results indicate that the LSTM remains the more reliable in three out of four subsets, particularly in stable or highly variable environments where temporal recurrence offers robust degradation modelling. The transformer showed its strengths in FD002 outperforming the LSTM but this advantage did not generalize to FD004, where the LSTM maintained lower errors.

5 Conclusion

This dissertation started with the goal to explore the use of deep learning models for Remaining Useful Life (RUL) prediction, specifically using the NASA dataset CMAPSS, with a particular focus on comparing recurrent and attention-based models. The study covered the full development pipeline, from dataset exploration and preprocessing to the implementation, training and evaluation of the LSTM and Transformer models. Starting with dataset where first it was performed an exploration of the features that were available regarding their distribution since this dataset is composed of 4 subsets each representing a different scenario. This stage raised some concerns regarding the contribution of some features, and so, for this work two different feature selection methods were applied. First one being the correlation and low variation method and the second was the Boruta method proposed on one of the state-of-the-art papers analysed for this work. Second one was regarding the RUL values, since literature mention two methods linear and piecewise RUL calculation, however no real comparison between their performance on the same model is made. This study therefore implemented and contrasted both methods. Lastly, it was necessary to prepare the data so it could be used by the LSTM and transformer model, due to time series nature on the problem, it was necessary to adapt the sensor readings in a sequence. For test purposes two window sizes, meaning the sequence length were used 50 and 60. When all the data was studied and pre-processed this work moved on to the implementation stage where two different models were developed. First a LSTM model and then a Transformer encoder only model. These models were then trained and evaluated using the RMSE and MAE metrics. This metric choice was mainly due to being the most used on other studies on the same field, facilitating the comparability with other work's results.

The results obtained on the experimental phase confirmed some important insights. Starting with the RUL calculation method proved to be decisive. The application of the piecewise method consistently improved model performance across all subsets. This helped stabilize the learning process and avoid systemic overestimation in early life cycles. Secondly, feature selection strategies had a meaningful impact, especially under complex operating conditions.

On one hand the correlation-based filtering performed well in simpler scenarios, on the other hand Boruta selection provided clear improvements in multi-condition datasets, particularly when combined with longer input windows.

On the topic of model performance, the LSTM achieved excellent results in FD001 and FD003, with RMSE values as low as 12.57 and 10.63 respectively. These results could compete with or sometime be better than other published models. For example, previous works have emphasized that LSTMs remain particularly strong under stable operating conditions with a single fault mode, as in FD001.

These results reinforce the suitability of recurrent networks for scalable operating conditions, both with single and multiple modes. The transformer, on the other hand, showed its strongest performance on FD002, reducing the error to 20.02. However, in FD004 the LSTM remained more accurate. Overall, the results demonstrate that while LSTMs provide a strong and reliable baseline, the potential of Transformer-based models becomes more evident when additional architecture enhancement, such as convolutional modules, multiscale attention or graph-based components, are introduced.

The motivation behind predictive maintenance is to reduce downtime, improve safety, and optimize resources. However, achieving these benefits requires more than just accurate models. Industrial adoption must also consider ethical aspects such as sensitive data protection, system resilience and environmental concerns. Algorithms that drive these solutions may create challenges of transparency, fairness and privacy. The “black box” nature of these models makes it difficult for users to understand how a prediction is produced, which in return can weaken trust in the results [109], [110], [111], [112]. Transparency has been highlighted as a core ethical principle for AI, since it helps to establish confidence in model outcomes [113], [114]. Bias is another important challenge. Datasets may not capture all possible operational conditions, creating data bias, while model design choices such as features selection or evaluation metrics can also introduce systematic errors [115]. This can affect both reliability and model generalization. In addition, the data used for these systems often contain sensitive operational and maintenance information making security and privacy protection critical [111], [116]. Research proposed encryption methods, anonymization strategies, and stricter access controls, while more recent approaches such as federal learning allow training across distributed devices without exposing raw data, thereby improving privacy [116], [117] Lastly, predictive maintenance has the potential to abide with sustainable goals. By reducing waste, improving resource allocation and optimizing energy use, contributing to more efficient and responsible industrial practices. This gains relevance since AI methods can be computationally intensive, so aligning them with sustainable practices is increasingly necessary [118], [119].

From this work it is also possible to point out two major contributions. Starting with the study of the preprocessing strategies for RUL prediction. As it was presented on Chapter 3.3 by exploring different techniques for feature selection, RUL calculation and sequence generation, this work highlights the importance and impact of preprocessing stage on the predictive models. Without this step the dataset would be too noisy and inconsistent, unable to provide good inputs for training. The experiments conducted here reinforce the idea that preprocessing is not just a technical detail, but a fundamental part of the overall modelling pipeline. The other contribution concerns the comparison of two deep learning models, the LSTM and the Transformer, across all four subsets on the CMAPSS dataset. While LSTMs have been considered a strong baseline for timeseries forecast, as shown in [87], [88]. The transformer, on the other hand, did not consistently outperformed the LSTM in this study, but still provided valuable insights into how attention-based models behave when applied to RUL prediction. For instance, in scenarios with more variability in operating conditions, the Transformer showed signs of capturing long range dependencies, although its performance was not superior to the LSTM. This side-by-side comparison does not claim that one architecture is universally better, but rather highlights their different behaviours, which can lead to future improvements. By presenting results across all four subsets, this study delivers a clearer picture of how each model responds to the challenges posed by different modes of operation in the CMAPSS dataset.

Naturally, this work has its own limitations. The Transformer architecture explored here was restricted to an encoder only design, without the integration of other modules that could improve feature extraction. Recent studies in the literature have shown that combining attention mechanisms with convolutional layers or graph structures can enhance performance by capturing both local and global dependencies, as mentioned in [85], [92], [100]. The absence of such hybrid designs in this dissertation means that there is still room for improvement particularly in scenarios where the sensor signals are highly correlated or contain subtle degradation patterns that are difficult to capture with self-attention alone.

For future work, the Transformer implementation could be extended towards hybrid models that combine different strengths from different architectures. One promising path would be to integrate recurrence and attention, leveraging the sequential modelling ability of LSTMs together with the global context awareness of Transformers. Another line of improvement lies in the implementation of a convolution transformer, which have already been proposed in the literature to enhance local feature extraction while preserving long-range dependencies. For instance, the authors of [98] introduced a gated convolutional unit into the Transformer encoder to better incorporate local context, while in [92] it is proposed a dual scale Transformer that uses temporal convolutional networks to address positional information loss. Similarly, the work of [100] where the authors combine a Transformer encoder with an adaptive graph convolutional network to capture both temporal and sensor wise relationships. Exploring these directions could help overcome the limitation that were observed in this study. This could provide more models for RUL prediction across different CMAPSS subsets.

To conclude, this research showcased the value of both LSTM and Transformers bring as tools for RUL prediction area. While LSTMs provide a reliable and competitive baseline, the transformer model represent a promising method, particularly when enhanced by additional mechanisms. It was, as well, confirmed the importance of preprocessing choices and model design in predictive maintenance, and its contribution to the ongoing effort of developing more accurate and robust prognostic models.

References

[1] M. Molęda, B. Małysiak-Mrozek, W. Ding, V. Sunderam, and D. Mrozek, “From Corrective to Predictive Maintenance—A Review of Maintenance Approaches for the Power Industry,” Jul. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/s23135970.

[2] D. Kolar, D. Lisjak, M. Pajak̨, and D. Pavković, “Fault diagnosis of rotary machines using deep convolutional neural network with wide three axis vibration signal input,” *Sensors (Switzerland)*, vol. 20, no. 14, pp. 1–13, Jul. 2020, doi: 10.3390/s20144017.

[3] L. Magadán, J. C. Granda, and F. J. Suárez, “Robust prediction of remaining useful lifetime of bearings using deep learning,” *Eng Appl Artif Intell*, vol. 130, Apr. 2024, doi: 10.1016/j.engappai.2023.107690.

[4] J. Lee, H. A. Kao, and S. Yang, “Service innovation and smart analytics for Industry 4.0 and big data environment,” in *Procedia CIRP*, Elsevier, 2014, pp. 3–8. doi: 10.1016/j.procir.2014.02.001.

[5] M. Fernandes, J. M. Corchado, and G. Marreiros, “Machine learning techniques applied to mechanical fault diagnosis and fault prognosis in the context of real industrial manufacturing use-cases: a systematic literature review,” *Applied Intelligence*, 2022, doi: 10.1007/s10489-022-03344-3.

[6] B. Bagheri, S. Yang, H. A. Kao, and J. Lee, “Cyber-physical systems architecture for self-aware machines in industry 4.0 environment,” in *IFAC-PapersOnLine*, May 2015, pp. 1622–1627. doi: 10.1016/j.ifacol.2015.06.318.

[7] H. Fox, A. C. Pillai, D. Friedrich, M. Collu, T. Dawood, and L. Johanning, “A Review of Predictive and Prescriptive Offshore Wind Farm Operation and Maintenance,” Jan. 01, 2022, *MDPI*. doi: 10.3390/en15020504.

[8] P. Do, V.-T. Nguyen, A. Voisin, B. Iung, and W. A. F. Neto, “Multi-agent deep reinforcement learning-based maintenance optimization for multi-dependent component systems,” *Expert Syst Appl*, vol. 245, p. 123144, Jul. 2024, doi: 10.1016/j.eswa.2024.123144.

[9] M. Bevilacqua and M. Braglia, “The analytic hierarchy process applied to maintenance strategy selection.” [Online]. Available: www.elsevier.com/locate/ress

[10] A. Gorenstein and M. Kalech, “Predictive maintenance for critical infrastructure,” *Expert Syst Appl*, vol. 210, Dec. 2022, doi: 10.1016/j.eswa.2022.118413.

[11] R. Ahmad and S. Kamaruddin, “An overview of time-based and condition-based maintenance in industrial application,” *Comput Ind Eng*, vol. 63, no. 1, pp. 135–149, 2012, doi: 10.1016/j.cie.2012.02.002.

[12] R. Sala, E. Francalanza, and S. Arena, “A review on three decades of manufacturing maintenance research: past, present and future directions,” *Prod Manuf Res*, vol. 13, no. 1, 2025, doi: 10.1080/21693277.2025.2469037.

[13] P. Poór, D. Ženíšek, and J. Basl, “Historical Overview of Maintenance Management Strategies: Development from Breakdown Maintenance to Predictive Maintenance in Accordance with Four Industrial Revolutions.”

[14] D. Li, Y. Wang, J. Wang, C. Wang, and Y. Duan, “Recent advances in sensor fault diagnosis: A review,” Jul. 01, 2020, *Elsevier B.V.* doi: 10.1016/j.sna.2020.111990.

[15] Y. J. Yoo, “Data-driven fault detection process using correlation based clustering,” *Comput Ind*, vol. 122, Nov. 2020, doi: 10.1016/j.compind.2020.103279.

[16] D. Kolar, D. Lisjak, M. Pajak¸, and D. Pavković, “Fault diagnosis of rotary machines using deep convolutional neural network with wide three axis vibration signal input,” *Sensors (Switzerland)*, vol. 20, no. 14, pp. 1–13, Jul. 2020, doi: 10.3390/s20144017.

[17] Y. Peng, A. Tan, J. Wu, and Y. Bi, “Hierarchical edge computing: A novel multi-source multi-dimensional data anomaly detection scheme for industrial internet of things,” *IEEE Access*, vol. 7, pp. 111257–111270, 2019, doi: 10.1109/ACCESS.2019.2930627.

[18] Z. H. Janjua, M. Vecchio, M. Antonini, and F. Antonelli, “IRESE: An intelligent rare-event detection system using unsupervised learning on the IoT edge,” *Eng Appl Artif Intell*, vol. 84, pp. 41–50, Sep. 2019, doi: 10.1016/j.engappai.2019.05.011.

[19] A. Abbas, “AI for Predictive Maintenance in Industrial Systems,” 2024.

[20] J. Osheyor Gidiagba, N. Kelvin Nwaobia, P. Winston Biu, C. Alex Ezeigweneme, A. Akpan Umoh, and C. Author, “REVIEW ON THE EVOLUTION AND IMPACT OF IOT-DRIVEN PREDICTIVE MAINTENANCE: ASSESSING ADVANCEMENTS, THEIR ROLE IN ENHANCING SYSTEM LONGEVITY, AND SUSTAINABLE OPERATIONS IN BOTH MECHANICAL AND ELECTRICAL REALMS,” *Computer Science & IT Research Journal*, vol. 5, no. 1, pp. 166–189, 2024, doi: 10.51594/csitrj.v5i.716.

[21] M. A. Rassam, M. A. Maarof, and A. Zainal, “A distributed anomaly detection model for wireless sensor networks based on the one-class principal component classifier,” *International Journal of Sensor Networks*, vol. 27, no. 3, pp. 200–214, 2018, doi: 10.1504/IJSNET.2018.093126.

[22] D. ElMenshawy and W. Helmy, “A correlation based approach to differentiate between an event and noise in internet of things,” *International Journal of Advanced Computer*

Science and Applications, vol. 9, no. 12, pp. 79–83, 2018, doi: 10.14569/IJACSA.2018.091212.

[23] B. Zhao, X. Li, J. Li, J. Zou, and Y. Liu, "An Area-Context-Based Credibility Detection for Big Data in IoT," *Mobile Information Systems*, vol. 2020, 2020, doi: 10.1155/2020/5068731.

[24] Y. Liu, T. Dillon, W. Yu, W. Rahayu, and F. Mostafa, "Noise Removal in the Presence of Significant Anomalies for Industrial IoT Sensor Data in Manufacturing," *IEEE Internet Things J*, vol. 7, no. 8, pp. 7084–7096, Aug. 2020, doi: 10.1109/JIOT.2020.2981476.

[25] F. De Caro, A. Vaccaro, and D. Villacci, "Adaptive wind generation modeling by fuzzy clustering of experimental data," *Electronics (Switzerland)*, vol. 7, no. 4, Apr. 2018, doi: 10.3390/electronics7040047.

[26] H. M. Hashemian and W. C. Bean, "State-of-the-art predictive maintenance techniques," in *IEEE Transactions on Instrumentation and Measurement*, Oct. 2011, pp. 3480–3492. doi: 10.1109/TIM.2009.2036347.

[27] P. Stavropoulos, A. Papacharalampopoulos, E. Vasiliadis, and G. Chryssolouris, "Tool wear predictability estimation in milling based on multi-sensorial data," *International Journal of Advanced Manufacturing Technology*, vol. 82, no. 1–4, pp. 509–521, Jan. 2016, doi: 10.1007/s00170-015-7317-6.

[28] C. Zhang, X. Yao, J. Zhang, and H. Jin, "Tool condition monitoring and remaining useful life prognostic based on awireless sensor in dry milling operations," *Sensors (Switzerland)*, vol. 16, no. 6, Jun. 2016, doi: 10.3390/s16060795.

[29] M. C. Garcia, M. A. Sanz-Bobi, and J. del Pico, "SIMAP: Intelligent System for Predictive Maintenance. Application to the health condition monitoring of a windturbine gearbox," *Comput Ind*, vol. 57, no. 6, pp. 552–568, Aug. 2006, doi: 10.1016/j.compind.2006.02.011.

[30] J. C. P. Cheng, W. Chen, K. Chen, and Q. Wang, "Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms," *Autom Constr*, vol. 112, Apr. 2020, doi: 10.1016/j.autcon.2020.103087.

[31] C. Gianoglio, E. Ragusa, A. Bruzzone, P. Gastaldo, R. Zunino, and F. Guastavino, "Unsupervised monitoring system for predictive maintenance of high voltage apparatus," *Energies (Basel)*, vol. 13, no. 5, Mar. 2020, doi: 10.3390/en13051109.

[32] N. Helwig, E. Pignanelli, and A. Schütze, "Condition Monitoring of a Complex Hydraulic System using Multivariate Statistics."

[33] N. Helwig, E. Pignanelli, and A. Schütze, "D8.1 - Detecting and Compensating Sensor Faults in a Hydraulic Condition Monitoring System," AMA Service GmbH, Dec. 2020, pp. 641–646. doi: 10.5162/sensor2015/d8.1.

[34] A. Lourenço *et al.*, "Adaptive time series representation for out-of-round railway wheels fault diagnosis in wayside monitoring," *Eng Fail Anal*, vol. 152, Oct. 2023, doi: 10.1016/j.engfailanal.2023.107433.

[35] N. Ghanbari, Y. Riyazi, F. A. Shirazi, A. Kalhor, and Y. Riyazi, "Enhanced Gearbox Fault Diagnosis with Fusion LSTM-CNN Network," 2023. [Online]. Available: https://www.researchgate.net/publication/377217896

[36] N. A. Raji, R. O. Kuku, A. O. Bakare, M. M. Ogunbiyi, and T. I. Morafa, "Comparative analysis of gearbox fault detection using ensemble learning techniques with vibration sensor data," *Journal of Production Engineering*, vol. 27, no. 2, pp. 1–9, Dec. 2024, doi: 10.24867/JPE-2024-02-001.

[37] S. Arshad Mayo, S. Rehman, and Z. Cai, "High-accuracy gearbox fault detection using deep learning on vibrational data," in *Journal of Physics: Conference Series*, Institute of Physics, 2024. doi: 10.1088/1742-6596/2853/1/012066.

[38] D. Coelho, D. Costa, E. M. Rocha, D. Almeida, and J. P. Santos, "Predictive maintenance on sensorized stamping presses by time series segmentation, anomaly detection, and classification algorithms," in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 1184–1193. doi: 10.1016/j.procs.2022.01.318.

[39] M. B. Kursa, A. Jankowski, and W. R. Rudnicki, "Boruta - A system for feature selection," *Fundam Inform*, vol. 101, no. 4, pp. 271–285, 2010, doi: 10.3233/FI-2010-288.

[40] A. K. Kar, "Bio inspired computing - A review of algorithms and scope of applications," Oct. 15, 2016, *Elsevier Ltd*. doi: 10.1016/j.eswa.2016.04.018.

[41] J. H. Suh, S. R. T. Kumara, and S. P. Mysore, "Machinery Fault Diagnosis and Prognosis: Application of Advanced SignaI Processing Techniques."

[42] Y. Peng, M. Dong, and M. J. Zuo, "Current status of machine prognostics in condition-based maintenance: A review," *International Journal of Advanced Manufacturing Technology*, vol. 50, no. 1–4, pp. 297–313, Sep. 2010, doi: 10.1007/s00170-009-2482-0.

[43] J. Hu and P. Chen, "Predictive maintenance of systems subject to hard failure based on proportional hazards model," *Reliab Eng Syst Saf*, vol. 196, Apr. 2020, doi: 10.1016/j.ress.2019.106707.

[44] Y. Lei, Li Naipeng, Lin Jing, and Dybala Jacek, "A Model-Based Method for Remaining Useful Life Prediction of Machinery," in *Intelligent Fault Diagnosis and Remaining*

Useful Life Prediction of Rotating Machinery, Elsevier, 2017, pp. 231–280. doi: 10.1016/b978-0-12-811534-3.00005-6.

[45] N. Li, Y. Lei, T. Yan, N. Li, and T. Han, “A wiener-process-model-based method for remaining useful life prediction considering unit-to-unit variability,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 3, pp. 2092–2101, Mar. 2019, doi: 10.1109/TIE.2018.2838078.

[46] J. Lee, “A Similarity-Based Prognostics Approach for Remaining Useful Life Estimation of Engineered Systems,” 2008. [Online]. Available: https://www.researchgate.net/publication/269167324

[47] A. Saxena, K. Goebel, Simon Don, and Eklund Neil, “Damage Propagation Modeling for Aircraft Engine Prognostics”.

[48] M. Altosole, G. Benvenuto, M. Figari, and U. Campora, “Real-time simulation of a COGAG naval ship propulsion system,” *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, vol. 223, no. 1, pp. 47–61, Jan. 2009, doi: 10.1243/14750902JEME121.

[49] T. Berredjem and M. Benidir, “Bearing faults diagnosis using fuzzy expert system relying on an Improved Range Overlaps and Similarity method,” *Expert Syst Appl*, vol. 108, pp. 134–142, Oct. 2018, doi: 10.1016/j.eswa.2018.04.025.

[50] F. Guastavino, A. Dardano, and E. Torello, “Measuring Partial Discharges under Pulsed Voltage Conditions,” 2008.

[51] J. Qiu, B. B. Seth, S. Y. Liang, and C. Zhang, “Damage mechanics approach for bearing lifetime prognostics,” *Mech Syst Signal Process*, vol. 16, no. 5, pp. 817–829, 2002, doi: 10.1006/mssp.2002.1483.

[52] N. Vafaei, R. A. Ribeiro, and L. M. Camarinha-Matos, “Fuzzy early warning systems for condition based maintenance,” *Comput Ind Eng*, vol. 128, pp. 736–746, Feb. 2019, doi: 10.1016/j.cie.2018.12.056.

[53] Q. Cao, A. Samet, C. Zanni-Merk, F. D. B. De Beuvron, and C. Reich, “An ontology-based approach for failure classification in predictive maintenance using fuzzy c-means and SWRL rules,” in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 630–639. doi: 10.1016/j.procs.2019.09.218.

[54] Marius Baban, Calin Florin Baban, and Beniamin Moisi, “A fuzzy logic-based approach for predictive maintenance of grinding wheels of automated grinding lines”.

[55] L. Breiman, “Random Forests,” 2001.

[56] Z. Li, D. Wu, C. Hu, and J. Terpenny, "An ensemble learning-based prognostic approach with degradation-dependent weights for remaining useful life prediction," *Reliab Eng Syst Saf*, vol. 184, pp. 110–122, Apr. 2019, doi: 10.1016/j.ress.2017.12.016.

[57] X. Tang, M. Xiao, Y. Liang, H. Zhu, and J. Li, "Online updating belief-rule-base using Bayesian estimation," *Knowl Based Syst*, vol. 171, pp. 93–105, May 2019, doi: 10.1016/j.knosys.2019.02.007.

[58] J. R. Ruiz-Sarmiento, J. Monroy, F. A. Moreno, C. Galindo, J. M. Bonelo, and J. Gonzalez-Jimenez, "A predictive model for the maintenance of industrial machinery in the context of industry 4.0," *Eng Appl Artif Intell*, vol. 87, Jan. 2020, doi: 10.1016/j.engappai.2019.103289.

[59] N. Z. Gebraeel, M. A. Lawley, R. Li, and J. K. Ryan, "Residual-life distributions from component degradation signals: A Bayesian approach," *IIE Transactions (Institute of Industrial Engineers)*, vol. 37, no. 6, pp. 543–557, Jun. 2005, doi: 10.1080/07408170590929018.

[60] J. Man and Q. Zhou, "Prediction of hard failures with stochastic degradation signals using Wiener process and proportional hazards model," *Comput Ind Eng*, vol. 125, pp. 480–489, Nov. 2018, doi: 10.1016/j.cie.2018.09.015.

[61] K. A. Kaiser and N. Z. Gebraeel, "Predictive maintenance management using sensor-based degradation models," *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, vol. 39, no. 4, pp. 840–849, 2009, doi: 10.1109/TSMCA.2009.2016429.

[62] M. Y. You, F. Liu, W. Wang, and G. Meng, "Statistically planned and individually improved predictive maintenance management for continuously monitored degrading systems," *IEEE Trans Reliab*, vol. 59, no. 4, pp. 744–753, Dec. 2010, doi: 10.1109/TR.2010.2085572.

[63] L. Liao, W. Jin, and R. Pavel, "Enhanced Restricted Boltzmann Machine with Prognosability Regularization for Prognostics and Health Assessment," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7076–7083, Nov. 2016, doi: 10.1109/TIE.2016.2586442.

[64] H. Shin, J. K. Lee, J. Kim, and J. Kim Sk T-Brain, "Continual Learning with Deep Generative Replay."

[65] M. A. Rassam, A. Zainal, and M. A. Maarof, "One-class principal component classifier for anomaly detection in wireless sensor network," in *Proceedings of the 2012 4th International Conference on Computational Aspects of Social Networks, CASoN 2012*, 2012, pp. 271–276. doi: 10.1109/CASoN.2012.6412414.

[66] H. Qiu, J. Lee, J. Lin, and G. Yu, "Robust performance degradation assessment methods for enhanced rolling element bearing prognostics," *Advanced Engineering Informatics*, vol. 17, no. 3–4, pp. 127–140, Jul. 2003, doi: 10.1016/j.aei.2004.08.001.

[67] F. P. Nasution, S. Sævik, and J. K. Ø. Gjøsteen, "Fatigue analysis of copper conductor for offshore wind turbines by experimental and FE method," in *Energy Procedia*, Elsevier Ltd, 2012, pp. 271–280. doi: 10.1016/j.egypro.2012.06.109.

[68] P. Bansal, S. A. Imam, and R. Bharti, "Speaker recognition using MFCC, shifted MFCC with vector quantization and fuzzy," in *International Conference on Soft Computing Techniques and Implementations, ICSCTI 2015*, Institute of Electrical and Electronics Engineers Inc., Jun. 2016, pp. 41–44. doi: 10.1109/ICSCTI.2015.7489535.

[69] X. Valero and F. Alias, "Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification," *IEEE Trans Multimedia*, vol. 14, no. 6, pp. 1684–1689, 2012, doi: 10.1109/TMM.2012.2199972.

[70] C. Peng, Y. Chen, W. Gui, Z. Tang, and C. Li, "Remaining useful life prognosis of turbofan engines based on deep feature extraction and fusion," *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-10191-2.

[71] S. Boral, S. K. Chaturvedi, and V. N. A. Naikan, "A case-based reasoning system for fault detection and isolation: a case study on complex gearboxes," *J Qual Maint Eng*, vol. 25, no. 2, pp. 213–235, Apr. 2019, doi: 10.1108/JQME-05-2018-0039.

[72] J. L. Kolodner, "An Introduction to Case-Based Reasoning*," 1992.

[73] F. Cauteruccio *et al.*, "Short-long term anomaly detection in wireless sensor networks based on machine learning and multi-parameterized edit distance," *Information Fusion*, vol. 52, pp. 13–30, Dec. 2019, doi: 10.1016/j.inffus.2018.11.010.

[74] J. Wang, L. Zhang, L. Duan, and R. X. Gao, "A new paradigm of cloud-based predictive maintenance for intelligent manufacturing," *J Intell Manuf*, vol. 28, no. 5, pp. 1125–1137, Jun. 2017, doi: 10.1007/s10845-015-1066-0.

[75] J. Bechta Dugan, S. J. Member IEEE Salvatore Bavuso Mark A Boyd, and M. Ieee, "Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems," 1992.

[76] R. Gao *et al.*, "Cloud-enabled prognosis for manufacturing," *CIRP Ann Manuf Technol*, vol. 64, no. 2, pp. 749–772, 2015, doi: 10.1016/j.cirp.2015.05.011.

[77] S. Garg, K. Kaur, S. Batra, G. Kaddoum, N. Kumar, and A. Boukerche, "A multi-stage anomaly detection scheme for augmenting the security in IoT-enabled applications," *Future Generation Computer Systems*, vol. 104, pp. 105–118, Mar. 2020, doi: 10.1016/j.future.2019.09.038.

[78] M. A. Hayes and M. A. Capretz, “Contextual anomaly detection framework for big sensor data,” *J Big Data*, vol. 2, no. 1, Dec. 2015, doi: 10.1186/s40537-014-0011-y.

[79] M. Roig, M. Catalan, and B. Gastón, “Ensembled outlier detection using multi-variable correlation in WSN through unsupervised learning techniques,” in *IoTBDS 2019 - Proceedings of the 4th International Conference on Internet of Things, Big Data and Security*, SciTePress, 2019, pp. 38–48. doi: 10.5220/0007657400380048.

[80] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, “Cyber-Physical Systems: The Next Computing Revolution,” 2010.

[81] H. Wang, J. Yang, L. Shi, and R. Wang, “Remaining Useful Life Prediction Based on Adaptive SHRINKAGE Processing and Temporal Convolutional Network,” *Sensors*, vol. 22, no. 23, Dec. 2022, doi: 10.3390/s22239088.

[82] A. Sherifi and W. Pratt, “Turbofan Engine Remaining Useful Life (RUL) Prediction Based on Bi-Directional Long Short-Term Memory (BLSTM),” 2024.

[83] A. Vakili, “A Comparative Analysis of Machine Learning Models in Prognostics and Prediction of Remaining Useful Life of Aircraft Turbofan Engines,” 2023.

[84] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, *Long Short-Term Memory Network for Remaining Useful Life Estimation*. IEEE, 2016.

[85] H. Zhang, Q. Zhang, S. Shao, T. Niu, and X. Yang, “Attention-Based LSTM Network for Rotatory Machine Remaining Useful Life Prediction,” *IEEE Access*, vol. 8, pp. 132188–132199, 2020, doi: 10.1109/ACCESS.2020.3010066.

[86] Y. Li, Y. Chen, Z. Hu, and H. Zhang, “Remaining useful life prediction of aero-engine enabled by fusing knowledge and deep learning models,” *Reliab Eng Syst Saf*, vol. 229, Jan. 2023, doi: 10.1016/j.ress.2022.108869.

[87] P. Malhotra *et al.*, “Multi-Sensor Prognostics using an Unsupervised Health Index based on LSTM Encoder-Decoder,” Aug. 2016, [Online]. Available: http://arxiv.org/abs/1608.06154

[88] V. P. Nemani, H. Lu, A. Thelen, C. Hu, and A. T. Zimmerman, “Ensembles of probabilistic LSTM predictors and correctors for bearing prognostics using industrial standards,” *Neurocomputing*, vol. 491, pp. 575–596, Jun. 2022, doi: 10.1016/j.neucom.2021.12.035.

[89] S. Tippareddy, “Predicting Remaining Useful Life of Turbofan Engines on CMAPSS and N-CMAPSS using Deep Recurrent Neural Networks,” San Jose State University, San Jose, CA, USA, 2024. doi: 10.31979/etd.cyfe-z4sj.

[90] J. P. Noot, M. Martin, and E. Birmele, “LSTM and Transformers based methods for Remaining Useful Life Prediction considering Censored Data,” *Int J Progn Health Manag*, vol. 16, no. 2, Jul. 2025, doi: 10.36001/IJPHM.2025.v16i2.4260.

[91] T. Sabri Syuhada, “Performance Analysis of Long Short-term Memory (LSTM) Model for Remaining Useful Life Prediction on Turbofan Engine”, doi: 10.52465/joetex.v3i1.585.

[92] J. Li *et al.*, “A Dual-Scale Transformer-Based Remaining Useful Life Prediction Model in Industrial Internet of Things.”

[93] A. Y. Zhou and A. Barati Farimani, “FaultFormer: Pretraining Transformers for Adaptable Bearing Fault Classification,” *IEEE Access*, vol. 12, pp. 70719–70728, 2024, doi: 10.1109/ACCESS.2024.3399670.

[94] S. Kim, Y. H. Seo, and J. Park, “Transformer-based novel framework for remaining useful life prediction of lubricant in operational rolling bearings,” *Reliab Eng Syst Saf*, vol. 251, Nov. 2024, doi: 10.1016/j.ress.2024.110377.

[95] D. D. Pezze, C. Masiero, D. Tosato, A. Beghi, and G. A. Susto, “FORMULA: A Deep Learning Approach for Rare Alarms Predictions in Industrial Equipment,” *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1491–1502, Jul. 2022, doi: 10.1109/TASE.2021.3127995.

[96] M. R. Babaei, A. D. Fentaye, and K. Kyprianidis, “Enhanced Anomaly Detection in Aero-Engines using Convolutional Transformers,” Jan. 2025. doi: 10.3384/ecp212.059.

[97] A. Vaswani *et al.*, “Attention Is All You Need.”

[98] Y. Mo, Q. Wu, X. Li, and B. Huang, “Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit,” *J Intell Manuf*, vol. 32, no. 7, pp. 1997–2006, Oct. 2021, doi: 10.1007/s10845-021-01750-x.

[99] O. Ogunfowora and H. Najjaran, “A Transformer-based Framework For Multi-variate Time Series: A Remaining Useful Life Prediction Use Case,” Aug. 2023, [Online]. Available: http://arxiv.org/abs/2308.09884

[100] M. Ma, Z. Wang, and Z. Zhong, “Transformer Encoder Enhanced by an Adaptive Graph Convolutional Neural Network for Prediction of Aero-Engines’ Remaining Useful Life,” *Aerospace*, vol. 11, no. 4, Apr. 2024, doi: 10.3390/aerospace11040289.

[101] D. K. Frederick, J. A. Decastro, and J. S. Litt, “User’s Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS),” 2007. [Online]. Available: http://www.sti.nasa.gov

[102] D. S. Abhinav Saxena, “Damage Propagation Modeling for Aircraft Engine Prognostics,” *Don Simon,*.

[103] A. Saxena, K. Goebel, D. Simon, and N. Eklund, “2008 INTERNATIONAL CONFERENCE ON PROGNOSTICS AND HEALTH MANAGEMENT Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation,” 2008.

[104] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” 2014.

[105] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, “Deep Learning.” Accessed: Mar. 25, 2025. [Online]. Available: http://www.deeplearningbook.org

[106] J. Chung, C. Gulcehre, and K. Cho, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.”

[107] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6980

[108] W. Zhang, J. Jia, X. Pang, J. Wen, Y. Shi, and J. Zeng, “An Improved Transformer Model for Remaining Useful Life Prediction of Lithium-Ion Batteries under Random Charging and Discharging,” *Electronics (Switzerland)*, vol. 13, no. 8, Apr. 2024, doi: 10.3390/electronics13081423.

[109] Femi Osasona, Olukunle Oladipupo Amoo, Akoh Atadoga, Temitayo Oluwaseun Abrahams, Oluwatoyin Ajoke Farayola, and Benjamin Samson Ayinla, “REVIEWING THE ETHICAL IMPLICATIONS OF AI IN DECISION MAKING PROCESSES,” *International Journal of Management & Entrepreneurship Research*, vol. 6, no. 2, pp. 322–335, Feb. 2024, doi: 10.51594/ijmer.v6i2.773.

[110] S. Du and C. Xie, “Paradoxes of artificial intelligence in consumer markets: Ethical challenges and opportunities,” *J Bus Res*, vol. 129, pp. 961–974, May 2021, doi: 10.1016/j.jbusres.2020.08.024.

[111] C. J. Liang, T. H. Le, Y. Ham, B. R. K. Mantha, M. H. Cheng, and J. J. Lin, “Ethics of artificial intelligence and robotics in the architecture, engineering, and construction industry,” Jun. 01, 2024, *Elsevier B.V.* doi: 10.1016/j.autcon.2024.105369.

[112] A. B. Brendel, M. Mirbabaie, T. B. Lembcke, and L. Hofeditz, “Ethical management of artificial intelligence,” Feb. 02, 2021, *MDPI*. doi: 10.3390/su13041974.

[113] J. Laux, S. Wachter, and B. Mittelstadt, “Trustworthy artificial intelligence and the European Union AI act: On the conflation of trustworthiness and acceptability of risk,” *Regul Gov*, vol. 18, no. 1, pp. 3–32, Jan. 2024, doi: 10.1111/rego.12512.

[114] K. W. Ramsey, “THE ETHICS OF ARTIFICIAL INTELLIGENCE (2011) Nick Bostrom Eliezer Yudkowsky Draft for Cambridge Handbook of Artificial Intelligence, eds.”

[115] A. Durresi, M. Tuceryan, M. Dundar, and Q. Hu, “THE PURDUE UNIVERSITY GRADUATE SCHOOL STATEMENT OF COMMITTEE APPROVAL.”

[116] D. Dhinakaran, S. M. Udhaya Sankar, D. Selvaraj, and S. E. Raja, “Privacy-Preserving Data in IoT-based Cloud Systems: A Comprehensive Survey with AI Integration.”

[117] L. Yuan, Z. Wang, and C. G. Brinton, “Digital Ethics in Federated Learning,” Oct. 2023, [Online]. Available: http://arxiv.org/abs/2310.03178

[118] A. John, “AI-Powered Predictive Maintenance: Optimizing Industrial Efficiency.” [Online]. Available: https://www.researchgate.net/publication/387428815

[119] D. Leuthe, T. Meyer-Hollatz, T. Plank, and A. Senkmüller, “Towards Sustainability of AI – Identifying Design Patterns for Sustainable Machine Learning Development,” *Information Systems Frontiers*, 2024, doi: 10.1007/s10796-024-10526-6.