Programa de Pós-Graduação em

Computação Aplicada

Doutorado Acadêmico

Jovani Dalzochio

MILPdM:
an architecture for predictive maintenance of assets in the military domain

São Leopoldo, 2024

Jovani Dalzochio

MILPDM:
An architecture for predictive maintenance of assets in the military domain

Tese apresentada como requisito parcial para a obtenção do título de Doutor pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Advisor:
Prof. Dr. Rafael Kunst

Co-advisor:
Prof. Dr. Jorge Luis Victória Barbosa

São Leopoldo
2024

D153m Dalzochio, Jovani.
MILPDM : Na architecture for predictive maintenance of assets in the military domain / Jovani Dalzochio. – 2024.
125 f. : il. ; 30 cm.

Tese (doutorado) – Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Comutação Aplicada, 2024.
“Orientador: Prof. Dr. Rafael Kunst
Coorientador: Prof. Dr. Jorge Luiz Victória Barbosa”

1. Foundation model. 2. Machine learning. 3. Military. 4. Predictive maintenance. I. Título.

CDU 004.4

Dados Internacionais de Catalogação na Publicação (CIP)
(Bibliotecária: Silvana Dornelles Studzinski – CRB 10/2524)

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) -Código de Financiamento 001 /"This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior -Brasil (CAPES) -Finance Code 001

(Esta folha serve somente para guardar o lugar da verdadeira folha de aprovação, que é obtida após a defesa do trabalho. Este item é obrigatório, exceto no caso de TCCs.)

This thesis is dedicated to my late mother, Cecília,
who instilled in me the value of education
through her support and encouragement.

*"Life slips away minute by minute,
and you can’t buy it at the supermarket.
So fight to live it, to give it content."*
— PEPE MUJICA

ACKNOWLEDGEMENTS

From the bottom of my heart, I thank everyone who contributed to completing this doctoral thesis. This challenging, sometimes lonely, but highly enriching journey would not have been possible without the support of such special people.

Firstly, I would like to express my gratitude to my late mother, Cecília, who always encouraged me to continue my studies. Although she wasn't physically present, her existence held significant importance. I also thank my brothers, Gerson and Joelma, who were fundamental in overcoming the challenges to reach this moment.

My advisor, Professor Rafael, whose wise and dedicated guidance was essential for developing this work. Our hours of conversation, your valuable advice, and your patience were fundamental to my academic and personal growth. I am grateful for your careful guidance, which contributed significantly to the success of this research.

To my friends, I want to express gratitude for your understanding during periods of absence and your constant support, which made all the difference in challenging times. Our meetings are essential for allowing me to unburden myself and make life easier.

To my colleagues at the IBM Watson Center in Munich, thank you for your responsiveness and crucial support during the sandwich period, providing resources and opportunities that enriched my academic experience.

The University of Vale do Rio dos Sinos (UNISINOS) and the Coordination of Superior Level Staff Improvement (CAPES) for the opportunity offered, the trust in my work, and the financial support that made this research possible.

To everyone who contributed to this academic journey, my sincere thanks. I share this milestone with each of you, and I thank you for being part of it.

ABSTRACT

Manutenção preditiva é um tema abordado em diferentes contextos como indústria, logística e saúde, onde o sensoriamento dos parâmetros dos equipamentos permite monitorar a degradação da saúde e antecipar falhas. Essa mesma abordagem tem sido usada no domínio militar, monitorando ativos como veículos militares. O monitoramento da degradação dos ativos gera benefícios econômicos semelhantes aos observados em outras áreas, reduzindo custos por meio da otimização do uso dos ativos monitorados. No entanto, os ativos que operam no domínio militar realizam tarefas críticas, onde as falhas podem gerar um alto custo material e humano. A aplicação de manutenção preditiva em equipamentos militares é desafiadora. Veículos e equipamentos são enviados para missões em cenários já conhecidos com alta disponibilidade de dados coletados. Entretanto esses equipamentos podem ser enviados a novos ambientes, onde não existem dados de operações anteriores para avaliação de degradação da saúde do equipamento. As abordagens encontradas na literatura para previsão de falhas aplicadas ao domínio militar focam no monitoramento dos equipamentos. Este trabalho apresenta uma abordagem mais ampla através do uso do MILPdM, uma arquitetura que visa a predição de falhas aplicadas ao domínio militar, considerando o dinâmico cenário de atuação de equipamentos militares. Nossa abordagem possui duas frentes distintas, primeiramente buscamos verificar a possibilidade de uso dos modelos de aprendizados de máquina para a predição de falhas, e a partir desse ponto, buscamos verificar a capacidade de predição em cenários desconhecidos, onde aplicamos *foundation models* para predição do tempo de vida em novos cenários, com uma comparação dos resultados com modelos tradicionais. Para testarmos a arquitetura quatro casos de usos são propostos, dois casos de uso para validar modelos tradicionais da predição de falhas, onde é empregado os algoritmos de aprendizado de máquina *long-short term memory* e *random forest*. Outros dois casos de uso para avaliar o uso de *foundation models* em cenários desconhecidos para o equipamento. Os resultados adquiridos da predição dos modelos treinados mostram que o MILPdM pode antecipar falhas com alta assertividade. Já para a capacidade de predição em cenários desconhecidos, o uso de *foundation model* se mostrou promissor, superando modelos de aprendizado tradicionais. Essesresultados mostram o potencial do uso do modelo de fundação em manutenção preditiva.

Keywords: Predictive maintenance. Machine learning. Foundation Model. Military.

ABSTRACT

Predictive maintenance is a topic addressed in different contexts such as industry, logistics, and healthcare, where sensing equipment parameters allows monitoring health degradation and anticipating failures. This same approach has been used in the military, monitoring assets such as vehicles. Monitoring asset degradation generates economic benefits similar to those observed in other areas, reducing costs by optimizing the use of monitored assets. However, assets operating in the military domain perform critical tasks, where failures can generate a high material and human cost. Applying predictive maintenance to military equipment is challenging. Vehicles and equipment are sent for missions in already known scenarios with high availability of collected data. However, this equipment can be sent to new environments where there is no data from previous operations to evaluate the degradation of the equipment's health. The approaches in the literature for failure prediction applied to the military domain focus on equipment monitoring. This work presents a broader approach through the use of MILPdM. This architecture aims to predict failures applied to the military domain, considering the dynamic scenario in which military equipment operates. Our approach has two distinct fronts. First, we seek to verify the possibility of using machine learning models to predict failures, and from that point on, we seek to verify the prediction capacity in new scenarios, where we test the new foundation models for predicting lifespan in new scenarios, with a comparison of results with traditional models. We propose four use cases to test the architecture. Two use cases validate traditional failure prediction models using long-short term memory and random forest machine learning algorithms. Two other use cases evaluate the use of foundation models in new scenarios. The results acquired from the prediction of the trained models show that MILPdM can anticipate failures with high accuracy. As for the prediction capacity in new scenarios, using the foundation model proved promising, surpassing traditional learning models. These results show the potential of using the foundation model in predictive maintenance.

Keywords: Predictive maintenance. Machine learning. Foundation Model. Military.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

The application of Industry 4.0 concepts is impacting several areas beyond the industry (DALZOCHIO et al., 2020). Advances in this domain are allowing the large-scale use of network-connected sensors and actuators for several different tasks. With the spread of sensors, more and more data are being generated, which opens up possibilities for use in applications, such as artificial intelligence algorithms, enabling the execution of autonomous tasks, such as predicting the degradation of the health of equipment (YAN et al., 2017).

Failure prediction is a matter of great interest and a high priority for the industry, particularly considering the current networked industrial plants (DALZOCHIO et al., 2020). This type of prediction allows the anticipation of failures in equipment and components, increased performance, and minimized periods of unscheduled downtime for maintenance, reducing their inefficiency (YAN et al., 2017). Implementing systems that anticipate these failures is a challenge for the industry (DALZOCHIO et al., 2020) and for other sectors, such as transport (ATAMURADOV et al., 2017), health (SHAMAYLEH; AWAD; FARHAT, 2020) and military.

The context of military use can take advantage of the application of failure prediction through foundation models (FM). Studies show FM plays a crucial role in a context where an asset that operates in certain conditions may, in future missions, operate in conditions where there is no prior data that could anticipate equipment degradation. To anticipate the equipment degradation, the learning model must have generalization capacity. Studies show that FM can be a solution for generalizing scenarios (JIN et al., 2023a) where, with a pretrained FM on a large benchmark dataset and performing fine-tuning, it is possible to obtain a model that can solve similar tasks (ZHOU et al., 2023a).

The military domain has characteristics and challenges that differ from other areas. In this chapter we present the motivation and definition of the problem, related to the application of failure prediction in the military domain, summarizing the objectives to be achieved and the methodology applied.

1.1 Motivation

Military operations presuppose the use of a large amount of armament, supplies, and equipment that need to be ready for the maneuvers on the battlefield. In order to achieve it, the maintenance management projects and operations (KOSZTYÁN; PRIBOJSZKI-NÉMETH; SZALKAI, 2019), whether preventive (KOSZTYÁN; PRIBOJSZKI-NÉMETH; SZALKAI, 2019; NORDAL; EL-THALJI, 2021; VILLA et al., 2021), predictive (FERNÁNDEZ-BARRERO et al., 2021; VILLA et al., 2021), or corrective (KOSZTYÁN; PRIBOJSZKI-NÉMETH; SZALKAI, 2019; NORDAL; EL-THALJI, 2021), are an essential part in the military domain. The troops rely on maintenance management to preserve the ordinary conditions of using the materials or restore them to combat or training. The military maintenance projects and operations include military land vehicles (WOLDMAN et al., 2015), Unmanned Aerial Vehicles (UAVs) (HRÚZ et al., 2021), military aircraft (BAYOUMI; MATTHEWS, 2020), watercraft and warships (FERNÁNDEZ-BARRERO et al., 2021), rail and communication material, tactical intelligence, electronic warfare, sensors material (ARMY, 2019), optical and individual equipment, military systems in general, and even non-tactical equipment. This results in an extensive, diverse, and complex military equipment ecosystem (FERNÁNDEZ-BARRERO et al., 2021) that, on the battlefield, needs accurate maintenance planning to optimize the supply chain.

However, keeping these assets operated and available is a costly task that needs to be optimized constantly. Only the maintenance investment made by the United States Department of Defense in 2019 achieved $78 billion (PESCHIERA et al., 2020). Part of this total cost was for the predictive maintenance (PdM) strategy, based on the periodic measurement of the variables that determine the conditions of the equipment while it is operating (FERNÁNDEZ-BARRERO et al., 2021). Nevertheless, to minimize maintenance-related costs, it is necessary to be accurate in the maintenance operations and projects, extracting as much of each resource as possible and avoiding wastes that unnecessary maintenance can generate (BABBAR et al., 2009). In this context of PdM in a modern military environment, intelligent concepts such as the Internet of Battlefield Things (IoBT) that brings the Internet of Things (IoT) technologies to the military scenario (RUSSELL; ABDELZAHER, 2018) can be helpful to implement a predictive maintenance policy.

Through this PdM strategy, sometimes called intelligent maintenance strategy (NORDAL; EL-THALJI, 2021), failure prediction is one of the facets of prognostics and health management (PHM). E. g., IoT technologies combined with time series analysis make the failure forecasting conceivable to estimate the remaining useful life (RUL) of specific equipment. Furthermore, several areas use the data resulting from the monitoring to put in place PdM strategies to optimize the use of equipment, reduce maintenance costs, minimize downtime, and increase equipment availability (LEE et al., 2014). Consequently, tasks such as accurate fault diagnosis and forecasting through PHM make it possible to optimize maintenance investment in PdM.

Some points must be considered when applying a PdM system, regardless of the context used, as in the industry (DALZOCHIO et al., 2020), transport (ATAMURADOV et al., 2017), or health (SHAMAYLEH; AWAD; FARHAT, 2020), for instance. A PHM system, applied for equipment, has in general four stages, widely discussed in the literature (LEI et al., 2018): (i) data acquisition, (ii) health indicator construction, (iii) health stage division, and (iv) remaining useful life prediction. Therefore, to implement a system of PHM, some points must be taken into account, such as, which critical component should be analyzed, which sensors to use, which characteristics to analyze, which prognostic methods, and the evaluation methods to use (ATAMURADOV et al., 2017). Besides that, a PdM system must be integrated with other areas concurrently, such as logistics and fleet management (KILLEEN et al., 2019). Each of the contexts cited has its unique characteristics. For example, we may search for increased productivity in industry; in transport, we may search for increased security; and the criticality of eventual failures in a healthcare context.

The military domain of a PdM system has its particularities and challenges, such as those related to the type of equipment and vehicles operated, the environment and operating conditions, the need to maintain the readiness of equipment and vehicles in cases of external aggression, and the security of the information. Also, the equipment maintenance in battle is related to damages from weapons, types of ammunition, combat vehicles, and the enemy attack modes on the battlefield (LI; ZHAO; PU, 2020).

1.2 Problem Definition

Equipment monitoring for failure prediction is receiving attention from different sectors in society, such as industry, healthcare, and the military. In the military domain, assets like military vehicles generate data that we can use to identify behavior changes to anticipate possible failures in run-time, avoiding unnecessary maintenance interventions. Failure anticipation is crucial, as assets operated in the military domain perform critical tasks, in which unexpected equipment failures result in high material and human costs.

In a dynamic battlefield, ensuring safe and high-quality access to sensor data is a challenge. The success of any operation relies heavily on the accuracy and reliability of the data accessed. When implementing a predictive maintenance system, one must consider such data restrictions. One moment, a vehicle operates in a desert climate on flat terrain. In a second moment, that exact vehicle may be operating in a tropical environment on rough terrain. It is only sometimes possible to access equipment behavior data in all these different contexts. This lack of data can affect the ability of traditional learning models to understand asset behavior in operation and accurately predict health degradation. Approaches found in the literature typically deal with condition monitoring, aiming at analyzing the equipment's behavior, but do not address data restrictions.

Based on these challenges, we have come up with the following research question:

> Is it possible to develop a failure prediction architecture to predict equipment degradation in unknown operational scenarios?

1.3 Objectives

This work proposes MILPdM, an architecture for predictive maintenance applied to the military context. The architecture defines the steps that must be applied to first predict equipment failures, and then predict equipment degradation in unknown operational scenarios.

In military operations, having access to high-quality data is challenging. Therefore, MILPdM must consider the lack of data issues, in this way, one objective of the architecture is, through the application of machine learning, to anticipate future failure for the maintenance team. In this sense, learning models must be able to make predictions, even when the mission scenarios in which the asset is operating are new and without previously collected data.

The main goal of this thesis is to design and propose an architecture for predictive maintenance applied to the military domain that can overcome the challenges of lack of data for training traditional learning models and predict equipment degradation in unknown operational scenarios.

To achieve these objectives, this work will focus on the following specific objectives:

- Conduct a systematic review of the literature covering the application of predictive maintenance in military scenario.
- Propose an architecture for PdM, named MILPdM, describing the steps necessary to implement a military PdM system, from data collection to anticipating the failures for the maintenance team.
- Obtain a dataset of a failed equipment for training traditional machine learning models.
- Create the machine learning models that are used by MILPdM in the failure prediction process.
- Elaborate scenarios for application of the architecture using the trained learning models together with the dataset previously obtained.
- Elaborate scenarios for application of the architecture using the foundation models to predict equipment degradation in unknown operational scenarios.
- Evaluate the architecture through the discussion of the results obtained from the foundation models and traditional learning models.

1.4 Contributions

Among the contributions of this work, we have carried out a systematic review of the literature involving the application of PdM in the military domain, highlighting challenges, machine learning techniques used, and application scenarios. We also have the creation of MILPdM, an architecture for PdM that considers the challenges of the military domain in its design. Specifically, concerning MILPdM, we can cite the following contributions:

- Failure prediction architecture.
- Adaptability to various military equipment.
- Recording prediction results together with feedback from the maintenance team, allowing for retraining the machine learning models when needed.

- Creation of history of machine learning models and their results obtained through asset monitoring, this history allows applying existing prediction models in new but similar scenarios.

- Evaluation of a novel approach that uses foundation models to predict equipment degradation in unknown operational scenarios.

1.5 Methodology

Figure 1 shows the steps to achieve the objectives described above. The first step will be to carry out a study on topics related to the MILPdM. This first study will serve as a basis for consolidating knowledge on topics such as existing maintenance policies and existing machine learning techniques applied in the military context.

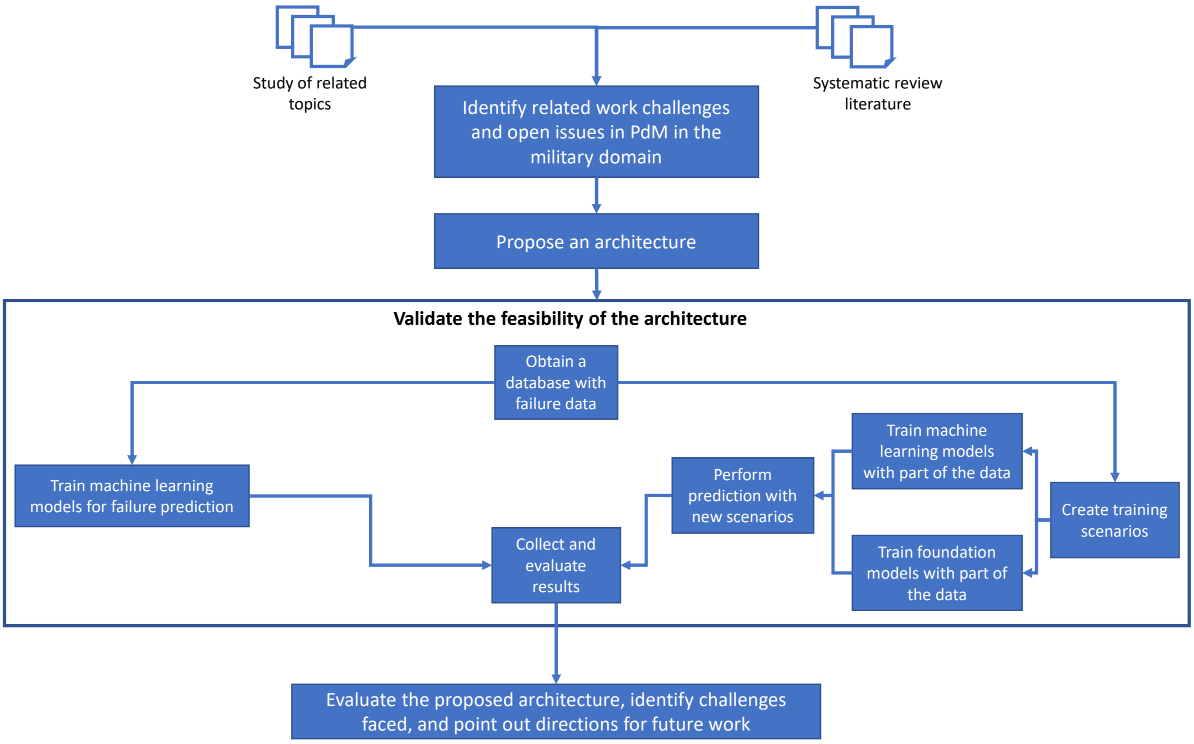


Figure 1: Methodology steps

The next step to achieve the proposed objectives is to carry out a systematic review of the literature to understand the use of PdM in the military domain and identify the particularities that differentiate the military domain from others. The systematic review of the literature also aims to identify the challenges and open issues in PdM in the military domain. Which military contexts PdM is applied and which techniques the authors use. Between these techniques, in recent years, machine learning is something that the proposed works use. What technologies and actions do the related works employ concerning data transmission.

After the systematic review, an architecture, called MILPdM, is proposed. It is composed of layers with different functions and objectives, which together help in the process of implementing a predictive maintenance policy, keeping in mind the need to adapt to new usage scenarios that exist in military use.

To validate the feasibility of the MILPdM architecture, it is necessary to carry out use cases to obtain and analyze the results. Thus, collecting a database for training traditional learning models for failure prediction tests and creating scenarios for training and testing foundation models and their prediction capacity in new scenarios.

In the end, we will do final considerations about the results obtained, listing points identified with challenges in the implementation of MILPdM and pointing out possible directions for the implementation of future works.

1.6 Text organization

This proposal is organized as follows: chapter 2 describes the concepts of the themes related to the proposed architecture. The topics covered include maintenance concepts, failure prediction, machine learning, and foundation models. Chapter 3 presented a systematic literature review to identify the works related to the theme of the proposal, explore the challenges and open possibilities. The works resulting from the literature review and is closest to the theme of this proposal are used as a comparison with the proposed approach. Chapter 4 presets the architecture MILPdM, its layers and components. Chapter 5 presents the ML model and FM results. Chapter 6 presents the proposal contributions and future work.

2 BACKGROUND

Keeping an asset in an operational state is not a trivial task. It is necessary to put in place a set of actions to maintain this asset in operation for as long as possible. In this section, we will present some of the concepts that which, when applied, allow equipment to be used for a longer period at a lower cost. We can say that to reach a high operational level, there are some possible paths, and initially, it is necessary to take into account the existence of two basic maintenance policies. A maintenance policy aimed at taking actions only after the resource or asset presents some type of problem is called corrective maintenance. The second policy seeks to anticipate failures, this anticipation is achieved through a maintenance policy called preventive maintenance. Each of the maintenance policies has its positive points, its negative points, and challenges related to its implementations.

First, it is important to point out that regardless of the policy adopted, it is necessary to have a broad understanding of the functioning of both the corrective maintenance policy and the preventive maintenance policy. This understanding allows to evolve and improve the maintenance strategy, always pursuing the objective of increasing the availability of the asset at the lowest possible cost. Only then, upon reaching an advanced stage of maintenance, is it possible to determine when a failure might occur, preventing the asset from going into a failure situation. To achieve this level of ability to anticipate failure, a set of practices and tools must be adopted.

In certain domains, such as the domain of military use, the application of maintenance policies is necessary, since many of the assets operated by the army require high availability and security in their use, such criticalities are better explored in the section 3 of related works. To meet the needs of the domain of use in military assets, a set of tools is used to achieve the ability to anticipate the occurrence of failures. These tools which, within their functions, constantly monitor the life condition of an asset, are part of a maintenance philosophy that we call predictive maintenance (PdM) or condition-based maintenance (CBM).

Along with the concept of CBM and PdM new acronyms usually appear as, Prognostics & Health Management (PHM), Remaining Useful Life (RUL), and Machine Learning (ML), where each acronym represents a concept within a CBM system. In the following sections, we will present each of these concepts, from the existing maintenance policies to the tools available to achieve maintenance capacity in which it is possible to anticipate failures. In the end, we present the concepts related to foundation models, which ensure the ability to make predictions in new scenarios.

2.1 Maintenance Policy

Historically, when we talk about maintenance policies and the tasks involved in the maintenance process, we have as an initial view the image of broken equipment undergoing some type of intervention so that this equipment returns to its normal operating state. Eventually, at a more advanced stage in the application of maintenance policies, we see periodic examinations being carried out, such examinations are carried out following the equipment manufacturer's standards or manuals (JARDINE; TSANG, 2013).

If we consider the vision of broken equipment undergoing an intervention by the maintenance team, we can locate this task as part of a corrective maintenance policy. As we move towards the process of carrying out periodic reviews, such activity is classified as a preventive maintenance task. Figure 2 presents these two views throughout history.

In addition to these two views, we can have a maintenance policy that fits the so-called predictive maintenance. The predictive maintenance policy is the result of several technological advances, both in the hardware areas (use of sensors) and in the software area with the development of robust algorithms for failure prediction. This view can be addressed with a specialization of preventive maintenance (KOBBACY; MURTHY, 2008). So, in the next section, we will present in detail the corrective and preventive maintenance policies.

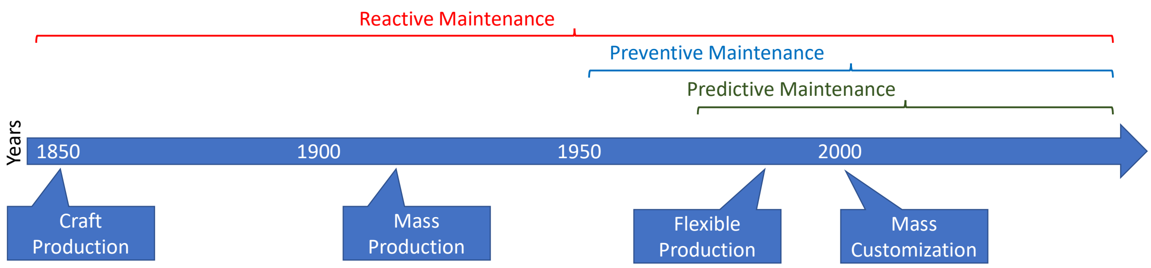


Figure 2: Evolution of maintenance policies over the years, adapted from Vogl, Weiss e Helu (2019)

2.1.1 Corrective Maintenance

Also known as reactive maintenance or breakdown maintenance (JARDINE; TSANG, 2013). Corrective maintenance has been used as a primary maintenance policy for hundreds of years (KOBBACY; MURTHY, 2008). Its application and use are based on the maintenance task of equipment when it has already failed and is in a non-operational state, in other words, it is a "Run-to-failure" policy where if a device has not broken, simply don't fix it, so as much as possible is extracted from a device without any intervention (MOBLEY, 2002).

If, on the one hand, in the corrective maintenance policy we have the maximum extraction of equipment, without unnecessary changes or maintenance, on the other hand, this type of policy can be dangerous. Such danger is a result of the unpredictability of the damage that a failure can cause to a complex system or equipment, and how that failure can propagate and affect other systems (JARDINE; TSANG, 2013). Thus, even if the corrective maintenance policy requires less attention and less investment in maintenance initially, turns out to be more costly in the long term, especially in a more complex system. In addition, even if the current maintenance policy is corrective, minimal tasks invariably continue to be carried out preventively, such as a periodic oil change, for example (MOBLEY, 2002).

2.1.2 Preventive Maintenance

Known for being a time-driven approach (JARDINE; TSANG, 2013), preventive maintenance covers the periodic inspection of parts and equipment that are in an operational state. Historically, it is after World War II that the prevention of failures begins to receive attention and investments. Fast forward to the early 1960s (BARLOW; HUNTER, 1960) the need for maintenance plans for more complex systems regularly is mentioned, using metrics characteristics such as the number of operating hours of equipment or operating cycles.

In general, the application of a preventive maintenance policy involves a higher maintenance cost, however, this cost is eventually rewarded by the decrease in corrective maintenance and by the decrease in losses caused by the occurrence of failures. This relationship between increased costs in performing preventive maintenance tasks and reduced costs in correction tasks, as the costs arising from any unscheduled shutdown of systems is not a constant, and the correct balance between these costs is a challenge inherent to the area (MOBLEY, 2002).

As already mentioned, preventive maintenance is efficient in the task of reducing failures during the operation of equipment and, in general, can be classified as periodic maintenance (at fixed intervals) or sequential maintenance (at different time intervals) (NAKAGAWA, 1986). We can expand these classifications in more detail as follows (WANG; PHAM, 2006):

- Age-dependent PM Policy: The most commonly used of maintenance policies, where interventions take place in the asset to change materials in a given time T or in case a failure occurs.

- Periodic PM Policy: Maintenance is performed at each pre-fixed interval or period of time.

- Failure Limit Policy: In this preventive maintenance policy, indexes such as failure rate or reliability are applied. If the assets reach predetermined levels, intervention is carried out, in this way the equipment always operates above a minimum level of confidence.

- Sequential PM Policy: In this maintenance policy, there is no fixed period or interval of maintenance, occurring at unequal intervals that become shorter as time goes by. Here it is taken into account that the older the equipment, the more frequent the need for maintenance.

- Repair Limit Policy: The repair limit policy can be two, by cost or by repair time. In case of equipment failure, if the cost limit or the repair time limit is reached, then the equipment is replaced, and the repair is not performed.

- Repair Number Counting and Reference Time Policy: There is no schedule for this type of maintenance, here a unit is replaced in the k^{th} failure where k is the policy decision variable.

As seen, preventive maintenance is commonly classified as time-base maintenance. Thus, the current state of the system is not taken into account and assets that are in operation end up being stopped for some inspection and preventive maintenance, respecting the stipulated maintenance schedule. This happens regardless of whether or not there is a real need for maintenance. For example, there are environments such as helicopter operations where half of the assets are removed from operation even though they are still operational (KOBBACY; MURTHY, 2008).

One way to mitigate this type of situation involves the development of prediction techniques for maintenance, which is currently implemented with a maintenance philosophy called condition-based maintenance (CBM) (KOBBACY; MURTHY, 2008). The next section presents the concepts involved in a maintenance policy that uses CBM.

2.2 Condition-Based Maintenance

Technological advances in recent years make possible constant and non-intrusive monitoring of the condition of equipment through vibration measurement, thermography, ferrography, and spectroscopy, laying the foundations for a Condition-Based Maintenance (JARDINE; TSANG, 2013), an alternative approach to the time-based one previously mentioned in the section 2.1.2 (JARDINE; TSANG, 2013). This monitoring is done in an automated way through the use of software and hardware, which are capable of detecting, isolating and, predicting the degradation of an asset until the moment when the failure occurs (KIM; AN; CHOI, 2017).

Between the 1950s and 1970s, industries such as automobile, aerospace, manufacturing, and military have applied the concepts of CBM, bringing several benefits in terms of costs (PRAJAPATI; BECHTEL; GANESAN, 2012), currently being involved and injecting resources into the development of technologies for CBM institutions like the Department of Defense (DoD), which encompasses other institutions such as the army, air force, navy, and marines. The data collected and analyzed from assets helps in the maintenance, performance, and forecast of the RUL, giving insights to help decision-making regarding the maintenance of the assets.

For complex systems that have requirements such as high availability, preventive maintenance by itself becomes complex and expensive to maintain (KIM; AN; CHOI, 2017), being the Condition-Based Maintenance system a more cost-effective maintenance strategy, and in the task of implementing CBM a PHM system is critical.

2.2.1 Prognostics & Health Management

PHM applied to machinery and equipment is an approach that uses real-time data to determine the operating conditions of an asset, determine its deterioration state, and predict upcoming failures (KIM; AN; CHOI, 2017), providing information that will help in the decision support activity (VOGL; WEISS; HELU, 2019).

To implement a PHM system that is effective in protecting the integrity of equipment and anticipating any failures, it is necessary to use a multidisciplinary approach, as shown in Figure 3. We present these tasks separated between the activities of observing the asset through sensing and data processing. The analysis activity involves anomaly detection, diagnosis, and failure prognosis tasks. Finally, the task of acting concerns a capacity to support the decision of the system of PHM.

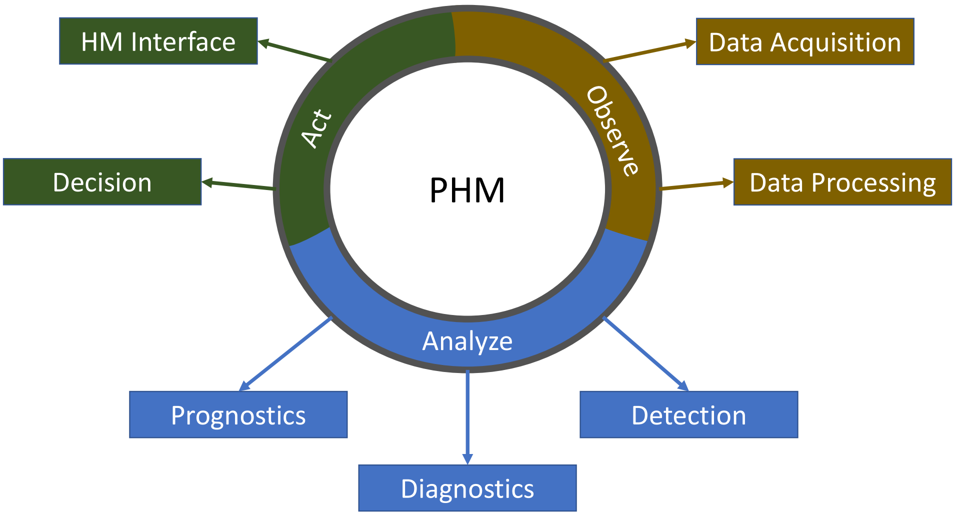


Figure 3: PHM multidisciplinary approach, adapted from Atamuradov et al. (2017)

In a PHM system like the one shown in Figure 3, we can prevent the occurrence of a failure by estimating the RUL, which is the final end of a larger set of steps. Such tasks, ranging from data collection through asset monitoring, detection, diagnosis, and finally to the prognosis process through understanding the degradation of equipment to failure (VOGL; WEISS; HELU, 2019) will be detailed in the next sessions.

2.2.2 Data Acquisition and Processing

The first step of a PHM system is responsible for collecting and analyzing the data used in the detection, diagnosis, and prognosis of failures. The collected data can have two different sources, being possible to obtain through the sensing of an asset or system, or through the capture of event data such as system logs and maintenance team intervention information (ATAMURADOV et al., 2017).

Specifically, in a condition monitoring maintenance system, the data are essentially those captured through the sensing of systems. The captured data may contain vibration, temperature, pressure, humidity, oil quality, and others information (ATAMURADOV et al., 2017). However, the importance of event data, despite involving human intervention and being more susceptible to errors, should not be overlooked, being important to ultimately validate the performance of indicators obtained through sensing (JARDINE; LIN; BANJEVIC, 2006).

Before the collected data to a PHM system use, it is necessary to guarantee its quality. As a quality, we can contemplate three distinct dimensions, being them: a data size sufficient to use; an accuracy of data that can correctly represent the real world; and finally, ensure the completeness of the data through the treatment of any gaps in the collections (OMRI et al., 2021).

After obtaining the data and guaranteeing its quality, the next step is to pre-process the raw data for the PHM system to use. Among the pre-processing objectives, we have treatment of the large amount of data collected in a PHM system, aiming, among other things, to clean the data by removing errors or noises (ATAMURADOV et al., 2017). After this data cleaning step, the process of extracting the main characteristics that represent the monitored system is carried out, generating a set of data that is later used in the creation of accurate models for detection, diagnosis, and prediction of failures (CARBERY; WOODS; MARSHALL, 2018).

We can be base the feature extraction process on three approaches, time-domain-based, frequency-based, and time-frequency-based techniques. In the time-domain-based approach, features such as mean, calculations performed on the data generate standard deviation, and high-order statistics from the time waveform. The frequency-domain-based approach transforms the signal in the frequency domain, facilitating the identification and isolation of certain frequency components of interest, enabling the identification of failures that would not be possible when compared to the time-base. Finally, we have the time-frequency-based approach, where it is possible to analyze both the time and frequency domains, making it possible to work with non-stationary waveform signals (JARDINE; LIN; BANJEVIC, 2006).

2.2.3 Detection

Considered the first task among the main PHM tasks (OMRI et al., 2021). Its function is to indicate when something outside the normal behavior of a monitored system happens (JARDINE; LIN; BANJEVIC, 2006). According to (ROY; DEY, 2018), we can divide the fault detectability and two notions, they are being *structural or intrinsic detectability* and *performance based fault detectability*.

Structural or intrinsic detectability refers to the signature of system failures, without a reference to fault diagnosis algorithms. The intrinsic property elucidates system behavior on faults and explores system limitations in fault detection. It is necessary to verify the intrinsic failure detectability before designing any failure detection algorithm. Performance based fault detectability, on the other hand, is defined in relation to a diagnostic algorithm used, quantifying the ability of this algorithm to detect failures.

2.2.4 Diagnosis

The process of diagnosing a failure is carried out through the recognition of patterns generated from the information mapping. This information is obtained through measurements of characteristics of failed systems. Some tools help in pattern recognition, the tools, when operated by a trained person, make pattern recognition as automatic as possible (JARDINE; TSANG, 2013).

Diagnosis plays a crucial role in the failure prognosis process, since a failure prognosis starts with its diagnosis, in addition, a properly functioning diagnostic system prevents failures from occurring in systems. However, implementing a diagnosis process is difficult due to the lack of standardized methods or guides to validate a diagnosis, in addition to the aforementioned need for trained and qualified personnel (VOGL; WEISS; HELU, 2019).

The approaches can be made based on statistical analysis and models of equipment failure, and with the use of machine learning, as can be seen in the following section.

2.2.4.1 Model-Based Approaches

In the task of predicting the behavior of a failure, it is necessary to compare the real condition of a system with an object model, that way we can achieve a model-based diagnostic approach (YAM et al., 2001). To diagnose a failure through the use of model-based approaches it is necessary to use a mathematical model that uses the physical specifications of the system.

Figure 4 shows the model-approach functioning, where residual generation methods, parameter estimation, and parity relations are used to obtain so-called signals or residuals, that point to the presence of a failure. After indicating the presence of a fault, the residues are evaluated to arrive at fault detection, isolation and identification. In this process, algorithms are used as a Kalman Filters or Petri net (JARDINE; TSANG, 2013).

Figure 4: Model-based process for diagnosis, adapted from Jardine e Tsang (2013)

2.2.4.2 Data-Drive Approaches

As it is not always possible to represent a system through a model, in these cases we can use strategies that are capable of learning by example, without the need for prior knowledge of the system. Learning in these cases is done using a large volume of data, and applying sophisticated machine learning algorithms. In this way it is possible through the establishment of standards, making correlations and viewing trends, detecting and performing fault diagnosis. Learning using a large volume of data to diagnose failures is called the data-driven approach (PECHT; KANG, 2019).

There are several approaches and techniques to perform fault diagnosis using a data-driven approach, but in general, an approach that uses machine learning can be represented in a set of steps shown in the figure 5. The realization of the prediction is the last of a sequence of steps that involve the generation of a dataset, training, and evaluation of the learning model.

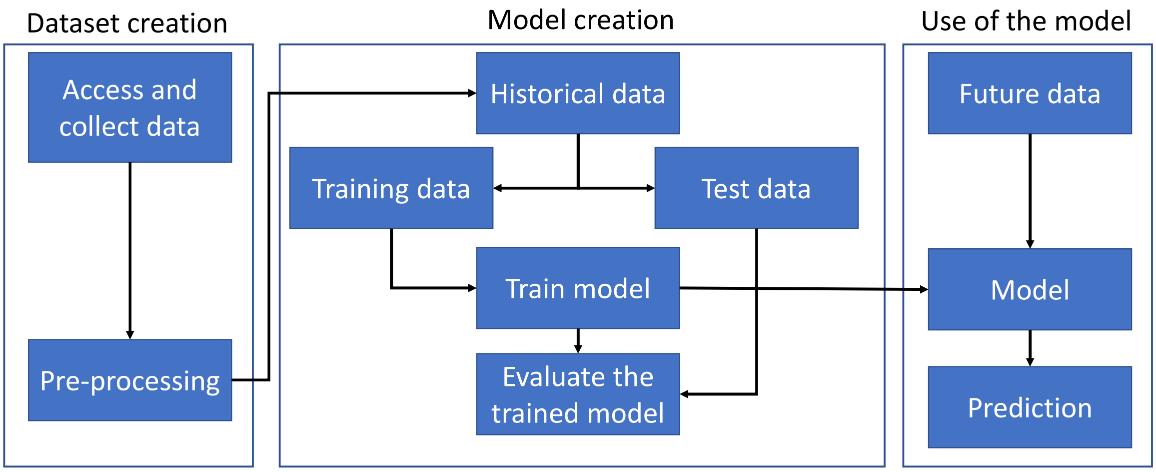


Figure 5: Machine-learning process for diagnosis, adapted from Machine Learning: Diagnostics and Prognostics (2019)

There are several techniques for performing fault diagnosis using machine learning, which can be classified between supervised and unsupervised learning. Among the learning algorithms used, we can mention the use of Naïve Bayes, Decision Trees, Random Forest, Neural network, Support vector machine, K-nearest neighbor. In addition, we have the application of deep learning techniques with several different architectures such as convolution neural network (CNN), the stacked autoencoder (SAE), deep belief network (DBN), deep Boltzmann machine (DBM), deep transfer learning network (DTLN). The techniques can be used separately or together through techniques such as Ensemble Learning (KUNDU; DARPE; KULKARNI, 2020; PECHT; KANG, 2019).

2.2.5 Prognosis

Classified as a more challenging task than diagnosis and still considered an element to be developed in PHM systems, being an emerging field of study (VOGL; WEISS; HELU, 2019).

The implementation of a prediction system brings several benefits, such as reductions in operating costs, extracting the most from each resource. For example, the oil change of a vehicle takes place at pre-determined periods, but through monitoring and forecasting, it can be done only when necessary. In the logistics area, it is possible to make maintenance resources available at the right time, helping with fleet management. Another relevant benefit concerns safety, anticipating failures, especially in critical systems such as an airplane, prevents accidents with large losses from occurring (KIM; AN; CHOI, 2017).

Implementing a diagnostic system is challenged. We have a lack of tools and methods for evaluating predictive algorithms, and for estimating the impact on the system as a whole. Other factors are the difficulty of dealing with multiple failures and implementing methods to estimate the RUL in real-time (VOGL; WEISS; HELU, 2019).

To overcome such challenges and achieve the benefits of a prediction system, it is necessary to make use of a set of tools and algorithms, which can generally be classified as Model-Based and Data-drive.

2.2.5.1 Model-Based Approaches

We use a model-based approach when there is a mathematical model about asset degradation available. To develop such a model, a broad knowledge of both the asset's operation and its behavior in failure mode is necessary. When the model is available, it plays a central role in predicting failures through the RUL (KORDESTANI et al., 2019). Among the advantages of using a model-based approach is the ability to incorporate the physical understanding of the system, and if the understanding of the system increases, the model can be adapted incorporating possible new parameters, thus increasing its accuracy (LUO et al., 2003).

There are several techniques in the literature for mathematical models and the prediction of failures, the most commonly used techniques using estimation and filtering, such as Kalman Filtering Based Prognosis, the Particle Filtering Based Prognosis, and the Model-Based Observers for Prognosis. All models mentioned have the characteristics of the use of a state-space mathematical model of a failure to estimate the RUL (KORDESTANI et al., 2019).

Compared with data-driven approaches, the model-based approach has several advantages, being able to perform long-term prediction. This capability comes from the characteristic of the base model that requires, in advance, an identification of the parameters involved. The propagation of these parameters values over time is a form of accurately determining the RUL. Furthermore, less data is needed for model-based predictive algorithms when compared to data-driven algorithms (KIM; AN; CHOI, 2017).

On the other hand, the construction of the models must observe some aspects. The first point is the model adequacy, that is if the model is capable of predicting the degradation of the system since as the complexity of the system increases, the more difficult it is to estimate the parameters. Estimating the parameters is the central task in building a prognostic model. Finally, to build the models it is necessary to obtain quality degradation data, which is not a simple task, as data with noise, for example, can induce prediction errors (KIM; AN; CHOI, 2017).

2.2.5.2 Data-Driven Approaches

Data-driven prognosis is preferred over model-based when there is a lack of knowledge of the functioning of the system being monitored (KORDESTANI et al., 2019). However, they require a large amount of data for the creation of prognostic models, being computationally expensive (HU et al., 2012). This data for the creation of the models comes from monitoring carried out previously and that measures the degradation of the asset under current or previous conditions of use (KIM; AN; CHOI, 2017).

Regarding the prediction of failures using data-drive, some characteristics must be observed, such as the fact that there is no guarantee that the model will anticipate a future failure if this failure is not related to the data monitored in the past. Thus, to determine what data and the amount of this data should be collected, it is necessary to have an understanding of the monitored system function, with the need for a large volume of data being common. Another feature is the practicality of implementation concerning the base model, since from a large volume of data and using machine learning and data mining tools it is possible to identify previously unknown correlations (KIM; AN; CHOI, 2017).

Data-driven approaches are commonly performed using artificial neural networks (KIM; AN; CHOI, 2017). However, there are a variety of techniques that can also be applied as linear regression, Markov base methods, probabilistic methods as Bayesian e Fuzzy based methods (KORDESTANI et al., 2019). A data-driven approach is capable of estimating equipment life (SI et al., 2011).

2.2.5.3 Remain Useful Life

The remaining useful life is the lifetime of an asset before a failure 6, being considered one of the key issues when addressing the use of a CBM and PHM system. Estimating the RUL brings a set of advantages that go beyond preventing a failure from occurring. Estimating the moment that a failure will happen allows maintenance planning to be carried out in advance, allowing to order parts for the equipment at the ideal time, avoiding an unnecessary spare parts inventory, and avoiding waiting for parts in case of not possessing. Estimating the RUL also allows the operator of an asset to optimize its maximum in normal condition, being able, in certain scenarios, to save energy and raw materials (SI et al., 2011).

Failure it is just one of the stages that the degradation process of equipment goes through. We can divide it into another three stages of degradation, thus totaling four stages. A system is classified at the stage of *Healthy* when it is operating normally, however with advancing the operating time and useful life of the asset, it is natural for a degradation process to occur, in this degradation process the indexes that measure health enter a descending curve. Depending on the equipment, the threshold values of the indexes used may vary for each determination of stage. After a period, the downward curve of system health enters a regime of *Caution* and, by

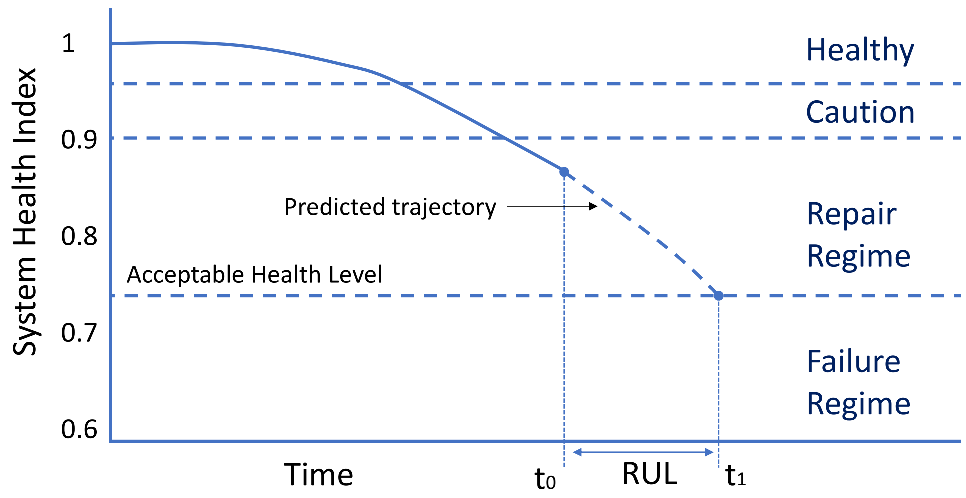


Figure 6: Functioning of Remain useful life, adapted from Xiongzi et al. (2011)

maintaining the health index degradation trajectory, enters into a regime of *Repair*. Eventually, the system will enter the last health stage of failure if in this repair period no action is performed, and the repair does not occur. Estimating the trajectory of the descending curve of equipment health from a pre-failure point to the moment in the future time when a failure occurs is the objective of RUL (XIONGZI et al., 2011).

There are several approaches described in the literature to determine the RUL of an asset. We can classify these approaches as methodologies or as techniques. Among the approaches classified as methodologies, there are the Model-Bases with the application of Statistics and Computational Intelligence(CI), the Analytical-Based approaches that make use of techniques for the physical representation of the failure, the Knowledge-Based approach through the combination of tacit domain experience and Computational Intelligence, and a Hybrid approach with the combination of more than one of the methodologies. Within the techniques applied to determine the RUL, there are the Statistics, with the use of methods such as the autoregressive moving average (ARMA), the Experience approach where judgment is made through tacit knowledge, Computational Intelligence Using Machine Learning Like Artificial Neural Networks, Physics-of-Failure with the use of techniques such as Continuum Damage Mechanics and finally the fusion approach with data fusion and the use of fuzzy methods (OKOH et al., 2014).

2.2.6 Decision and human machine interface

Time-based maintenance (TBM) decisions are generally made by specialists. These types of maintenance are carried out following the manufacturer's recommendations and technical manuals. However, as seen so far, in a CBM system we seek to optimize the use of existing resources, thus indicators are used that enable the detection of failures before their occurrence. This anticipation occurs through, for example, detection and measurement of asset anomalies or trend analysis of some monitored feature, thus enabling the prediction of failures (YAM et al., 2001). Based on a specific diagnosis or prognosis, a set of actions among a set of possible actions previously mapped must be taken. These actions can be at the operational level, where we carry out an intervention in the asset, or at the design level through, for example, replacement or even addition of new sensors (ATAMURADOV et al., 2017).

In the case of an operational action, a decision support system aims to assist the maintenance tasks choices. Within the tasks we can mention, for example, we have the definition of which systems need repair and which must be replaced by determining the security levels that these systems can achieve. A decision support system can also help in managing the supply chain estimating the consumption of materials and equipment to perform maintenance. At the end of the maintenance process, the decision support system can still help to define when a new intervention tends to occur, helping in the planning and scheduling of both material and human resources for the next intervention in the system (TSANG, 1995; KUNDU; DARPE; KULKARNI, 2020).

All information regarding the current status of the monitored equipment can be presented in a graphical interface (ATAMURADOV et al., 2017). In addition, a display and notification system is helpful in the operational control of maintenance, helping to plan maintenance tasks by providing information about when maintenance should be performed and providing data to streamline the maintenance process.

2.3 Machine Learning

Algorithms are part of a set of tools used to solve computer problems. Algorithms have a sequence of instructions that, from a given input, carry out a data transformation process and generate an output. However, there are tasks that conventional algorithms are not able to solve, such as classification, clustering, and prediction tasks. In this set of tasks, where no algorithm solves the problem, we can adopt a strategy that uses a large volume of data to serve as example data. By using this data, it is possible to detect and understand the existing patterns and, through approximations, make predictions. Predictions assume that the future has a pattern similar to past data used as an example (ALPAYDIN, 2014).

To make it clearer, we can mention two examples of tasks that are complex to solve through traditional algorithms. The first example task is the process of identifying and classifying emails as being of type *spam*. Although we have an input to the algorithm, which is the email itself represented as a text file, and we have the desired output, which is its classification, the classification process, when done conventionally, requires that we study all the features that identify that email as *spam* to elaborate a set of rules to be implemented in the algorithm. Creating this set of rules is challenging given the enormous variability of emails. Another example that fits as a challenging problem to solve by conventional algorithms is the indication of handwritten letters. Although there is a limited and known set of existing letters, the variability of their writing makes creating a set of rules to implement a traditional algorithm a complex task (ALPAYDIN, 2014; REBALA; RAVI; CHURIWALA, 2019).

Machine learning (ML) is the field of computing that studies algorithms for solving complex problems that conventional algorithms cannot solve. ML algorithms use a large volume of annotated data to learn, eliminating the need for an explicitly detailed design. The greater the volume of this data, the more assertive the algorithm becomes, as the algorithm seeks to create a model by processing this data. These models make predictions when new inputs are given to the algorithm. This process is called machine learning.

We can separate the learning process into four categories. When the data used for training has an annotation we call supervised learning, the same process with the use of unannotated data we call unsupervised learning, and the implementation of algorithms that consider the use of both annotated and unannotated data at the same time we call semi-supervised learning. Finally, we have the so-called reinforcement learning, which works with penalties and rewards (REBALA; RAVI; CHURIWALA, 2019). We will detail in the next subsections the four ways of training, covering the two types of tasks that an ML model can solve, the classification task and the regression task.

2.3.1 Supervised learning

As already exemplified in the task of classifying e-mails, in a supervised learning process, we give the algorithm a large amount of annotated data. Through this annotated data the algorithm identifies the key features, and through this carry out the learning process. Thus, when new data is made available to the algorithm, it must be able to predict the output based on the previously identified key characteristics (REBALA; RAVI; CHURIWALA, 2019).

Figure 7 shows the supervised learning process, where the training algorithm receives the data originating from the environment, culminating in the generation of prediction models. The model is created to minimize as much as possible the errors between the expected output values (noted in the data) and the realized ones (predicted by the algorithm). The minimization of the error is achieved by adjusting the model's parameters to determine the key characteristics so that, when the model receives the new environmental data, classification based on the key characteristics occurs (JO, 2021).

Supervised learning algorithms fall into two distinct categories, algorithms for solving classification problems and regression problems. In classification problems, the algorithm has the task of classifying something in a group of classes or categories, whereas in regression problems, the algorithm must be able to predict a value of a continuous variable. Each of the categories will be better addressed in the sections 2.3.4 and 2.3.5.

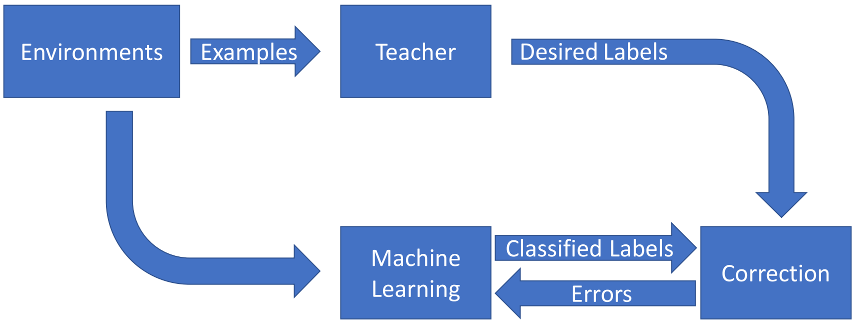


Figure 7: Supervised learning diagram, adapted from Jo (2021)

2.3.2 Unsupervised Learning

In unsupervised learning there are no annotated data, thus the algorithm has no prior knowledge of the correct answer. As there is no correct answer, the algorithm will identify trends or similarities in the data set provided, separating these data into clusters or groups, so when a new input is given, the model will analyze and generate an output based on similarity.

To exemplify this learning process, we can use as an example the way a child learns to use a Blocks Shape Sorter Toy. In this toy, there is a box with a set of holes in specific shapes and several independent block pieces. Although the child does not know the name of the shapes, he knows after a few attempts which blocks can be placed in the specific hole in the box, thus separating the blocks by their shape (REBALA; RAVI; CHURIWALA, 2019).

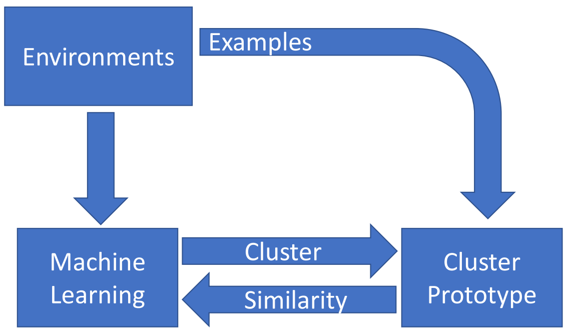


Figure 8: Unsupervised learning diagram, adapted from Jo (2021)

Figure 8 shows the training process by separating objects by similarity or clusters. Initially, depending on the similarity between the training items, the algorithm will optimize the prototype clusters, the cluster number being eventually pre-defined, and the updating of the cluster prototypes are iterated until their convergences. The objective of the algorithm will be to maximize the similarities between the cluster prototypes and the data items (JO, 2021).

2.3.3 Semi-supervised Learning

Figure 9 shows that semi-supervised learning has characteristics that approximate both supervised and supervised learning. This is because in semi-supervised learning a partially annotated dataset is provided. So, the clustering process still needs to occur as in unsupervised learning, in which case the annotated data will provide the annotations for those in the same cluster or group (REBALA; RAVI; CHURIWALA, 2019).

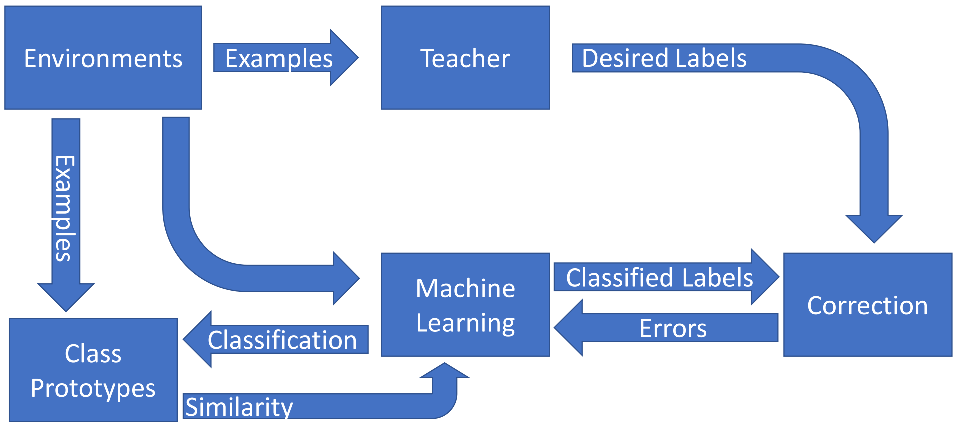


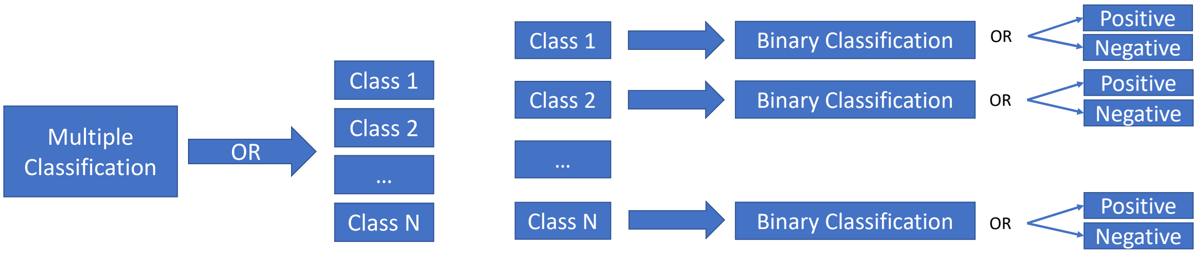
Figure 9: Semi-supervised learning diagram, adapted from Jo (2021)

The advantage of using this approach is that the process of annotating unlabeled data is automated, eliminating the need to manually annotate a large set of data. Furthermore, we use semi-supervised algorithms in both regression and classification tasks. For its implementation, it is necessary to apply supervised and unsupervised learning algorithms (REBALA; RAVI; CHURIWALA, 2019; JO, 2021)

2.3.4 Classification Problems

Classification algorithms are those that have the ability, within a set of classes, to identify and classify an individual by creating a model based on a set of training data. Classification models can be simple threshold values, regression techniques or other learning techniques such as neural networks or random forest (REBALA; RAVI; CHURIWALA, 2019).

We can use the classification process to perform a binary classification, where there are only two classes of items, which is the simplest form of classification. It is also possible to use classification for classes through multi-classification. In the case of multi-classification, there is the possibility of transforming it into a binary classification problem, through the decomposition of the problem. Figure 10 shows the functioning of multi-classification (JO, 2021).



(a) Multiple classification (b) Binary Classification

Figure 10: Multiple classification types

2.3.5 Regression Problems

Regression tasks also apply to supervised learning, in which the objective is the prediction of continuous values, such as predicting the value of a product, based on past values (JO, 2021; REBALA; RAVI; CHURIWALA, 2019). The utilization of a regression model takes place in two ways: to determine if a value will be reached, such as whether a company stock will reach a specific value, through logistic regression, or predict a value of a continuous variable through Linear Regression (REBALA; RAVI; CHURIWALA, 2019).

Furthermore, we use regressions to predict a value using univariate regression or to predict a set of values using multivariate regression. In univariate regression, we can mention as an example the prediction of the value of company stock, using current and past data, this prediction is also called time-series prediction. Now, if we want to predict the value of a set of company stock, which will have more than one continuous variable as output, then we have a multivariate regression. We can separate time series prediction as a specific type of prediction when using past and current measures to predict a value in the future (REBALA; RAVI; CHURIWALA, 2019).

2.3.6 Machine Learning Techniques

The literature already proposes several machine learning algorithms applicable to a series of problems, such as classification and regression. In this subsection, we present two algorithms with different approaches that this work use. A recurrent neural network, which uses a neural network approach, and a random forest, which uses a decision tree approach.

2.3.6.1 Recurrent neural networks

Recurrent neural networks (RNN) is a neural network that uses deep learning to solve problems. Through their ability to remember data processed in the past, RNN is well suited to solving tasks that involve sequential data input. We can cite natural language processing (NLP) as a task with sequence data, where it is necessary to understand the context of a word by looking at the previously processed words and words that will enter in the future. Tasks such as those related to genetic sequencing and in time series, by predicting a future value based on past data are also tasks where RNN can be applied. (REBALA; RAVI; CHURIWALA, 2019; GÉRON, 2019).

The functioning of an RNN is somewhat similar to a natural neural network (NN), having as the main characteristic that differentiates it from a NN the use of values from the past in the activation function. Figure 11 shows how past values impact the process to compute value Y, where do we have $a{<}t-1{>}$ as previously calculated activation value, $X^{<t>}$ the input value, $Y'^{<t>}$ the output value generated, W_x is weight input vector, W_y is weight vector for output vector, W_a is weight vector for activation input. The activation function for input is tanh and Sigmoid for output (REBALA; RAVI; CHURIWALA, 2019).

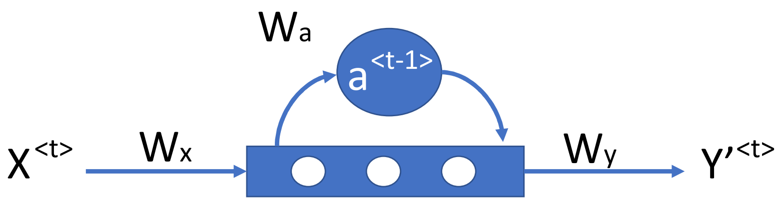


Figure 11: RNN learning process, adapted from Rebala, Ravi e Churiwala (2019)

In an RNN, during the training process, the error values generated by the network are used to adjust the weights through the loss function. A loss function, in this case, is the sum of the loss of each sequence in the network, the output of the RNN is the result of all sequences in the network. This weight adjustment process is done following the gradient descent concept to minimize the loss. This adjustment process at each stage of the network is called backpropagation (REBALA; RAVI; CHURIWALA, 2019).

One of the limitations of an RNN is capturing long-term dependencies, such as in situations where the input sequence is large. For example, in sentences where words that are in different parts of the text and that are related, in this case, due to the problem of Vanishing Gradients, this relationship between the words can, through the process of weight adjustments, be lost. The use of different activation functions minimizes Vanishing Gradients problems. However, other RNNs deal better with long-term dependence (REBALA; RAVI; CHURIWALA, 2019).

2.3.7 Long-Short Term Memory

While the RNN does not learn satisfactorily when there is a lag between the event input and the target signal that is bigger than 5 - 10 discrete time steps. The Long-Short Term Memory (LSTM) can learn with lags bigger than 1000 discrete time steps, solving tasks with long time lags, which until the LSTM proposal, other RNN were not able to solve. This is because it has a gradient learning-based characteristic, which allows considering long-term information becoming able to deal satisfactorily with time series, while a standard RNN uses information from nearby previous tasks as input to the neural network (GERS; SCHMIDHUBER; CUMMINS, 2000).

The memory of the network is the neuron's responsibility, which can allow information to be added or not in the cell through the use of gates. The LSTM cell is composed of a neuron and three types of gates, an input gate, a forgot gate, and an output gate. These gates may or may not allow information to be added An LSTM cell is composed of the following seven components:

- Forgot gate: denoted as f, is responsible for decide if the previous state information will be kept or forgotten.

$$f^t = \sigma(W_{fi}i^t) \tag{2.1}$$

- Input gate: denoted as g, is responsible for decides if a new information enter the cell, has a layer with a tanh function that creates a vector of values for the new candidates.

$$g^t = \sigma(W_{gi}i^t) \tag{2.2}$$

- Output gate: denoted as o, is responsible to decides if the internal state values is passed out to the hidden state in the next step.

$$o^t = \sigma(W_{oi}i^t) \tag{2.3}$$

- Input data: LSTM input data, denoted as x.

- Hidden state: denoted as h, and is used to determine what will be forget, enter, and exit in the next step

$$h^t = o^t \bigodot m^t \tag{2.4}$$

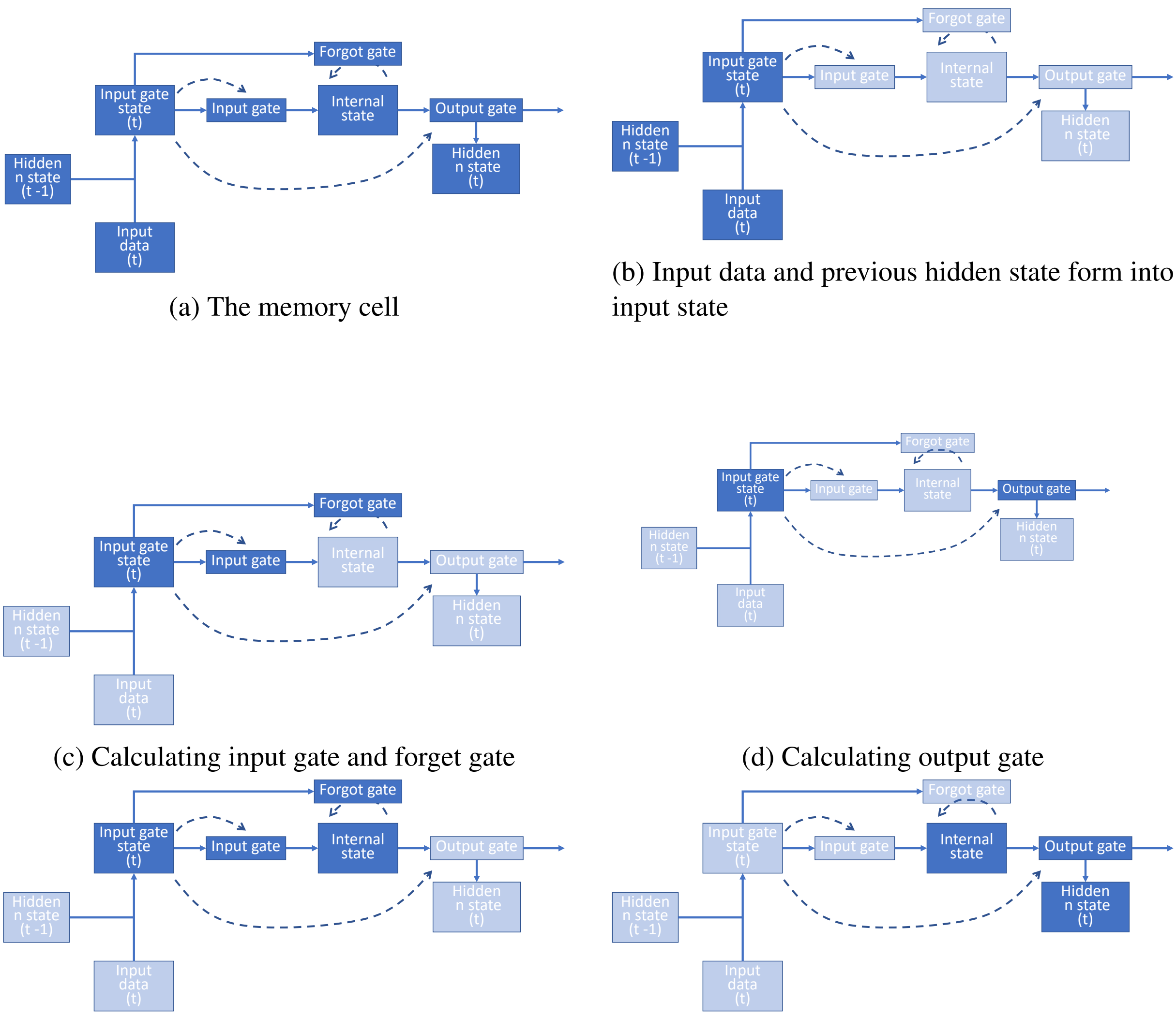
- Input state: denoted as i, is the combination of the hidden state and the current input

$$i^t = \sigma(W_{ix}x^t + W_{ih}h^{t-1}) \tag{2.5}$$

- Internal state: denoted as m, a value that has the memory function

$$m^t = g^t \bigodot i^t + f^t m^{t-1} \tag{2.6}$$

Figure 12 show the functioning of the LSTM memory cell. The construction of the input data using the previous hidden state described in Equation 2.5 is shown in 12 (b). Figure 12 (c) present the calculation of the input gate (Equation 2.2) and the forget gate (Equation 2.1). The output gate described in the equation 2.3) is presented in Figure 12 (d), meanwhile the equation



(a) The memory cell

(b) Input data and previous hidden state form into input state

(c) Calculating input gate and forget gate

(d) Calculating output gate

(e) Update internal state

(f) Output and update hidden state

Figure 12: The working flow of a LSTM memory cell, adapted from Wang e Raj (2017)

2.6) describing the update of the internal state is show in Figure 12 (e). Finally, the update of the output of the hidden state (Equation 2.4) are also presented in Figure 12 (f).

2.3.8 Random Forest

Random Forest (RF) is an algorithm capable of generating accurate classifiers and regressors. It works through the use of a combination of decision trees, where each tree receives sampled input vectors independently and with the same distribution. RF uses a random selection of data to split nodes, with an internal error, strength, and correlation estimators to identify the number of features used in each split and estimate the importance of each variable in classification or regression tasks (BREIMAN, 2001).

As Random Forest works with *n* decision trees, each tree receives a subset of data created from the original set to generate a classifier (bootstrapping), using about two-thirds of the data. This way RF maintains accuracy even when missing data, in addition, it can handle a large amount of data and variables, including being able to identify the variables that have greater weight, that is, it is even able to reduce the number of variables resulting in a decrease in the dimension of the data. At the end of the process, voting of the results of each tree is carried out, which will select the most voted class for classification or average for regression (LIAW; WIENER et al., 2002; BREIMAN, 2001).

As the bootstrap uses two-thirds of the data to create the classification or regression tree, one-third of the data is used to calculate the error obtained with the training data through the so-called out-of-bag (OOB). The error obtained (OOB) in each tree is used to determine the importance of each variable, changing the choice of one variable while the others remain the same. The importance of each variable is difficult to determine, especially when there are complex interactions between each variable Liaw, Wiener et al. (2002).

The information gained in each step is maximized by dividing the nodes, for this, we commonly use two forms of division. The Gini criteria are given by $gini = 1 - \frac{\Sigma}{j} p_j^2$ where p is the probability that we have a given data class in our dataset. The Entropy criteria is given by $entropy = \frac{\Sigma}{j} - p_j * log_2(p_j)$ with the value closest to zero being better. In general, the results obtained by both types, Gini and Entropy, tend to have little difference in the final result. Hartshorn (2016).

2.4 Foundation Models

First introduced by Bommasani et al. (2021), foundation model are trained with a large volume of data. Generally, this training uses self-supervision at scale and can be fine-tuned to perform a wide range of tasks. Figure 13 presents the process of collecting data through a curation process, using this data to train a model and adapt the model to specific tasks.

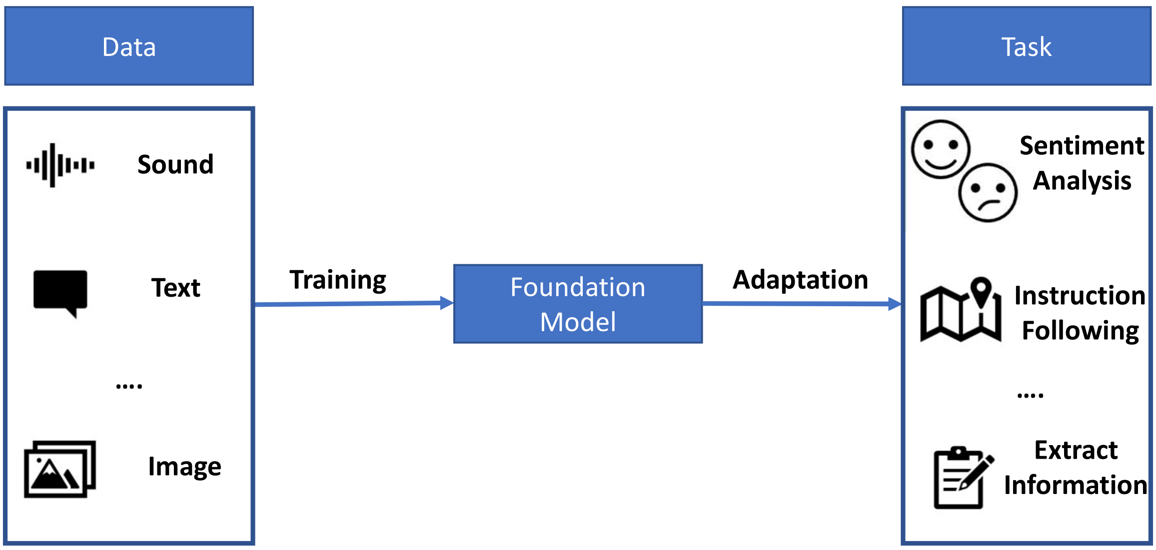


Figure 13: Multimodal foundation models, adapted from Bommasani et al. (2021)

Creation of foundation models is only possible through a pretraining approach using *trans-*

fer learning. This approach allows the use of knowledge learned in a specific task and adapts through fine-tuning it to perform another task. An example of this adaptability is using an image recognition model to perform, for example, the recognition of activities in videos (BOMMASANI et al., 2021).

Scale is of great importance in the feasibility of using foundation models. Access to a large volume of data allows the training of large models. However, it is necessary to have processing capacity for this data, which has been made possible by increased hardware capacity in recent years. Finally, a model architecture capable of using this large volume of data and processing capacity is needed. Transformers models have an architecture that uses the hardware's parallelism capacity to be trained (BOMMASANI et al., 2021).

In addition to *Transformer* and *Scale*, self-supervised learning (SSL) plays a fundamental role in developing foundation models. Cost does not impose a practical limit on the benefits of pre-training, and in self-supervised learning, the pre-training task is automatically derived from unannotated data. Self-supervised tasks are scalable and are designed to force the model to predict parts of the inputs, making the model richer than models trained on a more limited label space (BOMMASANI et al., 2021).

The research fields where foundation models are receiving the most attention in performing tasks are natural language processing (NLP), computer vision (CV), and graph learning (GL) (ZHOU et al., 2023a). Large language models (LLMs) have been developed to solve NLP tasks, which learn to solve tasks through the complex semantic knowledge acquired from large-scale text corpora. For example, the GPT-3 model and its ability to generate text and even snippets of computer programs (BOMMASANI et al., 2021).

However, new areas have been receiving attention from the scientific community in using foundational models, such as time series forecasting. Whether through reprogramming LLM to work with time series or creating specific foundation and general planning models using only (JIN et al., 2023a) time series data.

In this context, figure 14 shows how foundations model can be trained with time series data and readapted for general or specialized uses for specific domains, such as healthcare, finance, and transportation (JIN et al., 2023a).

The first Pretreined Foundation Model (PFM) for time series prediction capable of producing accurate predictions without additional training is TimeGPT (GARZA; MERGENTHALER-CANSECO, 2023). Presented last year, TimeGPT proposes an accessible and accurate PFM and reduces computational complexity and time consumption. TimeGPT works using a Transformer with self-attention mechanisms. The prediction uses a window of historical values, adding local positional coding to enrich the input. The architecture includes an encoder-decoder structure with multiple layers, each with residual connections and layer normalization. Finally, a linear layer maps the decoder output to the prediction window dimension.

TimeGPT model can process time series of varying frequencies and characteristics while adapting to different input sizes and forecast horizons. Even though it is not based on an existing

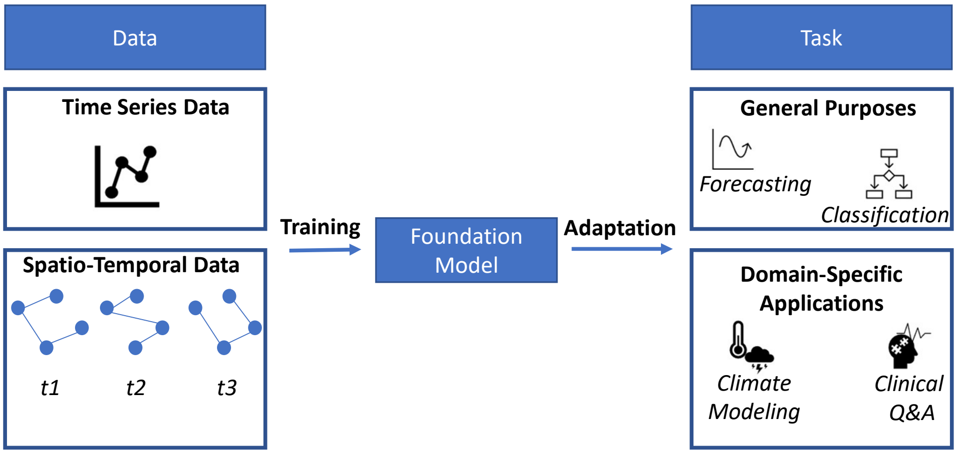


Figure 14: Time series foundation models, adapted from Jin et al. (2023a)

LLM, it follows the principle of training a large transformer model on a vast dataset. The difference is that the TimeGPT architecture is specialized in handling time series data and is trained to minimize prediction error.

Figure 15 presents an overview of the data input for generating time series inference. TimeGPT uses historical data about the target and additional exogenous variables as inputs to produce the forecasts.

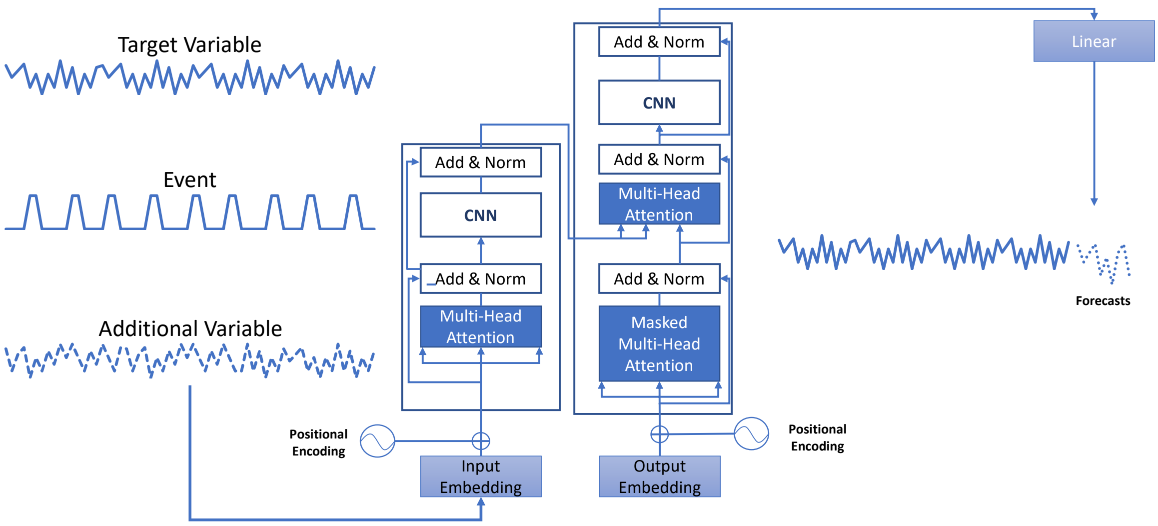


Figure 15: TimeGPT schema, adapted from Garza e Mergenthaler-Canseco (2023)

3 RELATED WORK

In predictive maintenance, numerous challenges are shared in different contexts, such as Industry 4.0 (DALZOCHIO et al., 2020). In addition, the benefits obtained are also similar, such as maintenance cost reduction and predictability in maintenance tasks. However, some of the desired benefits are specific to the military domain. While in Industry 4.0, the aim, in many cases, is to increase productivity, the use of PdM in the military field focuses its attention on tasks such as increase asset availability and vehicle fleet management, preventing a crucial area such as the air defense of a country from being lacking maintenance tasks.

Furthermore, maintenance must be treated as a strategic logistical function because its performance will directly affect the operation of the Force (ARMY, 2009). Because of that, in the upcoming sections, we discuss background aspects of predictive maintenance management in the military context and machine learning in the context of predictive maintenance to contextualize the readers before presenting details on the systematic literature review.

3.1 Predictive Maintenance Management in the Military Context

Maintenance management includes preventive, predictive, and corrective approaches. Preventive maintenance (KOSZTYÁN; PRIBOJSZKI-NÉMETH; SZALKAI, 2019) avoids the deterioration suffered by a piece of equipment. This approach anticipates functional failures in a planned, programmed, and controlled manner (FERNÁNDEZ-BARRERO et al., 2021). Predictive maintenance is the set of routine procedures involving systematic actions aiming to reduce or prevent failures or drop in material performance and reduce the possibility of breakdowns and degradation through inspections, tests, repairs, or replacements (ARMY, 2009). This model involves periodic measurement of the variables that determine the condition of the equipment while it is operating, prematurely detecting failures and developing actions to correct them (FERNÁNDEZ-BARRERO et al., 2021). Corrective maintenance intends to repair or recover the material damaged to put it back in usable condition (ARMY, 2009).

Predictive maintenance in the military context implements a type of maintenance that allows predicting the most appropriate time to perform maintenance activities and, in this way, get as close as possible to the useful life limit of parts and components, optimizing the cost-operation-maintenance triad. Failure prediction in the military domain involves the predictive control of maintenance, with the determination of a predictive point from which the probability of the equipment to fail assumes undesirable values, both on the technical and economic aspects. Moreover, the determination of the predictive point can be assessed in two ways: statistical analysis or symptom analysis (ARMY, 2009). For this reason, failure predictions are an essential part of keeping the state of readiness.

The literature describes several techniques to predict the appropriate time to perform maintenance. Current approaches use methods like the Autoregressive Moving Average (ARMA) or experience-based procedures, considering tacit knowledge, often applying computational intelligence using machine learning (ML) (OKOH et al., 2014). This systematic review investigates the challenges and opportunities of applying PdM in the military context.

3.2 Machine Learning in the Context of Predictive Maintenance

Implementing predictive maintenance tasks, such as estimating the remaining useful life of components, involves enabling the prediction and prognosis of failures. There are several approaches to applying predictive maintenance, using data-driven or physics models. In solutions that use a data-driven approach, successfully implementing these tasks demands monitoring, collecting, and storing data captured from asset sensing. These data are input to an ML algorithm responsible for predicting the health degradation of machines or equipment being monitored (DALZOCHIO et al., 2020).

In a more general perspective, machine learning is well known in the context of predictive maintenance in areas such as aviation (BEHERA et al., 2019), automotive (GIORDANO et al., 2021), and naval (PAL et al., 2019). Examples of these approaches include a Bayesian optimized discriminant analysis for a vertical machining center (BAJAJ et al., 2021), a deep Neural Network classifier in fault diagnosis using power transformers (ANDRADE LOPES; FLAUZINO; ALTAFIM, 2021), and a Merged-long-short term memory network (CHEN et al., 2021) for automobile maintenance.

In model-based approaches, characterized by the use of mathematical models of the system, it is necessary to compare the actual condition of a system with an object model to achieve a model-based diagnostic approach (TINGA et al., 2014). To diagnose a failure using model-based approaches is necessary to describe the physical behavior of degradation or failure modes. As in data-driven solutions, several approaches are viable. In this sense, we highlight the proposal of a new physical model to the application of methods such as particle filtering (PF) for performing prognoses, autoregressive moving-average (ARMA) techniques, Bayesian filtering algorithms, and empirically-based methods (LIN; LUO; ZHONG, 2018; VACHTSEVANOS; VALAVANIS, 2018).

Combining methods through hybrid approaches to improve prediction accuracy is a possibility explored in the literature (WANG, 2018; PAL et al., 2019). However, several challenges exist in predicting failures, as already pointed out in areas such as industry 4.0 (DALZOCHIO et al., 2020). Most of these challenges also apply to the military domain.

Authors have also explored other approaches besides failure prediction, including frameworks and architectures that establish methods to perform predictive maintenance (FERNÁNDEZ-BARRERO et al., 2021), feature modeling proposals (HUANG; LIU; TAO, 2020), or new maintenance policies (TINGA et al., 2020). In this article, we discuss these approaches and their challenges, giving greater attention to how they affect military operations and how to deal with them.

3.3 Research Methodology

The literature review highlights the gaps in a given research area and points to new research opportunities (KITCHENHAM, 2004). It can occur empirically or through applying a method that helps the researcher identify and analyze publications in the area of interest. In this paper, we chose to follow the research method proposed by Kitchenham et al. (2010) to develop a systematic literature review (SLR), applying four steps:

1. **The definition of the research questions:** definition of the general research question to be used in the literature search.
2. **The search process:** databases and string definition for the consultation and development of the search strategy.
3. **The studies selection:** criteria elaboration to determine the relevant papers for the study.
4. **The quality assessment:** quality criteria definition to be applied in the selected studies.

3.3.1 Research Questions

The research question is a part of any research. This paper seeks to identify the open opportunities in predictive maintenance in the military context, with a particular interest in ML applications. In order to define the research questions, we used our previous experience in predictive maintenance to carry out an initial search for its applications in the military domain. Samples extracted from the resulting papers were analyzed to evaluate the potential relevance of conducting an SLR in this field. A preliminary analysis indicated that the area is relevant and, therefore, based on these preliminary results, we elaborated the general research question (GRQ) that guides this work:

What are the challenges and open questions in applying predictive maintenance in military assets?

Based on the GRQ, we proposed the following specific questions (SQ):

- SQ1: What are the principles of PdM used in the context of military environments?
- SQ2: What are the open questions of applying predictive maintenance in a military context?
- SQ3: What are the scenarios that allow the application of PdM in the military context?
- SQ4: Which techniques are used in this context to predict failure?

The purpose of SQ1 is to verify whether it is feasible and how to apply PdM in military scenarios. SQ2 focuses on identifying the challenges of implementing a PdM policy within the military domain. SQ3 intends to discuss applying PdM in the several possible assets operated by the military forces, such as combat vehicles, ships, and airplanes. Finally, SQ4 aims to understand what techniques, methods, and algorithms can predict failures and what role each method plays in the failure prediction process.

3.3.2 Search Process

The first step of the search process is to build a search string (SS). The initial search, described in section 3.3.1, resulted in references that assisted the construction of the research question. The previous step identified keywords to help build an accurate SS. A crucial part of this process involved combining the resulting keywords using boolean operators and synonyms. In order to expand the scope of the search and decrease the possibility of relevant papers in the area not being selected, we created two search strings. In the first search string (SS1), we focus on finding the PdM implementations involving assets in the military context:

("Predictive Maintenance") AND ("Military Vehicle" OR "Military Aircraft" OR "Warships" OR "Artillery" OR "Army")

In the second search (SS2), we focus on finding papers related to machine learning to solve problems in the military domain:

("Machine Learning" OR "Deep Learning") AND ("Predictive Maintenance") AND "Military" AND ("Vehicles" OR "Fighter Aircraft" OR "Warships" OR "Artillery")

The second task of the search process is the database joint (figure 16). After constructing the SSs, the next step is to search and select the scientific databases that will serve as the data source for the queries. We selected electronic databases with journals and conferences where relevant works related to predictive maintenance and machine learning are published. Five electronic databases were selected: IEEE Xplore Library[1], Google Scholar[2], Springer[3], ACM Digital Library[4], and ScienceDirect[5]. Finally, the database joint grouped the results to start the papers selection process.

[1]https://ieeexplore.ieee.org/
[2]https://scholar.google.com/
[3]https://link.springer.com/
[4]https://dl.acm.org/
[5]https://www.sciencedirect.com/

3.3.3 Papers Selection Process

1. **Title and abstract analysis:** consists of reading the title and the abstract to remove publications that are not in the scope of this SLR.

2. **Introduction analysis:** papers that pass the first phase of the analysis have their introduction read. This step avoids the early removal of papers that may be relevant for the SLR but in which the contributions are not evident in the title.

3. **Full-text analysis:** in some cases, the title, the abstract, and the introduction are not enough to evaluate whether the manuscript fits the SLR goals, so a complete analysis of the text is essential.

After the complete text analysis, another analysis is also carried out following three exclusion criteria (EC) to remove those not considered relevant:

- EC1: If the papers are not directly related to PdM.

- EC2: If the papers do not cite the military context.

- EC3: If the papers presented results of surveys or reviews.

Two researchers executed the entire selection process to minimize the possibility of removing relevant work. In case of divergence about some papers, a third researcher was used as a termination criterion.

3.3.4 Quality Assessment

After the filtering application process, the papers were classified according to the predefined quality parameters, following Kitchenham's methodology (KITCHENHAM et al., 2010). In this step, we follow the questions to selected papers to meet the quality requirements (QR) presented in table 1.

Considering the specifications in the table, two researchers scored each selected work independently and compared their results. In case of discrepancies in the evaluations, a third researcher discusses the scores with the two referees and decides the more appropriate score. Next, we apply the criteria presented below to decide which papers should remain in the original corpora and which ones should be removed.

- Articles that generally present opinions and comments of a personal nature, and that do not present a methodology for validation.

- Articles that do not reach a cutoff score of 2.5 points; that is, works that reached half or less than half of the possible score are removed.

Table 1: Questions, descriptions, answers, and scores.

Description	Answer	Score
QR1: Is the purpose of the research presented?		
If the paper explicitly presents the research proposal.	Y (Yes)	1.0
If the research proposal is implicit.	P (Partial)	0.5
If there is no defined research proposal.	N (No)	0.0
QR2: Is there a research methodology presented?		
If the paper presents a methodology for the development of the work.	Y (Yes)	1.0
If the research proposal implies the use of a methodology.	P (Partial)	0.5
If the work does not present a defined methodology for the development of the research.	N (No)	0.0
QR3: Are the research results presented and discussed?		
If the paper generated results from the application of a use case, and these are presented and discussed in the paper, presented conclusions from the results.	Y (Yes)	1.0
If any results, even partially, are presented.	P (Partial)	0.5
If the work does not present results or use cases with practical applications of what is being proposed.	N (No)	0.0
QR4: Is the research context specifically in the military area?		
If the development of the work took place specifically in the military context.	Y (Yes)	1.0
If the research proposal is developed in another area, but there is a relationship with the military area.	P (Partial)	0.5
If there is no relation between the work presented and the military context.	N (No)	0.0

After applying the methodology for selecting works, filtering, and keeping in the original corpora the papers considered most relevant for this study, we read the selected studies again to answer the research questions presented in section 3.3.1. Section 3.4 presented the results.

3.4 Search Results

This section presents the results obtained through searching, selecting, and analyzing the selected papers. Figure 16 shows each step of the process, describing the number of papers selected at each stage until we reach the 50 articles analyzed. The results presented include the two search strings (SS1 and SS2), with the duplicated results already removed.

Following the SLR methodology, section 3.4.1 presents the paper filtering process. We discuss how each filter is applied until we reach the final number of papers applying the quality assessment filter. The results of the application of quality assessment are presented in section 3.4.2.

3.4.1 Search and Selection Processes

Figure 16 presents the filter process result of the selected papers, from the Initial Search (section 3.3.2) until the Full Text Filter (section 3.3.3). After the Initial Search in each research database, the database joint results in 2453 papers. Next, the Impurity Removal process

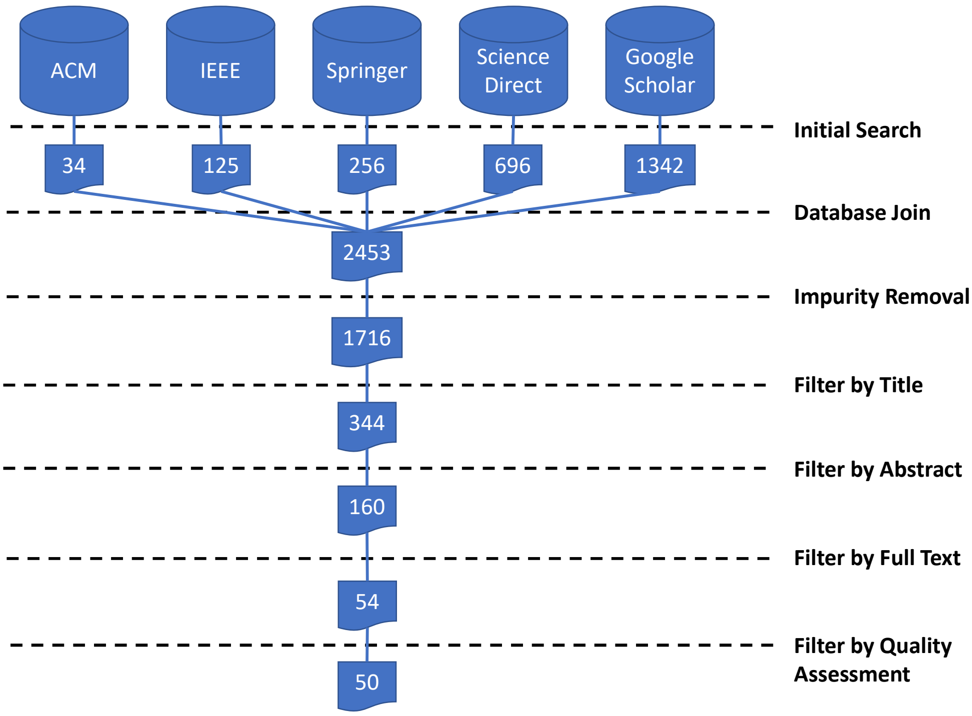


Figure 16: Search, selection, and filter processes of the selected papers

removed non-research papers, resulting in 1716 papers. The non-research papers texts are returned mainly in the Google Scholar results, which largely explains the number of results found in this database compared to the other databases like IEEE. The process of Filter by Title resulted in 344 selected papers, and the Filter by Abstract results in 160 papers. We observed that in these two processes (by Title and Abstract), those papers that quote the words in the search strings were removed mainly because the words just appear in the related works section or reference section of the papers. The papers resulting from the Filter by Introduction were then analyzed in more detail through a complete reading of the works in Filter by Full Text, resulting in 54 papers to begin the quality assessment.

The exclusion process removed high-quality papers from this SLR that did not fit this paper's purpose: Military Employment Material (MEM). The work developed by Leão et al. (2008) proposes a methodology to analyze the cost-benefit of applying PHM in aircraft. As mentioned by the authors, PHM enables the application of PdM. The methodology allows to analyze the cost-benefit of these PHM systems; however, the methodology considers the characteristics of civil aviation and not military aviation. Rui, Xiaofan e Yuhai (2018) work is restricted to proposing an approach to damage estimation and fatigue life prediction without clarifying the viability in the military context. The research conducted by Desell et al. (2014) is also restricted to application in civil aviation, where neural networks are employed to predict flight data parameters, achieving predictive ability and the potential to detect anomalous flights. Such results can be helpful to warn the pilot of possible problems and can be used to predict engine failures and other hardware. Following in the aviation context, it is of great importance to prevent engine failures, given their criticality and the severe consequences involved in possible failures. Wang

(2018) proposes a new prediction method to improve its accuracy. The proposal is limited to specific engine applications and does not explicitly mention its applicability in the military context.

In contrast to works that apply PdM, but not specifically in the military context, we have works like the paper presented by Furch, Nguyen e Glos (2017), where the detection and diagnosis of failures in bearings installed in military vehicles are proposed. However, the work does not advance to the concept of failure prediction. This way, we chose to remove the work from this SLR.

3.4.2 Quality Assessment to Select Relevant Papers

Table 2 presents the results to the quality assessment score, where the papers are classified based on the quality requirements (from QR1 to QR4). It also includes the references, years, and authors.

Table 2: Results of the Quality Assessment Score

Year	**Authors**	**SQ1**	**SQ2**	**SQ3**	**SQ4**	**Score**
2023	Min, Wood e Joo	Y	Y	Y	Y	4.0
2023	Purnama, Susmartini, and Herdiman	Y	Y	Y	Y	4.0
2023	Ulricson, Mickle e Sanders	Y	Y	Y	Y	4.0
2021	Balakrishnan et al.	Y	Y	Y	Y	4.0
2021	Fernandez et al.	Y	Y	Y	Y	4.0
2021	Novoa Paradela et al.	Y	Y	Y	Y	4.0
2020	Ranasinghe et al.	Y	Y	Y	Y	4.0
2020	Tinga et al.	Y	Y	Y	Y	4.0
2019	Peschiera et al.	Y	Y	Y	Y	4.0
2019	Pal et al.	Y	Y	Y	Y	4.0
2018	Cipollini et al.	Y	Y	Y	Y	4.0
2018	Homborg et al.	Y	Y	Y	Y	4.0
2018	Nixon et al.	Y	Y	Y	Y	4.0
2018	Vachtsevanos and Valavanis	Y	Y	Y	Y	4.0
2017	Banghart	Y	Y	Y	Y	4.0
2017	Le et al.	Y	Y	Y	Y	4.0
2014	Banks et al.	Y	Y	Y	Y	4.0
2012	Shao et al.	Y	Y	Y	Y	4.0
2009	Blechertas et. al	Y	Y	Y	Y	4.0
2009	Khatri et. al	Y	Y	Y	Y	4.0
2007	Cook	Y	Y	Y	Y	4.0

Table 2 continued from previous page

Year	Authors	SQ1	SQ2	SQ3	SQ4	Score
2007	Li et al.	Y	Y	Y	Y	4.0
2007	Li et al.	Y	Y	Y	Y	4.0
2005	Roemer et al.	Y	Y	Y	Y	4.0
2014	Tinga et al.	Y	Y	Y	Y	4.0
2002	Byington et al.	Y	Y	Y	Y	4.0
2023	Akrim et al.	Y	Y	Y	N	3.0
2022	Cho, Carrasco e Ruz	Y	Y	Y	N	3.0
2022	Shah et al.	Y	Y	Y	N	3.0
2020	Baker et al.	Y	Y	Y	N	3.0
2020	Huang et al.	Y	Y	Y	N	3.0
2020	Vidyasagar et al.	Y	Y	Y	N	3.0
2019	Behera et al.	Y	Y	Y	N	3.0
2019	Chan and Chin	Y	Y	Y	N	3.0
2019	Ducoffe et al.	Y	Y	Y	N	3.0
2019	Iannace et al.	Y	Y	Y	N	3.0
2019	Kála et al.	Y	Y	Y	N	3.0
2019	Tagliente et al.	Y	Y	N	Y	3.0
2019	Yiwei et al.	Y	Y	Y	N	3.0
2018	Lin et al.	Y	Y	Y	N	3.0
2015	Vali's et al.	Y	Y	Y	N	3.0
2015	Woldman et al.	Y	Y	Y	N	3.0
2013	Tinga et al.	Y	Y	N	Y	3.0
2012	Lall et al.	Y	Y	Y	N	3.0
2012	Lall et al.	Y	Y	Y	N	3.0
2010	Tinga	Y	Y	Y	N	3.0
2008	Babbar et al.	Y	Y	Y	N	3.0
2008	Siegela et al.	Y	Y	N	Y	3.0
2007	Lijun et al.	Y	Y	N	Y	3.0
2001	Boller	Y	Y	N	Y	3.0
2015	Tambe et al.	Y	Y	N	N	2.0
2011	McNaught et al.	Y	Y	N	N	2.0
2006	Rahej et al.	Y	Y	N	N	2.0
2002	Campos et al.	Y	Y	N	N	2.0

Figure 17 shows the scores obtained for each question. One can observe that, of the four specific research questions, the filtering process through the reading of the analyzed papers kept

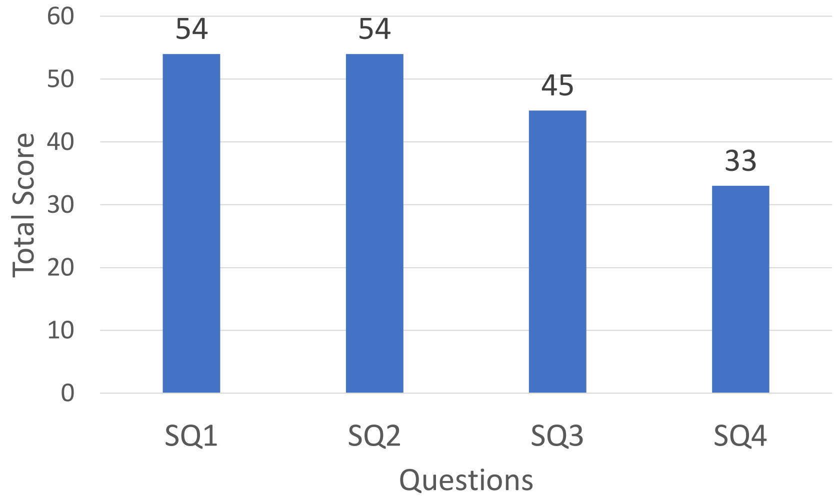


Figure 17: Total score by question

only the papers that present a straightforward research question and a well-defined methodology. The variation in the scores was due to questions about the results presented in the paper and the application context, which should directly involve the military area, not just as a possible area of application. In this way, we removed from the corpus those works that do not entirely satisfy two last questions, so we removed the papers (TAMBE et al., 2015; MCNAUGHT; ZAGORECKI; PEREZ, 2011; RAHEJA et al., 2006; CAMPOS; MILLS; GRAVES, 2002).

Figure 18 shows the selected papers of this SLR by year and the source database, being the y-axis of the figure the amount of paper and the x-axis being the year of the papers. It is possible to observe that although there has been an interest since the beginning of the 21st century, more than half of the selected papers have been published since 2017. We discussed each of the papers presented in the figure in the section 3.5.

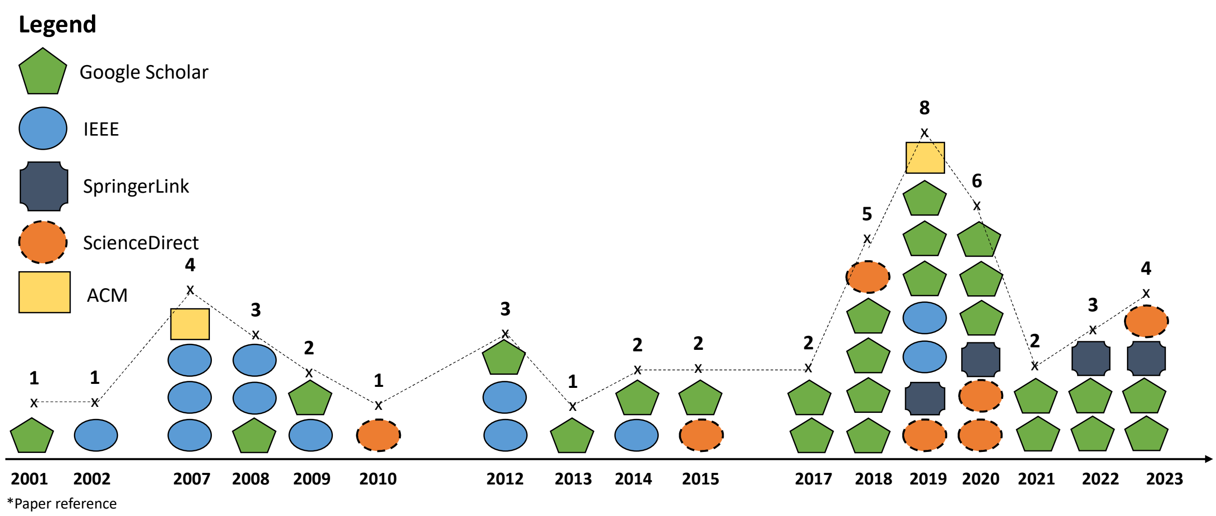


Figure 18: Selected papers by year and database

3.5 Answer to the research questions and discussion

In this section, the questions elaborated in 3.3.1 will be discussed based on the final corpora obtained after applying Kitchenham's methodology (KITCHENHAM, 2004). We applied the SLR methodology as a practical example (KITCHENHAM et al., 2010) to answer the RQ elaborated in Section 3.3.1. In this way, we analyzed the entire corpora considering each RQ, extracting from each selected article the characteristics that match the answers to each question.

As in the article selection process, presented in Section 3.3, at least two researchers analyzed each work to discuss RQs. In cases of divergence about any selected works, a third researcher assisted in the discussion, reducing the possibility of incorrect interpretations and loss of information. Ultimately, we organized the answer in this section, separating each RQ into a specific subsection.

3.5.1 The principles of PdM used in the context of military environments

This research aims to understand which principles must be applied by a military organization to achieve failure prediction capability. In this way, we strive to list the approaches presented in the literature to predict failures and how this translates into benefits for a military organization.

We also present table 3 to group the leading strategies and the works that used each method. Through reading selected works, we identified the approach proposed by each work. As predictive maintenance is a broad subject, the proposals presented in the literature use one or more approaches to achieve the proposed objectives. In this way, through the complete reading of the articles, we identified six main approaches used by the authors to reach the presented objectives in the area of failure prediction. We describe the characteristics of each approach in the description column of the table and the reference column identifying the related research.

We overview the concepts for failure prediction by analyzing the keywords of the works selected in the systematic review of the literature. Figure 19 summarizes our findings by showing keywords that appeared more than once. An important outcome of this figure is the keyword *condition-based maintenance*, which appears in nine of the selected works. Applying a condition-based maintenance policy is one of the possible approaches, which relies on collecting data and training learning models to predict failures. Approaches applied to these tasks will be detailed in this section.

3.5.1.1 Data acquisition

Data collecting is the first task one should perform towards failure prediction. The remaining phases of the process encompass the capture of this data and the transmission, subsequent analysis, and treatment. The need to perform these tasks becomes more evident when predicting failures using ML.

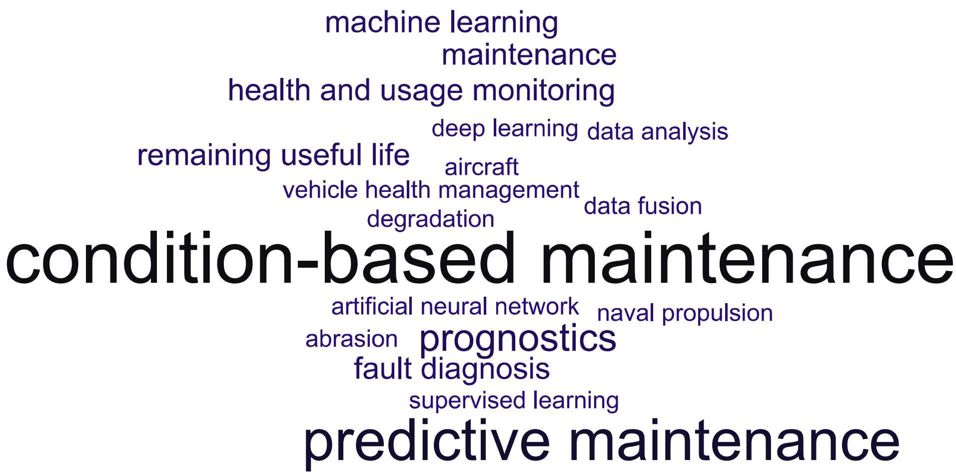


Figure 19: Word cloud from article keywords

Data plays a central role in the so-called data-driven method, where a large volume of historical data feeds the creation of predictive models (YIWEI et al., 2019). In the literature, several approaches are explored for dealing with data. In complex systems, data may be previously available for use, covering several pieces of equipment through the storage of thousands of variables, as proposed by Fernández-Barrero et al. (2021). However, before using the data for training failure detection models, pre-processing may be necessary. For instance, Iannace, Ciaburro e Trematerra (2019) used this pre-processing to prepare the data before training a learning model by applying data normalization.

Data acquisition depends on strategies such as interviews and questionnaires with experts, as proposed by Banks et al. (2014). In the authors' case study, the interviews consisted of questions answered by vehicle maintainers, field service representatives, and vehicle operators. They also apply statistics to infer which components have been replaced most often over consecutive years. Based on the results, they proposed a component reliability indicator. Babbar et al. (2009) uses a similar approach in the aeronautical sector. They collect data to predict the cannibalization of parts from discrepancy reports made by pilots and crew members. In addition, the authors treated outliers manually, requiring the help of experts to identify inconsistencies and going through the pre-processing process to be then used to train the models.

One of the widely used ways to acquire data is sensing equipment performance. There are architectures to assist in this task, such as the Health and Usage Monitoring System (HUMS) (DUCOFFE et al., 2019), the Vehicle Health and Usage Monitoring System (VHMUS) (LE et al., 2017), the Aircraft Condition Monitoring System (ACMS) (BABBAR et al., 2009), and Modern Signal Processing Unit (MSPU) (BLECHERTAS et al., 2009). Ranasinghe et al. (2020) installed sensors in armored personnel carrier vehicles and employed a HUMS architecture to collect data and annotate it for spatial and timing information. An on-board link performs the transmission via the controller area network (CANBus). Data is stored locally, converted into a standard format, compressed, and transmitted in encrypted form over a Wi-Fi network or a commercial 3/4g network.

Embedded sensors, commonly found in diesel engines, are used by Nixon et al. (2018) to estimate the useful life of the equipment. These onboard sensors, when installed in vehicles, are called VHUMS. A VHUMS can collect data such as engine RPM, engine temperature, throttle position, oil temperature, odometer, vehicle speed, fuel usage, and ambient air temperature. Other information, such as chemical conditions of the oil, can also be included for analysis (LE et al., 2017; VALIŠ; ŽÁK; POKORA, 2015). Tagliente, Ludwig e Marston (2019) employed diagnostics and system health (DASH) vehicle health management system (VHMS) on a military vehicle for data collection. DASH is a client-server application that implements common diagnostics, maintenance, and user interaction capabilities. It provides a highly adaptable platform for vehicles and weapons systems that performs tasks like file compression to work on a low bandwidth network. Modern aircraft, such as a Boeing 737, implement an onboard ACMS that collects data such as parameters for Takeoff, Cruise, and Landing flight mode and records the complete set of in-flight parameters. Babbar et al. (2009) combined data obtained from ACMS with information such as problem descriptions and corresponding corrective obtained from Pilot Reports, PreMat Reports, and Delay Cancellation.

MSPU, used by Blechertas et al. (2009) is a data capture system that helps the equipment sensing process. Military helicopters use the MSPU with an onboard sensor for vibration data acquisition and signal-processing equipment for health monitoring critical mechanical components. This data is processed in conjunction with other data collection systems such as the specialized laboratory data acquisition system, recording torque, speed, temperature, vibration, and acoustic emission monitoring.

In some cases, data may already exist but not be accessible before the maintenance process begins on the equipment. In this case, looking for data from other sources is possible. In this sense, Kála, Lališ e Vittek (2019) proposed to search this data in historical files or even in information based on past experiences, such as the required number of hours spent on unplanned maintenance tasks, and equipment data, such as age, cycles, and hours of use. The authors analyze the data using Pearson's correlation coefficient, helping identify variables contributing to particular outputs.

Although it is a crucial phase of the process, obtaining raw data is not always necessary. For example, when proposing a new prediction model, a possible approach is to consider existing datasets. For example, Pal et al. (2019) used public datasets to train artificial neural network (ANN) models. Behera et al. (2019) used the C-MAPSS datasets, made available by the prognostic center of excellence stationed at NASA Ames Research Center.

Predictive maintenance techniques do not always require sensors for data collection. The approach adopted by Tinga et al. (2020) allows the construction of usage profiles to estimate the average time interval. This approach allows optimizing the use of equipment without the need for high investment in sensor acquisition or development, eliminating the need to build datasets with an operation history.

Data collection can be performed in several different ways, depending on the problem the author intends to attack or the scenario in which the collection will take place. Each approach has its benefits and challenges involved. In section 3.5.2 we specifically present data acquisition challenges.

3.5.1.2 Model training

The increased computational capacity has enabled a leap in demand in areas such as predictive analytics, artificial intelligence, and ML. When applied to a large set of data, data-driven techniques give an organization the ability to detect behavior anomalies in assets, classify failures and estimate the equipment's useful life, supplement or even replace the use of physical models (BAKER et al., 2020). Several approaches are available for training ML models. In the literature, solutions like adapting existing algorithms, proposing new ones, or employing hybrid strategies by using algorithms simultaneously are common (FERNÁNDEZ-BARRERO et al., 2021; HUANG; LIU; TAO, 2020; BEHERA et al., 2019). These strategies are necessary because obtaining a model capable of predicting the health condition of an asset is not a trivial task, typically requiring experiments with multiple models to ensure the most efficient strategy (BAKER et al., 2020; SHAH et al., 2022).

The use of data by the learning models must eventually go through pre-processing, where new features are extracted from the data and data preparation is performed. Behera et al. (2019) presented the benefit of such tasks, where a feature engineering process can achieve results with greater accuracy when compared to using the original dataset. Such tasks aim to use higher quality training data, increasing the predictive capacity of the models (IANNACE; CIABURRO; TREMATERRA, 2019). For behavior prediction using supervised learning algorithms, it is necessary to have data labeled with failure behavior. In the absence of labeled data, there are strategies to overcome this problem, such as the artificial generation of data to simulate failures based on previously known failure values (FERNÁNDEZ-BARRERO et al., 2021).

Learning models have been gaining more attention in the last five years. In Subsection 3.5.1.3, we discuss how these models are integrated into solutions for predictive maintenance. We also analyze literature proposals in 3.5.4.

3.5.1.3 Approaches

Using predictive maintenance is challenging, and there is no universal solution or approach. Table 3 presents the approaches identified while reading the related research. The table summarizes these approaches into six groups: (I) Framework, (II) Model-based, (III) Data-driven, (IV) Features Modeling, (V) Maintenance Policy, and (VI) Agent software. The *Framework* approach presents research in which the objective is dealing with complex ecosystems where an architecture helps organize information flows for predictive maintenance.

In the *Model-based* approach, the focus of the research is to build the physical model of a system, which can be applied to estimate the degradation of equipment. In a *Data-driven* approach, there is the creation of models using a large volume of data, which, combined with artificial intelligence techniques, is capable of, among other things, estimating the useful life of the equipment. When we talk about *Feature modeling*, the objective is to improve the estimation of some prediction models through feature extraction techniques. The feature extraction can better represent the equipment's state and improve the models' accuracy.

Policies involving data collection, enabling data-driven approaches such as CBM, and model-based policies, are widely used in the literature, each with its pros and cons. However, new *Maintenance policy* proposals can fill existing gaps. The *Software Agent* approach concerns the approach where the authors propose software solutions that use and practice techniques for predicting failures.

In this context, frequent approaches found in the literature are the proposal of architectures or frameworks to implement predictive maintenance. Fernández-Barrero et al. (2021) presented Soprene, an architecture composed of three main areas, covering data preparation and use in the training and operation modules. Soprene is used to predict failures in military ship engines, where Lamas-López et al. (2022) uses its flexibility and wide range of methods, as it is scalable.

Ranasinghe et al. (2020) propose a framework that uses HUMS to obtain data. The framework aims to analyze the current state of health and estimate the degradation of a power train subsystem of an armored personnel carrier. A feature of HUMS is the ability to apply ML to the data collected to identify trends, perform inferences, and obtain insights into the large volume of data collected. The framework also introduces a virtual dynamometer, where data from several sensors calculate the torque on the motor and thereby estimate its degradation. This approach can be applied in other areas, such as aerospace and defense.

Cho, Carrasco e Ruz (2022) uses a framework previously developed by the authors for detecting failures in a new environment. In this new environment, improvements in data pre-processing, such as cleaning spikes or possible outlines and smoothing time series, show the adaptation framework's capacity. New strategies for estimating the RUL are also part of the work developed, employing techniques such as RNN and a time series decomposition model called Prophet to measure the precision of the RUL.

Lin, Luo e Zhong (2018) presented a combination of two methods for predicting aircraft structures subjected to fatigue loads, the probability–damage–tolerance (PDT) and the model-based particle filtering (PF). This type of strategy has the advantage of, on the one hand, extracting the benefits of each method and, on the other hand, mitigating their limitations. In the authors' proposal, the PF method performs system state (e.g., crack size) estimation in parallel with parameter identification of the prediction model as its function. In complement, the PDT method combines the result of the estimate obtained by the PF to predict the reliability of the structure. Together with the model-fusion framework, the authors propose the multi-objective decision-making model based on condition-based maintenance (MODM-CBM) to minimize

Table 3: Main approaches to failure prediction

Approach	Description	Reference
Frame-work	They present strategies, establishing methods and steps of the failure prediction process. They may cover tasks such as, data collection pre-processing data modeling, use or implementation, new methods of fault diagnosis and prognosis, and decision making.	Fernández-Barrero et al. (2021) Ranasinghe et al. (2020) Vachtsevanos e Valavanis (2018) Tinga et al. (2014) Blechertas et al. (2009) Cho, Carrasco e Ruz (2022)
Model-based	Describe the behavior of equipment through a physical model. From the physical model of a component, it is possible to simulate its uses mechanisms and apply strategies, such as making a prognosis by understanding how a failure progresses.	Yiwei et al. (2019) Woldman et al. (2015) Liu, Cartes e Quiroga (2007) Vališ, Žák e Pokora (2015) Tinga et al. (2014) Shao et al. (2012) Lijun et al. (2007)
Data-driven	Creating physical models of complex systems is not a trivial task. In this way, it is possible to apply a pattern recognition strategy, where a set of data representing the behavior of a device is provided to algorithms such as machine learning. A model that is able to predict the future state of the equipment from the data provided in the past is generated at the end of the process.	Iannace et al. (2019) Pal et al. (2019) Behera et al. (2019) Le et al. (2017) Nixon et al. (2018) Siegel, Ghaffari e Lee (2008) Babbar et al. (2009) Chan e Chin (2019) Lamas-López et al. (2022) Shah et al. (2022) Akrim et al. (2023) Min, Wood e Joo (2023) Ulricson et al. (2023)
Features modeling	Increased accuracy of prediction models can be achieved by improving the quality of data available for learning models. Strategies with Data-fusion and techniques for generating new features can help in the task of improving he accuracy of the models	Raheja et al. (2006) Huang, Liu e Tao (2020) Khatri et al. (2008) Shah et al. (2022)
Maintenance policy	Maintenance policies such as corrective maintenance and preventive maintenance are strategies used today. Condition monitoring-based policy is widely used to enable the ability to predict failures, however, new policies can help in the task of failure prediction.	Tinga (2010) Tinga et al. (2020) Banghart (2017) Kála, Lališ e Vittek (2019)
Software Agent	Implementation of the sets of techniques for the prediction of failures involves the development of software. The results obtained in practice need to be validated through metrics that point out the economic benefits of implementing failure prediction.	Roemer et al. (2005) Liu et al. (2007)

the fleet maintenance cost and maximize its availability through decision-making based on the current state of the structure.

Purnama, Susmartini e Herdiman (2023) presents an approach for predicting failures in radars in the defense sector where a Failure Mode Effect and Criticality Analysis (FMECA) method analyzes all possible losses and then calculates the risk priority number (RPN). FMECA collects all failure mode information from failure data and causes, effects, and risks associated with each component and subsystem represented by the RPN. The RPN is a value that evaluates the criticality of the system to consider the type of maintenance, whether general predictive and preventive or corrective maintenance. Hypertext Pre-processor (PHP) software provides visualization of information to assist in taking action, such as equipment health status and predictions of critical failures that may arise. It guides technicians on which components should be prioritized for maintenance or repair before a problem, a critical failure, or a system failure occurs.

As a way to increase fleet availability and avoid downtime due to the time for parts to be ordered and delivered, Ulricson, Mickle e Sanders (2023) propose a new approach to predictive maintenance. Sequential Pattern Mining (SPM) examines patterns and predicts when and which parts are most likely to fail based on historical data. The authors state that previous studies are limited to the commercial context. In the approach proposed by the authors, the assessment uses a military helicopter. With the use of SPM in the use case, collected maintenance data is analyzed to compare which Work Unit Codes occur most frequently and simultaneously with others, allowing parts to be ordered and replaced preventively, reducing costs and maintenance time.

Vachtsevanos e Valavanis (2018) present an approach for developing critical health management technologies with an emphasis on prognosis, using a test case to evaluate a new approach to an architecture for the implementation of failure diagnosis and prognosis. This architecture consists of the components of integrated Vehicle Health Management (IVHM), Condition-Based Maintenance (CBM), and PHM, achieving an end-to-end approach. The architecture also features Particle Filtering Based Prognostic, providing a long-term prognosis while accounting effectively for uncertainties. Particle filtering combines model-based techniques, data-driven methods, and a Bayesian estimation method to achieve accurate failure detection and RUL prediction. According to the authors, this combination of model-based and data-driven methodologies produces accurate, precise, and robust results.

Other approaches focus on data and its processing for later use by learning models through techniques such as data fusion and data mining. Data fusion automates the process and combines information from several sensors, achieving decision-making based on the state of an object. In the data-fusion process, tasks such as keeping data in units on a common time and unit basis, heterogeneous data integration, and data normalization are performed before creating the learning algorithms (RAHEJA et al., 2006). Huang, Liu e Tao (2020) use the data fusion approach. The authors compare the Joint Directors of Laboratories (JDL) fusion model and the Hierarchical fusion model. Although widely applied in military scenarios, JDL models are limited by the need for human interaction, leading to higher processing time. The Hierarchical fusion model, in contrast, is the mainstream model for multi-source perceptual information fusion modeling. Despite having many layers and suffering from a slow processing speed, the amount of information dramatically reduces as data passes through each model layer, leading to a good performance. Khatri et al. (2008) also focus on data by proposing an approach for generating additional features for use in diagnostic systems. The authors propose decomposition in an empirical way of measuring signals, which differs from the traditional means of obtaining features based on statics of the collected data. This new feature generation approach aims to improve the performance of classifiers applied to the data. In the case of the authors' experiments, three sets of new features are provided from vibration data, complementing the existing features.

There is a class of research works that, instead of analyzing the health degradation of equipment through condition monitoring, focuses on other aspects that impact failure prediction. One of these aspects is evaluating the existing condition-based and load-based maintenance policies, identifying limitations, and proposing new solutions. Authors also seek to optimize the time spent on each revision maintenance. In this context, Tinga (2010) introduce a novel maintenance concept by using a physical model in usage and load-based maintenance concepts. They claim it is unnecessary to monitor equipment's condition in these scenarios, but only equipment usage, through metrics like operating hours, cycles, loads, temperature, and electrical current. Using usage and load-based maintenance concepts in a test case showed that the approach is functional when a usage uncertainty scenario exists. Thus, incorporating the monitored usage or loads into the physical model provides a clear benefit in the prediction accuracy compared to other maintenance models, such as calendar time-based maintenance, behind only the performance presented by CBM. Still working with maintenance policies, Tinga et al. (2020) proposed a maintenance policy with lower technical demand when compared to traditional ones, focusing on specifying the maintenance interval more precisely based on equipment usage profiles. The premise of the authors' proposal is to define a limited number of usage profiles to assess the effects of usage on equipment degradation. The usage profile definition combines the equipment's performed tasks and the operational context, analyzing parameters such as mileage, speed, continuous hours of operation of a vehicle, and the type of operation terrain.

According to Banghart (2017), predictive maintenance relates to the probability of cannibalization, a concept so far not covered in the literature. In the process of cannibalization, one piece of equipment, such as an aircraft, provides parts for another. This cannibalization action maintains the equipment readiness when parts are not available. The authors propose a method to predict which parts will undergo cannibalization, anticipating time-consuming processes, such as reverse engineering, improving the supply chain, and identifying new suppliers. Thus, using methods such as Bayesian predicts cannibalization actions, as demonstrated in the test case presented by the authors. Kála, Lališ e Vittek (2019) proposed maintenance optimization and overhaul time by predicting unplanned tasks or work in the maintenance process. The authors use Maximum Likelihood Estimation (MLE) and Bayesian linear regression modeling to make this prediction, justifying the choice as two common approaches to solving this type of promise. The authors achieved an accuracy of 75% in predicting unplanned work.

Model-based is an approach frequently used in the literature for predicting equipment degradation, being an alternative solution to data-driven techniques, offering superior performance in degradation models. In this way, Yiwei et al. (2019) propose a new model-based prediction method divided into two steps, estimate the fatigue crack size and predict the evolution of the crack size using a new linearization method. The authors' approach differential is the model's ability to consider situations with unknown parameters, where the Extended Kalman Filter (EKF) is used to estimate such parameters. When compared with Monte Carlo methods, the results obtained with the new linearization method were equally efficient, correctly indicating the panels for replacement or repair. But the authors' linearization method presented a lower computational cost.

Lall, Lowe e Goebel (2012a) present a proposal where an EKF is employed to predict the remaining useful life of the BGA components. However, in the proposed approach, the use of EKF is made in a new way, using particle swarm optimization to robustly demonstrate and quantify the repeatability of the resistance spectroscopy measurements and the prognostic monitoring algorithms. The authors created five individual failure experiments using the pre-existing threshold value of data to determine a failure. The authors chose Particle swarm optimization to find an optimum set of parameters to apply on the 'left out' data set used for validation due to its robust ability to cover an unfamiliar optimization space.

Woldman et al. (2015) propose models to describe the abrasive mechanism of sand and how it causes wear through the movement of scratching a sprocket of a military vehicle. This model predicts the amount of abrasive wear. However, its modeling requires detailed knowledge of the wear mechanism since it needs the means to quantify the amount of wear as a function, considering the types of sands and the amount of time and kilometers the vehicle will operate in a given condition. The results show that the model can predict the magnitude of the service life. However, uncertainty in the input values, such as the amount of sand in the contact and the distribution of the driven kilometers over the various operating conditions, may lead to a less accurate prediction of the service life. A proposal using a Model-based approach applied to a permanent magnet synchronous motor, a component of a Navy Ship, is proposed by Liu, Cartes e Quiroga (2007). Vališ, Žák e Pokora (2015) propose using Wiener's process to analyze the presence of particles such as iron and lead in the oil and relate these particles to the degradation of equipment components, indicating which system is deteriorating, anticipating failures.

A failure prediction system implementation that uses a CBM policy requires a collaborative effort between different sectors of society, such as industry, academia, and government. This transition to a CBM policy, including its expansion to the military context, is presented by Blechertas et al. (2009). The authors present the steps involved in implementing a condition-based monitoring system and the techniques and practices to be adopted at each stage. Among the methods, we can highlight the multi-sensor approach and data fusion that significantly improves the robustness and accuracy of fault detection. In this way, the authors give an overview of the implementation process of a monitoring system. The authors exemplify the approach through a use case where the benefits achieved are presented, such as a decrease in maintenance test flight hours, a decrease in unscheduled mechanical components maintenance operations, decrease in replaced mechanical components costs.

Proposals for new physical models and the integration into frameworks to expand their applicability is an alternative, as proposed by Tinga et al. (2014). In this way, the stages of construction of the physical model to achieve the failure prediction can be schematized, passing through the stage of monitoring the equipment. It is not always possible to monitor the condition of the equipment by accessing data like vibration or oil analysis. It is possible to use load monitoring (strain gauges, thermocouples) or usage monitoring (operating hours, rotational speed, power setting, number of starts) to assess the condition of the equipment. The framework validation containing the physical model considers four test cases, which proved to be adaptable to the profile use of each applied system (vehicle, helicopter, and navy frigate), demonstrating increased predictability of failure.

Other approaches apply existing artificial intelligence algorithms. Iannace, Ciaburro e Trematerra (2019) propose a system that uses acoustic measurement data of the noise produced by an UAV to detect an imbalance in a quadrotor's propeller. The author proposes an artificial neural network used to achieve the ability to predict. Pal et al. (2019) also use an artificial neural network, comparing an artificial neural network with Principle Component Analysis (PCA) in predicting failures of a Combined Diesel-Electric and Gas (CODLAG) propulsion plant used in naval vessels. The results obtained show that ANN had better results than ANN with PCA.

Using the well-known Gradient Boosted Trees (GBT) and Random Forest (RF) algorithms, Behera et al. (2019) propose to estimate the RUL of a turbofan aircraft engine. The approach adopted by the authors differs in the aspect of the application, giving an ensemble tree over feature engineered dataset. In the feature engineering process, attributes were added to the dataset, such as statistical measures, an RUL decremented at each time step, and the target attribute, which contains operational status, such as warning and normal.

Le et al. (2017) used a Decision Tree to classify a land vehicle's oil condition into three distinct classes, being *normal, degraded, and unsuitable* reaching a high accuracy in the prediction. The authors also apply ML in the rule extraction stage, using a neural network and a Decision Tree (DT) to extract knowledge from the data, such as the impact of certain input features on the output. The authors' results demonstrate that using a rule extraction step can supplement the prediction capacity of the trained models. Nixon et al. (2018) uses various ML approaches, such as LDA-Naïve Bayes, RF, and Support Vector Machine (SVM) for prediction. Physical model data is grouped with log data to train the algorithms.

In addition to the so-called traditional machine learning methods, we have been following the evolution of applications that use self-supervised learning and various applications in the last year. However, the PdM area still presents a limited number of works that apply self-supervised learning. One of the first proposals is presented by Akrim et al. (2023). The approach investigates using SSL self-supervised learning and its ability to estimate RUL with scarce labeled data. The results presented show promising results, presenting an ability to surpass traditional models.

To demonstrate the benefits of failure prediction for military applications in general, Siegel, Ghaffari e Lee (2008) apply a methodology to assess and predict the health of a military vehicle's alternator using a logistic regression model. The results were promising, and the resulting method proved feasible for application in other components of military vehicles. A data-driven approach using double exponential smoothing is proposed by Babbar et al. (2009) to identify engine health deterioration and link this information to off-board maintenance procedures, helping in the decision-making process and making maintenance a more accurate process. The prognostic approach is based on double exponential smoothing, a smoothed time series prediction approach where older data tends to have less weight than newer data in an older time series. The authors' case study considered that past flight parameters assign confidence intervals to future flight parameters, helping in the maintenance decision-making process according to the projected values in each flight mode.

Virtual prototypes are an approach adopted to perform time series prediction and predict possible failures in the future. This approach adopted by Lijun et al. (2007) uses ADAMS, a platform that simulates a device's operation, gets the change rules, concludes the membership functions, confirms the current failures, and finds out the reasons for the corresponding failures. Shao et al. (2012) propose a virtual prototype for simulation and failure prediction for breech mechanism-based. The virtual prototype allows measuring the signal of velocity, accelerated velocity, and force. These measures enable the acquisition of fault development rules such as the rule of collision force, constraint force, and spring force. The analysis of this data allows simulating the abrasion process and estimating the life cycles of the automatic breech opening process.

Roemer et al. (2005) present a software prototype for the implementation and validation of a PHM system, which includes processes such as pattern definitions, metrics for detection, diagnosis, and prognostics, as well as a forum for exchanging information between users. The software is a response to the lack of methodologies to access PHM's technical and economic benefits. The proposal is validated through a use case in an F-35 aircraft. The advantage of the software is the ability, through plug and play, to collect and transmit data to the evaluation system. Another contribution is the development and evaluation of prognostic metrics considering the maintainer's point of view, in which the focus is on determining when maintenance should be performed and considering the field commander's point of view, who needs to know if an asset will be available for the mission.

Liu et al. (2007) followed the strategy of using software for fault detection, diagnostics, and prognostics. The authors discussed the application of a Software Agent solution on ships to reduce crew requirements, replacing such conditions by performing work tasks by software agents. Boring, repetitive, time-consuming, complex, and analytical tasks can be performed more accurately and reliably by software agents than by people, leaving higher-level tasks such as decision-making in charge of human agents. The authors conducted a case study on a shipboard power system using fault detection and diagnostics utilizing software agents. In this use case, agents were used for monitoring, detecting, and diagnosing, leaving the implementation of an agent to perform the failure prognosis as future work.

3.5.2 The challenges and open questions of applying predictive maintenance in the military context

This research question aims to contribute to the scientific community by unifying challenges and research questions in the use of PdM in the specific domain of military application. We propose a taxonomy (figure 20) to assist researchers in visualizing the challenges and open questions in the use of PdM in the military application domain.

The taxonomy is composed of two distinct components shown in the figure. The components in green are the areas of more comprehensive studies, and each green box is associated with several blue boxes. The blue boxes represent the challenges that we find by following the research questions. This form of organization avoids the duplication of blue boxes, helping to organize the taxonomy. Below each blue box is the reference number to the paper that presents the challenge. We divided the taxonomy into three major areas: (I) Data, (II) ML/Statistic Models, and (III) Others. In the following, each of these areas will be further explored and detailed.

3.5.2.1 Data

Starting with open issues related to *Data*, PdM faces challenges in *Data Acquisition* and *Interoperability*. When monitoring an asset, it is necessary to sense and obtain the data in real-time. However, the dataset adopted for a study can be difficult to obtain in the real world (FERNÁNDEZ-BARRERO et al., 2021; BALAKRISHNAN et al., 2021; CIPOLLINI et al., 2018). Also, real-time access to the asset is not always possible, making sensing unfeasible, so it is necessary to use other data sources such as environmental data or asset's operational data (HOMBORG; TINGA; MOL, 2018). Another critical situation regards components that affect an asset's security. In this case, it is not possible to let the component evolve to failure, thus, making it impossible to collect failure data to train ML models (VIDYASAGAR, 2020).

In addition, when acquiring the data, characteristics such as the quality and accuracy of the data collected must be taken into account (SHAO et al., 2012; YIWEI et al., 2019). Knowing what to monitor in a complex system (LIN; LUO; ZHONG, 2018; BOLLER, 2002), and being

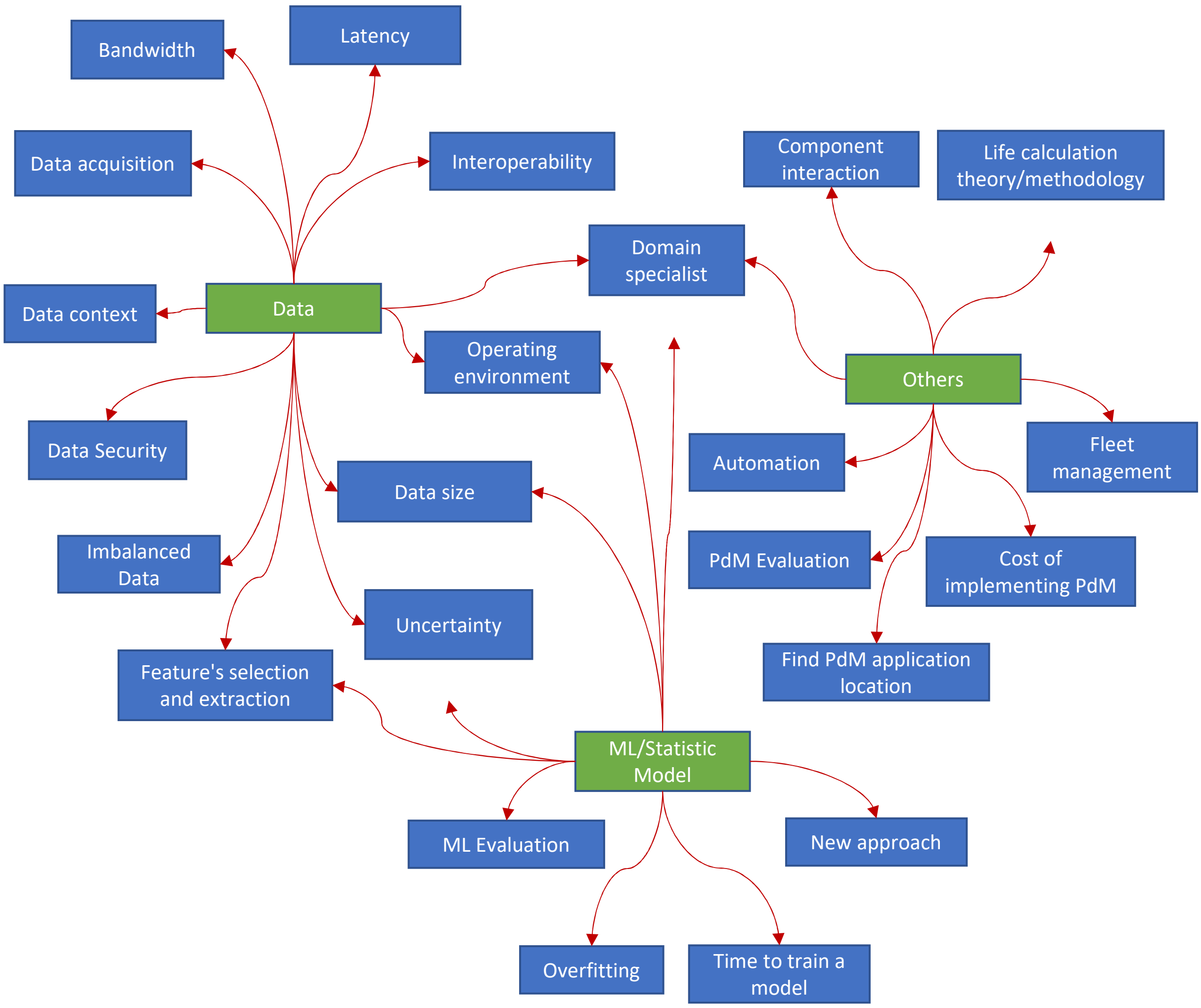


Figure 20: A Taxonomy to classify challenges and open issues in PdM in the military domain

able to relate the asset data to the operations performed at the time of collection (DUCOFFE et al., 2019) are challenges related to *Data Acquisition* which impact the implementation of a PdM system. Another characteristic of *Data Acquisition* is the variety of sensor manufacturers, which, despite having the same purpose, generate different degrees of errors and the possibility of a sensor malfunction, leading to incorrect data. Furthermore, data comes from diverse sources with different standards and formats, making it difficult to use it with data processing tools. Another challenge relates to the fact that sensors operate at different frequencies, making it necessary to determine an ideal transition frequency pattern to avoid data redundancy or lack of data (TAMBE et al., 2015; HUANG; LIU; TAO, 2020). Situations like the ones mentioned make it challenging to apply existing diagnostics architectures in new contexts, hindering the *Interoperability* of the fault diagnosis methods (BYINGTON; ROEMER; GALIE, 2002).

There are challenges and open questions related to *Data Context*, *Imbalanced Data*, and *Data Size*. A common feature of military assets such as vehicles and aircraft is their operation. These assets operate in a variety of environments and, sometimes, in extreme conditions. Collect the *Data Context* is of great relevance. The impossibility of collecting this data may compromise the creation of prediction models. In this case, alternative options such as the creation of physical models are an option (BABBAR et al., 2009; TINGA, 2013; WOLDMAN et al., 2015). In addition, there are cases where the asset is of high importance, affecting the collection of degradation data since it is impossible to assume the risk of operating such assets until the state of failure or degradation. This operational characteristic may lead to the creation of and *Imbalanced Database*, i.e., one composed of a higher proportion of normal operation data than failure-related data. (CHAN; CHIN, 2019; BAKER et al., 2020). These data collection limitations make it challenging to create a dataset with the necessary *Data Size*. Mitigating this problem typically involves creating simulated environments or implementing prototypes to collect data in a controlled scenario. (ROEMER et al., 2005). On the other hand, when large amounts of data are available, other kinds of challenges arise. High data density and high data collection rates may lead to network resource and storage consumption, demanding infrastructure-related investments (BAKER et al., 2020).

Other challenges and open questions are related to *Data Security*, *Bandwidth*, and *Latency*. One of the points of attention in applying PdM within the military domain concerns data security. The use of techniques such as data encryption can minimize security problems. Such security measure becomes even more important when there is the necessity of using civilian networks for data transmission. Ideally, the transmission of sensitive information should happen in military networks. Despite improving the transmission's security, the use of military networks can lead to other challenges, such as *Bandwidth* limitations and increased *Latency* (TAMBE et al., 2015; TAGLIENTE; LUDWIG; MARSTON, 2019).

Next, we have the challenges and open questions related to *ML/Statistic Models* and data availability and quality. Creating learning models with an adequate amount of labeled data for ML training is a complex task (CIPOLLINI et al., 2018). In addition to an appropriately sized dataset, it is essential to ensure that the available data covers all possible failure situations. However, in practice, knowledge about the failures of the monitored asset can be scarce, directly impacting the dataset quality, and consequently on the machine learning-based prediction model (BANGHART, 2017; MCNAUGHT; ZAGORECKI; PEREZ, 2011; AKRIM et al., 2023).

The *Operating Environment* and *Domain Specialist* challenges affect data capture and the creation and training of failure prediction models. Data captured under a given operating condition in the past may not reflect the operating condition in the future (YIWEI et al., 2019). For example, assets such as military aircraft have different operating conditions in takeoff scenarios, cruise flights, and landing. This operational characteristic may lead to the creation of a dataset that does not reflect all assets' operating scenarios (DUCOFFE et al., 2019; SIEGEL; GHAFFARI; LEE, 2008; BOLLER, 2002). In the naval context, proposals for new architectures seek to work in a modular way, aiming to adapt to new environments quickly (FERNÁNDEZ-BARRERO et al., 2021).

Additionally, there is no guarantee that a dataset collected in a vehicle or aircraft reflects the operation of others of the same model, mainly due to the extreme and distinct environments that the same vehicle model can operate (LALL; LOWE; GOEBEL, 2012b; TINGA, 2013; CIPOLLINI et al., 2018; KÁLA; LALIŠ; VITTEK, 2019; BEHERA et al., 2019). Sometimes, such extreme environments are only possible to obtain data by operating in simulated environments (LIJUN et al., 2007). Another challenge involves the complexity of some scenarios, making it necessary to have a *Domain Specialist* both to create prediction models and for data collection. *Domain Specialists* can assist in tasks such as defining metrics and characteristics to be monitored (NIXON et al., 2018) or providing an understanding of the physics involved in the failure to build models capable of estimating the RUL more accurately (TINGA, 2013; MCNAUGHT; ZAGORECKI; PEREZ, 2011). Thus, we increased the possibility of creating more accurate prediction models and threshold values (DUCOFFE et al., 2019).

The challenges and open questions related to *Feature Selection and Extraction* and *Data Uncertainty* are of great interest for training ML models to deal with big data-related problems. Selecting the most relevant features helps create more accurate prediction models (SHAO et al., 2012; TINGA, 2013). In some contexts, new features can be created to enhance prediction accuracy (KHATRI et al., 2008). Exploring new attributes is also necessary when data availability is limited (VALIŠ; ŽÁK; POKORA, 2015). On the other hand, there are scenarios where a wide range of features is available. In these cases, selecting the most critical features is necessary to avoid the problems generated by data excess (TINGA et al., 2014; HOMBORG; TINGA; MOL, 2018; TAGLIENTE; LUDWIG; MARSTON, 2019). Features can receive data gathered directly from sensors. Another possibility for feature creation is the fusion of information coming from multiple sensors. In this case, feature selection can be automatic or manual, depending on the specific characteristics of the ML model. Data fusion and feature selection processes are helpful when there are network connection limitations. They can decrease the dimensionality of the data and speed up the transmission of data collected from the asset (TAMBE et al., 2015;

BANGHART, 2017; HUANG; LIU; TAO, 2020). *Data Uncertainty* also impacts feature extraction. The main reason for that is that data may carry environmental noise. Dealing with data uncertainty involves pre-processing the dataset (HUANG; LIU; TAO, 2020). Uncertainty is an inherent problem of the prognosis task, requiring actions that, for example, improve the signal-to-noise ratio (VACHTSEVANOS; VALAVANIS, 2018).

3.5.2.2 Machine Learning Models

The resulting corpora also brought challenges and open questions related to *Machine Learning Models*, that involve issues like *Machine Learning Model Evaluation*, *Overfitting*, and *Model Training Time*. One can deal with failure prediction in different ways, either by data-dive, model-based, experience-based, or implementing hybrid approaches (BAKER et al., 2020). These processes involve tuning the models to achieve the best possible performance. Results achieved for a model or strategy pass through a *Machine Learning Model Evaluation* and can eventually improve by creating ensembles with other models (IANNACE; CIABURRO; TREMATERRA, 2019; BEHERA et al., 2019; BAKER et al., 2020). Even a standalone model can be tuned to achieve more satisfactory results. However, good metric values are not always an indication of good performance. Frequently, models that achieve high accuracy, for example, are facing problems like *Overfitting* (HUANG; LIU; TAO, 2020) or can demand a long *Time to Train a Model*, making the model unfeasible for application in a real-world scenario (DUCOFFE et al., 2019; PAL et al., 2019; BEHERA et al., 2019; HUANG; LIU; TAO, 2020). The literature deals with these problems by proposing new prediction models, either by incorporating new strategies for the task of monitoring the condition of an asset or proposing the use of new tools or ML approaches in the failure prediction process (LIU; CARTES; QUIROGA, 2007; TINGA, 2010; LALL; LOWE; GOEBEL, 2012a; BALAKRISHNAN et al., 2021; AKRIM et al., 2023).

3.5.2.3 Others

Another perspective highlighted in the corpora analysis regards the *Cost of Implementing* PdM. The literature claims that this cost should be considered to determine where the prediction will be most effective. For example, predicting the failure of a non-crucial component does not necessarily mean that an overall system failure will be predicted (TINGA et al., 2020). Solving this issue generally involves interviewing and using the experience of *Domain Specialist* to decide which pieces of equipment should be monitored (BANKS et al., 2014). Domain Specialists also play a role in correctly understanding and modeling the *Component interaction* failure of one component over another. The task of determining whether the PdM of an asset is viable takes into account the costs involved in the process of implanting the prediction system. This task is vital to avoid unnecessary costs, especially in the military environment where budgets

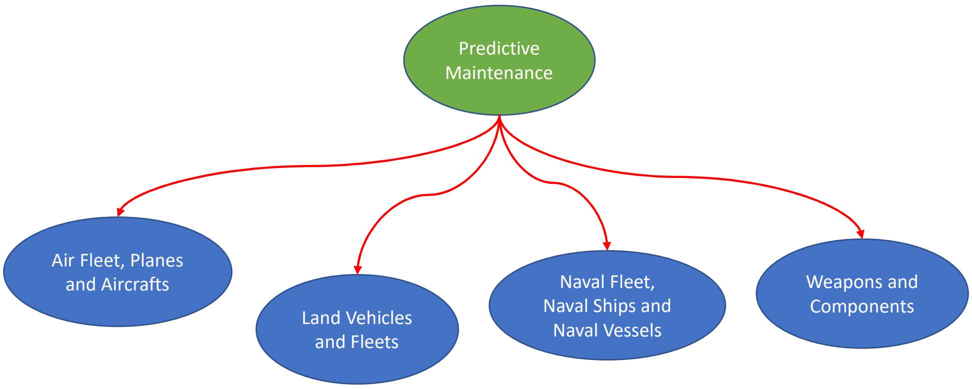


Figure 21: Contexts of use in military environment

are limited (TINGA et al., 2014). Developing effective PdM strategies should take into account the cost as a factor (RAHEJA et al., 2006) and verify the possibility of reusing technologies and standards (CAMPOS; MILLS; GRAVES, 2002). The development of new maintenance policies, such as a policy based on profile usage, is an alternative that eliminates the costs related to sensor acquisition and data collection, which are fundamental in the CBM approach (TINGA et al., 2020).

The *Automation* of detection, diagnostics, and prognostics process through the use of agents to replace human operators is a solution commonly applied to decrease the number of people involved in the maintenance processes (LIJUN et al., 2007). There are also challenges related to *Fleet Management*. Deciding on the right moment for one of these assets to be maintained is crucial for the military operation. Such a decision must consider the fleet's availability and ensure that the asset will be unavailable for the shortest period possible (COOK, 2007; LIN; LUO; ZHONG, 2018; PESCHIERA et al., 2020).

3.5.3 Scenarios of PdM application in the military domain

There are several scenarios in which predictive maintenance is applied, involving the implementation and evaluation of effectiveness (ROEMER et al., 2005). This section we focus on practical applications, such as what type of vehicle or asset applies PdM. One of the most cited areas is aviation, given the severe consequences of any failure occurring. Nevertheless, we also see applications in land and water vehicles domains, according to figure 21.

3.5.3.1 Aerial Vehicles

In the aviation context, scenarios such as the prediction of failures, specific components studies, fleet availability, cannibalization, and monitoring, receive the most attention from the community. As an airplane is composed of several systems and sub-structures, in general, each work concentrates on a specific component, as Yiwei et al. (2019), which focuses on the analysis of the aircraft fuselage, more specifically on fatigue crack propagation in fuselage panels. Structure monitoring is also an approach adopted by Boller (2002); the focus is on monitoring old planes, and building load models since these planes tend to receive updates and operate in different environments and missions than initially proposed in the vehicle's design.

In addition to monitoring the fuselage, it is essential to consider fleet availability before performing maintenance as proposed by Lin, Luo e Zhong (2018), and deal with the long-term Military flight and maintenance planning problem as Peschiera et al. (2020). A feature on airplanes is the cannibalization of other vehicles as a way of supplying parts. Banghart (2017) propose a PdM approach taking into account such characteristics. Min, Wood e Joo (2023) propose and evaluate using a multiple machine learning tool that uses 33 aircraft operated by the US Air Forces to minimize aircraft downtime and predict aircraft failures.

Other authors propose solutions for predicting aircraft engine failures, such as the experiments on a Boeing 747 engine (BABBAR et al., 2009), or focus on identifying problems on maintenance processes. Another approach monitors the airplane's operation (landing, takeoff, and cruise), analyze the health of aircraft engines by monitoring parts conditions (VIDYASAGAR, 2020). Balakrishnan et al. (2021) propose a new solution for monitoring the health of an aero-engine and estimate the remaining useful life of a turbofan engine using a data-driven approach and Shah et al. (2022) apply an ensemble model for failure prediction in a turbofan engine. Research conducted within the US Navy proposed a plug-and-play prognostic solution to monitor gas turbine engines, and gearbox systems of aircrafts (BYINGTON; ROEMER; GALIE, 2002). Also in the aviation context, PdM can be applied to monitor the operation of helicopters (KHATRI et al., 2008; BLECHERTAS et al., 2009; TINGA, 2013; TINGA et al., 2014; DUCOFFE et al., 2019). Similar to the approach used with airplanes, before putting a helicopter into service, it is crucial to consider the availability of the fleet (COOK, 2007). UAV are relatively new in the military context when compared to aircraft. However, they are also the subject of research (IANNACE; CIABURRO; TREMATERRA, 2019).

3.5.3.2 Land Vehicles

In the land vehicle domain, we have a series of approaches in different types of vehicles and different components. Just as in aviation, where the engine receives significant attention, this also happens in land vehicles. Some approaches use the sensors installed in specific engines to estimate the RUL (NIXON et al., 2018), or by monitoring the oil condition to estimate the ideal time for changing (LE et al., 2017). As the vehicles operate in adverse conditions, it is necessary to research models of component failure prediction considering the action of the environment, as in combat vehicles that operate in places like the desert and are exposed to sand and its abrasive action (SHAO et al., 2012; WOLDMAN et al., 2015).

To demonstrate the feasibility of predicting failures in different environments that a vehicle operates, a methodology for the multi-purpose vehicle alternator can be used (SIEGEL; GHAF-

FARI; LEE, 2008). The M109A7 / M992A3 Family of Vehicles are used as a case, exploring the challenges of implementing vehicle monitoring in the military environment (TAGLIENTE; LUDWIG; MARSTON, 2019). Ranasinghe et al. (2020) used an armoured personnel carrier vehicle to assess the health of its power train system and validate the proposed architecture vehicle is used. For the monitoring stainless steel, without a specific a vehicle, can be used as a use-case example (HOMBORG; TINGA; MOL, 2018).

3.5.3.3 Naval Vehicles

PdM also applies to the naval context. Pal et al. (2019) and Cipollini et al. (2018) propose the use of ML to monitor the condition of the propulsion system of a frigate. The air compressor and the advanced carbon dioxide removal unit are the systems that Baker et al. (2020) evaluate the potential of an on-board PHM. The authors' objective is to use existing sensors to deploy a PdM system using ML techniques. Chan e Chin (2019) propose solutions to deal with unbalanced data, using the propulsion system of a naval asset as a use case.

A naval turbine is evaluated by Tinga (2013). Attention is given to operational conditions, the selection of ideal parameters, and how these tasks impact the results of a prediction system. In modern Navy all-electric ships, there is an amount of data flowing that humans cannot process, so agents are proposed to monitor the systems continuously (LIU et al., 2007). Fernández-Barrero et al. (2021) proposed a modular architecture that applies to various naval platform equipment and assets and that is capable of predicting the health of equipment in the future, and Lamas-López et al. (2022) apply the architecture to naval assets.

3.5.4 Techniques Used in the Predict Failure Context

There is no strategy capable of simultaneously solving all problems related to PdM activity. Thus, each scenario has its challenges and requires different approaches. For this reason, we explore model-based methods, machine learning, and deep learning techniques in this section.

In model-based approaches, Yiwei et al. (2019) proposed prognostic tasks to predict the fatigue crack growth evolution in fuselage panels, estimating the size of the crack using data collected from the plane applying the Paris' law crack growth model. Another model-based approach is the creation of physical models. According to Liu, Cartes e Quiroga (2007), developing fault prediction models is challenging since systems and equipment can be very complex and dynamic. Physical models can quantify the degradation of a system through a relationship with its use, as demonstrated in the approaches adopted by Tinga (2013) and Tinga et al. (2014).

Woldman et al. (2015) proposed creating a physical model to predict the amount of abrasive wear. This proposed model integrates into a framework with a usage-based maintenance concept. The usage-based maintenance concept uses parameters such as equipment operation hours, kilometers traveled, and terrain of operation. This way, it is possible to quantify the abrasive action in different vehicles with different application contexts. Shao et al. (2012) also create a physical model to simulate the abrasion process in a specific mechanism.

Vališ, Žák e Pokora (2015) apply a stochastic diffusion model based on a Wiener process to estimate the lifetime of a system. In the case of use presented by the authors, the objective is to optimize the engine oil use time. The presence of particles during the operation is analyzed to achieve the objective proposed by the authors. One of the particles analyzed is iron. For the authors, the problem of iron particle levels observed has a time dependence, thus making it appropriate to apply a diffusion model.

In data-driven approaches, table 4 presents several ML techniques and models proposed to predict failure in the military context. Figure 22 shows the rising interest in using ML, considering papers published between 2017 to 2021, while there is a decrease in the number of works that present a model-based approach for failure prediction.

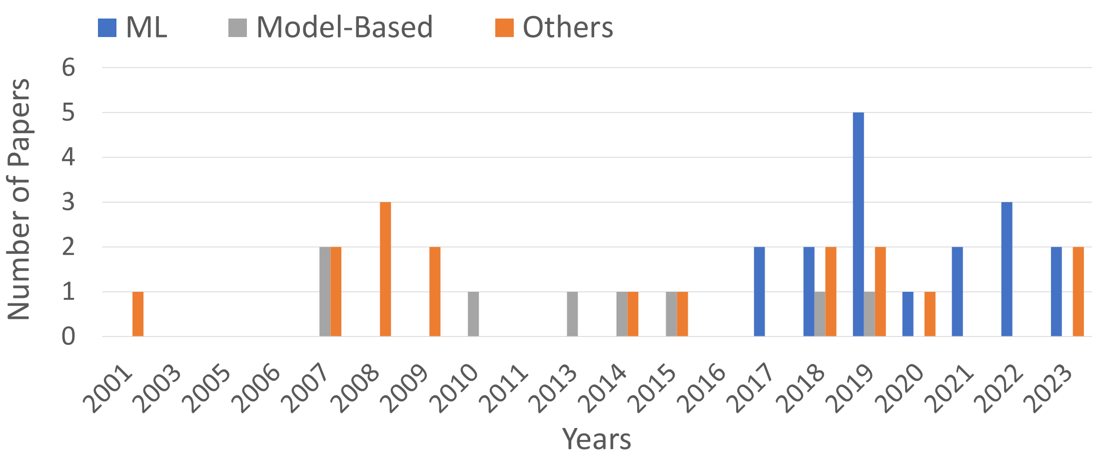


Figure 22: Use of techniques over the years

Nixon et al. (2018) evaluates the possibility of using previously collected monitoring data consumed by the physical model to predict component degradation. The authors use LDA-Naïve Bayes, Random Forest, and Support Vector Machine to classify degradation clusters and their distances to failure moment. The authors' approach combines a classification model that uses Linear Discriminant Analysis (LDA) for subspace creation and dimensionality reduction with a Naïve Bayes classifier trained using the transformed data. LDA maximizes the separation between classes in an optimized subspace in this combination. Naïve Bayes then learns where each degradation class is most likely to be found in that subspace. New observations are first transformed into the subspace learned by the LDA model, and then the probability of association with each class is calculated based on the estimated distributions. The results showed that the combined LDA-naïve Bayes classifier performed better than RF classifier and a SVM classifiers. Behera et al. (2019) use RF to estimate the remaining useful life of a turbofan engine. The authors compare the results with GBT, showing that RF is robust compared to GBT and performs well for RUL estimation. The authors added six new features related to the operating condition to the ML models to improve the classification performance.

Table 4: ML algorithms by paper

Ref.	Title	Year	ML algorithms
Le et al.	Condition monitoring of engine lubrication oil of military vehicles: a machine learning approach	2017	NN, DT
Banghart	Identification of Reverse Engineering Candidates utilizing Machine Learning and Aircraft Cannibalization Data	2017	BN
Nixon et al.	A machine learning approach to diesel engine health prognostics using engine controller data	2018	RF, SVM, LDA-Naïve Bayes
Cipollini et al.	Condition-Based Maintenance of Naval Propulsion Systems with supervised Data Analysis	2018	NN, KM, EM, BM, LM
Ducoffe et al.	Anomaly detection on time series with wasserstein GAN applied to PHM	2019	GAN
Pal et al.	Condition based maintenance of turbine and compressor of a codlag naval propulsion system using deep neural network	2019	DNN
Iannace et al.	Fault diagnosis for UAV blades using artificial neural network	2019	NN
Chan e Chin	Health stages diagnostics of underwater thruster using sound features with imbalanced dataset	2019	ELM
Behera et al.	Ensemble trees learning based improved predictive maintenance using iiot for turbofan engines	2019	RF, GBT
Huang, Liu e Tao	Mechanical fault diagnosis and prediction in IoT based on multi-source sensing data fusion	2020	RBFNN, ENN, BPNN, PNN, FNN, WNN
Peschiera et al	A novel solution approach with ML-based pseudo-cuts for the Flight and Maintenance Planning problem	2020	GBRT, SVR, MLPR, QR, LR, DTR
Novoa Paradela et al.	Predictive Maintenance of Naval Assets Using Machine Learning Techniques	2021	LSTM, LR
Shah et al.	Comparative Study on Estimation of Remaining Useful Life of Turbofan Engines Using Machine Learning Algorithms	2022	DT, RF, XGBoost, LSTM, CNN
Cho, Carrasco e Ruz	A RUL Estimation System from Clustered Run-to-Failure Degradation Signals	2022	LSTM, GRU ESNs
Akrim et al.	Self-Supervised Learning for data scarcity in a fatigue damage prognostic problem	2023	SSL

As a part of a solution to improve the maintenance planning, Peschiera et al. (2020) applied a set of supervised learning methods, such as Linear Regression (LR), Decision Tree Regression (DTR), Multi-layer Perceptron Regression (MLPR), Support Vector Regression (SVR), Quantile Regression (QR), and Gradient Boosted Regression Trees (GBRT), to predict characteristics in optimal or near-optimal solutions before actually solving the problem with a mathematical programming model. A set of supervised learning methods is also applied by Shah et al. (2022) to predict the lifetime of a turbofan engine, comparing the results with a CNN-LSTM ensemble model.

Banghart (2017) explore the use of Bayesian Network (BN), whether to predict the risks of cannibalizing airplane parts using real-world maintenance data or as part of a generic PdM model using Dynamic Bayesian Networks (DBN). The authors discuss the types of knowledge needed to build this model since a DBN needs a wide variety of data and knowledge to create the model. To classify the integrity stage of a propellant, Chan e Chin (2019) proposed a multi-level ridge regression Extreme Learning Machine (ELM). ELM is known for being quick to train and for its level of accuracy. Because it is a supervised learning algorithm, it needs labeled data. However, there are scenarios where the data obtained are unbalanced, with many noisy samples and few useful samples. The multi-level ridge regression ELM method is employed to avoid problems with imbalanced data.

Ducoffe et al. (2019) employed Generative Adversarial Networks (GAN) to detect time series anomalies. The authors present a modification in the GAN to circumvent the mode collapse problem, where the generator is rewarded if it produces good realistic samples, not being encouraged to produce other samples that may be as good for the discriminator as those already found. In this way, the generator learns only to reproduce a small fraction of the dataset variability. To solve this problem, Ducoffe et al. (2019) propose to use the 1-Wasserstein distance to learn the distribution of the dataset directly. The Wasserstein distance is a tool based on the theory of optimal transport to compare data distributions, applied in image processing, computer vision, and ML. The method proved capable of detecting anomalies but required improvement to detect anomalies that are not in the training phase.

Iannace, Ciaburro e Trematerra (2019) applied a Neural Networks (NN) classification model to detect unbalanced blades in a UAV propeller through the sound emitted by blades. The authors chose NN because of their ability to generalize and respond to unexpected inputs. If the network identifies a pattern not associated with a label, then the label least different from the input is selected, exercising the ability to generalize. The NN with only one hidden layer obtained high accuracy, above 97%. Le et al. (2017) chose NN for monitoring engine lubrication oil. The authors compare neural network results with a Decision Tree through a proposal for classifying oil degradation in the *Normal*, *Unsuitable*, and *Degraded* categories.

Huang, Liu e Tao (2020) proposed a simulation model to predict mechanical failure diagnosis. A set of fusion algorithms based on neural networks were evaluated: Back-Propagation Neural Network (BPNN), Radial Basis Function Neural Network (RBFNN), Elman Neural Network (ENN), Probabilistic Neural Network (PNN), Fuzzy Neural Network (FNN), and Wavelet Neural Network (WNN). The authors present the basic concepts, a comparison between the application characteristics of each algorithm, in addition to applying it to a previously existing dataset. The best accuracy results were obtained by the BPNN, while the ENN achieved the lowest accuracy in the tests performed by the authors.

Figure 23 presents each model-based and data-driven technique and each technique's objective within the research. The model-based approach mainly performs the prognostic task, given the characteristic of representing equipment through creating a physical model. From this physical model, it is possible to simulate wear according to the equipment used, such as the environment to which it is exposed, hours of use, or cycles of use.

A data-driven approach encompasses the use given by the model-based, eventually losing some precision and demanding a large amount of data. However, data-driven approaches can play other roles within a predictive maintenance policy, such as classifying and identifying reverse engineering candidates utilizing ML (BANGHART, 2017). As no single approach can solve all problems, the increase in interest seen in figure 22 shows the machine learning ability to solve several problems encountered in applying predictive maintenance policies in the military domain.

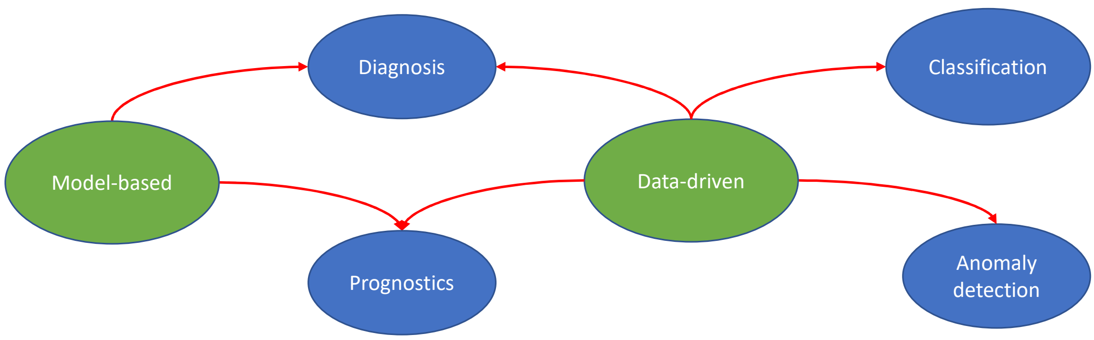


Figure 23: Use of techniques by approach

3.6 PFM in time series forecasting

It is observable that the use of FM has gained substantial attention in recent years. With LLMs, tasks involving NLP, such as text generation, are evolving rapidly. In addition to tasks involving NLP, we have witnessed the rapid evolution of FM for CV and graph learning (GL) (ZHOU et al., 2023a) tasks. However, tasks that involve Time series forecasting are critical. In this sense, we have seen the emergence of the first works that explore the use of pretrained models trained for prediction in this context in the last year.

As the prediction of equipment health degradation using a Data-driven approach involves the ability to predict time series, we searched the literature to see how PFM or FM are being used to predict time series and how this applies to the failure prediction context and predictive maintenance. We used *Google Scholar*[6] as a search database for papers, initially applying the search string *("foundation model" + "time series" + "predictive maintenance")*. As the search results returned a small number of papers, we conducted a new search, removing the term "predictive maintenance" from the search string. The final search string was as follows: *("foundation model" + "time series")*.

This subsection presents a short review of the literature on time series prediction resulting from the search. We separate it into a PFM subsection from NLP and CV and a subsection for PFM for time series. We also add a section with PFM for time series forecasting.

3.6.1 NLP and CV PFM for time series forecast

Applying PFM for NLP and CV in time series prediction tasks is part of the model proposed by (ZHOU et al., 2023b). The model uses frozen pre-trained language mode and, to increase performance, adapters for specific tasks such as anomaly detection. Adapters offer superior performance to state-of-the-art methods.

Time-LLM (JIN et al., 2023b) appears as a framework that adapts the use of LLM for time series forecasting while keeping the backbone model intact. This work presents the new concept of LLM reprogramming for time series, with declarative prompts as input context to guide LLM reasoning. Using a Prompt-as-Prefix (PaP) to enrich the input context and direct the transformation of reprogrammed input patches is presented as a new idea. TIME-LLM proved capable in both few-trial and zero-trial learning scenarios. The results surpass state-of-the-art forecasting methods in both long-term and short-term forecasting. The tests use the Electricity Transformer Temperature (ETT) as a testing basis.

With an approach to training a PFM for Time Series Data with self-supervised learning using the UCR Archive as a training base, the TimeCLR method (YEH et al., 2023a) is proposed. The TimeCLR method is based on SimCLR, a self-supervised pre-training method for computer vision using contrastive learning, which was later extended to a time series of human activities.

TimeCLR adds improvements by augmenting time series data through jittering, smoothing, and magnitude warping techniques. TimeCLR also uses the single augmentation function to generate positive pairs instead of all augmentation functions. The authors compare the results of TimeCLR with alternative methods, such as LSTM and Gated Recurrent Unit Network (GRU), across 128 datasets in the UCR Archive. The results show that TimeCLR outperforms other methods.

LLM4ST (CHANG; PENG; CHEN, 2023) is a framework for time-series forecasting integrated with LLM. The LLM4TS architecture follows a two-stage fine-tuning process that aligns LLMs with time-series data characteristics and then concentrates on time-series forecasting tasks. The model incorporates two Parameter-Efficient Fine-Tuning (PEFT) techniques to increase robustness and versatility: Layer Normalization Tuning and LoRA. PEFT improves the

[6]https://scholar.google.com/

adaptability of pretrained LLMs to time series data without distorting the inherent features.

LLM4TS framework uses GPT-2 as the backbone model, and its evaluation uses seven multivariate time-series datasets: Weather, Traffic, Electricity, and four ETT sets (ETTh1, ETTh2, ETTm1, ETTm2). LLM4TS sets new benchmarks in long-term forecasting and representation learning in the authors' assessment. It also excels in few-shot learning, making it the best choice for real-world scenarios with limited data availability.

3.6.2 PFM for Time Series Forecasting

Following the trend of using PFM from NLP and CV in the time series prediction task, using PFM with a time series database has recently been gaining attention. We have seen the use of PFM to predict time series built for specific domains, such as health (ORTEGA CARO et al., 2023; ZHANG et al., 2023), network security (GUTHULA et al., 2023), whether (CHEN et al., 2023), time series classification (YEH et al., 2023b), and forecasting pixel-level surface reluctance (SMITH; FLEMING; GEACH, 2023).

PFM for general purposes is also being developed, as the PatchTST (NIE et al., 2022) and TSMixer (EKAMBARAM et al., 2023). PatchTST is a model based on transformers designed to predict long-term time series. To address the limitations of other models in capturing local semantic information, PatchTST introduces a patching mechanism, which extracts local semantic information. The model also features a channel-independent design, which enables each series to learn its attention map to improve forecasting accuracy. In a channel-independence, each input token only contains information from a single channel, which works well with CNN and linear models and is now being applied to Transformer-based models.

The advantages of PatchTST are a reduction in time and space complexity, the capability of learning from the longer look-back window, and the capability of representation learning. A use case on a traffic dataset with 862-time series presents the results obtained with PatchTST, achieving state-of-the-art forecasting accuracy.

TSMixer is a model designed for multivariate time series forecasting. It is lightweight and uses the MLP-Mixer architecture. TSMixer has two online reconciliation heads that can improve forecasts by considering the hierarchical patch-aggregation's time series properties and cross-channel correlation. Unlike transformer-based models, TSMixer enhances the learning capability of simple multi-layer perceptron structures.

TimeGPT (GARZA; MERGENTHALER-CANSECO, 2023) is a time series model built on the Transformer architecture, incorporating self-attention mechanisms that utilize a historical value window, enhancing the input by introducing local positional encoding. The model adopts an encoder-decoder structure with multiple layers, incorporating residual connections and layer normalization at each level. A linear layer maps the decoder's output to the forecasting window dimension. The underlying rationale is that attention-based mechanisms capture past events' nuances, allowing for accurate extrapolating potential future distributions.

TimeGPT uses a training dataset of 100 billion data points from weather, finance, economics, health care, demographics, energy, web traffic, sales, transportation, banking, and IoT sensor data. As far as we know, TimeGPT is the first PFM model developed for the time series forecasting context, being able to predict across a diverse array of domains and applications without additional training.

Lag-Llama (RASUL et al., 2023) model is designed for direct probabilistic forecasting, unlike TimeGPT, which built the model using conformal prediction to quantify uncertainty after point-forecasting emissions.

In the Lag-Llama approach, the authors train only a single model with a large corpus of time series. The proposal focuses on univariate probabilistic forecasts, which are simpler than the multivariate case. As the model uses Transformer-based architectures and each dataset has specific frequencies, a general method for vectorizing time series is presented, considering the specific frequency of the data sets that are part of the corpus.

The Lag-Llama formation process is done from all public datasets from the Monash Time Series Repository (GODAHEWA et al., 2021) and additional datasets used in other research. The datasets have different frequencies and are from different domains. Lag-Llama's zero-shot performance surpasses or compares favorably to supervised baselines. As the model size increases, its performance stabilizes and improves across hyperparameter specifications.

Having a model that uses as a base on pretraining a patched-decoder style attention model on a large time-series corpus, (DAS et al., 2023) presents the Pretrained Decoder for Time-series (PreDcT), a time-series foundation model for forecasting. When applied to previously unseen forecasting datasets with different temporal granularities, the Zero-Shot model, also known as PreDcT (ZS), can obtain accuracy close to state-of-the-art zero-shot.

The proposed model has as key elements a time series corpus built using Google Trends and a corrected decoder style attention architecture that can be efficiently pre-trained on this time series corpus. Smaller in parameter size and pre-trained data size than the latest LLM, the model can serve predictions whose zero-shot performance approaches the accuracy of fully supervised approaches on a diverse set of time series data.

The model evaluation uses seven public databases (ETTh1, ETTh2, ETTm1, ETTm2, Wiki, ILI, and TourismL). The results demonstrate that a single pre-trained model can come close to or surpass the performance of baselines on the benchmarks even when the baselines are specially trained or tuned for each specific task.

In all models, preliminary results point to superior performance compared to traditional SOTA algorithms. Furthermore, it demonstrates that the prediction of time series using PFM is a field of research that still needs to be explored. As far as we know, no experiments use PFM in a predictive maintenance context, and our work differs in using a new approach to PdM.

3.7 Related work comparison

This SLR investigated the current state of predictive maintenance use in the military context through a systematic literature review. This investigation focused on the papers in which the authors mention that applying the PdM in the military area is possible. We seek to bring to light the challenges faced in implementing the solutions proposed by the authors in the selected papers and identify the specific application scenarios and techniques or algorithms used by the implementations presented.

We identified an increasing scientific community's attention in the use of PdM in the military context, primarily through machine learning models, but still getting less attention than areas such as the industry (DALZOCHIO et al., 2020). In addition, we see that there are a variety of operating environments and assets employed in the military environment that can employ PdM. So, we have a range of challenges for each scenario and asset monitored, which can be limited to specific scenarios or common to a wide range of other situations. Some of these general challenges are related to data, including data acquisition and feature extraction. However, there are challenges specific to contexts, such as operating environment, latency, and level of fleet availability.

The use of machine learning to predict failures in the military domain has gained greater attention from the scientific community, especially in the past five years. We have identified proposals of techniques and models that aim to predict failures. However, there is no single solution capable of solving all the challenges of all contexts, especially when we consider the complexity of the military ecosystem, the variety of environments, and existing assets that we can monitor in a PdM policy. Therefore, there are some gaps in the maintenance management in the military domain and possibilities to improve the time of failure prediction in the complex maintenance military projects and operations. Because of that, there are possibilities for new proposals in the PdM field of study, either by new learning models that can learn more quickly and with more accuracy, by optimizing existing learning models, or by selecting and customizing learning models for specific pieces of equipment.

Based on the growing interest from the scientific community in applying machine learning techniques in the military PdM context brings a range of new challenges, whether they are related to data capture and treatment or the creation and training of machine learning models. These challenges can be seen as open research fields that may receive attention from the community in the future to improve maintenance management planning, life cycle management of military employment systems and materials, reduce the supply chain, and serve as a basis for the conceptual formulation of future complex projects in the military context. This literature review contributes to the literature in understanding the perspective of the predictive maintenance system, specifically in the military domain. Thus, exploring the challenges in the area, the types of techniques used to deliver a PdM system, and the equipment and vehicles that are likely to receive a monitoring system to perform fault prediction.

Based on the results obtained by SRL, we selected the articles with have more similarity with the proposed work. Table 5 presents characteristics of the proposed approaches, where in the first column the author of the work is presented, and in the nest column if the proposal presents any framework or architecture. The column called generic aims to identify whether in the proposed works the authors consider the proposed solution applicable only to the specific scenario presented or if is possible to be applied in other scenarios. In the communication column we describe what each author proposes in the sense of transmitting the collected data for posterior analysis, such as cybersecurity and quality of service issues. The RUL column classifies whether the proposed work is capable of estimating the remain useful life of equipment. The column of approaches identifies those workers who chose for a model-drive, data-drive, or using both approaches in the proposed solution with a hybrid approach. Finally, we classify the works that are presented with a real-time solution, that is, they use data collected in real time to monitor the equipment.

Nixon et al. (2018) presents a hybrid proposal for the implementation of engine failure predictions using multiple techniques with sensor data already existing in the equipment. The data is captured periodically, with low-bandwidth (sample rate) data, and is stored locally to be sent constantly to a separate database. The authors propose the use of a data-drive approach that classifies the failure in one of multiple failure modes, as each mode has a training dataset to estimate the RUL, after determining the type of failure, a classifier is used to determine the fault type and then estimate the RUL.

As a secondary objective, the authors propose a software framework that orchestrates sensor data, preparing and processing it for a variety of machine learning techniques. The proposed framework is described as a group of Python scripts that automate the training and evaluation process of machine learning algorithms.

Woldman et al. (2015) proposes, through a use case in a sprockets of a military vehicle, the implementation of PdM concepts in a sandy conditions environment. The authors use a model-base approach, building a physical model to predict the amount of abrasive wear. This model is part of a framework used in a test on a Combat Vehicle 90 infantry fighting vehicle.

The results obtained in the use case demonstrate an ability to predict the magnitude of the useful life, but uncertainty in the model inputs, such as amount of area and contact area brings uncertainty to the RUL prediction. However, the authors point out that the main benefit of the proposed method is that changes in vehicle operation, such as sand variety and terrain irregularity, can be directly translated into changes in sprocket life in a quantitative sense.

Tinga et al. (2014) also uses a model-base approach for PdM in a scenario where military systems operate in dynamic ways, with varying uncertainties. In the proposed approach, a physical model, in combination with monitoring the use of systems, helps to reduce uncertainties. Figure 24 shows the framework for the proposed model, where the relationship between the use of the system associated with a physical model to estimate the RUL is presented.

In the use case presented, the framework is applied in 4 different scenarios. The first scenario

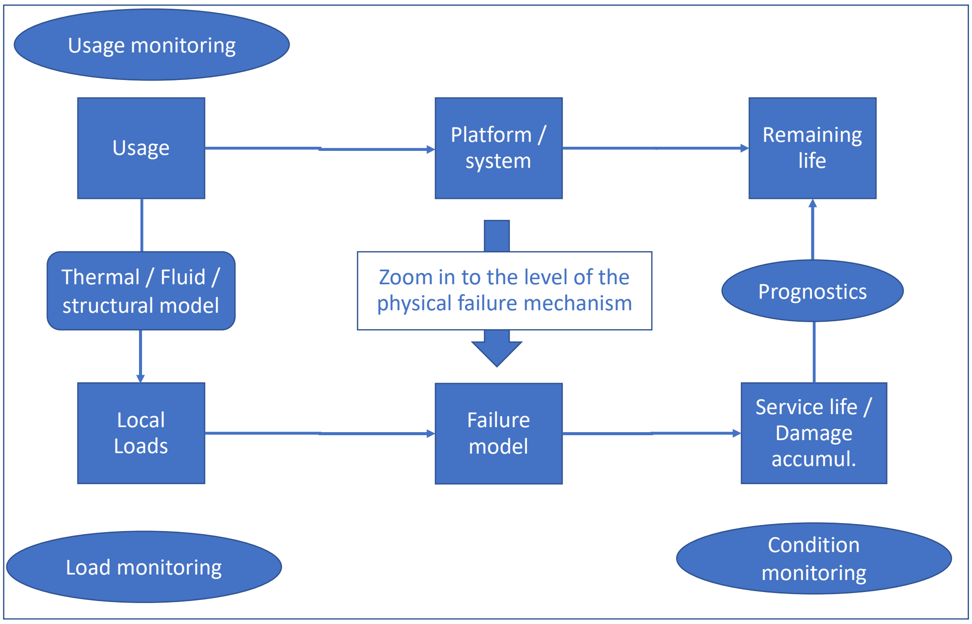


Figure 24: Framework overview, adapted from Tinga et al. (2014)

uses a CV90 infantry vehicle and the abrasion suffered on the sprocket wheel when operating in a sandy environment. In the second scenario, an NH-90 helicopter is utilized, as a focus on the leakage in the landing gear shock absorbers. The third scenario uses frigates, where the usage of the gas turbines is monitored to predict failure. The fourth scenario seeks to monitor the corrosion degradation process by monitoring its condition, not load or usage monitoring. In all scenarios the authors claim to have increased the predictability of failures.

Pal et al. (2019) propose the use of deep neural networks with Principle Component Analysis (PCA-ANN) to predict turbine and compressor failures in a Combined Diesel-Electric and Gas used in naval vessels. The experiments carried out use data from a repository previously made available to the community. The data were applied in the training of neural networks constructed using a variety of architectures as a test to find the ideal architecture for the neural network.

The authors compare the results obtained using the proposed ANN with Principle Component Analysis with an ANN. In all simulated scenarios, with different architectures, the ANN obtained minor errors, demonstrating that ANN outperforms PCA-ANN.

Tagliente, Ludwig e Marston (2019) describes how Diagnostics And System Health (DASH), integrated with M109A7/M992A3 Family of Vehicles, processes locally collected data for sending over low bandwidth networks. Such a tool is important for the data collected to be made available offload, allowing the implementation of CBM systems.

In addition, the authors point to the need to consider cybersecurity aspects when designing the offload strategy, since commercial data networks are not a reality in the military domain, which imposes restrictions on connection availability. These connections are usually made using satellites, through an intermittent and low bandwidth connection. Moving this data to other nodes requires cybersecurity measures to avoid intercepting the data or injecting false

CBM data. Currently the authors are in the early stages, as data become available, predictive models can be developed and such systems can anticipate failures.

Behera et al. (2019) proposed an Ensemble Trees Learning Based approach using data-driven prognostic method for predicting failure in Turbofan Engines. As a use case, a dataset called C-MAPSS was used, a dataset widely used by the academic community for PHM. The use of this dataset allows a comparison of the results obtained with previous benchmarks for RUL estimation, where the authors claim that their approach achieved significantly better results.

The algorithms used are Random Forest and Gradient Boosted Trees, where both obtained competitive results, however, in scenarios where there is a need for high real-time performance, it is preferable to use Random Forest. As an indication for future work, the authors suggest the use of neural networks and deep learning, in addition to relating the operation operations to optimize the RUL prediction.

Balakrishnan et al. (2021) propose a solution to monitor the health of an aircraft engine using Whale Optimization Algorithm based Artificial Neural Network (WOANN), which according to the authors, has not yet been used to predict aircraft engine failures. Data from 47 flights were used, containing eight different engines, and data were collected from both healthy and defective engines. The method can be used in squares for day-to-day monitoring of engine health, in near real-time.

The authors compare the proposed solution with other commonly used algorithms such as k-nearest neighbors algorithm and artificial neural network based on back propagation. As a result, the authors point to a lower error of the proposed WOANN method when compared to existing algorithms.

Fernández-Barrero et al. (2021) present a distributed architecture to handle with the wide variety and complexity of pieces of military equipment, validating the architecture using warships use case. For this, the SOPRENE program was created, which aims to allow the use of the prediction solution in large range of naval equipment, in a horizontally and vertically scalable way. The proposed solution must be able to detect and diagnose failures never before experienced by any ship in the fleet. Such tasks must be integrated within the logistical and operational decision process. The data used by the program is available in a data supervision and analysis center and gathers operational information from each vessel and part. CBM data such as vibration and laboratory oil analysis are also used.

Figure 25 shows a modular architecture, where each piece can be removed without compromising its flow. The architecture is divided into three main parts, the data preparation module being responsible for processing the input data, which involves activities such as structuring the data, cleaning, filtering and storing the data. This data is then sent to the two other parts of the architecture, the operation and the training.

Both operation and training are divided between predicting the future state of the engine, detecting anomalies in that future state, and diagnosing faults. As the architecture is flexible in the prediction technique, the end user can use regression techniques, neural networks, define training data, window size of historical data. This is done because each piece of equipment requires a different approach.

The architecture was tested in two scenarios, Diesel Engine for Propulsion and Diesel Engine for Power Generation where linear models and LSTM were used for failure prediction, obtaining generally good results. However, some open issues are the lack of a real-time operation, being dependent on complex processing being done offline in a centralized datacenter.

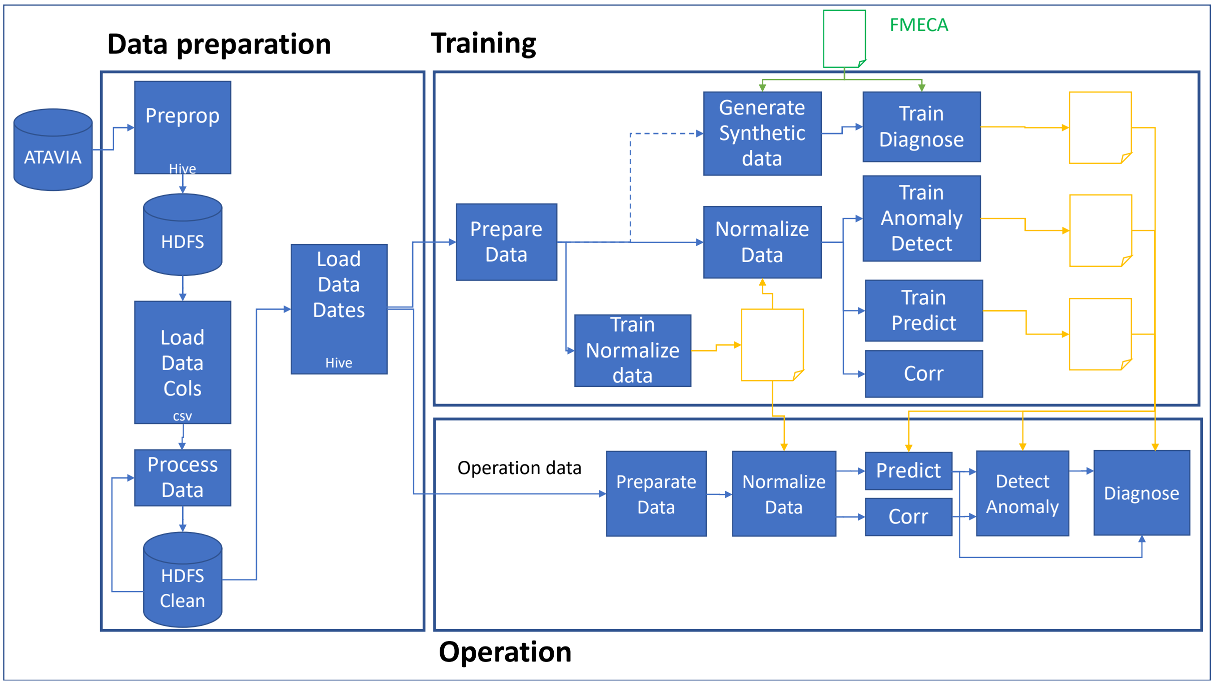


Figure 25: SOPRENE architectural overview, adapted from Fernández-Barrero et al. (2021)

Ranasinghe et al. (2020) presents an architecture for health and usage monitoring system (HUMS) along with diagnostic and prognostic algorithms that use data gathered from a sensor network embedded in a ground-based tracked armored personnel carrier (APC) to assess the health of its power train system. The authors also present a virtual dynamometer that is used to estimate the engine torque output, which is considered to be the primary indicator of engine health, and when used in conjunction with other sensed variables, virtual dynamometers are applied to determine the maximum torque output from the engine.

Figure 26 shows three steps, the Sensor network step is responsible for collecting data from the various sensors installed at the component level, giving this data a sense of location in space and time through the use of the global navigation satellite system. The next part is preprocessing the data, filtering, fusing, and analyzing the data. The data are then used by the next step, to identify speeches and make predictions, allowing the prediction of RUL of the components.

The results obtained by the authors show how the use of several sensors, and data fusion to construct a model-based to the virtual sensor to measure the force, torque, or power generated

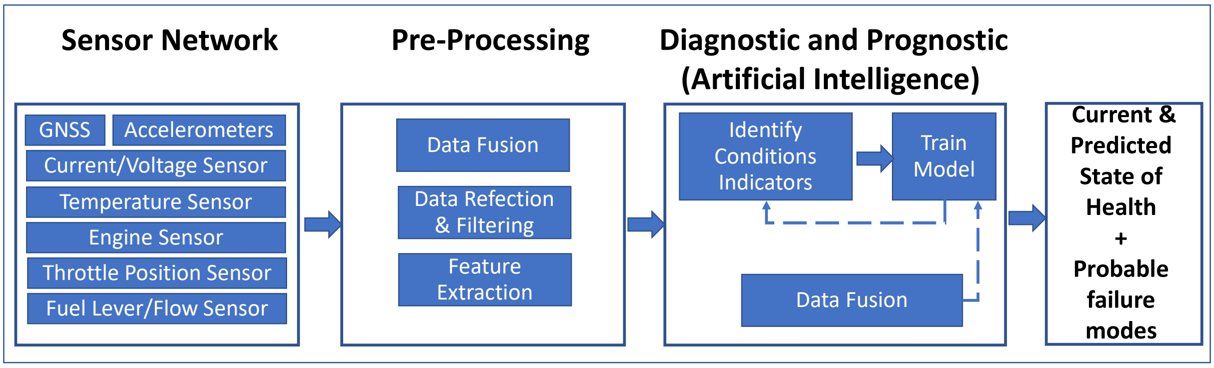


Figure 26: HUMS overview, adapted from Ranasinghe et al. (2020)

by the engine of a vehicle helped in the development of diagnostic and prognostic algorithms for the power train. The next step is the use of digital twins of vehicle systems, with this real-time measurements from sensors as inputs and produce predictions or estimations of the system responses to various inputs and external conditions.

Akrim et al. (2023) presents, to the best of our knowledge, one of the first works that use Self-Supervised Learning in the RUL estimation process. The authors' objective in the work presented is to overcome the challenge of lack of data, and to this end, they propose using Self-Supervised Learning. As mentioned by the authors of the work, the use of Self-Supervised Learning has shown promise in several areas but still presents a limited number of proposals for the task of prognosis. The authors develop the research using aluminum alloy panels subject to fatigue cracks as a use case. The model is trained with a large amount of unannotated run-to-failure time series data, and then fine-tuning is performed with a small amount of annotated data.

Figure 27 presents the scheme for training and fine-tuning the model. The authors selected the Deep Gated Recurrent Unit, which comprises a stack of GRU layers, as the fundamental model for deep prediction. The choice was made due to the sequential properties and positive regressive performance.

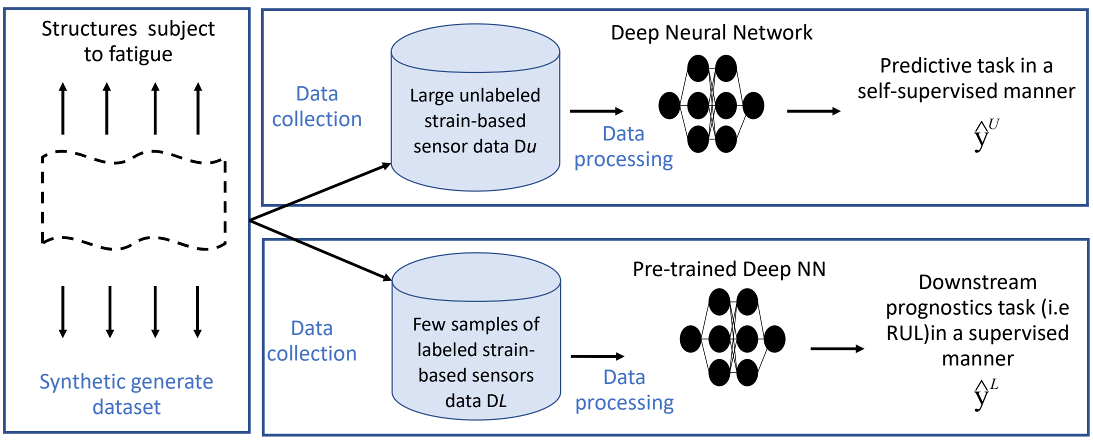


Figure 27: Self-Supervised Learning framework overview, adapted from Akrim et al. (2023)

The results showed that self-supervised learning is efficient in prognosis and can improve the performance of RUL estimation when only a limited amount of labeled data is available. The pre-trained models outperformed the non-pre-trained models in the RUL prediction task, reducing the computational costs of training.

Based on the related work described and highlighting the characteristics of each work, it is clear that, despite being one of the critical steps in a domain of use such as the military, the challenges related to new environments were not addressed. Suppose the issue of simulating the degradation of a vehicle in a new scenario needs to be addressed. In that case, the prediction capacity is compromised, with only one of the related works employing PFM for training in new scenarios similar to existing ones, as shown in the table 5. The approach we propose stands out in tackling the problem of predicting failures with a data-driven approach considering asset performance scenarios, where there may be a need to estimate the RUL of an asset in new environments, where only data from similar environments are available.

Table 5: Comparison with related work

Reference	**Framework**	**Generic**	**RUL**	**Approach**		**PFM**
				Model-drive	**Data-drive**	
Nixon et al. (2018)	Yes	No	Yes	No	Yes	No
Woldman et al. (2015)	Yes	Yes	Yes	Yes	No	No
Tinga et al. (2014)	Yes	Yes	Yes	Yes	No	No
Pal et al. (2019)	No	No	No	No	Yes	No
Behera et al. (2019)	No	No	Yes	No	Yes	No
Balakrishnan et al. (2021)	No	No	Yes	No	Yes	No
Fernández-Barrero et al. (2021)	Yes	Yes	Yes	No	Yes	No
Ranasinghe et al. (2020)	Yes	No	Yes	Yes	Yes	No
Akrim et al. (2023)	No	Yes	Yes	No	Yes	Yes
Proposed approach	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**Yes**

4 MILPDM ARCHITECTURE

This chapter will present the MILPdM, an architecture model for PdM. The first section presents the proposed architecture, detailing the functioning of each layer and its sub-processes. In the second section, we show a case study of time-series predictions through the use of consolidated learning algorithms in the literature. The obtained results demonstrate the feasibility of the proposed prediction of models, making the failure prediction possible.

Figure 28 presents a high-level visual description of the proposed architecture. This architecture is divided into six interconnected layers, the *Private cloud* layer, the *Communication* layer, the *Data collection* layer, the *Processing* layer, *Simulation* layer, and the *Decision-making* layer.

The objective of the architecture is to cover all stages of the failure prediction process, with each of the five proposed layers having distinct, interconnected, and dependent functions on each other. However, our primary focus will be on the private and processing layers for the approach presented here. In these two layers, the difference lies in relation to architectures proposed in the literature, through the prediction of RUL with traditional learning models and the ability to use foundation models in prediction in new scenarios.

Furthermore, the presented architecture covers the entire failure prediction process, so the following subsections detail how each layer works and interacts with each other, enabling a failure prediction system.

4.1 Data collection

The first layer, called data collection, is responsible for maintaining direct contact with the physical world, collecting data that the other layers will use. Sensors connected to military equipment such as vehicles, planes, or ships perform data collection. Monitoring takes place constantly, and data is sent to the real-time prediction system.

The sensors used to monitor an asset are varied, depending on the monitored parameter. The monitoring of a vehicle's engine health uses data from sensors, such as oil viscosity or vibration, either independently or jointly, by using data from more than one sensor simultaneously. As in an operating scenario, assets move dynamically, the number of monitoring assets can change frequently, and the number of sensors collecting real-time information can abruptly scale, so a WSN is used to control this dynamic and heterogeneous environment.

Activities such as filtering, treatments, and data preprocessing can be performed even at the time of data collection. This preprocess allows the use of resources available at the collection site to reduce the dimensionality of the data and its volume. These actions are helpful to mitigate problems that can occur when transmitting a large volume of data, given that an asset operating in the military context can operate in places where access to a data network is limited.

The data sent by the data collection layer, with the information that gives context to the

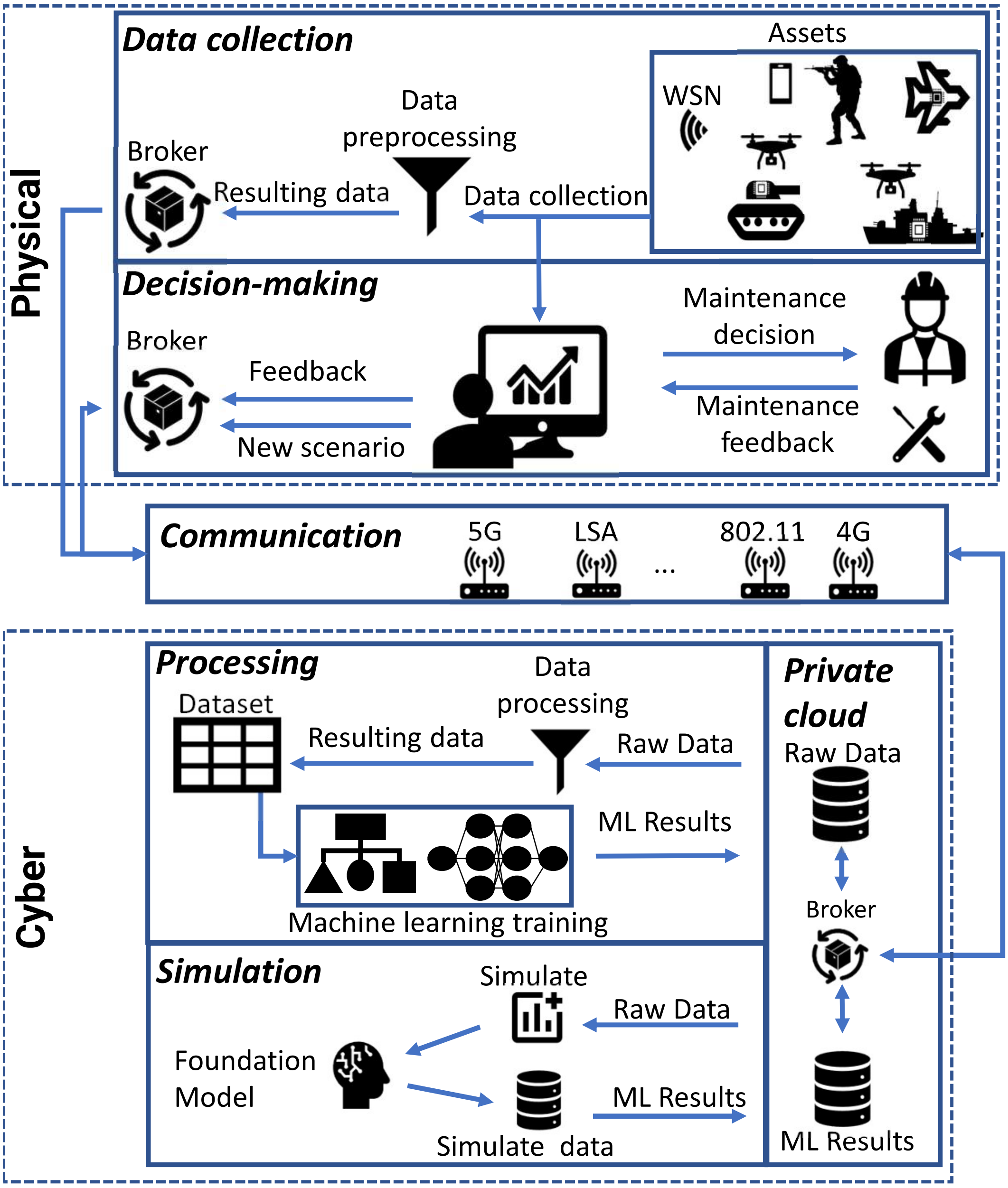


Figure 28: Architecture overview

data, must be stored in a database. It is this context, of knowing which sensors a given data was collected and which asset or component the sensor is monitoring, that allows the construction of more robust failure prediction models, and as a consequence, with greater accuracy in the task of predicting the health degradation of the monitored assets.

This data is then sent to a private cloud database. Given an environment with possible infrastructure limitations, like no cable connections, the sending of data can be made by a wireless network through the use of a broker that ensures access to a heterogeneous wireless network with quality of service (QoS) (KUNST et al., 2016). All the data stored in the cloud is made available for use in the remaining layers of the architecture.

4.2 Processing

In the Processing layer, the monitoring data captured by the data collection layer and stored in the private cloud is processed and then used as a dataset input for the failure prediction models. The prediction model does not use the original data that comes from the data collection layer, but the previously processed data. This task is essential when dealing with a large volume of data, creating new features from a dataset, or reducing its dimensionality. However, we do not rule out the use of the original data, being always necessary to understand the context in which the system is operating to create the prediction model.

There is a wide range of algorithms proposed in the literature that have the objective of predicting failures. We can mention recurrent neural network algorithms, such as the long short-term memory (LSTM), and decision tree algorithms like random forest (RF). The LSTM algorithms have as a characteristic the ability to handle time series, which is due to the ability to consider the weight of each input for a longer time within the neural network through the use of so-called gates.

Unlike LSTM, which is a sub-type of a neural network, RF is an algorithm that uses decision trees to solve classification tasks, and in the case under study, to solve regression tasks. Both LSTM and RF are widely used in the literature for the task of time series prediction (DALZOCHIO et al., 2020).

The training of LSTM and RF algorithms allows informing a series of parameters, and the training of each algorithm uses a range of values and combinations of different parameters. The application of the hyperparameter process to training allows training the LSTM and RF in a variety of models to find the model capable of predicting with the less possible error. The decision-making layer receives the prediction results of the best model and, if necessary, the decision-making layer can also receive the dataset, where the maintenance team can follow the evolution of asset degradation.

4.3 Decision-making

The decision-making layer involves analyzing machine learning-based failure prediction. One of the goals of this layer is to be user-friendly, making data processing and predictive model results available visually, for example, using dashboards that allow monitoring equipment health status. The monitoring process typically involves the definition of normal operation thresholds, which is supported by MILPdM architecture. Another important aspect of the proposed architecture is offering insights for the maintenance team. Towards that, MILPdM allows the implementation of warning systems triggered whenever a failure is imminent. In this way, the responsible teams are notified in advance, enabling decision-making to avoid, when possible, a failure to occur. When it is impossible to prevent, it seeks to mitigate the consequences.

In the feedback process, the prediction information of the models used to carry out the maintenance is confirmed or not by the maintenance team, and the private cloud stores this information. This task helps in the learning process, informing the assertiveness of the models in case of confirmation of the need for maintenance or case of a false alert, retraining the models with new data. As the operating environment is dynamic, the feedback step is also essential to ensure that the learning models cover as many situations as possible. This process allows the reuse of models already trained in the future, applying in scenarios and assets similar to those previously monitored.

In the military domain, the anticipation of failure can prevent an asset from entering a mission if a failure creates a risk to the troops and planning fleet maintenance and maintaining a minimum operational state. A decision-making system can give valuable information in these situations, making the decision-making layer crucial in a failure prediction system.

4.4 Communication

The communication layer is represented the most varied forms of existing mobile network operators for data transmission. However, a variety of kinds of resources are present as the proposed architecture is intended for military use. In addition to 4G and 5G LTE-based networks or IEEE 802.11 and IEEE 802.22 base stations, military frequencies availability are required. The licensed shared access pool of military frequencies is a network resource to be allocated whenever necessary (KUNST et al., 2016).

All data transfers between layers need the communication layer, and to access the communication layer, it is necessary to go through a multilevel resource broker. The broker is the result of research previously developed by the authors of this paper, and its application is aimed at ensuring the quality of service in data transmission in heterogeneous network scenarios, taking into account the needs of the network for military purposes (KUNST et al., 2016). An evolution of the broker, designed for real-time video surveillance applications for smart cities and military use in border control, already exists (KUNST et al., 2018).

Figure 29 shows the broker architecture with three levels : (I) Update Level, (II) Resources Level, and (III) Decision Level. The Update Level is responsible for collecting parameters from network operators, allowing the broker to make appropriate decisions on sharing network resources. The broker applies a usage profile concept, where the network usage profile is defined through the network usage history, minimizing the possible effect of abnormal traffic behavior.

Resource Level divides all network users into two distinct classes. The first class is the primary user, composed of users who have a license determined by a regulatory agency to use a frequency spectrum. The second class is the secondary users, who make use of the available network opportunistically. The broker also must know that there are three types of possible frequencies in the same geographic area, those of exclusive use, those of shared use, and those of exclusively shared use that serve as the basis for the Licensed Shared Access (LSA). LSA is an approach that grants to a limited number of devices individual licenses to access network

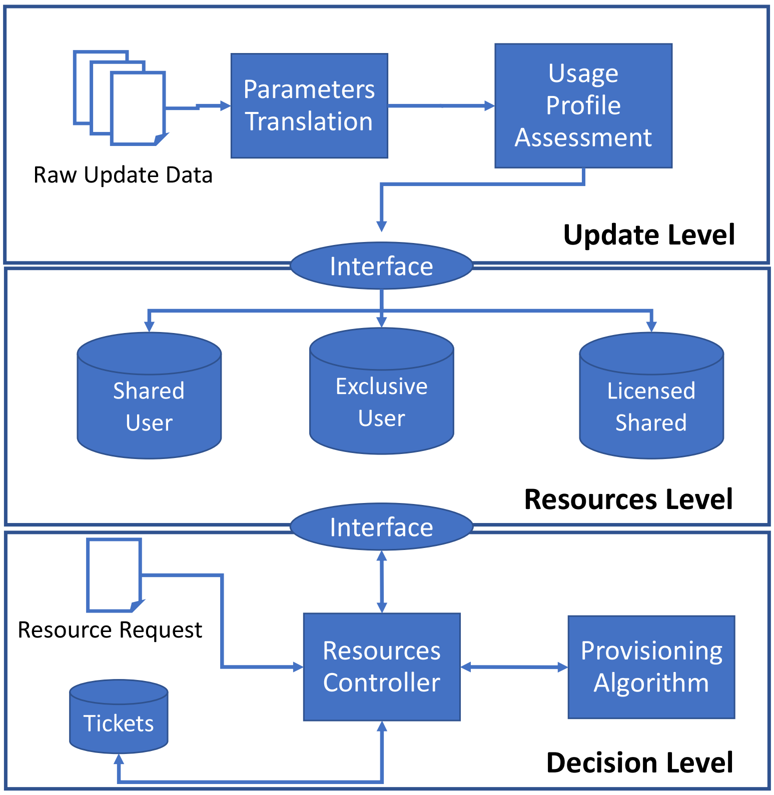


Figure 29: Broker levels and components, adapted from Kunst et al. (2018)

resources that are already allocated to one or more incumbent users.

Three databases store the data provided by the Update Level, a database containing primary user information, a database containing secondary users, and the LSA pool database which stores information about exclusively shared access frequencies. The LSA pool database information is accessed, for example, when primary users need to complement their network resources. The data from the three databases are then made available to the Decision Layer.

It is at the Decision Level that requests for network resources are processed. This level has four components. The resource request component is responsible for deciding which network resources will be designated for sharing, taking into account the QoS needs in addition to defining the priority of the request. In the resource pricing component, the use of a favorite model is proposed, enabling network sharing, considering that the available networks will coexist for a long period. This model allows reciprocity and payment control is not necessary, simplifying the LSA process.

The resources assessment model, the third component of the Decision Level, evaluates through simulations the number of resources controlled by each mobile network operator, applying models consolidated in the literature. Finally, we have the Resource Provisioning and Resource Controller algorithms. The Resource Controller component, which is responsible for receiving the resource request, uses the information stored in the databases of the Resource Level to define which are the resources that serve the requested demand through the execution of the Resources Controller Algorithm. The result of the algorithm is an array of candidates.

This array is the input to the Resources Provisioning Algorithm, which will define the most suitable one according to demands such as QoS and transmission cost.

In a dynamic environment like the military domain, that data can be collected in real-time, stored in a private cloud, and then consumed by a command and control center safely, and with a QoS, we use the broker. In this way, all information that travels through the network passes through a layer that aims to guarantee the delivery of information.

4.5 Private Cloud

Considering the critical nature of data within the military domain, adopting a private cloud infrastructure emerges as a strategy to ensure the security and privacy of information access. The control supports this strategic decision by governing data technologies and protocols to create a more robust cloud architecture. However, it is important to note that this heightened security posture comes at a higher cost of deployment when compared to commercially available cloud alternatives (ĎULÍK; JUNIOR, 2016).

All communication between the private cloud and other layers of the MILPdM architecture takes place through the broker. As it is a private cloud, there is a need to guarantee access for data storage and data access. Thus, redundancy in communication is necessary, using both wired and wireless networks, such as satellites. Each network type has different capabilities.

As each network has a different capacity, the broker is responsible for defining, within the needs of each request, which available network will be used. Training learning models require a large amount of data, while storing the results requires less network capacity. The broker's responsibility is to define which network will meet the needs of each request.

All the data needed to train machine learning models is stored in the private cloud. This includes raw data collected from the physical environment. The processing layer uses the private cloud to access this data to predict failures. Finally, the decision-making layer stores the prediction results and feedback from the maintenance team in the private cloud.

4.6 Simulation

The simulation layer is responsible for predicting the remaining useful life of equipment in unknown operational scenarios. It is necessary to fine-tune the PFM using previously stored data from assets already monitored in different contexts to achieve this capacity of predicting remaining useful life. Three steps compose the simulation process.

The first step uses actual data stored in *Private Cloud* to create new scenarios. Creating new scenarios works using a large amount of data from different contexts, such as a vehicle operating in high-temperature scenarios in a desert, in rugged terrain, or with variations in altitude. Meanwhile, the vehicle has little data available in scenarios such as a tropical environment of high temperature and high humidity. This step prepares the data available for fine-tuning the foundation model.

In the foundation model step, we use artificial intelligence (YOON; JARRETT; SCHAAR, 2019) to simulate behavior in unknown operational scenarios, as in our example, in a tropical environment. This layer does not generate new data in contexts without data available. However, it helps prevent equipment or vehicles from being used in situations that might not perform well, allowing for more accurate mission planning.

The purpose of the PFM in this layer is to generate data and predict the behavior in unknown operational scenarios. According to section 3, PFM has been used in tasks that generate text and images, demonstrating promising results. The expansion of the use of PFM in the last year resulted in the launch of the first models rated for temporary series. In this work, we will evaluate the feasibility of using TimeGPT PFM to predict data in different scenarios for which the model was trained. To validate this capacity, we compare results from LSTM neural network models in this work.

5 RESULTS

This section presents the obtained partial results by the PdM architecture. As presented above, the proposed architecture is composed of five layers. However, the focus of the executed experiments is in the processing layer and simulation layer.

Therefore, we divided it into two sections, the first presenting the results obtained on the processing layer and the second on the private cloud layer.

5.1 Processing Layer Results

At this stage, the goal is to assess the feasibility of the processing layer of the architecture. This layer performs a series of tasks to predict a failure, including data processing, data preparation, the training of learning models, and finally, failure prediction.

All the algorithms used in this work for data processing, graphics generation, and the LSTM and RF learning algorithms are built using Python programming language. Specifically for data manipulation and dataset processing, the *Python Data Analysis Library* is used. For tests with the LSTM *Keras* library was used, and for the RF test, the *scikit-learnig* library was chosen.

We evaluate the architecture through two cases, the first use case uses data previously collected and made available in the IMS bearing dataset. The IMS dataset contains the health degradation data of an asset from its normal state until the moment of its failure and is a widely used database in the literature for diagnosis and prognostic tasks. The second use case we created by collecting data from a path model.

5.1.1 Case study 1: Bearing dataset

In more detail, the IMS dataset has run-to-failure data of rolling in an AC motor and was made by the University of Cincinnati and released in 2014 (LEE et al., 2007). Figure 30 show the AC motor.

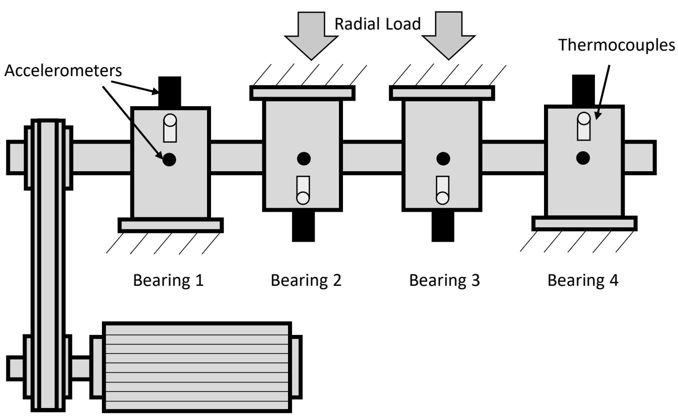


Figure 30: The bearing test of IMS dataset (QIU et al., 2006)

The dataset divides into three other smaller datasets, where each of these smaller datasets has a set of files representing a data collection for one second. In the results presented in this work, only the data from dataset one was used, which is composed of data collected from eight vibration sensors installed in four bearings, thus having two sensors per bearing. The sensor collection frequency is 20Khz, and each collection happens at intervals of five or ten minutes.

The dataset has 2156 files, resulting in more than 44 million data stored, where each data has eight records, one for each sensor. Due to the large amount of data and the computational limitation for training learning algorithms with the complete dataset, it is necessary to carry out a set of actions to reduce the dataset and simultaneously guarantee the prediction capacity of the learning algorithms. Figure 31 shows the process of reducing the dataset until the final choice of the learning model.

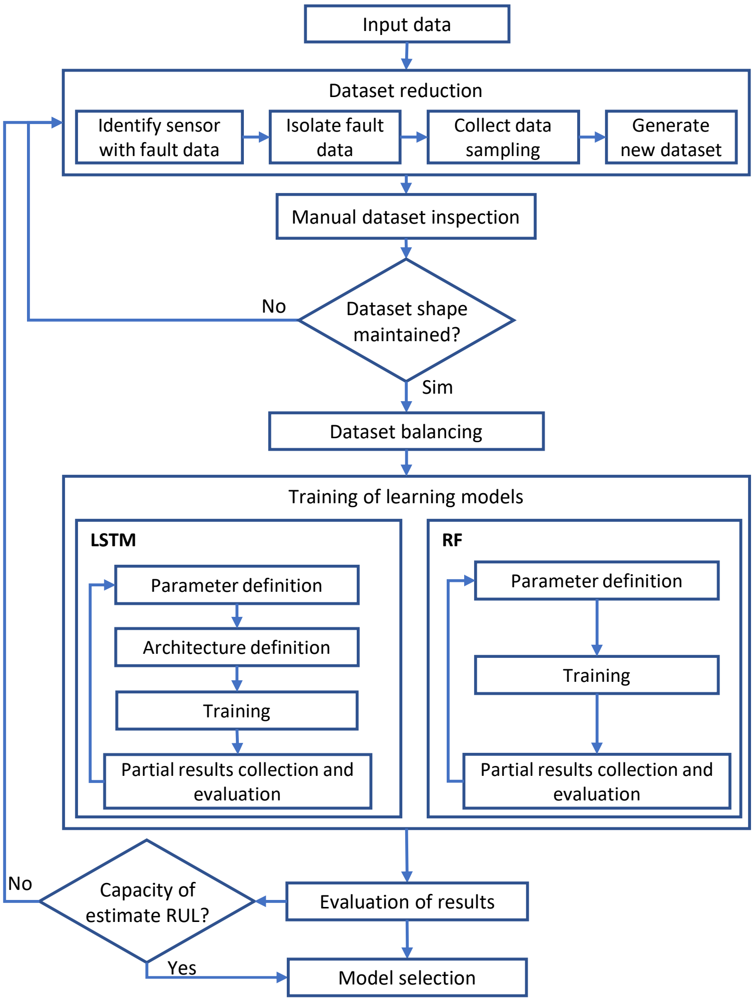
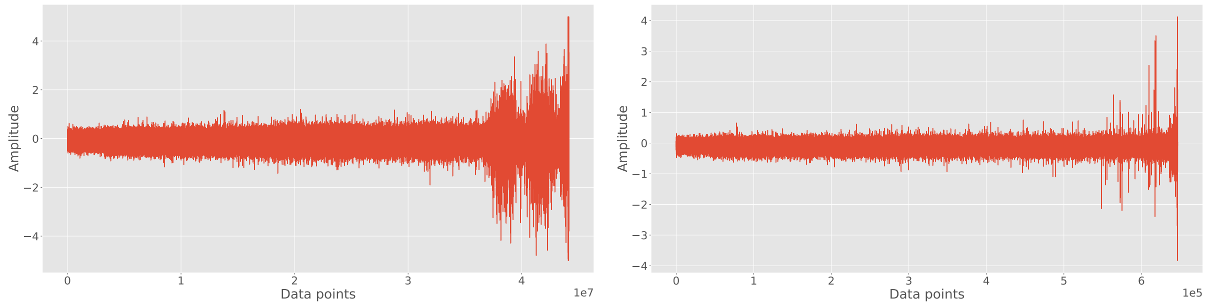


Figure 31: Failure prediction flowchart

As the first step of the process, we studied the original dataset to perform the reduction. The dataset has sensing information from 8 accelerometers that, in pairs, monitor four bearings. The first step is to identify the sensors that have fault data. Analyzing the dataset, we identified that one of the defects occurs specifically in bearing 3. This way, we remove all data from sensors not attached to bearing 3. Thus, the second task of preprocessing the data was eliminating records from the other sensors and working with only one sensor from bearing 3, reducing the dataset by 75%. Figure 32a presents the original dataset with more than 44 million data from one of the bearing number 3 sensors that we use in the experiments of this work.

With the dataset ready, containing the data from a bearing 3 sensor, the next step is the data size reduction through the collection of data sampling. The data size reduction allows the models to be computationally viable to perform the training and viable to perform predictions. We perform several tests empirically to generate a database with a smaller volume of data, allowing the creation of models in a viable time. Figure 32b shows the final result of the tests, where a total of 646,800 points compose the final dataset.



(a) Original (b) After data reduction

Figure 32: Dataset

As can be seen, the figure 32b presents a shape close to the original dataset (32a). This visual inspection is carried out to determine whether the data reduction process was able to generate a dataset smaller than the original and with the same behavior. If the dataset resulting from the reduction process is the same as the original dataset, the dataset reduction process must be repeated.

The last step before using the data for training is the dataset balancing. As can be seen from the shape of the graph shown in Figure 32b, from the beginning of the collection to the data point around 400 thousand, it has a behavior with few changes. To make the training process less costly, we chose data from the 400 thousand record to the end of the dataset as training data. This decision maintains the dataset with coverage in the behavior considered normal and, from the 500,000 records to the end of the dataset, the behavior is considered to be a failure. We use the data resulting from the processing to train the LSTM and RF ML algorithms.

The algorithm training stage generally involves defining parameters, training, collecting, and evaluating results. The process is repeated to minimize errors. At the end of the training, the results are evaluated. If the ability to estimate the RUL is not achieved, the process returns to the dataset generation stage until models capable of estimating the RUL are reached.

The first step in LSTM training is to use a hyperparameter technique to find the best parameters combination. There are several parameters to train an LSTM network, such as the number of epochs or batch size. Thus, we define the structure of the network, such as how many layers and how many neurons should each layer have. In this process, each parameter of the learning model receives a set of possible values to be trained, always aiming to reduce the error generated by the network.

Figure 33 presents the architecture used in this work, with two LSTM layers, with one with 50 neurons. After preliminary tests did not identify a decrease in error with a more significant number of layers of neurons, we define the number of neurons as 50. There is also a dropout layer after the LSTM layer to avoid overfitting. After defining the network layers, to achieve the lowest possible root mean square error (RMSE), it is necessary to define the best parameterization of the model. RMSE is the metric adopted to determine the best model, where the goal is to get as close as possible to the value of zero.

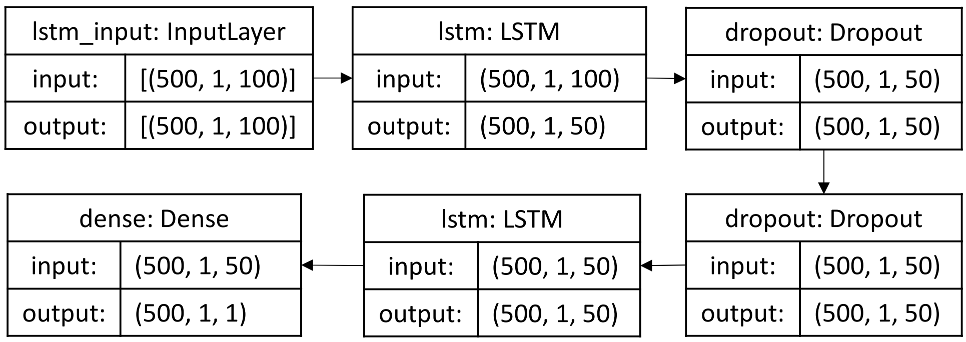


Figure 33: LSTM neural network architecture

Using a lag of 1 shows the worst choice for the lag parameter. The best results were mostly with the trained models with a value of 100 for the lag, with a small difference to the error obtained with lag 10. Figure 36 presents more clearly the error difference.

The model created with parameters of 100 for lag, 100 for epochs, 500 for batch size achieves the lowest RMSE value, as shown in the input layer in figure 33, and 50 for the value of the neurons in the LSTM layer, reaching a value of 0.15015 for the RMSE. The slightest error is slightly below the overall average of lag 100. Figure 34 shows that the loss stabilizes near the middle of the values zero and 200, which confirms the fact that models with 100 epochs have, in general, the smallest errors.

Figure 35a presents the result of the prediction performed by the model with the lowest RMSE, constructed according to the architecture shown in the Figure 33. The original test values are shown in the red line, with more remarkable aptitude, and predicted values in blue with lesser amplitude. In the figure, it is possible to observe that the predicted value generally follows the original dataset, and its use in architecture is promising.

As observed in the experiments performed with the LSTM, the RF presented the best results when the lag value used is equal to or greater than 10, reaching an RMSE of 0.1675. In addition to lag 10, the model that obtained the lowest RMSE value is trained with the values of 0.1 for the *Min samples split* parameter, *Max features* as "auto" and the parameter *Max depth* as 10. Figure 35b presets the result of the prediction generated by the model with the lowest RMSE,

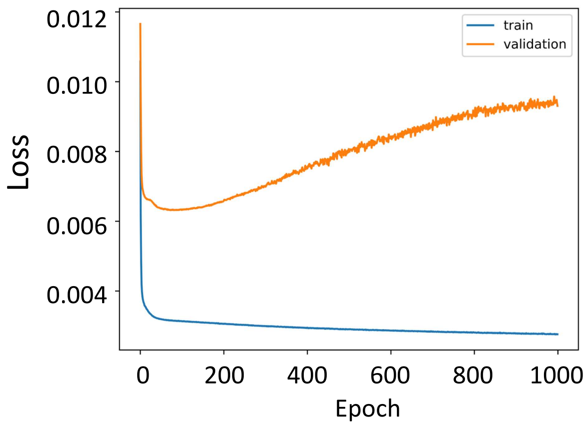
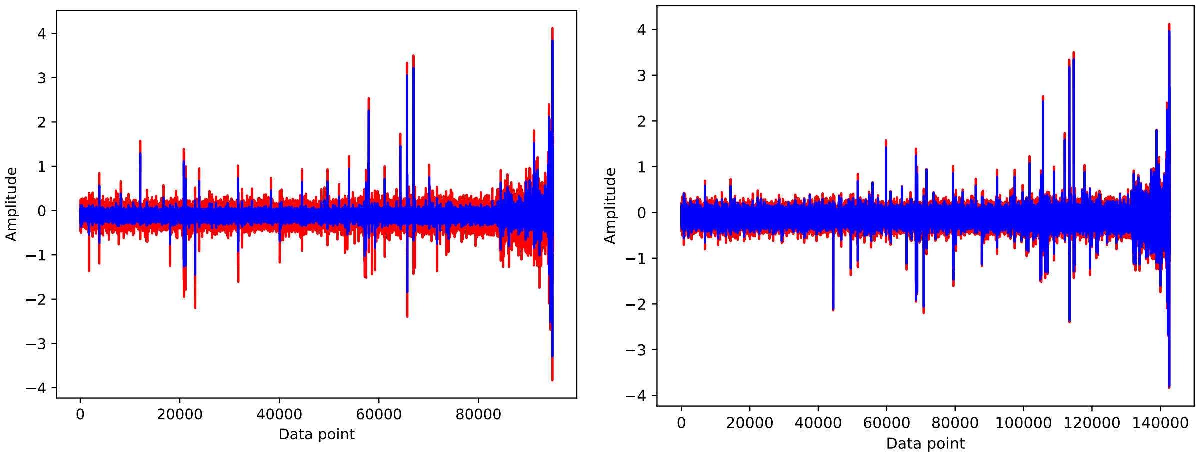


Figure 34: LSTM Loss vs epoch

where we have the predicted data in the blue line and the data from the original dataset in the red line.



(a) LSTM model

(b) RF model

Figure 35: Test vs Predict data point by model

Figure 36 presents the obtained RMSE values for all tested models, grouped by the lag value of 1, 10, and 100. As can be seen, the models had slight RMSE variation in the hyperparameter process. That is, the parameters of the models had less influence on the RMSE of the models than the lag value. This fact is even more evident in the results obtained by the predictions of the RF models, where the variation within each lag value happens only after the fourth decimal place of the RMSE. In addition, when training models with lag parameters as 100, there is an increase in the computational cost for training the models, which is not justified. Given that, on average, the RMSE obtained with a value of 10 for the lag has a similar RMSE.

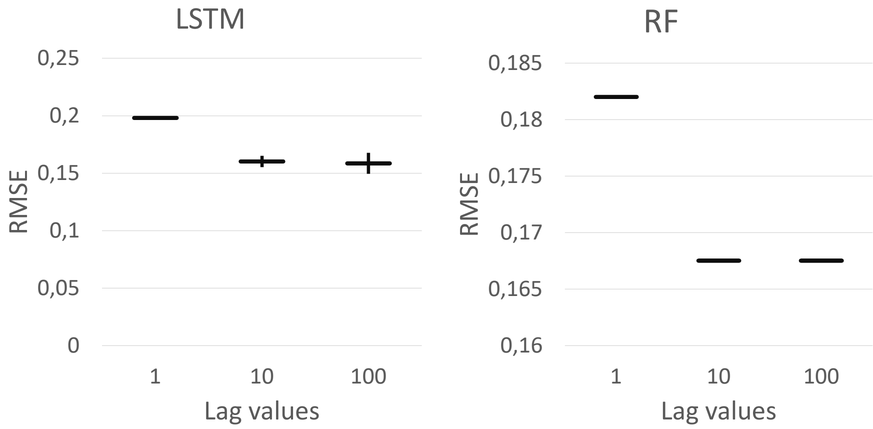


Figure 36: Mean and standard deviation of LSTM and RF models

5.1.2 Case study 2: Truck model

To create the truck use case, we used a Raspberry Pi model 3 with a Sense Hat (PI, 2015) for data collection, using a power bank to supply energy. Figure 37 shows the sensor positioned on a military truck model with a ratio of 1:12 to a real truck, and how we store the data through a Message Queuing Telemetry Transport protocol (MQTT) Broker. With the sensor attached to the truck model, we simulate scenarios of uses.

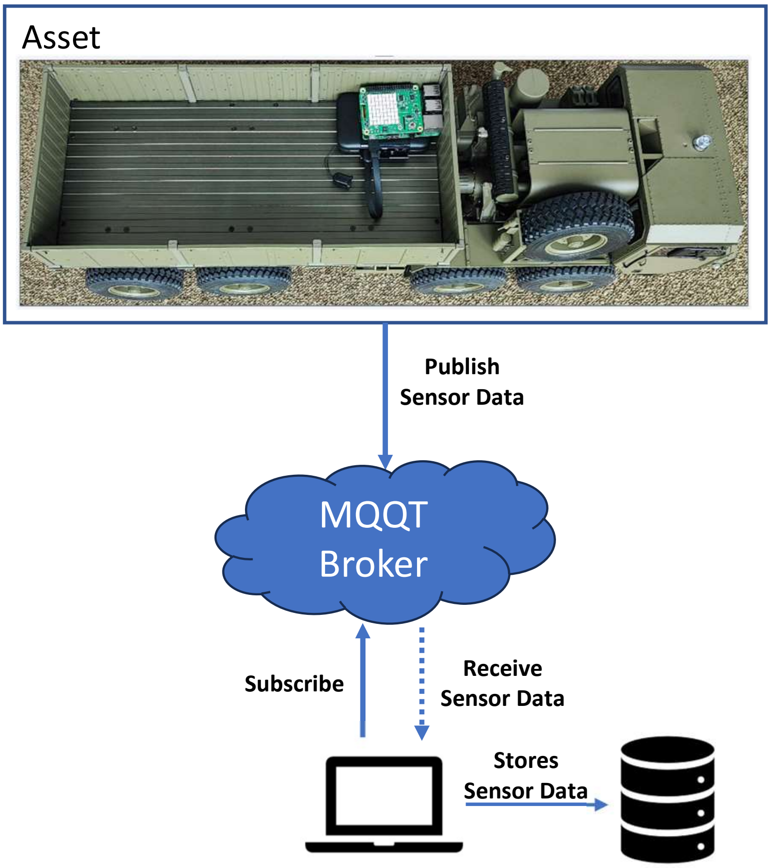


Figure 37: Truck with sensors

Figure 38 presents the flow between collecting data from the truck, processing the data, performing the training process, and collecting the results.

An MQTT broker sends the data collected by the sensors attached to the truck to the edge device. The data is preprocessed at this stage and then sent to cloud storage. We simulated use cases and collected data with the truck displaying different behaviors, such as an empty truck, a loaded truck operating on uneven terrain, and a tire failure simulation. For data collection with tire failure, we simulated a bubble in the tire, where the defect gradually gets more significant for each collection performed until a failure point.

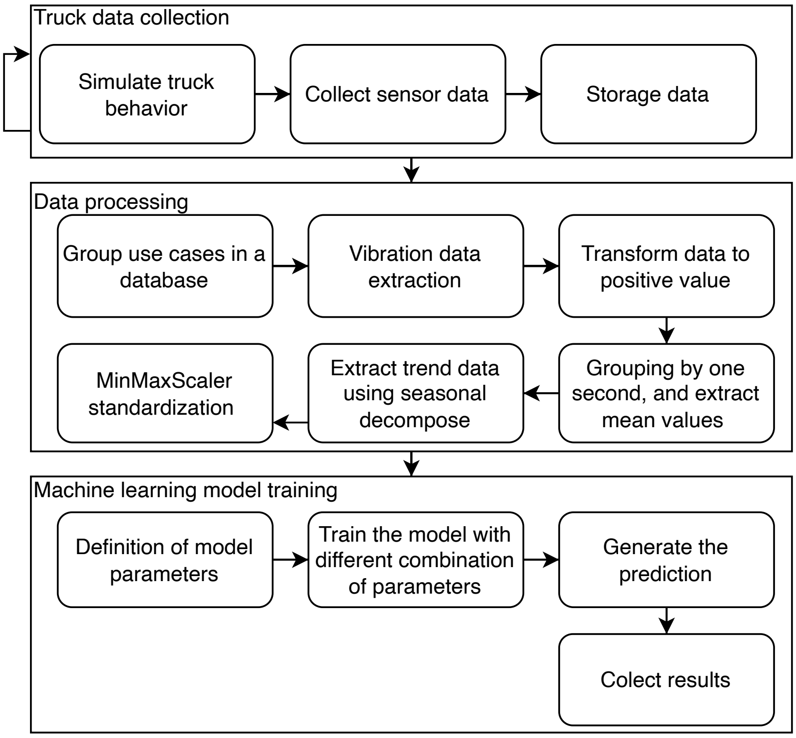
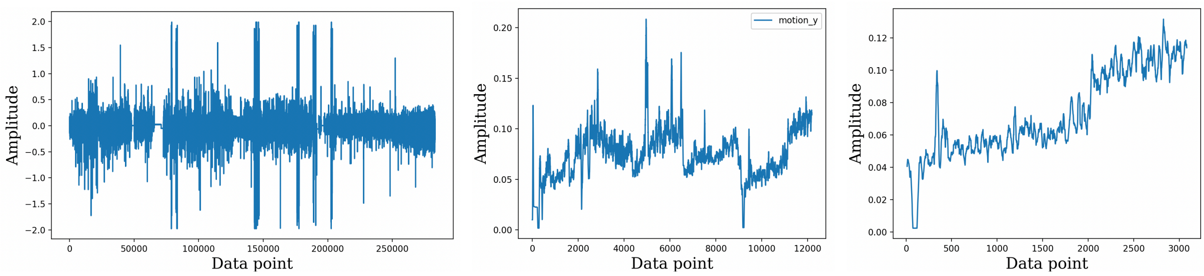


Figure 38: Flow between obtaining data and collecting results

We train the learning models with a dataset of several collections with different behaviors, intending to create a dataset representing different behaviors. Figure 39a presents the complete dataset, with 280679 points. Each second of the collection has between 20 and 30 points.



(a) Training dataset

(b) Final dataset

(c) Tire failure dataset

Figure 39: Datasets

The training process occurs in two stages. In the first phase, we process the dataset and extract features. At the end of this process, a new dataset is generated for use in the next model training phase. We detail these two phases in the following two subsections.

5.1.2.1 Data processing Phase

The data processing phase starts with the feature extraction process, in which we group the use cases into a single dataset and extract the vibration data from that dataset. Next, we reduce the dimensionality of the dataset so that each data point represents one second of collection. We perform two steps for dimensionality reduction. First, we obtain the absolute value of each data point, changing the negative values to positive values. In the second step, we obtain the mean values of a second data collection, apply the seasonal decompose strategy, and obtain the trend value. Ultimately, we normalize the dataset. Figure 39b presents the dataset after processing the data, with 12188 points, each corresponding to one second of collection. This dataset is the model training dataset.

To evaluate the response of our model to failures, we collected data from the truck in a normal state until reaching the moment when the wheel was no longer able to turn correctly, making it impossible to use the truck. Figure 39c presents the vibration data of the vehicle with the defect after going through the same data processing performed on the test database. The defect was increased at each collection, which the tendency for increased vibration can observe.

5.1.2.2 Training Phase

The training process's first step is defining the failure prediction strategy. We execute two distinct strategies. In the first strategy, the learning model is trained to predict the next value of the time series, and then the predicted value is used as input for predicting the next value, and so on. However, after the first few seconds, the model tends to generate an incorrect value, mainly when the model input contains only previously predicted values without actual values. Figure 40 shows the result of this first strategy.

However, when we predict only one value at a future point far from the current point, the result is more accurate, so we adopt the second strategy for failure prediction. Figure 41a presents the test results described in this section. We performed the tests seeking to predict 1 second in the future, 10 seconds in the future, 120 seconds in the future, and 240 seconds in the future.

The smallest root mean square error (RMSE) is obtained when predicting one second into the future, and the most significant error occurs when predicting a point 240 seconds into the future. That is, the further into the future the point to be predicted, the greater the error. Thus, according to experts, we use the value of 120 seconds in the future, which allows prediction in advance concerning the current moment.

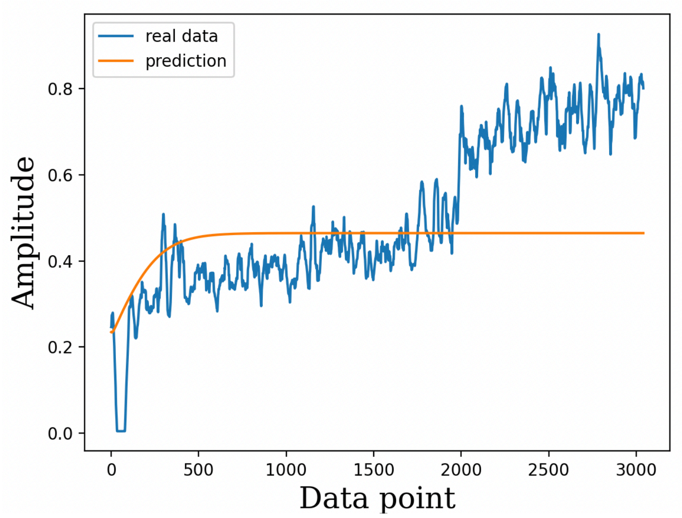
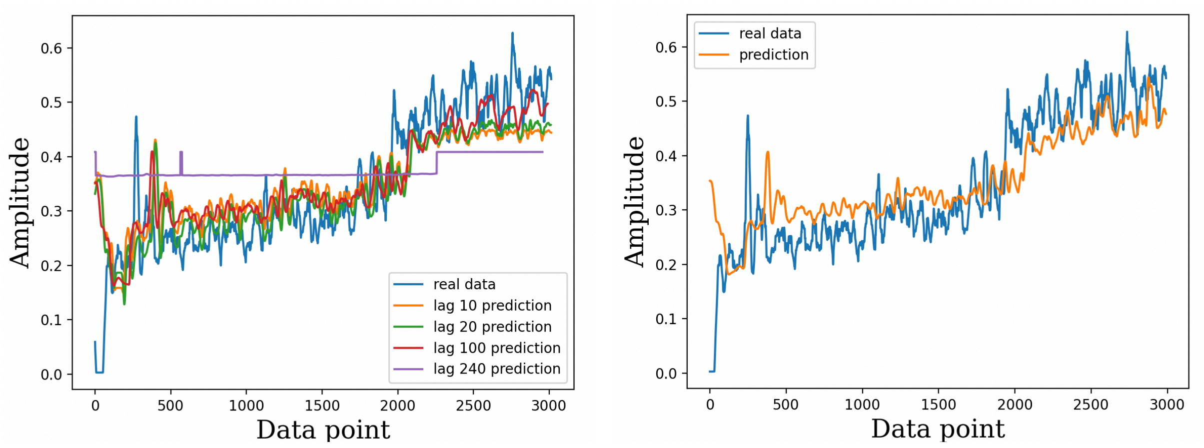


Figure 40: Result of first predictions strategy

After defining the prediction strategy, the next step in LSTM training is to define the network structure, such as how many layers and neurons each layer should have. Furthermore, several parameters can be used to train an LSTM network, such as the number of epochs or batch size. Thus, a hyperparameter technique was used to find the best parameters combination. In this process, each parameter (Table 6) of the learning model receives a set of possible values to be trained, always aiming to reduce the error generated by the network.



(a) By Lag

(b) Best Model

Figure 41: Test vs Predict data point by LSTM model

Experiments to define the number of layers of the LSTM with the smallest error are performed with a fraction of the possible parameters. When defining the number of LSTM layers, we train the model with the parameters mentioned in the table 6.

The lowest obtained RMSE value was achieved by the model created with parameters of 100 for lag, 32 for epochs, 15 for batch size, and 50 for the value of the neurons in the LSTM layer, reaching a value of 0.0707 for the RMSE.

Table 6: Training parameters

Model	Parameters	Interval
LSTM	Window	[10, 20, 100, 240]
	Cross-Validation	[10-folds]
	Epochs	[15, 25]
	Batch Size	[32, 100]
	Neurons	[25, 50]
RF	Window	[10, 20, 100, 240]
	Cross-Validation	[10-folds]
	Estimators	[10, 25, 100]
	Min Samples Split	[2, 5]
	Max Depth	[None, 10]
	Max Features	[sqrt, log2]

Figure 42 shows that the loss stabilizes near the middle of the values 1 and 10, which confirms the fact that models with more than 15 epochs have, in general, a higher RMSE value.

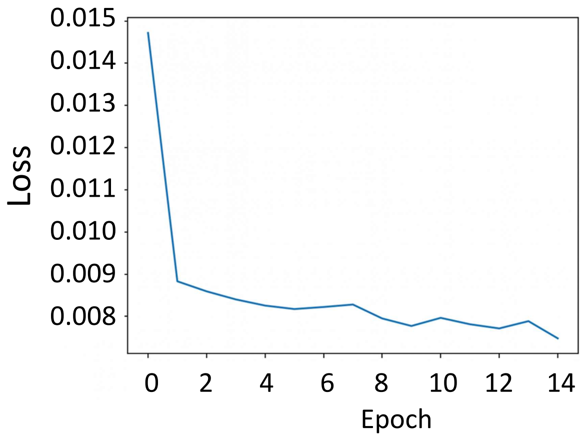
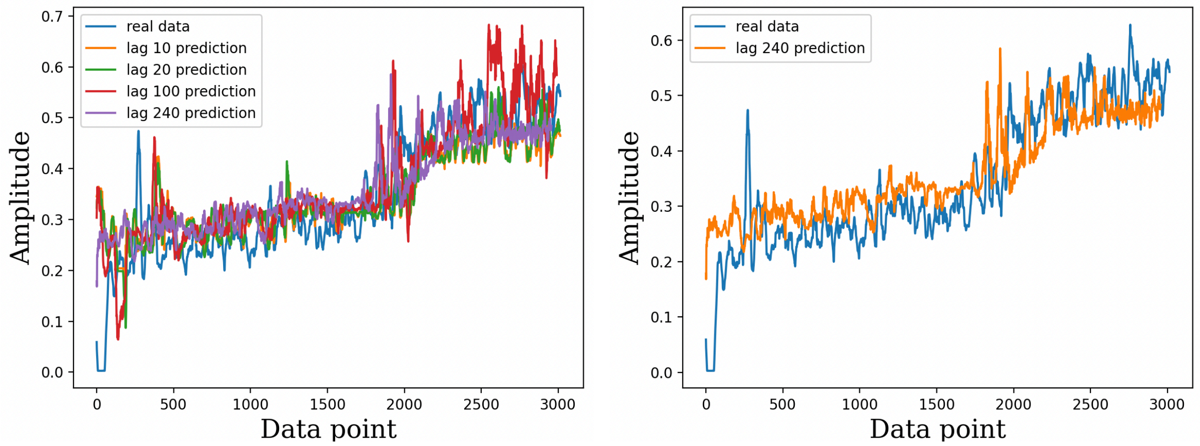


Figure 42: LSTM Loss vs epoch

Figure 41b presents the result of the prediction performed by the model with the lowest RMSE. The original test values are shown in the blue line, and the predicted values are in orange. In the figure, it is possible to observe that the predicted value generally follows the original dataset, and its use in architecture is promising. The RMSE applying the trained model to the tire failure use case is 0.0707.

To compare the result of the LSTM model, we used the RF implementation for the regression task, using the hyperparameters strategy to find the model with the lowest RMSE. Moreover, we tested with different lag values for each set of parameters, following the same strategy used in the LSTM training. Figure 43a presents the best model for the different lag values (10, 20, 100, 240).

RF best results are achieved when the lag value is 240, reaching an RMSE of 0.08639. In addition to lag 240, the model that obtained the lowest RMSE value is trained with the values of 2 for the *Min samples split* parameter, *Max features* as "sqrt" and the parameter *Max depth*



(a) By Lag (b) Best Model

Figure 43: Test vs Predict data point by RF model

as 10. Figure 43b presents the result of the prediction generated by the model with the lowest RMSE applied to the tire problem use case, where we have the predicted data in the orange line and the data from the original dataset in the blue line. The RMSE applying the trained model to the tire failure use case is 0.05951.

The RF model presented a lower RMSE when compared to the LSTM, but both RF and LSTM showed satisfactory results in terms of failure anticipation capability. As seen in the graphs presented, the prediction line follows the trends of the actual value.

5.2 Simulation Layer Results

Obtaining data on the degradation behavior of an asset from the normal state of health until the moment of failure is a challenging task. Furthermore, IoT devices tend to generate a large volume of data, requiring data reduction processing. This section presents the dataset used in each use case and how it was processed and used to train the learning models.

5.2.1 Case study 1: Turbofan engine

In the first use case, we use the New Commercial Modular Aero-Propulsion System Simulation (N-CMAPSS) prognostics dataset (ARIAS CHAO et al., 2021). N-CMAPSS simulates the actual degradation of a turbofan engine. The dataset was recently released and has been used in several previous works (NEMANI et al., 2023; PATER; MITICI, 2023; TIAN; YANG; JU, 2023). Figure 44 shows a diagram of a turbofan engine unit.

The dataset comprises eight sub-datasets; each has a set of variables divided into five types: scenario descriptor, measurements, virtual sensors, model health parameters, and auxiliary data. Each sub-dataset has a distinct set of units with different failures, affecting the flow (F) and efficiency (E). The failures involve five rotating subcomponents: fan, low-pressure compressor

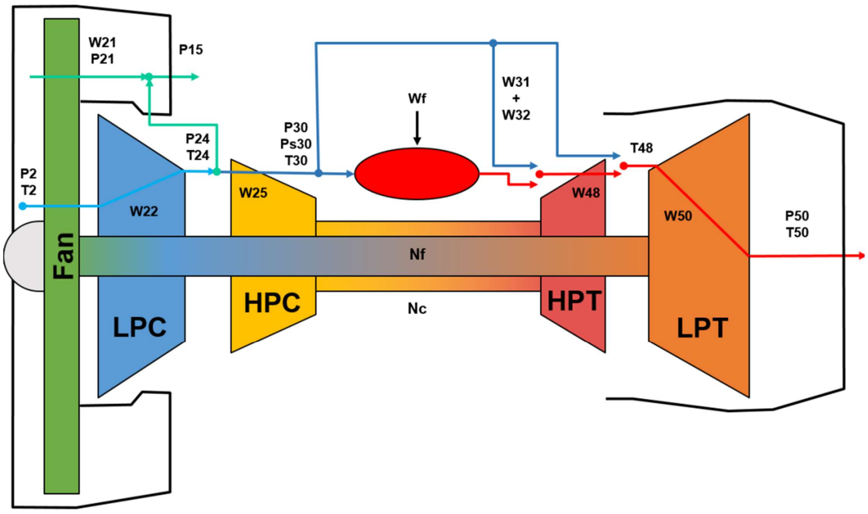


Figure 44: Turbofan engine unit, from Arias Chao et al. (2021)

(LPC), high-pressure compressor (HPC), low-pressure turbine (LPT), and high-pressure turbine (HPT). Table 7 presents an overview of the data.

Table 7: N-CMAPSS Dataset Overview

Name	Fan		LPC		HPC		HPT		LPT		Size
	E	F	E	F	E	F	E	F	E	F	
DS01							X				4.6M
DS02							X		X	X	6.5M
DS03							X		X	X	9.8M
DS04	X	X									10.0M
DS05											6.9M
DS06					X	X					6.8M
DS07			X	X	X	X			X	X	7.2M
DS08	X	X	X	X	X	X	X	X	X	X	35.6M

5.2.1.1 Feature engineering and modeling

As the N-CMAPSS comprises sub-datasets, since each sub-dataset comprises millions of data, it is necessary to reduce its size. To reduce dimensionality, we perform a set of tasks on the data so that, in the end, a set of data that is computationally viable to be used is generated.

Figure 45 shows the dimensionality reduction process performed. All sub-datasets are merged into a new dataset as part of this process.

After dimensionality reduction, we used random forest regression to determine the importance of each feature and select those that to be used as inputs to the machine learning models. Figure 46 presents the 15 most essential attributes obtained with the DS08 sub-dataset, where closer to 1, the greater the importance of the attribute for estimating the remaining life cycles.

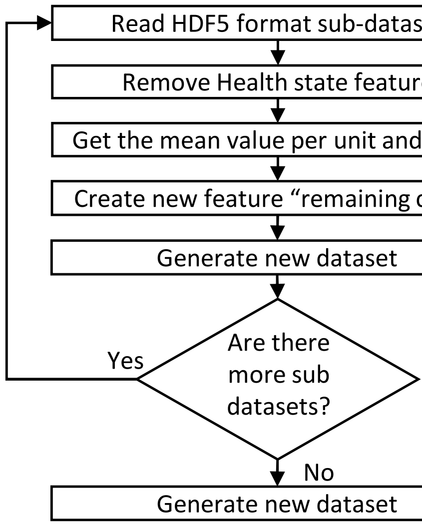


Figure 45: Flow between obtaining data and collecting results

The figure presents data from the DS08 sub-dataset as all possible failure behaviors exist in the N-CMAPSS dataset.

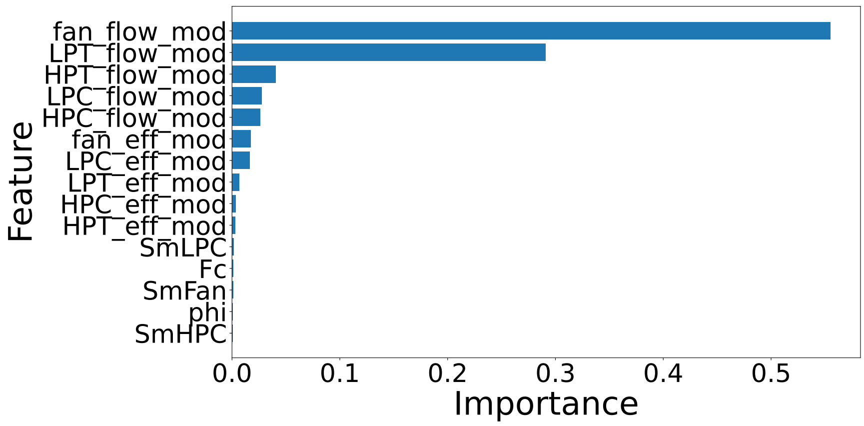


Figure 46: Dataset DS08 feature importance

Table 8 presents the selected features. The selected features fall within the unobservable model health parameters θ. These parameters fall into the class known as quality parameters and are used to simulate the deteriorated behavior of the system.

In conjunction with the features in the table 8, we use auxiliary information such as flight class in training the models. The dataset has three flight classes: fight class 1 to short flights at low altitudes and speed, flight class 2 to longer flights at higher altitudes, and flight class 3 with the longest and highest flights.

Figure 47 shows the final result of the dataset pre processing. In the figure, one can observe the different cycles of each unit das 61 existentes in the dataset and which feature is most important for measuring health degradation. The lower the value of each feature, the fewer usage cycles remain for the monitored unit.

Figure 48 shows the remaining cycles of just the first turbofan of the 61 turbofan engine

Table 8: Selected features

Type	Symbol	Description
Model health parameters	fan_eff_mod	Fan efficiency modifier
	fan_flow_mod	Fan flow modifier
	LPC_eff_mod	LPC efficiency modifier
	LPC_flow_mod	LPC flow modifier
	HPC_eff_mod	HPC efficiency modifier
	HPC_flow_mod	HPC flow modifier
	HPT_eff_mod	HPT efficiency modifier
	HPT_flow_mod	HPT flow modifier
	LPT_eff_mod	LPT efficiency modifier
	LPT_flow_mod	HPT flow modifier

units that make up the complete dataset. It is possible to see how the value of the *HTP_eff_mod* feature drops as the number of cycles remaining in the unit decreases. The drops in the value demonstrate the high correlation between remaining cycles and the health degradation features presented in the table 8.

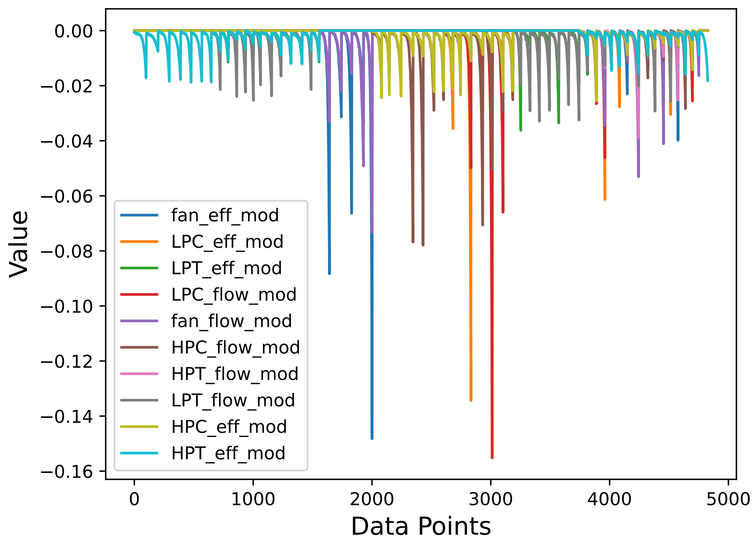


Figure 47: Final N-CMAPSS Dataset

5.2.1.2 Model training

To train the learning models, we separated the dataset into two parts. The training part used only data relating to class 1 and class 2 flights. Therefore, the generated model does not know the behavior of the data on class 3 flights.

As data input, we have the variables in the table 8 together with the flight class. The model's output is the new feature *remaining_cycles*. We train the LSTM models using the hyper parameterization process with the parameters shown in table 9.

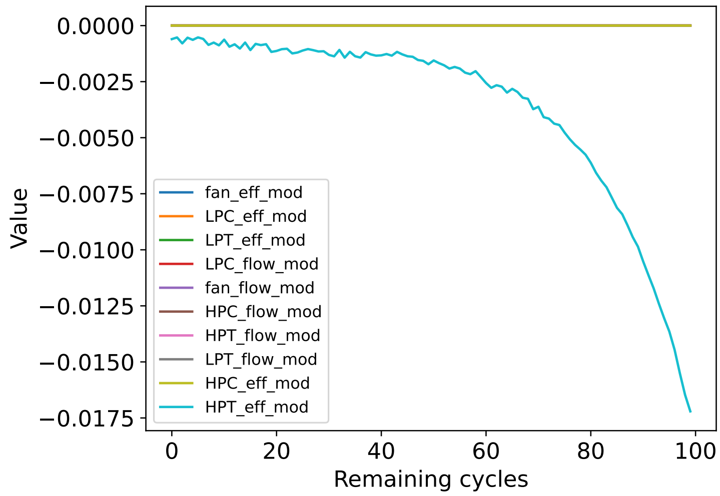


Figure 48: Remaining cycles from one unit of N-CMAPSS DataSet

Algorithm 1 presents the process of training and collecting the result, where X represents the attributes, and Y represents the value to be predicted. Use case 2 uses the same algorithm and parameters in the model training process.

Table 9: LSTM parameters

Parameter	Values
Neurons	25, 50
Epochs	10, 25, 50
Batch	16, 32, 50
Lag	1, 5, 10, 25

We only use the class 1 and 2 flight data when training the machine learning models. The model resulting in the lowest RMSE predicts the remaining cycles of class 3 flights. This way, the learning model predicts the remaining flight cycles in a scenario not in the training data.

The TimeGPT model uses the same data set, with class 1 and class 2 flights for training. Furthermore, we use the fine-tuning process to customize the model to our context. After this process, a small data set with class 3 was made available for the model to predict the time series.

One of the characteristics of the TimeGPT model is the need for a *timestamp* feature, so we created a feature with fake data to meet this demand. However, as in this use case, the useful lifetime is calculated by the number of cycles remaining. After obtaining the TimeGPT prediction values, we again convert the values of type *timestamp* to the remaining number of cycles until failure.

5.2.2 Case study 2: Bearing dataset

In this use case, we use the same dataset presented in section 5.1.1. The dataset comprises three sub-datasets. Dataset 1 has two accelerometers allocated for each bearing (x- and y-axes)

Algorithm 1 LSTM manual parameterization

$splits \leftarrow TimeSeriesSplit(n\_splits = 3)$
$neurons \leftarrow [25, 50]$
$epochs \leftarrow [10, 25, 50]$
$batch\_size \leftarrow [16, 32, 50]$
$lags \leftarrow [1, 5, 10, 25]$
for n in $neurons$ **do**
    for e in $epochs$ **do**
        for b in $batch\_size$ **do**
            for l in $lags$ **do**
                for $train, test$ in $kfold.splits(X, Y)$ **do**
                    $model \leftarrow Train(n, e, b, l, train, test)$
                    $rmse \leftarrow Evaluation(model)$
                    $results.append(rmse)$

and one accelerometer per bearing for data sets 2 and 3. All failures occurred after surpassing the designated lifespan of the bearings, which exceeds 100 million revolutions.

Upon concluding the testing in dataset 1, an inner race defect manifested in bearing 3, along with a roller element defect in bearing 4. In dataset 2, an outer race failure was observed in bearing 1, while dataset 3 documented an outer race failure in bearing 3.

5.2.2.1 Feature engineering and modeling

Working only with raw data is computationally challenging, requiring a long time to train the models, so dimensionality reduction and feature extraction are necessary. Figure 49 shows the feature extraction process.

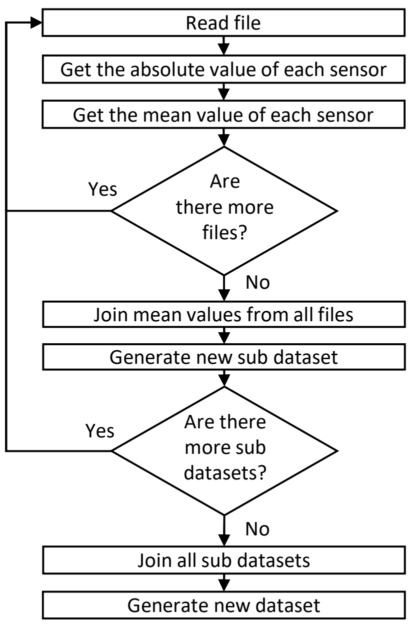


Figure 49: Flow between obtaining data and collecting results

As the first processing step, we extract the absolute value for each value collected from the dataset. This way, we only work with positive values. In the second step, we extract the mean value from each sensor from each file. Each file is transformed into a single value in the new dataset.

Next, we merge all sub-datasets into just one. As dataset 1 has two sensors in each bearing, we use the value of one of the sensors. Thus, our dataset has four features corresponding to one sensor per bearing. Ultimately, we generate a new feature, representing the number of collections remaining until the moment of failure. This new feature cycle performs the same role as in use case 1 (5.2.1). Our objective is for the models to be able to predict the number of cycles remaining until the moment of failure. As each collection occurs every ten minutes, each cycle represents ten minutes.

Figure 50 shows the final result of the database. We name each sensor according to the bearing in which the sensor is placed. It is possible to observe each of the three datasets and, through the sensor's increased vibration amplitude, identify which bearing has a fault.

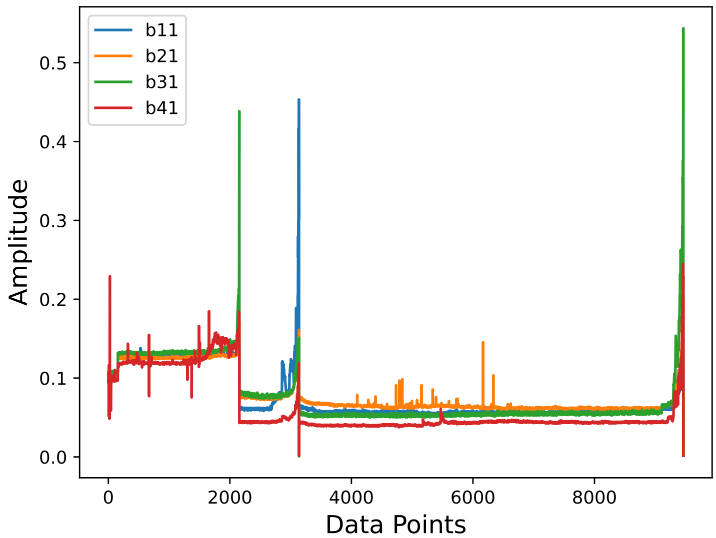


Figure 50: Final Bearing Dataset

5.2.2.2 Model training

The training parameters of TimeGPT and LSTM models follow the same values as those used for Use Case 1 (5.2.3.1). As the bearing dataset comprises three sub-datasets, the model training uses data from Dataset 1 and Dataset 2, excluding data from Dataset 3. The objective is to verify the prediction capacity of LSTM and TimeGPT using the values from the dataset that were not used in training.

5.2.3 Simulation Results

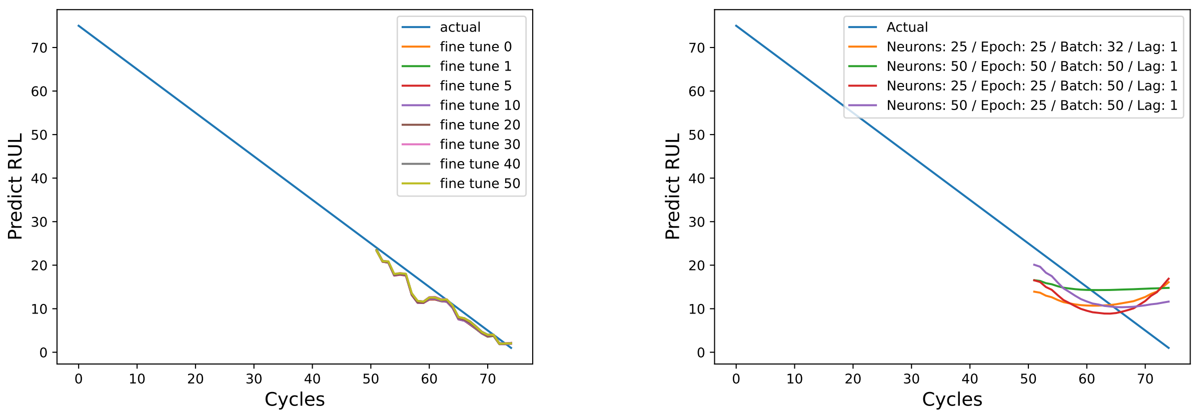
All tests with the dataset and algorithms were performed in the Visual Studio Code environment, using Python version 3.9.18. For the LSTM we used the *Keras* library end for the TimeGPT we used the *nixtlats* library.

For the LSTM model, we define the number of layers through empirical tests, where we use a fraction of the parameters of the hyperparameterization process for each new layer added to the model. Once we identified the stabilization of the reduction in the RMSE value, the number of layers of the architecture was defined. This process occurs in both use cases.

5.2.3.1 Case study 1: Turbofan engine

After generating a training dataset containing only flights from class 1 and class 2, we added a class 3 unit to the dataset so that the TimeGPT model can estimate the following 24 remaining cycles. To do this, we use the features presented in the table 8 with exogenous variables in conjunction with the flight class.

In the TimeGPT model, it is possible to determine how many fine-tuned cycles can be used. In this scenario, we gradually increased the fine-tuning steps, starting with 0 and going through 1, 5, 10, 20, 30, 40, and 50. As we did not detect a decrease in the error, and the increase in the number of fine-tuning steps tune generates a higher computational cost, we chose not to increase the fine-tuning steps. Figure 51a shows the remaining cycles and the lines predicted by each model with different fine-tuned values. It is possible to observe no significant change in the prediction when using different fine-tuned values. The model with the lowest RMSE was obtained using a fine tune of 40 steps, reaching a value of 0.02140.



(a) TimeGPT

(b) LSTM

Figure 51: Prediction of remaining cycles

To compare the results obtained with TimeGPT, we used the LSTM model. We use class 1 flight data and class 2 flight data to train the model, keeping class 3 flight data out of the training process. After selecting the data, we train the model with the hyperparameterization process. The model with parameters 25 for Neurons, 25 for Epochs, 32 for Batch, and 1 for Lag achieves the lowest RMSE. This model is now unaware of the behavior of the class 3 scenario. For all training, we use a time series cross-validator with value 3. Figure 52 shows the structure of the LSTM model used to achieve the lowest RMSE value.

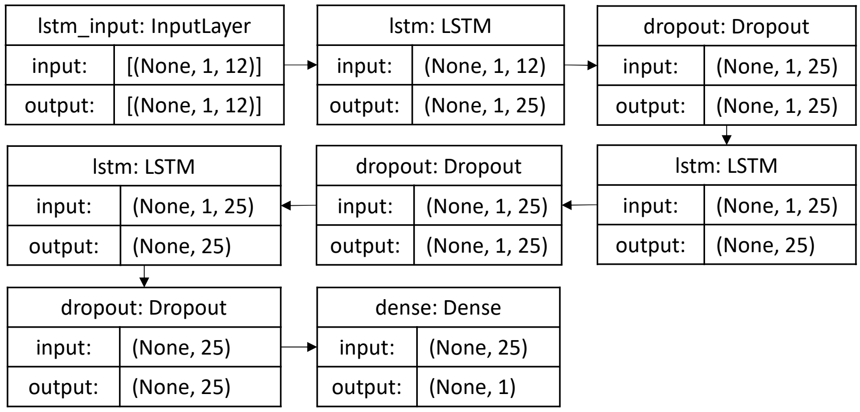


Figure 52: LSTM model structure

We use the selected model to predict the remaining cycles of a unit of a class 3 flight. In this scenario, the RMSE value obtained by the model trained with the following parameters: 25 for neurons, 25 for epoch, 32 for batch size, and 1 for lag is 0.07371. Figure 51b shows the models with the lowest RMSE values.

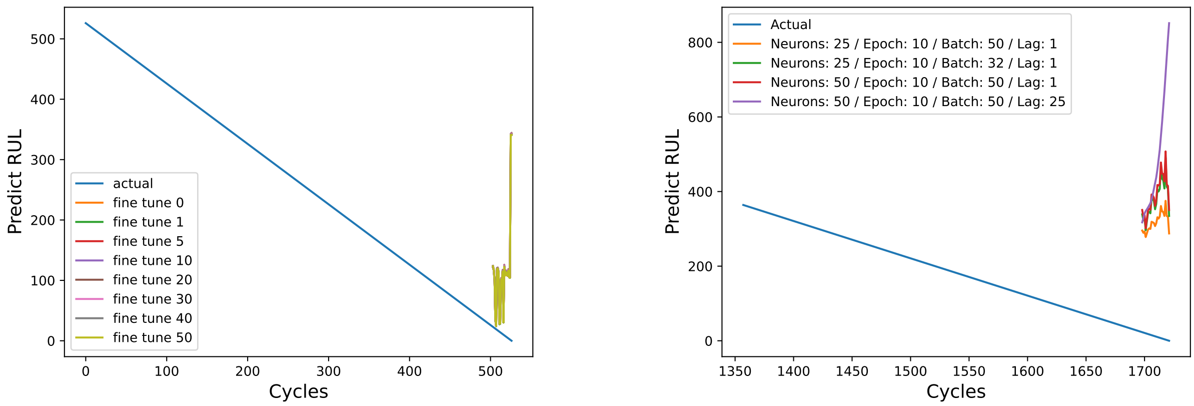
In this use case, it is possible to identify how the TimeGPT prediction values are closer to the actual values until the end of the test dataset. At the same time, the LSTM model tends to move away from the actual values as time progresses. In this case, it is essential to highlight the existence of the number of features in the dataset, providing more information to the models. In actual application scenarios, the use of functional models becomes a viable option in the context of predictive maintenance. More accurately anticipating the remaining useful life of the equipment.

5.2.3.2 Case study : Bearing dataset

In the second use case, the LSTM and TimeGPT models are trained with two of the three subdatasets. This approach ensures that the models are not aware of the failure behavior of the third dataset, allowing us to test their ability to predict remaining cycles.

Figure 53a shows the results obtained for each fine-tune value. The use of this parameter did not have a significant impact on the final result. The model with fine-tune value Zero received an RMSE of 0.1978, significantly higher than the 0.02140 of use case 1. This result can be attributed to the dataset comprising only vibration data and lacking contextual information.

Using the parameter values of 25 for Neurons, 10 for Epoch, 50 for Batch Size, and 1 for Lag, we achieved the RMSE value of 0.2340, the lowest among all the trained models. Figure



(a) TimeGPT (b) LSTM

Figure 53: Bearing dataset results

54 shows the structure of the LSTM model used to achieve the lowest RMSE value.

Figure 53b presents the four models with the lowest RMSE values, with all models having the same number of layers. As can be seen, following the behavior of TimeGPT, the result of the LSTM model is significantly higher than the result of Use Case 1.

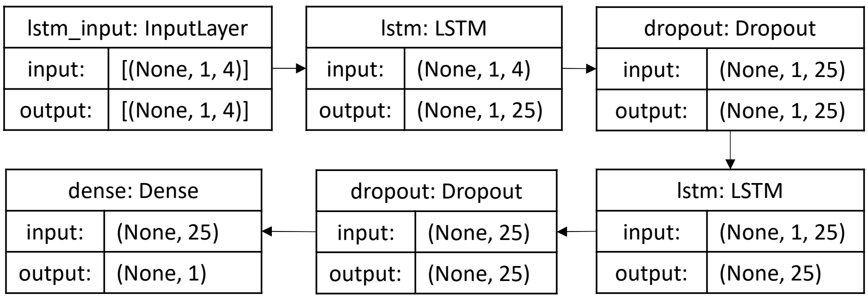


Figure 54: LSTM model structure

Unlike the turbofan use case, here, even the TimeGPT model did not present a result that was as close to the actual value but proved superior to the LSTM model. It is essential to highlight that in this use case, there is a smaller number of features in the dataset, which results in a smaller amount of context information for the models. However, in actual application scenarios, using foundation models using only sensor data becomes a viable option in predictive maintenance. We can monitor possible future failures and assist in planning the use of the equipment.

When evaluating the results obtained in the two use cases, we see that context information helps the models generate more accurate results than in the one with just sensor data. The use of datasets with only sensor data proved to be viable but with lower accuracy. In these cases, using a foundation model proved to be more accurate. Therefore, when exposing assets in new contexts or scenarios, using TimeGPT with its pre-trained model on a large set of time series data proved superior in its ability to generalize past data from other contexts for prediction in future scenarios..

Figure 55 shows the standard deviation of the RMSE obtained in the training of each LSTM and TimeGPT model. The standard deviation indicates a more significant impact of the parameters on the LSMT models. At the same time, the fine-tuning does not significantly impact the results obtained by TimeGPT. It is also possible to observe that the TimeGPT models obtained a lower RMSE value than the LSTM model, indicating the potential of using FM in the context of failure prediction.

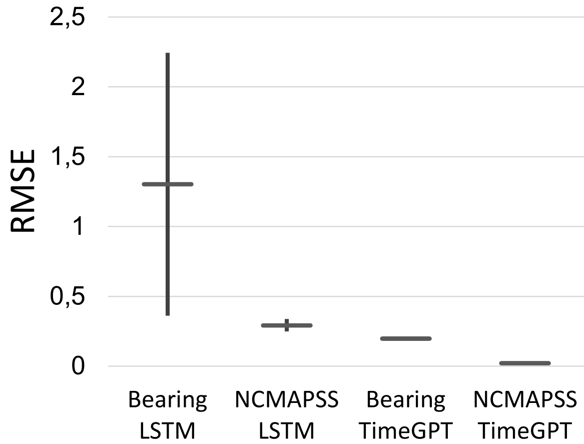


Figure 55: RMSE standard deviation of LSTM and TimeGPT models

6 CONCLUSION

This thesis presents a predictive maintenance architecture for the military domain called MILPdM. The architecture aims to anticipate failures of military equipment through the use of artificial intelligence algorithms. The architecture presented aims to mitigate problems of lack of data, considering the adverse environment in which military equipment operates and the challenges in simulating the vehicle's behavior in each scenario. In this sense, MILPdM incorporates FM into its failure prediction process, thus presenting an architecture capable of generalizing existing data and applying a new context, generating the ability to anticipate equipment degradation in new scenarios.

6.1 Contributions

The most significant contribution of the MILPdM is in its design, which considers the application of architecture for failure prediction considering the challenges of the military domain in its design. In this architecture we consider the use of FM in the failure prediction process and we use the broker proposed by Kunst et al. (2018) to guarantee QoS, considering the dynamics of a network operating in environments with the presence of enemies, such as on a battlefield.

MILPdM store the results of prediction models together with feedback from the maintenance team, which makes it possible to monitor the performance of the models, and, if necessary, retrain the models according to each scenario. This history makes it possible to reuse already trained models in cases where assets operate in environments already experienced in the past. That is if there is a model that has already received feedback from the maintenance team, reuse in a similar scenario is possible, and if necessary, the models can be optimized with the new data.

The architecture presents promising results for using FM in the failure prediction process using data in contexts not previously used in model training. The use of TimeGPT surpassed the results of traditional models in predicting time series, obtaining lower RMSE values. The results show potential in including FM in time series prediction tasks to achieve failure prediction capacity.

In a complementary way, the architecture allows estimate the useful life of an asset on the battlefield in run-time by sensing the assets and sending the data through the broker to a private cloud. The availability of data in the private cloud allows the command and control center to have access to data in run-time, performing monitoring and making the necessary decisions according to the current state of the health of the asset in operation.

Among other contributions, we can mention the systematic review of the literature, which helps the academic community to identify the challenges related to the application of PdM in the military domain, in addition to the approaches to achieve the ability to predict the degradation of assets, and in which assets PdM is tested and with which machine learning techniques. In addition to the systematic review, in this work, we present the use case of the truck model. This use case allowed the construction of a new dataset.

6.2 Future Work

Future directions involve establishing a threshold value for failure notifications, exploring alternative machine learning models for comparison, and using new FM (RASUL et al., 2023) for time series prediction. As the architecture proposes to operate in run-time, the training time is a pivotal metric for determining optimal models. Evaluating the broker's ability to determine the best network for data transmission must also be considered so that it is possible to deliver a run-time architecture.

With the evolution of the architecture and the development of new use cases, MILPdM will be able to predict the behavior of different assets in different scenarios. This capability enables new possibilities for using MILPdM, such as the inherent capabilities in this architecture lay the groundwork for developing learning models capable of offering alternative mission scenarios based on fleet availability and mission types. In other words, if we can determine how each asset behaves in different environments, the architecture can receive mission parameters, such as operating scenario and type of terrain, among other needs, and suggest the available asset that best fits the parameters of each mission. This developed capability can increase the chances of mission success and mitigate losses.

Finally, a prospective stage in the processing layer involves the classification of failure types and the identification of maintenance actions necessary to restore the health condition of the asset.

6.3 Accepted articles

This section highlights previously published articles relevant to the thesis topic and their contribution to the field of study.

6.3.1 Predictive maintenance in the military domain: a systematic review of the literature

In the (DALZOCHIO et al., 2023a) study, we seek to highlight the challenges, principles, scenarios, techniques, and open questions of PdM in the military domain. To achieve this objective, we conducted a systematic literature review. We organize the findings of the literature into challenges for applying PdM and scenarios where its use is crucial. These contributions will help the research community to understand the applications of PdM in the military domain.

6.3.2 MILPdM: A Predictive Maintenance Architecture for the Military Domain

In the (DALZOCHIO et al., 2023b) work, we present the MILPdM architecture, demonstrating its viability through use cases describing the vehicle's health degradation. This result obtained in the experiments demonstrates that MILPdM can anticipate failures with high assertiveness.

REFERENCES

AKRIM, A. et al. Self-supervised learning for data scarcity in a fatigue damage prognostic problem. **Engineering Applications of Artificial Intelligence**, v. 120, p. 105837, 2023.

ALPAYDIN, E. **Introduction to machine learning**. 3rd. ed. 2014. MIT Press.

ANDRADE LOPES, S. M. de; FLAUZINO, R. A.; ALTAFIM, R. A. C. Incipient fault diagnosis in power transformers by data-driven models with over-sampled dataset. **Electric Power Systems Research**, v. 201, p. 107519, 2021.

ARIAS CHAO, M. et al. Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. **Data**, v. 6, n. 1, p. 5, 2021.

ARMY, B. Administrative standards relating to weapons (normas administrativas relativas ao armamento - nara). , 2009.

ARMY, U. Army regulation 750-1 army materiel maintenance policy. , p. 225, 2019.

ATAMURADOV, V. et al. Prognostics and health management for maintenance practitioners-review, implementation and tools evaluation. **International Journal of Prognostics and Health Management**, v. 8, n. 060, p. 1–31, 2017.

BABBAR, A. et al. Advanced diagnostics and prognostics for engine health monitoring. In: IEEE AEROSPACE CONFERENCE, 2009., 2009. **Anais. . .** 2009.

BAJAJ, N. S. et al. A bayesian optimized discriminant analysis model for condition monitoring of face milling cutter using vibration datasets. **Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems**, v. 5, n. 2, p. 021002, 2021.

BAKER, W. et al. Degrader analysis for diagnostic and predictive capabilities: a demonstration of progress in dod cbm+ initiatives. **Procedia Computer Science**, v. 168, p. 257–264, 2020.

BALAKRISHNAN, N. et al. Aero-engine health monitoring with real flight data using whale optimization algorithm based artificial neural network technique. **Optical Memory and Neural Networks**, v. 30, n. 1, p. 80–96, 2021.

BANGHART, M. Identification of reverse engineering candidates utilizing machine learning and aircraft cannibalization data. **International Journal of Aviation, Aeronautics, and Aerospace**, v. 4, n. 4, p. 5, 2017.

BANKS, J. C. et al. Platform degrader analysis for the complex systems for the design and application of condition based maintenance. In: IEEE AEROSPACE CONFERENCE, 2014., 2014. **Anais. . .** 2014.

BARLOW, R.; HUNTER, L. Optimum preventive maintenance policies. **Operations research**, v. 8, n. 1, p. 90–100, 1960.

BAYOUMI, A.; MATTHEWS, R. Condition-based maintenance to predictive maintenance: a use case on selected usarmy military aircraft. **International Journal of COMADEM**, p. 3–8, 2020.

BEHERA, S. et al. Ensemble trees learning based improved predictive maintenance using iiot for turbofan engines. In: ACM/SIGAPP SYMPOSIUM ON APPLIED COMPUTING, 34., 2019. **Proceedings...** 2019.

BLECHERTAS, V. et al. Cbm fundamental research at the university of south carolina: a systematic approach to us army rotorcraft cbm and the resulting tangible benefits. In: AHS INTERNATIONAL SPECIALISTS' MEETING ON CONDITION BASED MAINTENANCE, 2009. **Proceedings...** 2009.

BOLLER, C. Ageing aircraft, health monitoring and maintenance. In: AEROINDIA SEMINAR ON ADVANCED AEROSPACE TECHNOLOGIE, BANGALORE,INDIA, 2002. **Anais...** 2002.

BOMMASANI, R. et al. On the opportunities and risks of foundation models. **arXiv preprint arXiv:2108.07258**, 2021.

BREIMAN, L. Random forests. **Machine learning**, v. 45, n. 1, p. 5–32, 2001.

BYINGTON, C. S.; ROEMER, M. J.; GALIE, T. Prognostic enhancements to diagnostic systems for improved condition-based maintenance [military aircraft]. In: IEEE AEROSPACE CONFERENCE, 2002. **Proceedings...** 2002.

CAMPOS, F. T.; MILLS, W. N.; GRAVES, M. L. A reference architecture for remote diagnostics and prognostics applications. In: IEEE AUTOTESTCON, 2002. **Proceedings...** 2002.

CARBERY, C. M.; WOODS, R.; MARSHALL, A. H. A bayesian network based learning system for modelling faults in large-scale manufacturing. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL TECHNOLOGY (ICIT), 2018., 2018. **Anais...** 2018.

CHAN, T. K.; CHIN, C. S. Health stages diagnostics of underwater thruster using sound features with imbalanced dataset. **Neural Computing and Applications**, v. 31, n. 10, p. 5767–5782, 2019.

CHANG, C.; PENG, W.-C.; CHEN, T.-F. Llm4ts: two-stage fine-tuning for time-series forecasting with pre-trained llms. **arXiv preprint arXiv:2308.08469**, 2023.

CHEN, C. et al. An integrated deep learning-based approach for automobile maintenance prediction with gis data. **Reliability Engineering & System Safety**, v. 216, p. 107919, 2021.

CHEN, S. et al. Prompt federated learning for weather forecasting: toward foundation models on meteorological data. **arXiv preprint arXiv:2301.09152**, 2023.

CHO, A. D.; CARRASCO, R. A.; RUZ, G. A. A rul estimation system from clustered run-to-failure degradation signals. **Sensors**, v. 22, n. 14, p. 5323, 2022.

CIPOLLINI, F. et al. Condition-based maintenance of naval propulsion systems with supervised data analysis. **Ocean Engineering**, v. 149, p. 268–278, 2018.

COOK, J. Reducing military helicopter maintenance through prognostics. In: IEEE AEROSPACE CONFERENCE, 2007., 2007. **Anais...** 2007.

DALZOCHIO, J. et al. Machine learning and reasoning for predictive maintenance in industry 4.0: current status and challenges. **Computers in Industry**, v. 123, p. 103298, 2020.

DALZOCHIO, J. et al. Predictive maintenance in the military domain: a systematic review of the literature. **ACM Computing Surveys**, v. 55, n. 13s, p. 1–30, 2023.

DALZOCHIO, J. et al. Milpdm: a predictive maintenance architecture for the military domain. In: IEEE INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND APPLICATIONS (ICMLA), 2023., 2023. **Anais...** 2023.

DAS, A. et al. A decoder-only foundation model for time-series forecasting. **arXiv preprint arXiv:2310.10688**, 2023.

DESELL, T. et al. Evolving neural network weights for time-series prediction of general aviation flight data. In: INTERNATIONAL CONFERENCE ON PARALLEL PROBLEM SOLVING FROM NATURE, 2014. **Anais...** 2014.

DUCOFFE, M. et al. Anomaly detection on time series with wasserstein gan applied to phm. **PHM Applications of Deep Learning and Emerging Analytics. International Journal of Prognostics and Health Management Reviewed (Special Issue)**, 2019.

ĎULÍK, M.; JUNIOR, M. Ď. Security in military cloud computing applications. **Science and Military Journal**, v. 11, p. 26–33, 2016.

EKAMBARAM, V. et al. Tsmixer: lightweight mlp-mixer model for multivariate time series forecasting. **arXiv preprint arXiv:2306.09364**, 2023.

FERNÁNDEZ-BARRERO, D. et al. Soprene: assessment of the spanish armada's predictive maintenance tool for naval assets. **Applied Sciences**, v. 11, n. 16, p. 7322, 2021.

FURCH, J.; NGUYEN, T. T.; GLOS, J. Diagnostics of gear fault in four-speed gearbox using vibration signal. In: INTERNATIONAL CONFERENCE ON MILITARY TECHNOLOGIES (ICMT), 2017., 2017. **Anais...** 2017.

GARZA, A.; MERGENTHALER-CANSECO, M. Timegpt-1. **arXiv preprint arXiv:2310.03589**, 2023.

GÉRON, A. **Hands-on machine learning with scikit-learn, keras, and tensorflow**: concepts, tools, and techniques to build intelligent systems. 2019. O'Reilly Media.

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. A. Learning to forget: continual prediction with lstm. **Neural Computation**, v. 12, p. 2451–2471, 2000.

GIORDANO, D. et al. Dissecting a data-driven prognostic pipeline: a powertrain use case. **Expert Systems with Applications**, v. 180, p. 115109, 2021.

GODAHEWA, R. et al. Monash time series forecasting archive. **Neural Information Processing Systems Track on Datasets and Benchmarks**, 2021.

GUTHULA, S. et al. netfound: foundation model for network security. **arXiv preprint arXiv:2310.17025**, 2023.

HARTSHORN, S. Machine learning with random forests and decision trees: a visual guide for beginners. **Kindle edition**, 2016.

HOMBORG, A.; TINGA, T.; MOL, J. Listening to corrosion. In: AVT-305 RESEARCH SPECIALISTS' MEETING ON SENSING SYSTEMS FOR INTEGRATED VEHICLE HEALTH MANAGEMENT FOR MILITARY VEHICLES, 2018. **Anais. . .** 2018.

HRÚZ, M. et al. The use of uav with infrared camera and rfid for airframe condition monitoring. **Applied Sciences**, v. 11, n. 9, p. 3737, 2021.

HU, C. et al. Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. **Reliability Engineering & System Safety**, v. 103, p. 120–135, 2012.

HUANG, M.; LIU, Z.; TAO, Y. Mechanical fault diagnosis and prediction in iot based on multi-source sensing data fusion. **Simulation Modelling Practice and Theory**, v. 102, p. 101981, 2020.

IANNACE, G.; CIABURRO, G.; TREMATERRA, A. Fault diagnosis for uav blades using artificial neural network. **Robotics**, v. 8, n. 3, p. 59, 2019.

JARDINE, A. K.; LIN, D.; BANJEVIC, D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. **Mechanical systems and signal processing**, v. 20, n. 7, p. 1483–1510, 2006.

JARDINE, A. K.; TSANG, A. H. **Maintenance, replacement, and reliability**: theory and applications. 2013. CRC press.

JIN, M. et al. Large models for time series and spatio-temporal data: a survey and outlook. **arXiv preprint arXiv:2310.10196**, 2023.

JIN, M. et al. Time-llm: time series forecasting by reprogramming large language models. **arXiv preprint arXiv:2310.01728**, 2023.

JO, T. **Machine learning foundations**: supervised, unsupervised, and advanced learning. 2021. Springer Nature.

KÁLA, M.; LALIŠ, A.; VITTEK, P. Optimizing calculation of maintenance revision times in maintenance repair organizations. In: INTERNATIONAL CONFERENCE ON MILITARY TECHNOLOGIES (ICMT), 2019., 2019. **Anais. . .** 2019.

KHATRI, H. et al. New features for diagnosis and prognosis of systems based on empirical mode decomposition. In: INTERNATIONAL CONFERENCE ON PROGNOSTICS AND HEALTH MANAGEMENT, 2008., 2008. **Anais. . .** 2008.

KILLEEN, P. et al. Iot-based predictive maintenance for fleet management. **Procedia Computer Science**, v. 151, p. 607–613, 2019.

KIM, N.-H.; AN, D.; CHOI, J.-H. Prognostics and health management of engineering systems. **Switzerland: Springer International Publishing**, 2017.

KITCHENHAM, B. Procedures for performing systematic reviews. **Keele, UK, Keele University**, v. 33, n. 2004, p. 1–26, 2004.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering–a tertiary study. **Information and software technology**, v. 52, n. 8, p. 792–805, 2010.

KOBBACY, K. A. H.; MURTHY, D. P. **Complex system maintenance handbook**. 2008. Springer Science & Business Media.

KORDESTANI, M. et al. Failure prognosis and applications—a survey of recent literature. **IEEE transactions on reliability**, 2019.

KOSZTYÁN, Z. T.; PRIBOJSZKI-NÉMETH, A.; SZALKAI, I. Hybrid multimode resource-constrained maintenance project scheduling problem. **Operations Research Perspectives**, v. 6, p. 100129, 2019.

KUNDU, P.; DARPE, A. K.; KULKARNI, M. S. A review on diagnostic and prognostic approaches for gears. **Structural Health Monitoring**, p. 1475921720972926, 2020.

KUNST, R. et al. A resources sharing architecture for heterogeneous wireless cellular networks. In: IEEE CONFERENCE ON LOCAL COMPUTER NETWORKS, 2016. **Anais. . .** 2016.

KUNST, R. et al. Improving network resources allocation in smart cities video surveillance. **Computer Networks**, v. 134, p. 228–244, 2018.

LALL, P.; LOWE, R.; GOEBEL, K. Particle swarm optimization with extended kalman filter for prognostication of accrued damage in electronics under temperature and vibration. In: IEEE CONFERENCE ON PROGNOSTICS AND HEALTH MANAGEMENT, 2012., 2012. **Anais. . .** 2012.

LALL, P.; LOWE, R.; GOEBEL, K. Prognostication of accrued damage in board assemblies under thermal and mechanical stresses. In: IEEE 62ND ELECTRONIC COMPONENTS AND TECHNOLOGY CONFERENCE, 2012., 2012. **Anais. . .** 2012.

LAMAS-LÓPEZ, F. et al. Predictive maintenance of naval assets using machine learning techniques. In: STO-MP-SAS-OCS-ORA-2021 NATO CONFERENCE. AIML-03 SECTION, 2022. **Proceedings. . .** 2022.

LE, V. T. et al. Condition monitoring of engine lubrication oil of military vehicles: a machine learning approach. In: AUSTRALIAN INTERNATIONAL AEROSPACE CONGRESS: AIAC 2017, 17., 2017. **Anais. . .** 2017.

LEÃO, B. P. et al. Cost-benefit analysis methodology for phm applied to legacy commercial aircraft. In: IEEE AEROSPACE CONFERENCE, 2008., 2008. **Anais. . .** 2008.

LEE, J. et al. **Bearing data set**. NASA Prognostics Data Repository, NASA Ames Research Center, Moffett Field, CA.

LEE, J. et al. Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. **Mechanical systems and signal processing**, v. 42, n. 1-2, p. 314–334, 2014.

LEI, Y. et al. Machinery health prognostics: a systematic review from data acquisition to rul prediction. **Mechanical Systems and Signal Processing**, v. 104, p. 799–834, 2018.

LI, X.; ZHAO, X.-d.; PU, W. An approach for predicting digital material consumption in electronic warfare. **Defence Technology**, v. 16, n. 1, p. 263–273, 2020.

LIAW, A.; WIENER, M. et al. Classification and regression by randomforest. **R news**, v. 2, n. 3, p. 18–22, 2002.

LIJUN, C. et al. Research on quality evaluation and fault forecast system for complicated equipments. In: INTERNATIONAL CONFERENCE ON ELECTRONIC MEASUREMENT AND INSTRUMENTS, 2007., 2007. **Anais...** 2007.

LIN, L.; LUO, B.; ZHONG, S. Multi-objective decision-making model based on cbm for an aircraft fleet with reliability constraint. **International Journal of Production Research**, v. 56, n. 14, p. 4831–4848, 2018.

LIU, L.; CARTES, D. A.; QUIROGA, J. Modeling and simulation for condition based maintenance: a case study in navy ship application. In: SUMMER COMPUTER SIMULATION CONFERENCE, 2007., 2007. **Proceedings...** 2007.

LIU, L. et al. Fault detection, diagnostics, and prognostics: software agent solutions. **IEEE Transactions on Vehicular Technology**, v. 56, n. 4, p. 1613–1622, 2007.

LUO, J. et al. Model-based prognostic techniques [maintenance applications]. In: AUTOTESTCON 2003. IEEE SYSTEMS READINESS TECHNOLOGY CONFERENCE., 2003. **Proceedings...** 2003.

MCNAUGHT, K.; ZAGORECKI, A.; PEREZ, A. G. Knowledge elicitation for predictive maintenance modelling with bayesian networks. , 2011.

MIN, H.; WOOD, R.; JOO, S.-J. Machine learning methods and predictive modeling to identify failures in the military aircraft: aicraft quality control. **International Journal of Industrial Engineering: Theory, Applications and Practice**, v. 30, n. 5, 2023.

MOBLEY, R. K. **An introduction to predictive maintenance**. 2002. Elsevier.

NAKAGAWA, T. Periodic and sequential preventive maintenance policies. **Journal of Applied probability**, p. 536–542, 1986.

NEMANI, V. et al. Uncertainty quantification in machine learning for engineering design and health prognostics: a tutorial. **Mechanical Systems and Signal Processing**, v. 205, p. 110796, 2023.

NIE, Y. et al. A time series is worth 64 words: long-term forecasting with transformers. **arXiv preprint arXiv:2211.14730**, 2022.

NIXON, S. et al. A machine learning approach to diesel engine health prognostics using engine controller data. In: ANNUAL CONFERENCE OF THE PHM SOCIETY, 2018. **Anais...** 2018.

NORDAL, H.; EL-THALJI, I. Lifetime benefit analysis of intelligent maintenance: simulation modeling approach and industrial case study. **Applied Sciences**, v. 11, n. 8, p. 3487, 2021.

OKOH, C. et al. Overview of remaining useful life prediction techniques in through-life engineering services. **Procedia Cirp**, v. 16, p. 158–163, 2014.

OMRI, N. et al. Towards an adapted phm approach: data quality requirements methodology for fault detection applications. **Computers in Industry**, v. 127, p. 103414, 2021.

ORTEGA CARO, J. et al. Brainlm: a foundation model for brain activity recordings. **bioRxiv**, p. 2023–09, 2023.

PAL, P. et al. Condition based maintenance of turbine and compressor of a codlag naval propulsion system using deep neural network. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND APPLICATIONS (AIAP-2019), 6., 2019. **Anais...** 2019.

PATER, I. de; MITICI, M. Developing health indicators and rul prognostics for systems with few failure instances and varying operating conditions using a lstm autoencoder. **Engineering Applications of Artificial Intelligence**, v. 117, p. 105582, 2023.

PECHT, M. G.; KANG, M. Machine learning: diagnostics and prognostics. In: ______. **Prognostics and health management of electronics**: fundamentals, machine learning, and the internet of things. 2019.

PESCHIERA, F. et al. A novel solution approach with ml-based pseudo-cuts for the flight and maintenance planning problem. **OR Spectrum**, p. 1–30, 2020.

PI, R. Raspberry pi 3 model b. **online].(https://www. raspberrypi. org**, 2015.

PRAJAPATI, A.; BECHTEL, J.; GANESAN, S. Condition based maintenance: a survey. **Journal of Quality in Maintenance Engineering**, 2012.

PURNAMA, H. C.; SUSMARTINI, S.; HERDIMAN, L. Implementation of predictive maintenance on defense radar using fmeca method and php software to support maintenance action decision making. In: BOROBUDUR INTERNATIONAL SYMPOSIUM ON SCIENCE AND TECHNOLOGY 2022 (BIS-STE 2022), 4., 2023. **Proceedings...** 2023.

QIU, H. et al. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. **Journal of sound and vibration**, v. 289, n. 4-5, p. 1066–1090, 2006.

RAHEJA, D. et al. Data fusion/data mining-based architecture for condition-based maintenance. **International Journal of Production Research**, v. 44, n. 14, p. 2869–2887, 2006.

RANASINGHE, K. et al. Vehicular sensor network and data analytics for a health and usage management system. **Sensors**, v. 20, n. 20, p. 5892, 2020.

RASUL, K. et al. Lag-llama: towards foundation models for time series forecasting. **arXiv preprint arXiv:2310.08278**, 2023.

REBALA, G.; RAVI, A.; CHURIWALA, S. Machine learning definition and basics. In: **An introduction to machine learning**. 2019. Springer.

ROEMER, M. J. et al. Validation and verification of prognostic and health management technologies. In: IEEE AEROSPACE CONFERENCE, 2005., 2005. **Anais...** 2005.

ROY, T.; DEY, S. Fault detectability conditions for linear deterministic heat equations. **IEEE control systems letters**, v. 3, n. 1, p. 204–209, 2018.

RUI, J.; XIAOFAN, H.; YUHAI, L. Individual aircraft life monitoring: an engineering approach for fatigue damage evaluation. **Chinese Journal of Aeronautics**, v. 31, n. 4, p. 727–739, 2018.

RUSSELL, S.; ABDELZAHER, T. The internet of battlefield things: the next generation of command, control, communications and intelligence (c3i) decision-making. In: MILCOM 2018-2018 IEEE MILITARY COMMUNICATIONS CONFERENCE (MILCOM), 2018. **Anais...** 2018.

SHAH, E. et al. Comparative study on estimation of remaining useful life of turbofan engines using machine learning algorithms. In: **Intelligent systems and sustainable computing**: proceedings of icissc 2021. 2022. Springer.

SHAMAYLEH, A.; AWAD, M.; FARHAT, J. Iot based predictive maintenance management of medical equipment. **Journal of Medical Systems**, v. 44, n. 4, p. 1–12, 2020.

SHAO, X. J. et al. Abrasion lives simulation and prediction of key parts for breech mechanism based on pro/e and adams. In: ADVANCED MATERIALS RESEARCH, 2012. **Anais...** 2012.

SI, X.-S. et al. Remaining useful life estimation–a review on the statistical data driven approaches. **European journal of operational research**, v. 213, n. 1, p. 1–14, 2011.

SIEGEL, D.; GHAFFARI, M.; LEE, J. Failure prediction of critical components in military vehicles. **Proceedings of the Society for Machinery Failure Prevention Technology**, 2008.

SMITH, M. J.; FLEMING, L.; GEACH, J. E. Earthpt: a foundation model for earth observation. **arXiv preprint arXiv:2309.07207**, 2023.

TAGLIENTE, D. A.; LUDWIG, A.; MARSTON, D. Condition based maintenance+ data collection and offloading. In: IEEE AUTOTESTCON, 2019., 2019. **Anais...** 2019.

TAMBE, S. et al. An extensible cbm architecture for naval fleet maintenance using open standards. In: INTELLIGENT SHIP SYMPOSIUM, BOSTON, USA, 2015. **Anais...** 2015.

TIAN, H.; YANG, L.; JU, B. Spatial correlation and temporal attention-based lstm for remaining useful life prediction of turbofan engine. **Measurement**, v. 214, p. 112816, 2023.

TINGA, T. Application of physical failure models to enable usage and load based maintenance. **Reliability Engineering & System Safety**, v. 95, n. 10, p. 1061–1075, 2010.

TINGA, T. Predictive maintenance of military systems based on physical failure models. **CHEMICAL ENGINEERING**, v. 33, 2013.

TINGA, T. et al. Advanced predictive maintenance concepts based on the physics of failure. In: **Optimal deployment of military systems. technologies for military missions in the next decade**. 2014. TMC Asser Press.

TINGA, T. et al. Dynamic maintenance based on functional usage profiles. **Journal of quality in maintenance engineering**, 2020.

TSANG, A. H. Condition-based maintenance: tools and decision making. **Journal of quality in maintenance engineering**, 1995.

ULRICSON, A.; MICKLE, I. C.; SANDERS, I. T. Sequential pattern mining for maintenance prediction modeling for military helicopters. In: IIE ANNUAL CONFERENCE. PROCEEDINGS, 2023. **Anais...** 2023.

VACHTSEVANOS, G. J.; VALAVANIS, K. P. A novel approach to integrated vehicle health management. **NATO STO: Brussels**, 2018.

VALIŠ, D.; ŽÁK, L.; POKORA, O. Contribution to system failure occurrence prediction and to system remaining useful life estimation based on oil field data. **Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability**, v. 229, n. 1, p. 36–45, 2015.

VIDYASAGAR, K. Optimization of aero engine utilization through improved estimation of remaining useful life (rul) of on condition (oc) parts. **International Journal of Mechanical Engineering and Technology**, 2020.

VILLA, V. et al. Iot open-source architecture for the maintenance of building facilities. **Applied Sciences**, v. 11, n. 12, p. 5374, 2021.

VOGL, G. W.; WEISS, B. A.; HELU, M. A review of diagnostic and prognostic capabilities and best practices for manufacturing. **Journal of Intelligent Manufacturing**, v. 30, n. 1, p. 79–95, 2019.

WANG, H.; PHAM, H. **Reliability and optimal maintenance**. 2006. Springer Science & Business Media.

WANG, H.; RAJ, B. On the origin of deep learning. , 2017.

WANG, Y.-C. Prediction of engine failure time using principal component analysis, categorical regression tree, and back propagation network. **Journal of Ambient Intelligence and Humanized Computing**, p. 1–9, 2018.

WOLDMAN, M. et al. Abrasive wear based predictive maintenance for systems operating in sandy conditions. **Wear**, v. 338, p. 316–324, 2015.

XIONGZI, C. et al. Remaining useful life prognostic estimation for aircraft subsystems or components: a review. In: IEEE 2011 10TH INTERNATIONAL CONFERENCE ON ELECTRONIC MEASUREMENT & INSTRUMENTS, 2011. **Anais...** 2011.

YAM, R. et al. Intelligent predictive decision support system for condition-based maintenance. **The International Journal of Advanced Manufacturing Technology**, v. 17, n. 5, p. 383–391, 2001.

YAN, J. et al. Industrial big data in an industry 4.0 environment: challenges, schemes, and applications for predictive maintenance. **IEEE Access**, v. 5, p. 23484–23491, 2017.

YEH, C.-C. M. et al. Toward a foundation model for time series data. In: ACM INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 32., 2023. **Proceedings...** 2023.

YEH, C.-C. M. et al. Multitask learning for time series data\\with 2d convolution. **arXiv preprint arXiv:2310.03925**, 2023.

YIWEI, W. et al. A model-based prognostics method for fatigue crack growth in fuselage panels. **Chinese Journal of Aeronautics**, v. 32, n. 2, p. 396–408, 2019.

YOON, J.; JARRETT, D.; SCHAAR, M. Van der. Time-series generative adversarial networks. **Advances in neural information processing systems**, v. 32, 2019.

ZHANG, D. et al. Brant: foundation model for intracranial neural signal. In: THIRTY-SEVENTH CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 2023. **Anais...** 2023.

ZHOU, C. et al. A comprehensive survey on pretrained foundation models: a history from bert to chatgpt. **arXiv preprint arXiv:2302.09419**, 2023.

ZHOU, T. et al. One fits all: universal time series analysis by pretrained lm and specially designed adaptors. **arXiv preprint arXiv:2311.14782**, 2023.