Kadripathi Kunchitigara Palya Narasimhaiah

Real-Time Fault Diagnosis in Aircraft Landing Gear: A Novel Two-Tier ML Approach with Intelligent Sensor Data Management and Explainable AI

School of Aerospace, Transport, and Manufacturing
Applied Artificial Intelligence

MSc
Academic Year: 2022 - 2023

Supervisor: Dr Dmitry Ignatyev
Associate Supervisor: Adolfo Perrusquia
August 2023

School of Aerospace, Transport, and Manufacturing
Applied Artificial Intelligence

MSc

Academic Year 2022 - 2023

Kadripathi Kunchitigara Palya Narasimhaiah

Real-Time Fault Diagnosis in Aircraft Landing Gear: A Novel Two-Tier ML Approach with Intelligent Sensor Data Management and Explainable AI

Supervisor: Dr Dmitry Ignatyev
Associate Supervisor: Adolfo Perrusquia
August 2023

This thesis is submitted in partial fulfilment of the requirements for the degree of MSc in Applied Artificial Intelligence

Academic integrity declaration

I declare that:

- the thesis submitted has been written by me alone.
- the thesis submitted has not been previously submitted to this university or any other.
- that all content, including primary and/or secondary data, is true to the best of my knowledge.
- that all quotations and references have been duly acknowledged according to the requirements of academic research.

I understand that to knowingly submit work in violation of the above statement will be considered by examiners as academic misconduct.

Abstract

This research introduces a pioneering approach for intelligent real-time fault diagnosis in aircraft landing gear actuation systems. The continuous monitoring of landing gear health is crucial for ensuring safety, minimizing accidents, and optimizing maintenance schedules and costs. The complexity and interconnectivity of modern aircraft systems pose significant challenges to real-time fault detection, particularly when sensors malfunction or are uncalibrated. To address these challenges, we propose an innovative two-tier machine learning framework specifically designed for diagnosing faults in landing gear systems, emphasizing hydraulic failure modes. Using a simulation-based approach, we generate data that mirrors complex hydraulic failures, laying a solid foundation for our machine-learning models. The primary model is geared towards sophisticated fault classification. Meanwhile, the secondary imputation model excels in managing missing or inconsistent data from malfunctioning or uncalibrated sensors. Remarkably, it employs and optimizes data even in the presence of these sensor issues, which traditionally compromise health assessments. By harnessing redundant sensors, a deviation from typical data omission strategies, we observed enhanced accuracy in primary classifiers; for instance, Decision Trees' accuracy increased from 95.88% to 98.76% after imputation. Further enriching this approach, we integrate Explainable AI (XAI) techniques, ensuring transparency and elucidating predictions for stakeholders. Notably, this research accentuates the pivotal roles of crucial sensors temperature and pump speed sensors in assessing landing gear health, championing their adoption in aviation systems. This study has profound implications, fostering heightened safety measures and spearheading cost-effective, data-driven advancements in the aviation sector.

Keywords: Intelligent real-time fault diagnosis, Aircraft landing gear actuation systems, Two-tier machine learning framework, Hydraulic failure modes, Simulation-based approach, Primary and secondary imputation model, Redundant sensors, Explainable AI (XAI)

Acknowledgements

I wish to convey my deepest gratitude to all who have played an indispensable role in the realization of this thesis, titled "Real-Time Fault Diagnosis in Aircraft Landing Gear: A Novel Two-Tier ML Approach with Intelligent Sensor Data Management and Explainable AI."

First and foremost, my heart brims with appreciation for my parents, who saw beyond our circumstances to recognize the potential in my dreams. To my father, whose unwavering emotional and financial support has been a pillar of strength, and to my mother, whose sacrifices, both seen and unseen, have been the wind beneath my wings – I owe the very foundation of this achievement.

I am deeply indebted to my main supervisor, Dr Dmitry Ignatyev, whose guidance, expertise, and tireless dedication have been the guiding force throughout this journey. I also extend my sincere gratitude to my associate supervisor, Adolfo Perrusquia, for his invaluable insights, patience, and commitment to this work. He was always on his toe to get the best out of me!

My appreciation extends to my peers, classmates, and especially my sister – your support, enriching discussions, and shared moments of enlightenment have been the sources of both inspiration and solace. Even as the youngest among you, I have felt the warmth of your collective embrace.

On a philosophical note, I bow in gratitude to the Divine, for I believe in the universe's benevolent design and the guiding hand of God, which has constantly directed me towards the path of wisdom and perseverance.

I wish to take a moment to acknowledge my own journey. The dedication, late nights, perseverance, and countless hours spent immersed in research weren't easy. This accomplishment serves as a testament to my passion and determination.

Lastly, to you, the reader of this thesis – thank you for your time, interest, and the honor of your attention. Your engagement breathes life into these pages and actualizes the efforts poured into this research.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
ICAO	International Civil Aviation Organization
PHM	Prognostics and Health Management
XAI	Explainable Artificial Intelligence
ATA 32	Air Transport Association Chapter 32: Landing Gear
TapAir	Hydraulic fluid-to-air ratio
RPM	Revolutions Per Minute
EDA	Exploratory Data Analysis
KNN	K-Nearest Neighbors
XGBoost	Extreme Gradient Boosting
GBM	Gradient Boosting Machine
L1	Lasso Regression
L2	Ridge Regression
SHAP	SHapley Additive exPlanations

1 Introduction

1.1 Background of the Problem

Aircraft, epitomising the zenith of modern engineering, comprise an intricate matrix of systems and subsystems functioning cohesively to guarantee a secure flight. At the heart of this matrix lies the landing gear system, an indispensable, non-redundant element of an aircraft's architecture. Acting as a conduit between the aircraft and the ground, it encompasses a range of dynamic components, including the landing gear, wheels, brakes, shock absorbers, retraction mechanisms, control valves, and supplementary systems. These systems coalesce to facilitate secure landings and take-offs.

The efficient functioning of landing gear systems is underpinned by an intricate matrix of mechanical, electrical, and hydraulic elements, all harmoniously orchestrated for peak performance. Notably, each component within an aircraft is usually complemented by a backup or redundant system to ensure continuity of operation in the event of a malfunction. For instance, should a primary thruster fail, the aircraft can seamlessly switch to an auxiliary thruster, enabling the flight to proceed to the nearest suitable landing site. This principle of redundancy applies to various subsystems, such as navigation, communication, power supply, and fuel systems, among others. However, for a paramount system like the landing gear, there exists no alternative or backup mechanism to fall back on in the event of failure. Consequently, it becomes imperative to constantly monitor the health of the landing gear in real-time.

Traditionally, real-time health assessments have been conducted using both model-driven and data-driven approaches. However, utilising AI with data-driven techniques for monitoring health has proven to be more efficient, prompting a shift within the aviation industry towards diagnosing health using data-driven AI methodologies. These AI-powered ML models rely heavily on data procured from sensors. If a sensor malfunctions or sustains damage, the input data to the model is inevitably compromised, which can lead to inaccurate health status assessments by the model. For instance, should a system be functioning optimally but the sensor suffers from wear and tear, the health assessment system may issue a false alarm. Conversely, a faulty sensor might lead the model to incorrectly predict a component failure, when in reality, the component is functioning as intended. This research endeavours to pioneer an innovative advanced two-tier model approach that accounts for data from malfunctioning or uncalibrated sensors, subsequently channelling this data into an ML classification model to yield accurate health assessments. This methodology markedly diverges from conventional ML practices.

In the current research, undertaken in collaboration with our industrial partner, AIRBUS, we endeavour to construct a proof of concept for a data-driven fault diagnosis technique. This methodology is tailored to effectively detect and pinpoint anomalies in the Landing Gear Actuation System at the component level. Notably, the approach is proficient in managing data from malfunctioning or uncalibrated sensors. Beyond the confines of this proposal, our research aims to evaluate and contrast the efficacy of the classification model aligned with this two-tier model strategy. An integral facet of our research is the incorporation of explainable AI, which stands paramount in augmenting result interpretability. This enhancement is pivotal in fostering the method's endorsement amongst technicians, engineers, and regulatory institutions.

1.2 Importance of the Problem

The health monitoring of landing gear presents not only a technological challenge but is also a crucial issue of passenger safety and economic efficiency. Malfunctions can precipitate accidents, endanger lives, disrupt flight schedules, and lead to substantial maintenance and repair costs. Every year, nearly 18% of all aviation accidents can be traced back to landing gear failure underscoring an urgent need for improved diagnostic systems. According to the International Civil Aviation Organization (ICAO), 756 accidents occurred during the landing phase due to faulty landing gear from 2013 to 2022. Of these, 51 were fatal, resulting in 2072 fatalities [1].

Currently, the Prognostics and Health Management (PHM) of these systems have been achieved through model-based methods and data-driven approaches. Model-based methods involve creating virtual models of the current system mathematically, to evaluate the system's performance. Although this approach requires an accurate mathematical model, time, and significant computational power, it has its drawbacks. These methods require an in-depth understanding of the system and its behavior, which may not always be feasible, especially in complex systems like an aircraft's hydraulic system. They may also fail to predict failures accurately on time due to unforeseen circumstances or factors not included in the model. For example, in the Saudi Airlines A330-200 incident in 2018, the aircraft had to make an emergency landing due to a failure in the nose landing gear. This failure was detected and reported by model-based health monitoring 6 mins after the casualty occurred [2].

On the other hand, data-driven methods require less prior knowledge of the system and can uncover hidden information through data processing and analysis. However, the conventional way of health monitoring by ML models poses a challenge when it encounters faulty and uncalibrated sensor data/signals.

In the A343 Helsinki Finland incident in 2009, a 'too hot temperature error' in the landing gear detected during aircraft takeoff turned out to be a hydraulic system leak into the flight. Although the aircraft landed safely, one of the two hydraulic lines was empty when the aircraft stopped [3]. This is a clear example of misinformation fed to a data-driven AI model by a faulty sensor. Similarly, this incident highlights the need for advanced real-time, data-driven robust health monitoring at the aircraft sub-component level.

1.3 Overview of the Research Objective

In this intricate scenario, the research aims to design a two-tire intelligent, robust, and data-driven machine learning methodology for real-time fault diagnosis in the landing gear actuation system. The focus is primarily on hydraulic failure modes. The proposed approach strives to accurately detect faults at the component level, manage multimode failure cases, and handle data from faulty and uncalibrated sensors.

The effectiveness of machine learning algorithms for this newly proposed model will be evaluated and compared. Additionally, the potential of integrating explainable AI techniques will be examined, which is crucial for improving result interpretability and hence, the method's acceptability by technicians, engineers, and regulatory bodies. The study will also highlight critical sensors for the health assessment of the landing gear, a pivotal aspect of any fault detection system.

1.3.1 Objectives of the research

To achieve the primary aim of the research, the following objectives have been delineated:

a) **Data Simulation for Multimodal Failures**: Harness the power of simulation techniques to gather data across a plethora of multimodal hydraulic failure scenarios, replicating real-world conditions.
b) **Failure Scenario Definition and Benchmarking**: Delve deep into the specific faulty scenarios, accurately defining them and establishing distinct benchmark parameters to discern and validate each scenario's nuances.
c) **Development of a Two-Tier ML Model**:
    - Institute a primary model dedicated to the intricate task of fault classification.
    - Complement with a secondary imputation model, serving as an adept data preprocessing layer. This module identifies and rectifies null values and outliers in the input signal, leveraging redundant sensor data. The refined data then seamlessly transitions to the primary model, setting the stage for accurate predictions.
d) **Classifier Evaluation**: Methodically assess the performance of an array of classifiers, determining their compatibility and effectiveness in the context of this research.
e) **Transparent Decision-Making with Explainable AI**: Empower the chosen classifier to elucidate its decision-making process. By integrating

explainable AI mechanisms, we ensure transparency, fostering trust in the model's outputs.

f) **Sensor Significance Analysis**: Unearth and spotlight the sensors that are indispensable for the health assessment of the landing gear. This endeavor promises to guide future endeavors, influencing the strategic design and integration of sensors within the aircraft landing gear architecture.

Building on the aforementioned objectives, it's imperative to approach our research with a clear understanding of its inherent limitations and the assumptions underpinning it. In the following section, we delve into these aspects to ensure clarity and transparency throughout the investigative process.

1.3.2 Limitations and Assumptions

I. **Simulation Over Real-Time Data**: Our primary limitation lies in the absence of real-time data pertaining to the extension and retraction of landing gear. While actual world data would undeniably be a more accurate medium for training the model, we have sought to approximate this by relying on simulation. Within this simulation, we've endeavoured to mimic real-world scenario signals, incorporating noise and anomalies into the signal to lend it authenticity.

II. **Scope of Failure Scenarios**: For the purpose of this research, our scope is restricted to only hydraulic failures, encompassing 12 multimodal failure case scenarios. In the real-world context, there's the potential for a wider range of failure modes, including but not limited to structural failures. These structural failures are not encompassed within the purview of our current investigation.

III. **Sensor Malfunction Parameters**: Our analysis assumes two primary modes of sensor malfunction:

a. Faulty or non-operational sensors, which produce null values.

b. Uncalibrated sensors, yielding values that are unrealistically high or low in comparison to typical readings.

It's crucial to note that in real-world applications, additional malfunctioning modes could exist. For instance, sensor signals may experience interference due to external factors such as electrical or magnetic fields. Our model, in its current state, does not account for such complexities.

Having elucidated our objectives and the inherent limitations, it's worth reiterating the broader implications of this research. The aviation industry stands at a pivotal juncture where every innovation, no matter how incremental, can lead to monumental shifts in safety, efficiency, and overall passenger experience. By realising the goals set out for this study, we envisage a profound contribution to the realm of aviation. Enhanced safety protocols, fewer flight disruptions, and diminished maintenance costs are not just aspirational targets; they are keystones for a future where air travel is consistently reliable, efficient, and above all, safe.

The ramifications of this research extend beyond the technical domain. They resonate deeply with the societal and economic fabric of aviation. In an industry that moves millions every day, a nuanced approach towards improved safety and efficiency is tantamount to a giant stride forward.

As we transition to the subsequent sections, the literature review will shed light on previous endeavours, studies, and innovations that have shaped our current understanding. This exploration will anchor our research within the broader discourse, providing a foundation upon which we build our study.

1.4 Structure of the thesis

This thesis delves into the nexus of landing gear health diagnostics and advanced machine learning methods:

- **Chapter 1** introduces the backdrop, significance, and objectives of landing gear fault diagnosis while also touching on its constraints.
- **Chapter 2** presents a literature review, identifying gaps in the current knowledge and justifying this study.
- **Chapter 3** details the methodology, from data generation and feature extraction to the use of various classification and imputation models, highlighting the role of explainability in decisions.
- **Chapter 4** offers a glimpse into the research's risks and presents a timeline for its progression.
- **Chapter 5** provides the results of the models, offering evaluations within the realm of landing gear diagnostics.
- **Chapter 6** delves into explainable AI techniques, using a case study for emphasis.
- **Chapter 7** shifts to discussing the economic implications of key sensors in aircraft maintenance.

- **Chapter 8** concludes with insights, while **Chapter 9** sketches out future research avenues, including real-world data integration and environmental factor considerations.

The main narrative is complemented by references and appendices, with an ethical approval letter and research specifics. This work aims to blend machine learning with aircraft diagnostics for a safer and economically efficient aviation landscape.

2 Literature Review

2.1 Historical Overview of Landing Gear Health Monitoring

The landing gear system, being a critical component of aircraft, has been the subject of extensive research and development over the past decades. Historically, health monitoring of the landing gear was primarily based on periodic inspections and maintenance schedules [4]. However, with the advent of technology, real-time health monitoring systems have gained prominence.

Phillips et al. (2011) discussed the evolution of landing gear health monitoring systems, highlighting the transition from manual inspections to automated systems [5]. The study emphasized the importance of real-time monitoring in enhancing aircraft safety and reducing maintenance costs.

Boniol et al. (2016) provided a comprehensive review of the mechanical and hydraulic components of the landing gear system. They discussed the challenges associated with monitoring these components and underscored the need for advanced diagnostic systems [6].

2.2 Model-Driven vs. Data-Driven Approaches in Aircraft Health Monitoring

2.2.1 Model-Driven Approaches

Historically, model-driven techniques, grounded in mathematical or physical models, have been the mainstay. These models, derived from fundamental principles, offer predictions based on well-established scientific laws.

Kang et al. (2023) delved into the intricacies of model-driven techniques for predicting landing gear failures. Their research highlighted the challenges of modeling complex interactions within the landing gear system. They argued that while these models provide a structured framework, their rigidity can sometimes be a limitation, especially when faced with unforeseen system behaviors [7].

Chen et al. (2020) presented a comprehensive model of an aircraft's hydraulic system. Their study demonstrated the efficacy of model-driven approaches in predicting system behavior under various conditions but also underscored the challenges in achieving high model fidelity [8].

2.2.2 Data-Driven Approaches

With the proliferation of sensors and advancements in computational techniques, data-driven methodologies have gained significant traction.

David and Nita (2020) showcased the potential of deep learning algorithms in aircraft health monitoring. Their study emphasized the superior performance of data-driven models, especially in identifying nuanced faults that traditional models might overlook. They highlighted the adaptability of these models, especially when trained with diverse and extensive datasets [9].

Dangut et al. (2023) took a critical look at data-driven health monitoring systems in aviation. Their research underscored the importance of data quality and robust preprocessing techniques. They pointed out that while data-driven models are powerful, their efficacy is heavily contingent on the quality of the input data. Faulty sensors or inconsistent data can significantly compromise the accuracy of these models [10].

While both methodologies have their strengths, the performance of then in real-time on board systems discourse seems to be leaning towards the potential of data-driven techniques. The adaptability, scalability, and pattern recognition capabilities of these models make them particularly suited for modern aircraft health monitoring systems. However, as *Zhao et al. (2022)* pointed out, the success of these models hinges on the quality of data, emphasizing the need for robust data acquisition and preprocessing systems [11].

2.3 AI and ML in Aircraft Health Monitoring: A Focus on Hydraulic Systems

The integration of Artificial Intelligence (AI) and Machine Learning (ML) in aircraft health monitoring with the advancement of data-driven approach, especially concerning hydraulic systems, has been a transformative force in recent years.

Jacazio et al. (2018) explored the application of ML algorithms specifically tailored for hydraulic system diagnostics. Their research highlighted the potential of data-driven models in detecting subtle anomalies within the hydraulic flow and pressure data, which traditional methods might overlook [12].

Kenan and Zhao (2023) further emphasized the advantages of using deep learning techniques, such as Convolutional Neural Networks (CNNs), for analyzing time-series data from hydraulic sensors. Their methodology demonstrated superior accuracy in predicting hydraulic system failures, especially in scenarios with complex, non-linear patterns [13].

Swischuk and Allaire (2019) discussed the challenges posed by sensor drift, calibration errors, and outright failures in hydraulic systems. Their study revealed that even minor discrepancies in sensor readings could lead to significant misdiagnoses, potentially compromising aircraft safety [14].

2.4 The Advent of Explainable AI (XAI) in Aircraft Health Monitoring

The black-box nature of many AI models, especially deep learning architectures, has been a point of contention, particularly in critical applications like aviation.

Shukla et al. (2020) championed the need for transparency in AI-driven aircraft health monitoring systems. Their research underscored the importance of understanding the decision-making process of AI models, especially when human lives are at stake [15].

2.5 Limitations in Previous Works

The vast landscape of aviation research has seen numerous studies focusing on the health monitoring of aircraft systems, particularly the landing gear. However, a closer examination of the existing literature reveals certain limitations:

1. **Scope of Research**: While many studies have delved into the intricacies of the landing gear system, their scope has often been restricted. For instance, provided an in-depth analysis of the mechanical and hydraulic

components but did not venture into the realm of sensor malfunctions and their implications.

2. **Sensor Malfunctions**: The challenge posed by sensor malfunctions, a critical aspect highlighted in the introduction, has been acknowledged but not comprehensively addressed. *Swischuk and Allaire (2019)* discussed the implications of faulty sensors but stopped short of proposing a robust solution to handle such anomalies [14].

3. **Hybrid Methodologies**: The debate between model-driven and data-driven approaches has dominated the discourse. However, the potential of hybrid methodologies, which combine the strengths of both approaches, remains largely unexplored.

4. **Real-world Application**: Much of the existing research has been theoretical, with limited real-world application or testing. The gap between laboratory findings and real-world scenarios is evident.

2.6 Gap in Current Knowledge

The current body of knowledge, while extensive, reveals a discernible gap:

1. **Handling Faulty Sensor Data**: Despite the advancements in AI and ML for aircraft health monitoring, there's a pressing need for methodologies that can effectively handle and rectify faulty sensor data in real-time.

2. **Two-Tier ML Model**: The concept of a two-tier ML model, as highlighted in the introduction, remains a novel idea. Current research has not ventured into the development of such a model that first rectifies anomalies in sensor data before making health assessments.

3. **Explainable AI**: The integration of explainable AI in aircraft health monitoring is still in its nascent stages. While AI models can make predictions, their decision-making processes remain opaque, making it challenging for technicians and engineers to trust and interpret the results.

2.7 Justification for the Current Study

The landing gear, emblematic of engineering prowess, has been at the center of events that highlight the vulnerabilities in our current systems. Instances like the A343 Helsinki Finland incident and the Saudi Airlines A330-200 event are not just historical markers but stark reminders of the pressing need for advanced solutions.

In the age of data-driven decision-making, the traditional techniques of machine learning are rapidly evolving. The old methods, which often struggled with handling highly correlated redundant sensor data, are giving way to more sophisticated and intelligent data management systems. The introduction of explainable AI is a game-changer. It not only enhances the accuracy of predictions but also demystifies the decision-making process, fostering trust and facilitating easier interpretation by technicians and engineers.

The need for such advancements is palpable. As aircraft systems grow more complex, the demand for transparent, reliable, and efficient health monitoring systems becomes paramount. This research is not just about refining methodologies; it's about revolutionizing the way we perceive and interact with aviation systems.

The aspiration to provide a proof of concept, to showcase how the integration of explainable AI and intelligent sensor data management can transform the aviation industry, is the driving force behind this research. We envision a future where every flight is not just a journey but a testament to innovation, safety, and unwavering trust. In this pursuit, we aim to redefine the paradigms of aviation safety and set new benchmarks for the industry.

3 Methodology

In the realm of predictive maintenance, the precision and clarity with which one can predict a system's health can dramatically influence operational efficiency, safety, and costs. As technological advancements continue to surge, the domain has seen a pivotal shift towards leveraging sophisticated machine learning models to harness data-driven insights. However, challenges such as data imperfections often impede the application of these models in real-world scenarios. Addressing these concerns requires a systematic, well-thought-out approach, which our research offers.

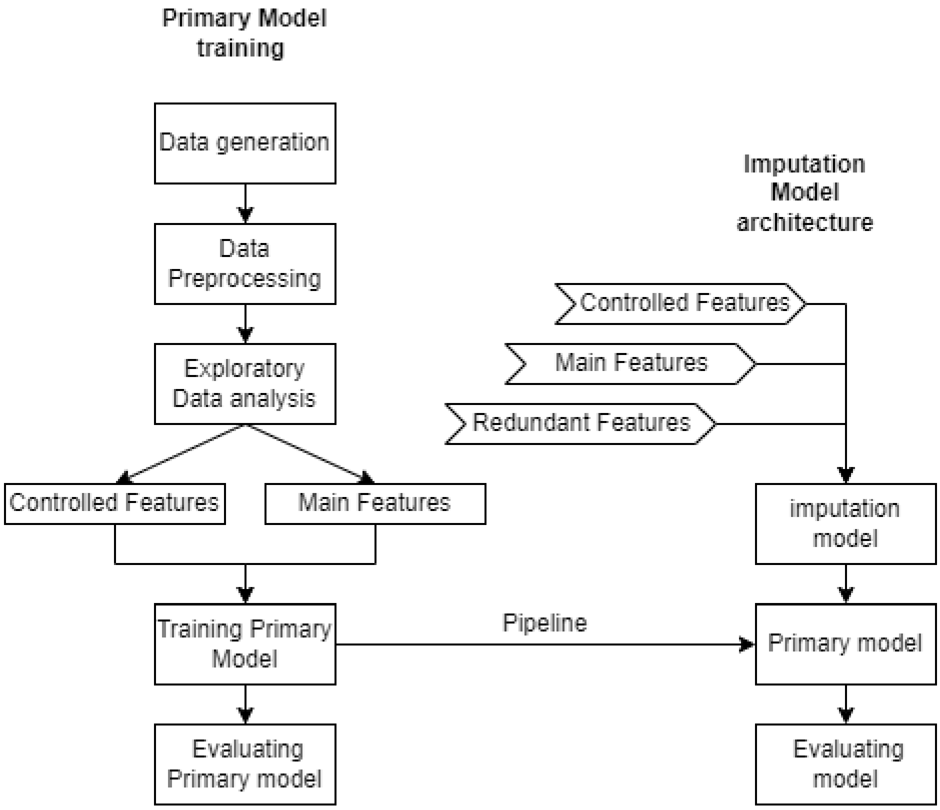

Figure 3-1 High level system architecture

To facilitate a seamless understanding of our proposed approach, we begin our methodology with a high-level schematic representation in the Figure 3-1. This diagrammatic overview elucidates the interplay between our two-tier model system: the primary classification model, dedicated to fault classification, and the secondary imputation model, designed to handle data imperfections through intelligent preprocessing techniques. With this visual aid, readers can garner a holistic understanding of the data flow, model interactions, and the sequence in which the system operates, before diving deep into each methodological component.

With this foundation, let's proceed to elucidate the intricacies of our methodology, commencing with the data acquisition phase.

3.1 Data generation

The cornerstone of our research methodology is the meticulous collection of data that portrays a broad range of operational states associated with landing gear extension and retraction systems.

The absence of real-time faulty landing gear operations data poses considerable challenges, that underscore the pivotal role of simulation tools. Such tools not only present an avenue to simulate the dynamics of landing gear operations with precision but also facilitate the accumulation of crucial data for the training of ML models. In our approach, we utilize the "Landing Gear Model in Simscape" provided by Simulink MATLAB. This tool, a product of Steve Miller's team's expertise, stands out for its accuracy and comprehensive representation of the system dynamics.

For the robust training of our ML model, it's essential to simulate a wide array of scenarios, particularly those indicative of failure states. Our model was adapted to encompass 370 distinct failure scenarios, systematically categorized into 12 defined failure types. Given that real-world environments are often affected by noise, our simulations deliberately introduce noise to sensor readings, enhancing the realism of our dataset.

Our design encompasses both singular mode and multimode failure conditions, ensuring the dataset captures a broad spectrum of system behaviors, vital for the machine learning model's efficacy.

I. Single-mode scenarios encompass:

a) Standard operational mode (No fault condition).

b) Pump malfunction (Pump failure condition).

c) Elevated temperature readings (Very high temperature condition).

d) Degraded pump performance (Faulty pump condition).

e) Hydraulic fluid compromises (Oil leakage condition).

II. Multi-mode failure scenarios amalgamate these singular conditions as follows:

f) Degraded pump performance concurrent with elevated temperatures (Faulty pump and very high temperature).

g) Pump malfunction in tandem with very high-temperature readings (Pump failure and very high temperature).

h) Compromised hydraulic fluid accompanied by high temperature conditions (Oil leakage and very high temperature).

i) Hydraulic fluid breaches coupled with pump malfunctions (Oil leakage and pump failure).

j) Degraded pump operations simultaneous with hydraulic fluid compromises (Faulty pump and oil leakage).

k) Triple anomaly of hydraulic fluid breaches, pump malfunction, and elevated temperatures (Oil leakage, pump failure, and very high temperature).

l) Degraded pump operations alongside hydraulic fluid breaches and high temperature readings (Faulty pump, oil leakage, and very high temperature).

In the upcoming subsections, we will elaborate on the simulation's overview, delve into fault and noise injections, and finally highlight the distinct features of our collected data.

3.1.1 Overview of simulation

The employed simulation model accurately reflects the **ATA 32 standard in aircraft systems**, which covers all aspects of landing gear, including hydraulics, structure, brakes, and steering. For this study, however, we focus solely on the hydraulic system. A top-level layout of the simulation component is provided in Figure 3-2 below.

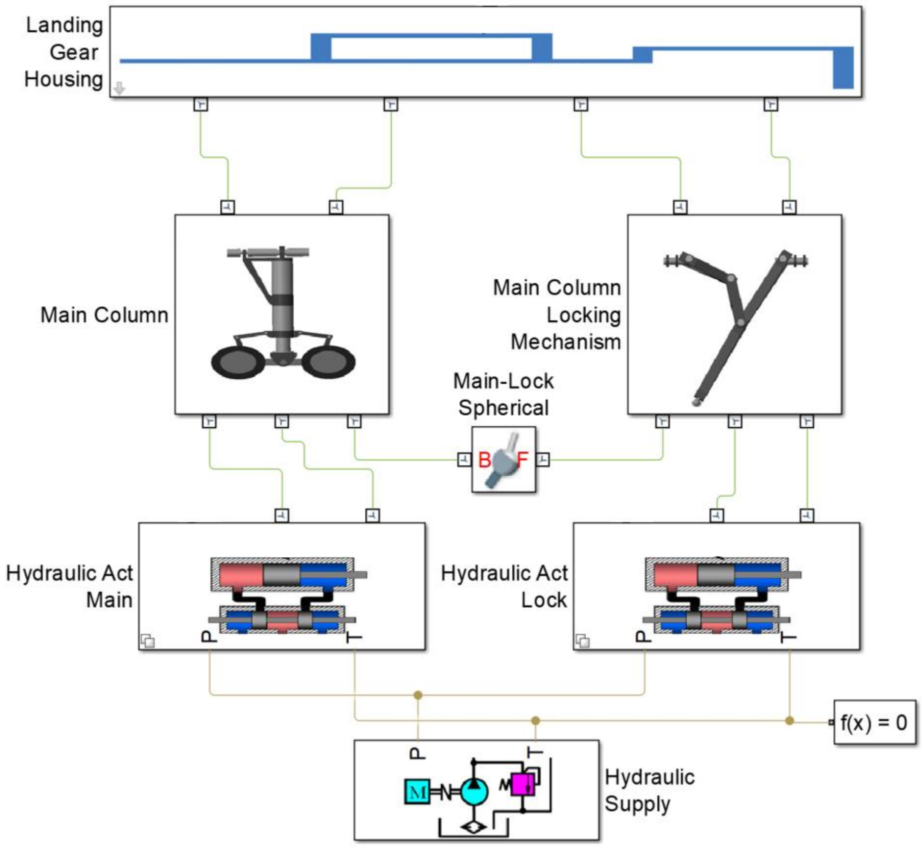

Figure 3-2 Block diagram of the simulation

It comprises a single hydraulic pump powered by an electric motor. The hydraulic reservoir supplies fluid to the main actuator, responsible for extending and retracting the landing gear based on pilot commands. Upon deployment, a secondary or locking actuator activates, securing the main landing gear in place, which is crucial for safety. The simulation also includes several sensors that measure factors like pressure, angular movement, valve status, and extension levels. A time series snapshot of the landing gear extension and retraction cycle simulation is given below figure 3-3 for better understanding.

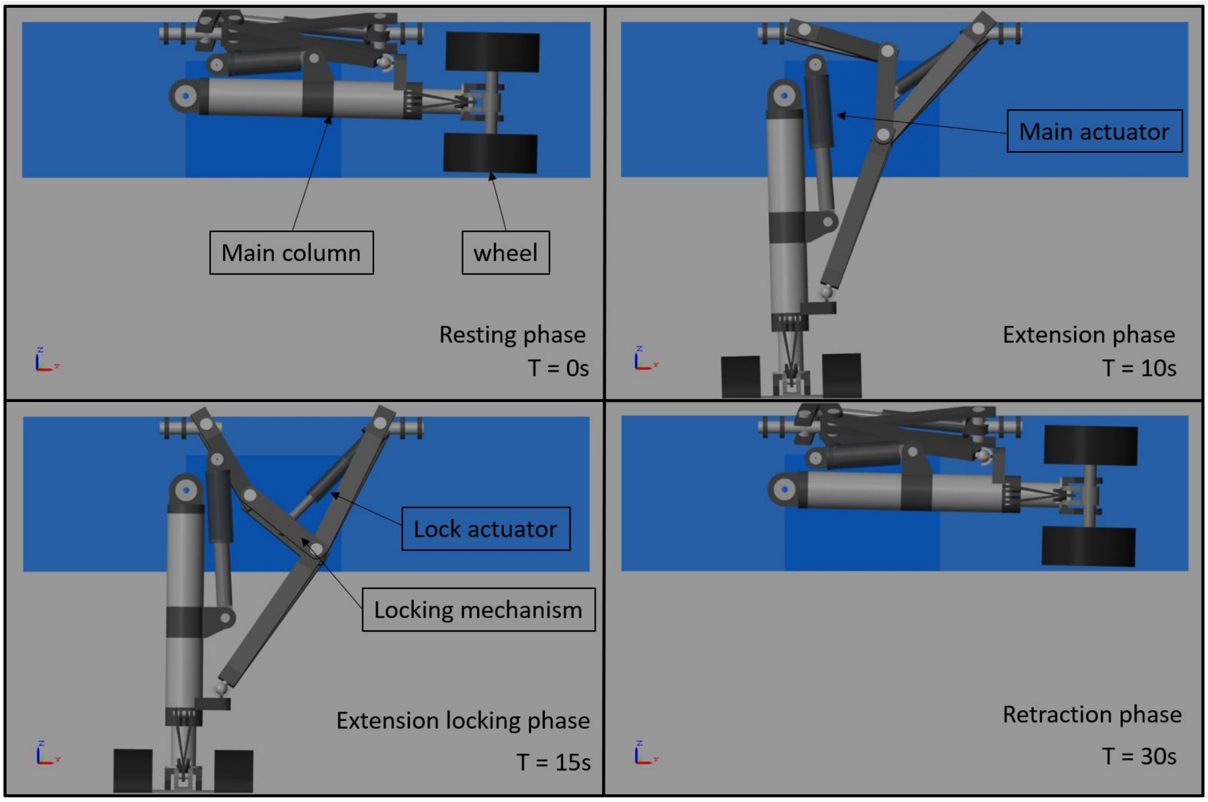

Figure 3-3 Timeseries snapshot of a simulation showing the different phase of Landing gear operation

This simulation incorporated a feature that allows us to adjust the temperature of the hydraulic fluid. This is to study how temperature changes might affect the system's performance. Additionally, the TapAir feature lets us modify the hydraulic fluid-to-air ratio, simulating various real-world conditions, which will be discussed in detail in the Fault Injection sub section later. To simulate different failure cases, we'll be adjusting three primary parameters: the **pump speed**, the **fluid temperature**, and the **TapAir ratio**.

3.1.2 Fault Injection into the Simulation

Our research delves deeply into simulating conditions within a hydraulic system, both faulty and non-faulty. We focus on three core operational parameters: TapAir ratio, pump speed, and system temperature. We explore a range of scenarios, from typical normal operations to compound faults which is depicted in the figure 3-4.

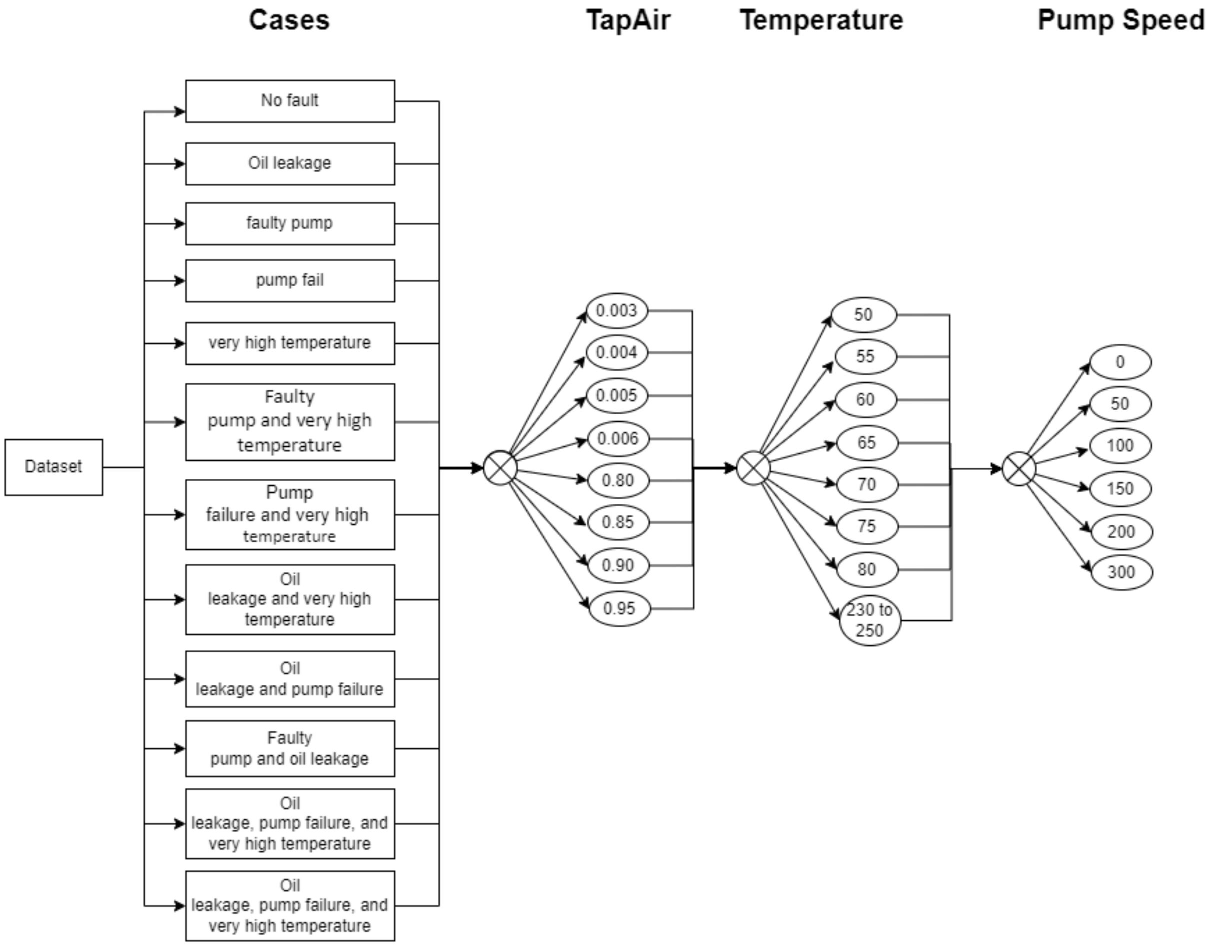

Figure 3-4 Fault injection strategy by different combination of 3 controlled parameters

To provide clarity, we have defined 12 scenarios below, of which 11 are faulty:

a) **No Fault**: This baseline scenario depicts the hydraulic system's routine functioning. It operates at temperatures oscillating between 50 to 80 degrees Celsius, which aligns with the conventional operating range of hydraulic systems. The TapAir ratio, indicative of the air-to-oil proportion in the reservoir, varies from 0.003 to 0.006, synonymous with a system in good health. The pump maintains a standard speed of 300 RPM, denoting efficient performance.
b) **Oil Leakage**: An increased TapAir ratio (0.80 to 0.95) indicates significant oil displacement, suggesting leakage.
c) **Faulty Pump**: The pump speed drops to 50-200 RPM, indicating malfunction, while other parameters remain standard.

d) **Pump Failure**: The pump speed hits 0 RPM, indicating complete cessation, with other parameters unchanged.
e) **Very High Temperature**: System temperature surges to 223-250°C, hinting at potential overheating or cooling failure.
f) **Faulty Pump and Oil Leakage**: A combined scenario of reduced pump speed (50-200 RPM) and elevated TapAir ratio (0.80 to 0.95).
g) **Faulty Pump and Very High Temperature**: Reduced pump function paired with significantly high temperatures.
h) **Oil Leakage and Very High Temperature**: Combines significant oil leakage with elevated temperatures.
i) **Oil Leakage and Pump Failure**: Portrays complete pump halt coupled with oil displacement.
j) **Pump Failure and Very High Temperature**: Highlights a halted pump and skyrocketing temperatures.
k) **Oil Leakage, Pump Failure, and Very High Temperature**: Combines pronounced oil leakage, complete pump halt, and extreme temperatures.
l) **Faulty Pump, Oil Leakage, and Very High Temperature**: Captures a malfunctioning pump, significant oil leakage, and heightened temperatures.

3.1.3 Sensor Noise Introduction

Within our utilized simulation, sensors were characterized as ideal, devoid of any noise. Contrarily, in practical applications, sensors invariably exhibit noise in their outputs. To enhance the fidelity of our simulation to real-world scenarios, we incorporated components that superimpose noise onto the sensor signals. Specifically, we introduced a 15% white noise to each sensor's output. This adjustment ensures our simulation more accurately reflects the inherent interference often encountered in actual sensor systems. Figure 3-5 plot of an ideal signal, free from noise, against one subjected to our introduced noise, illustrating the tangible modifications can be seen.

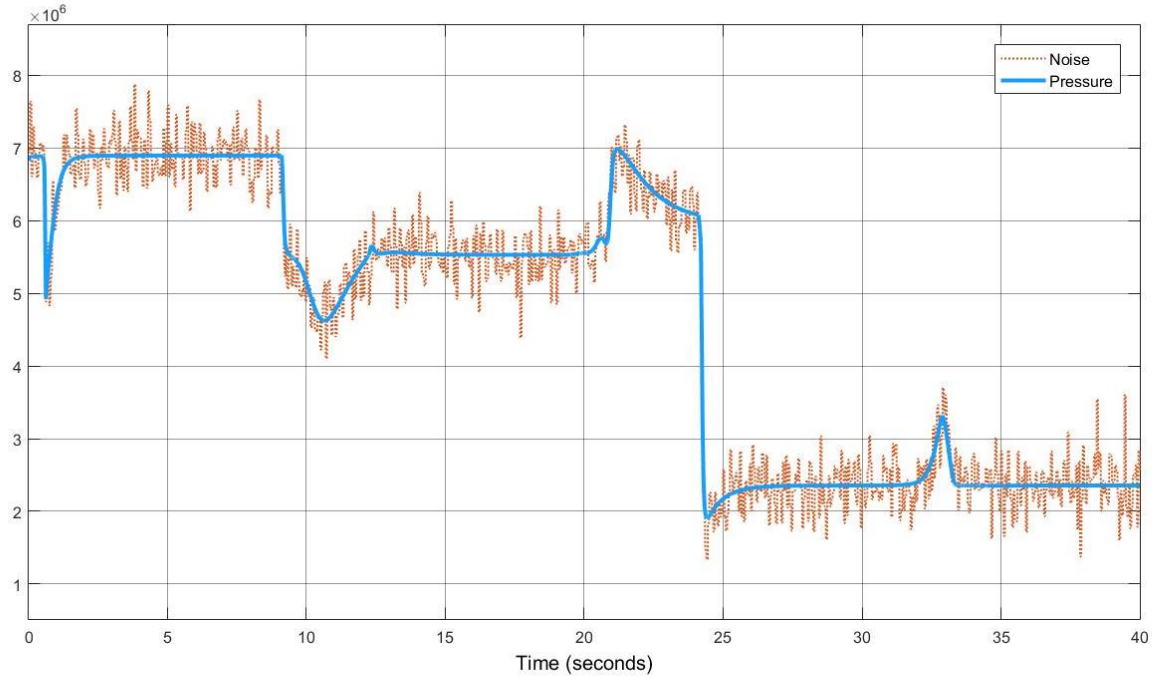

Figure 3-5 Comparison of signal without and with noise injection in pump pressure.

3.1.4 Features of the collected Data

The comprehensive dataset extracted from a series of simulations consists of 10 features, one of which is 'health,' the target variable for our classification model. These features serve as inputs for the machine learning model. Detailed information about each feature, including the corresponding sensor and its location within the simulation, is provided in the table 3-1 below:

Table 3-1 List of all features in the collected data and their details

	Feature	Sensor	Location
1	Time	Independent feature	
2	Main Actuator Pressure 1	Pressure sensor	Main Actuator
3	Main Actuator Pressure 2	Pressure sensor	Main Actuator valve

4	Pump Pressure	Pressure sensor	Hydraulic tank
5	Main Column Angle 1	Rotary encoder	junction of landing gear housing and main actuator
6	Main Actuator Position 2	Linear Variable Differential Transformers (LVDT)	Main Actuator
7	Pump Torque	Optical Torque Sensors	Mounted on top of pump motor
8	Temperature	Input parameter	
9	Pump Speed	Input parameter	
10	Health	Condition label	

Further association of these features and their importance in assessing the health of the system will be explored in the next section: Exploratory data analysis and Data pre-processing.

3.2 Exploratory data analysis and Data pre-processing

Exploratory Data Analysis (EDA) is a fundamental step in the data science pipeline. It involves the visual and quantitative inspection of data to understand its structure, relationships, anomalies, and patterns.

3.2.1 The Importance of Exploratory Data Analysis (EDA) in Handling Redundant Sensor Data

Exploratory Data Analysis (EDA) is more than just an initial step in the data analysis pipeline; it's the foundation upon which the entire research is built. When dealing with complex systems like aircraft landing gear, the sheer volume and intricacies of sensor data can be overwhelming. EDA provides a lens to view, understand, and decipher these intricacies.

I. Unveiling Redundancies:

a. In aircraft systems, redundancy is deliberately built for safety. Multiple sensors often measure similar or related physical quantities. Through EDA, we can identify these redundancies by observing high correlations between features.

b. By visualizing and quantifying relationships, EDA allows us to pinpoint which sensors are producing redundant information. This knowledge isn't just academic; it's vital for maintenance, cost savings, and ensuring efficient data processing.

II. **Informing Data Strategy:**

a. Traditional Machine Learning wisdom often advises dropping highly correlated features to prevent multicollinearity, which can destabilize some models. However, in the real world, especially in safety-critical systems like aircraft, every piece of data is invaluable.

b. By using EDA to understand the nature and extent of these correlations, we can devise innovative strategies that harness, rather than discard, this redundancy.

III. **Leveraging Redundancy for Robustness:**

a. The novel approach in this study exemplifies the above point. Instead of discarding correlated sensor readings, we are training primary models with main features and then using the correlated, redundant features to train secondary imputation models.

b. This method ensures two critical outcomes:

Improved Model Robustness: If a primary sensor fails, the secondary, correlated sensor can provide vital backup, ensuring continuous monitoring.

Efficient Use of Data: Instead of discarding valuable sensor data, this approach incorporates it, enhancing the model's richness and reliability.

3.2.2 Data profile and distribution

I. Data Size:

Number of Samples: 671,907

Number of Features: 10

II. Data Distribution

The histograms below in the figure 3-6 provide a visual representation of the data distributions for each numerical feature in the dataset:

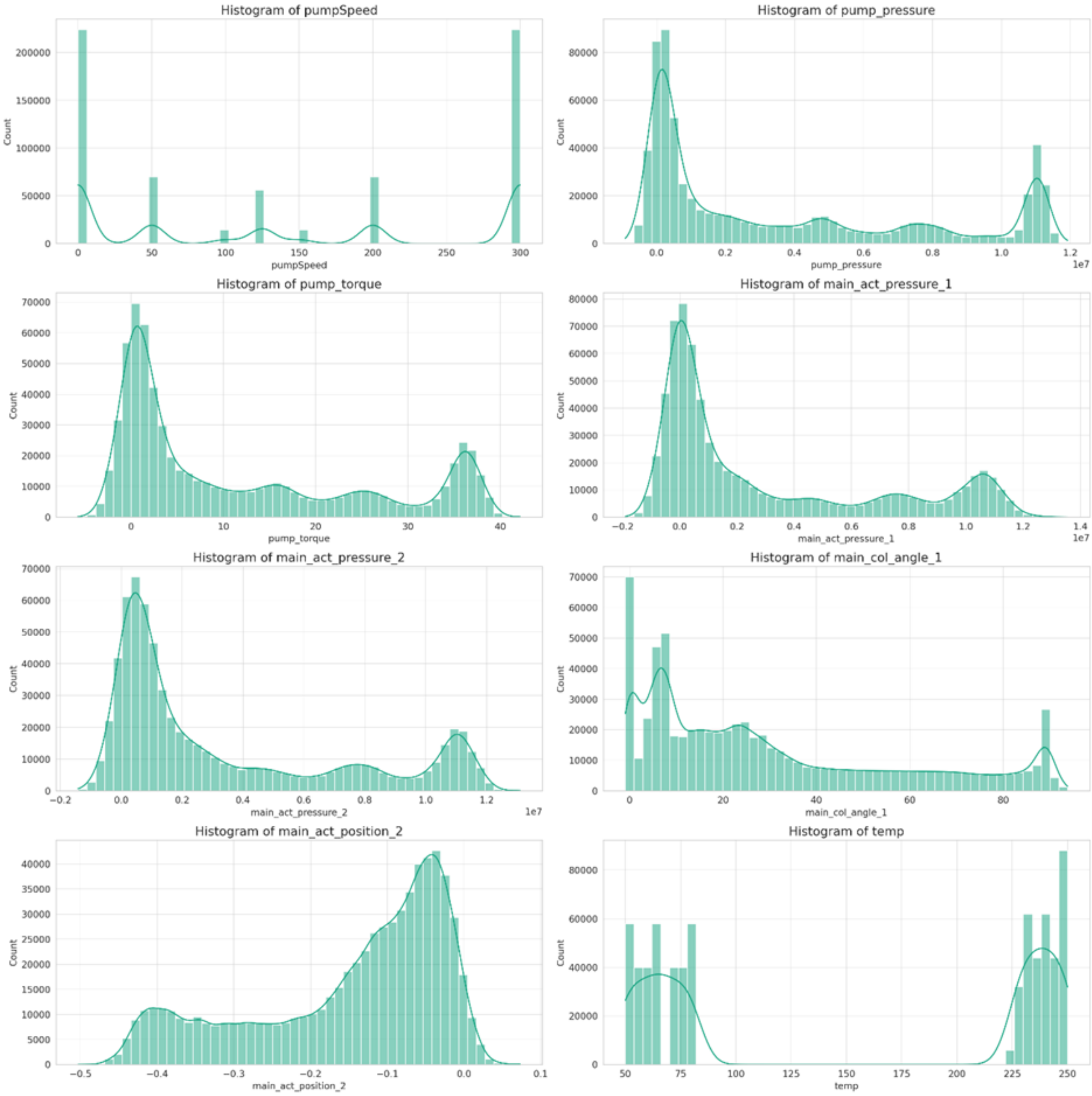

Figure 3-6 Feature distribution in the dataset

The histograms in the figure 3-6 above provide insights into the distribution of each numerical feature:

a) **pumpSpeed**: Appears to be multi-modal, indicating that there are specific speed values at which the pump frequently operates.

b) **pump_pressure**: Has a similar distribution as the main actuator pressures, being right-skewed.

c) **pump_torque**: Right-skewed distribution with a high frequency of lower values.

d) **main_act_pressure_1 & main_act_pressure_2**: Both show a right-skewed distribution with a high frequency of values near the lower end and a long tail towards the higher values.

e) **main_col_angle_1**: Displays a bimodal distribution, with two prominent peaks, one near 0 and the other near 90.

f) **main_act_position_2**: Centers around -0.1 to 0.1, with a roughly symmetrical distribution.

g) **temp**: Shows a multi-modal distribution, with multiple peaks observed.

III. Health distribution

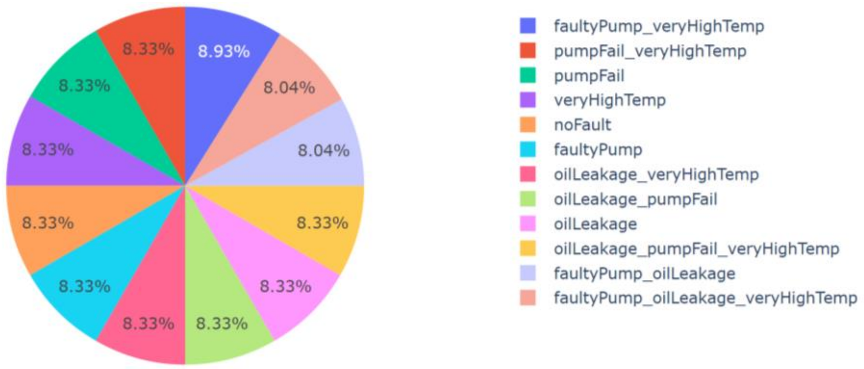

Figure 3-7 The pie chart illustrates the distribution of different health conditions

It's evident that there's a relatively even distribution among the various categories as seen in the pie chart from the figure 3-7, with each category having a representation of around 54,000 to 60,000 samples.

3.2.3 Correlation Analysis

In our analysis, the correlation values are derived using the Pearson Correlation Coefficient, a measure of linear association between two variables; denoted as *r*. The formula for the Pearson Correlation Coefficient is:

$$r = \frac{\sum(x_i - \overline{x}) \,.\, (y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2 .\ \sum(y_i - \overline{y})^2}}$$

(Equation 3-1)

Where r = Correlation coefficient

x_i = values of x variable in a sample

$\overline{x}$ = mean of x variable in a sample

y_i = values of y variable in a sample

$\overline{y}$ = mean of y variable in a sample

The coefficient can vary between -1 and 1, providing insights into the nature of the relationship, coefficient 1 Indicates a perfect positive linear relationship, whereas -1 indicates negative linear relationship and 0 suggests no linear association between the two variables

Refer to Figure 3-9 below, which displays a heatmap visually representing the correlation coefficients among the numerical features of our dataset.

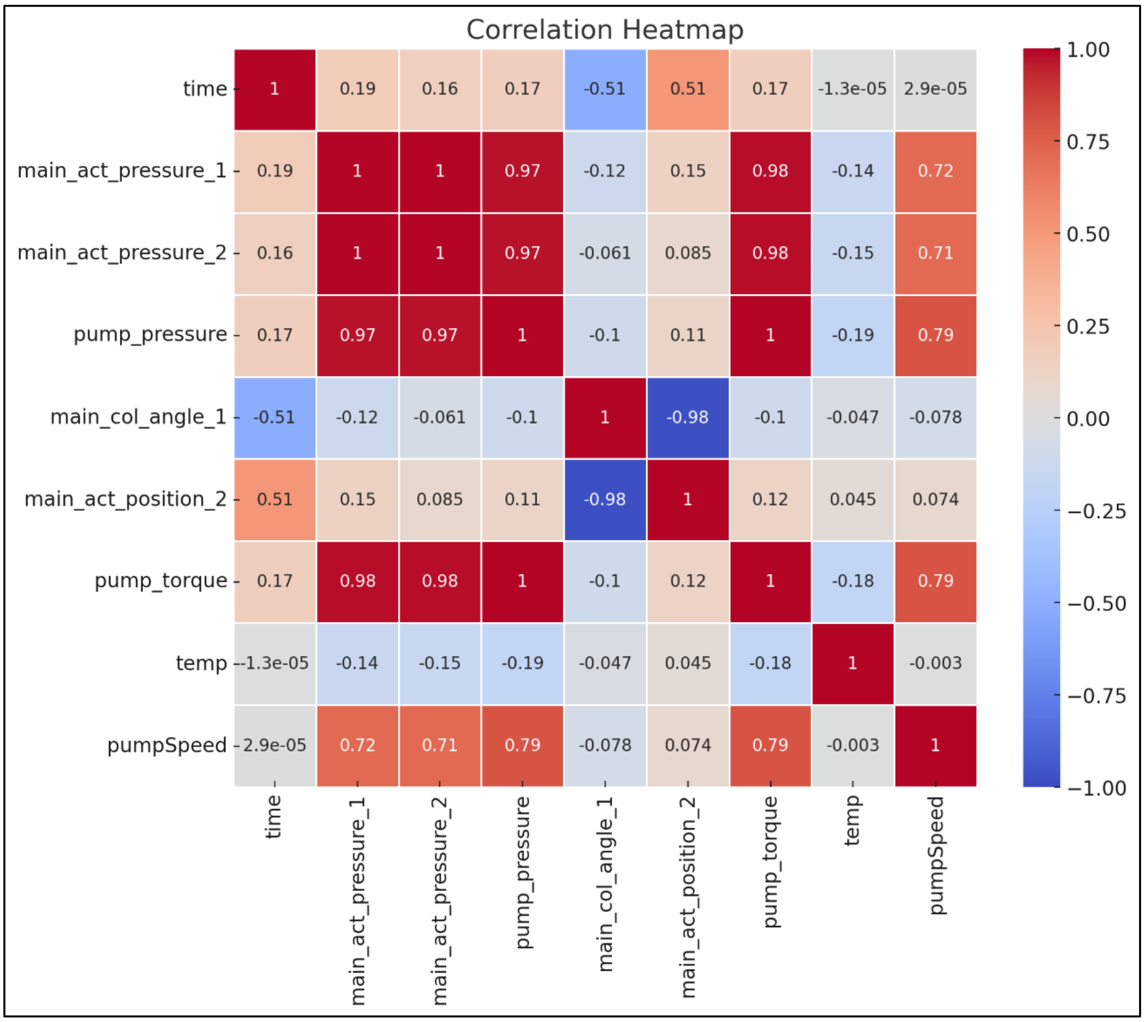

Figure 3-8 Correlation Heatmap

From the heatmap in Figure 3-9, there are several key takeaways:

The correlation heatmap elucidates the intricate relationships among the dataset's features:

1. **High Correlations:** Feature pairs like **pump_pressure** & **pump_torque**, **main_act_pressure_1** & **main_act_pressure_2**, and **main_col_angle_1** & **main_act_position_2** exhibit strong correlations.
2. **Distinct Features:** Some features, notably **pumpSpeed** and **temperature**, show limited correlation with others, underscoring their unique significance.

Spotting and Understanding Redundancies:

- **Pump Pressure and Pump Torque:** With a correlation of 0.998381, these two exhibit an almost perfect positive relationship. A time series plot of pump pressure and torque of one of the cases is plotted in the figure 3-10, where they exhibit the identical pattern. This suggests that pump pressure can precisely predict pump torque, potentially rendering one redundant in predictive modeling.

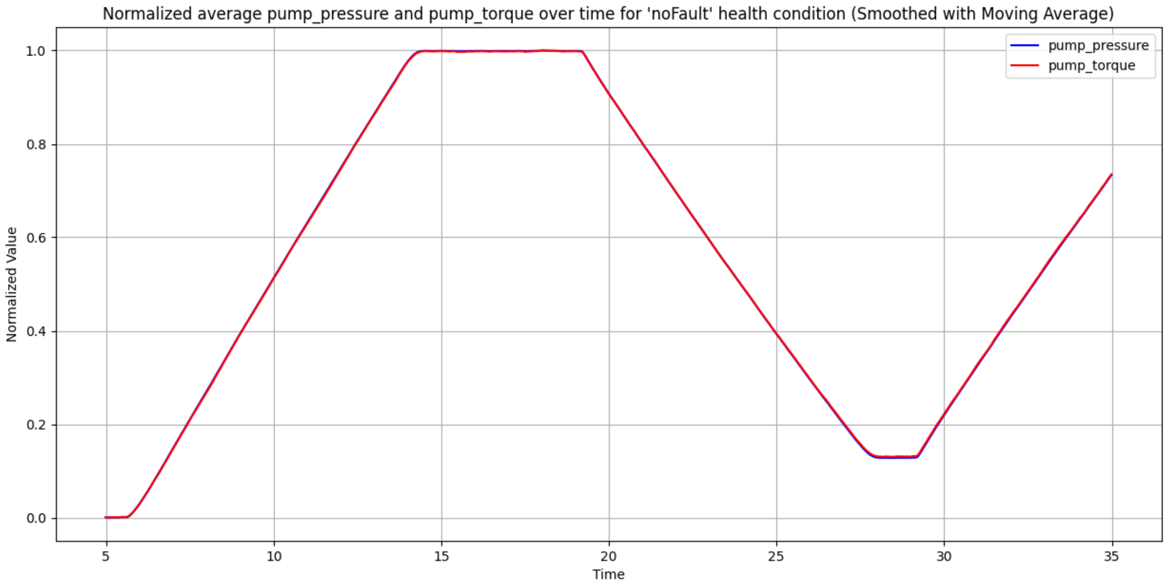

Figure 3-9 Moving averages of normalized pump pressure and pump torque.

- **Main Actuator Pressure 1 and Main Actuator Pressure 2:** A robust positive correlation of 0.996350 signifies that one can reliably predict the other, hinting at potential redundancy. A time series plot of Main Actuator Pressure 1 and Main Actuator Pressure 2 of one of the cases is plotted in the figure 3-11, where they exhibit nearly the identical pattern.

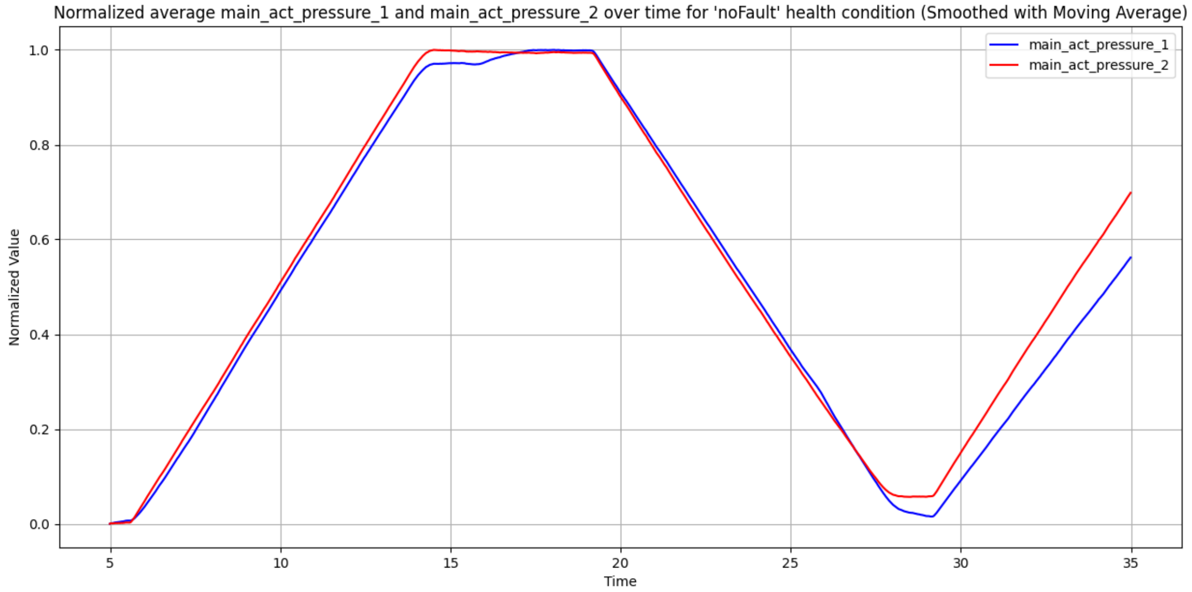

Figure 3-10 Moving averages of normalized main actuator pressures 1 and 2.

- **Main Column Angle 1 and Main Actuator Position 2:** A significant negative correlation of -0.983905 suggests an impeccable inverse relationship. As one variable's value rises, the other's falls, indicating mutual predictability. A time series plot of Main Column Angle 1 and Main Actuator Position 2 of one of the cases is plotted in the figure 3-12, where they exhibit exact mirror pattern.

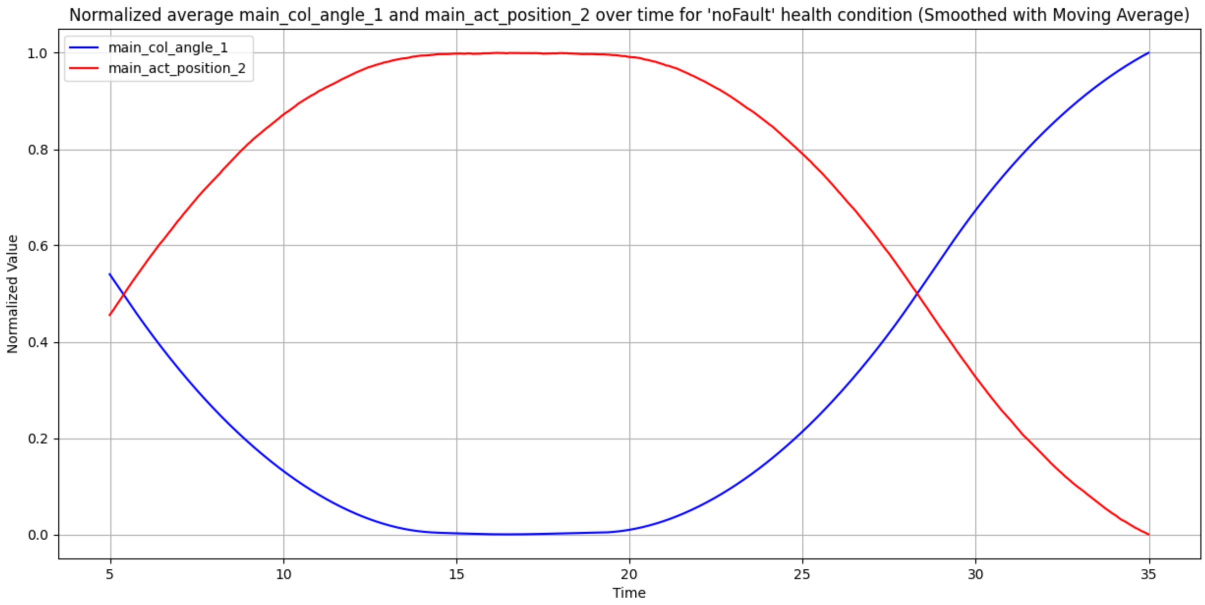

Figure 3-11 Moving averages of normalized main column angle 1 and Main Actuator Position 2.

- Features like **temp** and **pumpSpeed** have relatively low correlations with other features, suggesting they are more unique and independent without any redundancy. We call these features as crucial or controlled features. Note that these two are the input variables that we used in simulation to simulate wide range of failure scenarios.

3.2.4 Data Pre-processing

The raw data retrieved from the simulations does not adhere to consistent time intervals. To rectify this inconsistency, we employed a downsampling technique, standardizing the data to fixed time steps at a frequency of 50 Hertz. This strategy is a familiar approach in time-series data processing and closely mimics data harvested from real-world sensors operating at a 50 Hz frequency. Consequently, this operation furnished a comprehensive dataset consisting of 671,908 entries across 23 columns.

3.2.5 Rationale Behind Feature Selection and Tagging

Discarding the Temporal Element - Our primary focus is to train a model for classification that isn't influenced by temporal dynamics. As such, the 'time' feature from the dataset is omitted. This decision ensures our model evaluates the health of the system based on its current state, rather than its historical trajectory. By abstracting away from time, our model becomes robust to various temporal disturbances and focuses on the inherent properties of the system.

Target Variable - Health of the System: The 'health' feature serves as our target variable. This choice aligns with our overarching objective: understanding and predicting the system's health status.

Categorizing the Features:

1. **Controlled Features: Pump Speed and Temperature:** These are essential input parameters that play a pivotal role in determining the system's health. Altering these parameters in simulations, especially in

conjunction with the 'TapAir' parameter, creates various failure scenarios. Their significance is accentuated by the fact that they don't have corresponding redundant features, making them indispensable in monitoring the landing gear's health.

2. **Main Features: Pump Pressure, Main Actuator Pressure 1, and Main Column Angle 1:** These features offer direct, tangible insights into the system's current state.

3. **Redundant Features: Pump Torque, Main Actuator Pressure 2, and Main Actuator Position 2:** These features, due to their high correlations with main features, can mirror vital system information. Their significance lies in their backup capabilities. In scenarios where a primary sensor might fail or malfunction, these redundant sensors ensure that monitoring remains uninterrupted, thereby enhancing system reliability.

3.2.6 A Novel Approach to Redundancy

Traditional data science methodologies often discard highly correlated features to simplify models and avoid multicollinearity. However, our approach acknowledges the real-world importance of redundancy, especially in safety-critical systems like aircraft landing gears.

Here's our innovative two-pronged strategy:

A. **Harnessing Main and Controlled Features:** The primary models are trained using these features, capturing the core characteristics of the system.

B. **Leveraging Redundant Features:** Secondary imputation models are trained using these features. Their primary role is to act as intelligent backups. Should a main sensor fail, the model can seamlessly draw information from these correlated, redundant sensors, ensuring uninterrupted monitoring.

In conclusion, our methodology in feature selection and tagging isn't merely a series of data-driven choices. It's a reflection of a deeper understanding of the system's workings and an acknowledgment of the realities of sensor-based monitoring. By intelligently utilizing redundancies and categorizing features, we're laying the foundation for a model that's both robust and resilient.

To ensure a rigorous modeling process, our dataset is methodically divided: 70% for training, laying the foundation for model building; 20% for testing, offering an initial assessment of the model's capabilities; and the final 10% reserved for evaluation. This validation dataset remains entirely unseen by the model during its training phase. Consequently, it serves as an excellent tool for gauging true model performance, providing insights into its real-world applicability and reliability.

Having established the foundational structure by data preprocessing, we now transition to the heart of our research endeavor: the modeling process. The choice of model and its architecture plays a paramount role in harnessing the insights embedded within the data. Let's delve into the specifics of the model selected for training and its underlying rationale.

3.3 Primary model – Classification model

The objective of the primary model is to predict the health status of the aircraft's landing gear system based on given features. To achieve this, we've considered a range of classifiers, each with unique mathematical foundations, to compare and study the optimal predictive accuracy and robustness offered by them.

3.3.1 Logistic Regression

Principle: Logistic Regression is a statistical methodology employed to elucidate the relationship between a dichotomous dependent variable and one or more independent variables. While traditional logistic regression is designed for binary classification, its extension to multiclass classification, often termed as Softmax Regression or Multinomial Logistic Regression, allows for modeling the relationship between multiple categories and predictor variables. It provides a mechanism to estimate the probability of an instance belonging to each of the categories.

Mathematical Framework: The equation for Softmax Regression is:

$$\boldsymbol{P(y = i \mid X) = \frac{e^{(W_i x + b_i)}}{\sum_{j=1}^{K} e^{(W_j x + b_j)}}}$$

(Equation 3-2)

Where

- *P(y=i / X) is the probability that instance X belongs to class i*
- W_i and b_i are the weight and bias terms for class *i*.
- *K* is the total number of classes.

Rationale for Deployment: Given the inherent multidimensionality of the dataset under scrutiny—comprising a multiplicity of features dictating the health status of the aircraft's landing gear—Logistic Regression is proffered as a preliminary measure. It is imperative to initiate this analysis to discern the potential linear separability embedded within the dataset. Should the features delineate the target variable via a linear decision boundary, it imparts critical insights into the intrinsic architecture of the data.

3.3.2 Polynomial Regression

Principle: Despite its nomenclature suggesting regression, when juxtaposed with logistic regression, Polynomial Regression, traditionally used for regression tasks, can be adapted for multiclass classification when combined with Softmax Regression. By introducing polynomial and interaction terms to the base feature set, it can capture non-linear relationships, making it adept at handling complex classification scenarios.

Mathematical Framework: Polynomial regression augments the linear model paradigm by introducing polynomial features. To elucidate, for an individual feature x, instead of the mere presence of x, one might encounter terms such as x, x^2, x^3, …. For datasets characterized by multiple features, interaction terms further enrich the feature set, e.g., $x_1 . x_2$, $x_1^2 .x_2$ and so forth.

Rationale for Deployment: Contemplating the non-linear characteristics inherent to the dataset, it becomes evident that a rudimentary linear decision boundary may prove inadequate. Polynomial regression, by virtue of its feature transformation capability, is adept at capturing these non-linear intricacies, making it an indispensable tool for datasets typified by complex, multidimensional interactions such as the one under study.

3.3.3 Decision Tree

Principle: A Decision Tree operates on a hierarchical decision-making approach, systematically breaking down a dataset into smaller and smaller subsets until the subsets reach a level where the decision can be made in a straightforward manner. It essentially mimics a tree structure, wherein each internal node represents a feature, each branch symbolizes a decision rule, and each leaf node corresponds to an outcome.

Mathematical Framework: Decision Trees utilize several algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes amplifies the homogeneity of resultant sub-nodes. Here in this research we are employing Gini impurity as a loss function to split the nodes

Gini Impurity: It calculates the amount of uncertainty or disorder in a set. The formula for the Gini impurity is:

$$\boldsymbol{Gini}(\boldsymbol{p}) = \boldsymbol{1} - \sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{n}} p_i^2$$

(Equation 3-3)

Where p_i is the probability of choosing an item from class i

Rationale for Deployment: The inherent strength of a Decision Tree lies in its simplicity and visual interpretability. Given the non-linear and multi-dimensional nature of the aircraft's landing gear dataset, a Decision Tree can effectively handle such complexities without necessitating any assumptions about the data's structure or distribution. Furthermore, it can manage both numerical and categorical data, making it especially pertinent for datasets like the one under examination.

3.3.4 K-Nearest Neighbors (KNN)

Principle: K-Nearest Neighbors (KNN) is an instance-based learning algorithm, which means it doesn't intrinsically generate a model. Rather, it classifies based on the majority class among its 'k' most proximate instances from the training dataset. Think of KNN as a system where, when given an unclassified observation, the algorithm searches the training dataset for the 'k' training examples that are closest to the point and returns the output value that has the most occurrences among the k-neighbors.

Mathematical Framework: The efficacy of KNN rests upon distance metrics. Given a new observation, the algorithm computes its distance to all other points in the dataset and selects the 'k' closest ones. The prevalent distance metrics include:

Euclidean Distance: For two points $P(x_1,y_1)$ and $Q(x_2,y_2)$, the Euclidean distance is:

$$d(P,Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

(Equation 3-4)

Rationale for Deployment: KNN is inherently non-parametric, meaning it makes no explicit assumptions about the functional form of the data transformation, making it suitable for the multi-dimensional nature of the dataset in question. Given the non-linear properties exhibited by the aircraft's landing gear data, KNN can be a strategic fit. Its ability to adapt easily to changes and its capacity to handle both numerical and categorical features further justify its consideration.

3.3.5 Random Forest

Principle: Random Forest is an ensemble learning method that operates by constructing a 'forest' of decision trees at training time and outputting the mode (classification) of the classes for individual trees. By aggregating the predictions of numerous decision trees, Random Forest tends to achieve higher accuracy and avoids the overfitting issue that single decision trees can suffer from.

Mathematical Framework: The key mathematical principles behind Random Forest include:

Bootstrap Aggregating (Bagging): Multiple subsets of the original dataset are created using a process called bootstrapping (sampling with replacement). A decision tree is grown on each of these subsets. This ensures diversity among the trees and helps in reducing variance, leading to more stable predictions.

Feature Randomness: In traditional decision trees, at each split, the best split amongst all features is chosen. In contrast, each tree in a Random Forest picks the best split among a random subset of features. This introduces further diversity and results in uncorrelated trees which, when averaged, can reduce variance.

Rationale for Deployment: Given the non-linear and multi-dimensional nature of the aircraft's landing gear dataset, Random Forest is particularly well-suited. Here's why:

a) **Handling High Dimensionality**: Random Forest can manage large datasets with higher dimensionality. It can handle thousands of input variables and identify the most significant ones, making it ideal for our dataset.

b) **Robust to Outliers**: The ensemble nature of Random Forest makes it tough for outliers to sway the outcome significantly.

c) **Mitigating Overfitting**: By averaging out the biases of individual trees, the variance is reduced, leading to a more generalized model.

3.3.6 XGBoost

Principle: XGBoost stands for Extreme Gradient Boosting. It's an optimized distributed gradient boosting library designed to be efficient, flexible, and portable. Like Random Forest, XGBoost is also an ensemble technique, but rather than bagging, it uses boosting to convert weak learners into strong learners. At its core, XGBoost constructs an additive model in a forward stage-wise fashion, with the objective of minimizing a given loss function.

Mathematical Framework: XGBoost relies heavily on the gradient boosting framework. Key concepts include:

1. **Gradient Boosting**: Given a differentiable loss function, the model is built in a stage-wise fashion by optimizing the loss function. In each stage, a set of decision trees are fitted to the negative gradients (called "pseudo-residuals") of the loss function.

2. **Regularization**: Unlike the traditional Gradient Boosting Machine (GBM), XGBoost incorporates L1 (Lasso Regression) and L2 (Ridge Regression) regularization terms in its objective function, making the learning process more regularized and helping to reduce overfitting.

3. **Handling Missing Data**: XGBoost has an inherent routine to handle missing values, assigning them to whichever direction increases the purity most.

Rationale for Deployment:

1. **Efficiency in Handling Large Datasets**: Given the intricate nature of the aircraft landing gear dataset, XGBoost's ability to handle large datasets efficiently makes it a prime choice.

2. **Capability with Non-linear Data**: XGBoost's gradient boosting framework inherently can capture non-linear relationships, aligning well with the non-linear nature of the dataset.

3. **Regularization**: The built-in L1 and L2 regularization in XGBoost help prevent overfitting, ensuring the model generalizes well to unseen data.

4. **Parallel Processing**: XGBoost's ability to perform parallel computations on a single machine makes it faster than other boosting algorithms.

The selection of classifiers is strategic, aiming to capture both linear and non-linear relationships within the dataset. While Logistic Regression and Polynomial Regression provide foundational insights, Decision Trees, KNN, Random Forest, and XGBoost delve deeper, exploring intricate feature interactions and patterns.

By leveraging this diverse set of classifiers, we ensure a comprehensive understanding of the data, paving the way for robust predictions. Subsequent sections will delve into a detailed performance analysis, comparing the efficacy of these classifiers in the context of aircraft landing gear health prediction.

3.4 Secondary model – Imputation model

3.4.1 The Crucial Role of the Imputation Model

In the vast realm of sensor data, inconsistencies, missing values, and faulty readings are inevitable. Addressing these challenges is our Imputation Model, a sophisticated secondary layer engineered to ensure data integrity and consistency.

3.4.2 Why the Imputation Model Matters:

Traditional data handling might discard or simply average out faulty or missing data. However, in complex systems like aircraft landing gears, every piece of data holds significance. The Imputation Model is our strategy to salvage and intelligently fill these data gaps. Its primary role is to ensure that downstream machine learning processes receive a dataset that's not just complete, but also reliable and robust.

3.4.3 Decoding the Imputation Model Architecture:

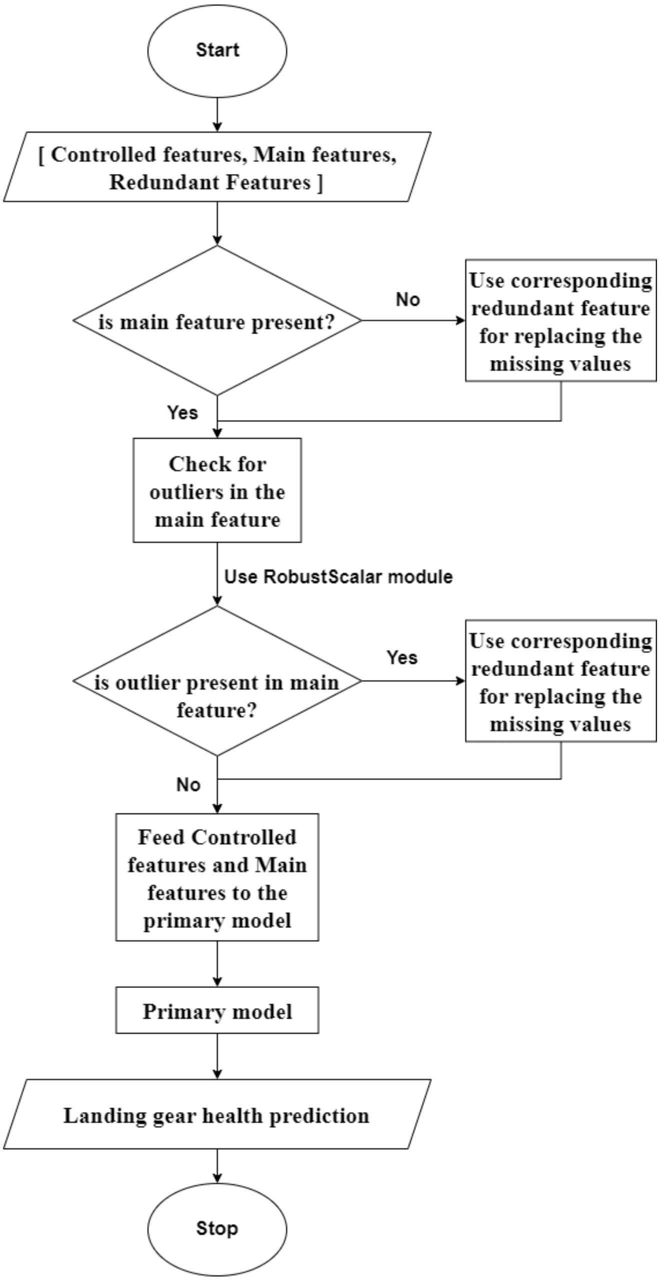

Figure 3-12 Data flow logic in Imputer model to primary model

The model depicted in the figure 3-13 leverages a pre-defined data structure, encompassing:

Controlled Features: pumpSpeed, temp

Main Features: pump_pressure, main_act_pressure_1, main_col_angle_1

Redundant Features: pump_torque, main_act_pressure_2, main_act_position_2

The model takes in 8 parameters from controlled, main and redundant feature. Simple imputer logic checks the data reliability, it checks whether main features are present in the input parameters;

if no – it replaces the normalised values of the corresponding missing values form the redundant feature;

if yes – it passes the data to the next layer to check the outliers.

In the consequent layer, the 8 parameters are again checked for outliers using Robust scalar which is trained using the test dataset to identify the extreme high or low values that cannot possibly the output of the sensor. If there is an outlier then it imputes the values as discussed in the earlier logic. Then this processed data is fed to primary model for classification.

3.4.4 From Imputation to Outlier Handling:

Post-imputation, the data undergoes scaling to identify outliers, which are often the result of uncalibrated sensors. When such anomalies are detected in main features, the model substitutes these values with the corresponding normalized data from redundant features.

3.4.5 Feeding the Primary Model:

Once preprocessed, the dataset—now distilled to controlled and processed main features—feeds into our primary machine learning model. This model has been meticulously trained on these five features, making it adept at assessing the health of the landing gear system. For a comprehensive assessment, we integrate the primary model with the imputation model, validating their combined prowess using a separate dataset.

3.5 Implementing Explainability to the Decision

3.5.1 Rationale for Incorporating Explainable AI

While traditional machine learning models have ushered in a new era of data-driven diagnostics in aviation, their intrinsic "black box" nature poses challenges in terms of transparency and interpretability. Given the critical implications of predictions related to aircraft health, especially the landing gear, there's a pressing need for models that are not only accurate but also interpretable. This led to the decision to incorporate explainable AI (XAI) into our methodology.

3.5.2 Integration of SHAP for Interpretability

To achieve explainability, We employed a decision tree classifier’s feature importance to derive SHAP values (SHapley Additive exPlanations) because there exists a well-defined framework for decision tree specifically that gives the forced plots. SHAP values provide a unified measure of feature importance, allowing us to discern which features play pivotal roles in the model's predictions.

The methodology involved:

- Initializing the SHAP explainer with our trained decision tree model.
- Computing SHAP values for instances in our dataset.
- Visualizing these SHAP values through force plots to understand the contribution of each feature to the model's decision.

3.5.3 Analysis of Specific Instances

For a deeper dive into the model's decision-making process, specific instances from the dataset were analyzed using the computed SHAP values. This allowed us to understand how the model arrived at its predictions and which features were the most influential in driving these decisions.

3.5.4 Methodological Significance

The integration of XAI into our methodology serves a dual purpose:

- **Transparency**: By demystifying the model's predictions, we can provide clear insights into the factors influencing aircraft landing gear health assessments.
- **Trustworthiness**: With interpretable results, stakeholders – be it technicians, engineers, or regulatory bodies – can place greater trust in

the model's predictions, ensuring that maintenance and interventions are evidence-based and targeted.

3.6 Performance Evaluation criteria of the model

In determining the efficacy of various classifiers, a consistent evaluation metric is crucial. For this study, the Confusion Matrix, a widely-used tool in classification tasks, has been employed.

The Confusion Matrix juxtaposes actual versus predicted classifications, offering insights into true positives, true negatives, false positives, and false negatives. From this matrix, several performance metrics can be derived:

- **Precision**: Ratio of correctly predicted positive observations to the total predicted positives.
- **Accuracy**: Ratio of correctly predicted observation to the total observations.
- **Sensitivity (or Recall)**: Ratio of correctly predicted positive observations to all actual positives.
- **F1 Score**: Weighted average of Precision and Recall.
- **Specificity**: Ratio of correctly predicted negative observations to all actual negatives.

For this research, 'Accuracy' has been spotlighted as the primary performance metric. Extracted directly from the Confusion Matrix, accuracy provides a concise measure of a classifier's overall correctness.

We evaluated the performance of six different classifiers: Logistic Regression, Polynomial Regression, Decision Tree, KNN, Random Forest, and XGBoost. The following are the results based on accuracy:

4 Results and Discussion

4.1 Overview

In this chapter, we present and discuss the results obtained from our two-tier model, which comprises a primary classification model and a secondary imputation model. This innovative approach was specifically designed to handle the intricacies of our dataset, derived from aircraft landing gear simulations. The results are evaluated based on the accuracy metrics of various classifiers applied to the dataset. Table 4-1 comprises the performance of the compound working of our proposed two-tire model, in the later subsections will dive deep into each layer's performance and their impact in the proof of concept.

Table 4-1 Performance of the model

Algorithm	**Test Accuracy**	**Validation Accuracy**
Logistic Regression	72.82%	72.09%
Polynomial Regression	83.91%	86.72%
Decision Tree	98.76%	93.93%
KNN	94.22%	92.87%
Random Forest	99.11%	92.98%
XGBoost	98.64%	87.83%

4.2 Primary Classification Model Results Evaluation

4.2.1 Logistic Regression:

- **Test Accuracy**: 72.82%
- **Validation Accuracy**: 72.09%

Logistic Regression, being a linear model, makes assumptions about the linear relationship between the input features and the log odds of the output. The obtained accuracy indicates that while the model has managed to capture some patterns in the data, it may be oversimplifying the problem. The similarity between test and validation accuracies suggests that the model is neither overfitting nor underfitting significantly but might be underperforming due to the inherent non-linearities in the data. The confusion matrices of this classifier with test and validation dataset is given the below Figure 5-1.

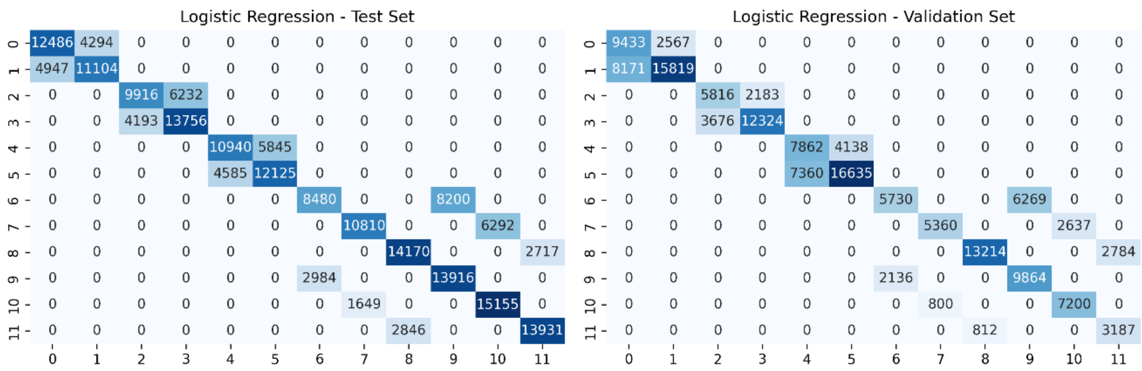

Figure 4-1 Confusion Matrices of Logistic Regression

4.2.2 Polynomial Regression:

- **Test Accuracy: 83.91%**
- **Validation Accuracy: 86.72%**

Discussion: Polynomial Regression introduces higher-degree terms, allowing the model to capture non-linear patterns. The noticeable jump in accuracy from the standard Logistic Regression suggests that the dataset indeed contains non-linearities which the Polynomial Regression is better equipped to handle. The consistent performance on both test and validation sets indicates that the model generalizes well. The confusion matrices of this classifier with test and validation dataset is given the below Figure 5-2.

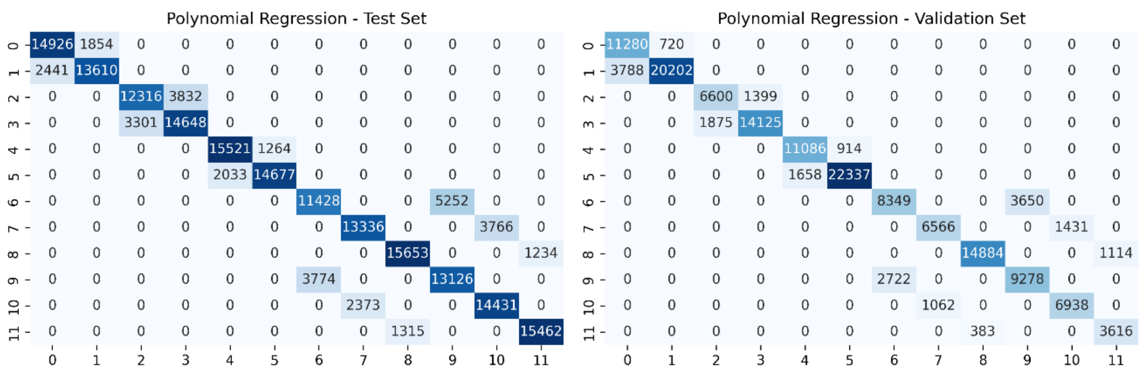

Figure 4-2 Confusion Matrices of Polynomial Regression

4.2.3 Decision Tree:

- **Test Accuracy: 98.76%**
- **Validation Accuracy: 93.93%**

Discussion: Decision Trees are non-linear models that split the data based on feature thresholds. The high test accuracy indicates that the model has learned the training data intricacies. However, the difference between the test and validation accuracy suggests the model might be overfitting to the training data, thereby not generalizing as effectively to unseen data. Decision Trees are notorious for being prone to overfitting, especially when not regularized. The confusion matrices of this classifier with test and validation dataset is given the below Figure 5-3.

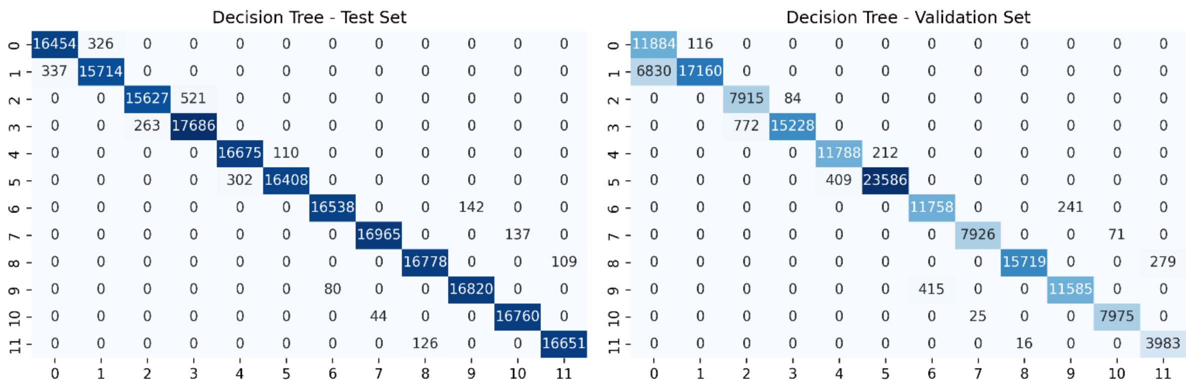

Figure 4-3 Confusion Matrices of Decision Tree

4.2.4 KNN

- **Test Accuracy: 94.22%**
- **Validation Accuracy: 92.87%**

Discussion: K-Nearest Neighbors (KNN) classifies data points based on the majority class among its 'k' neighbors. The model's high accuracy on both datasets indicates it captures the data's underlying structure well. Given the nature of your dataset with sensor readings, it's plausible that similar readings (or neighbors) often correspond to similar health statuses, making KNN a suitable choice. The confusion matrices of this classifier with test and validation dataset is given the below Figure 5-4.

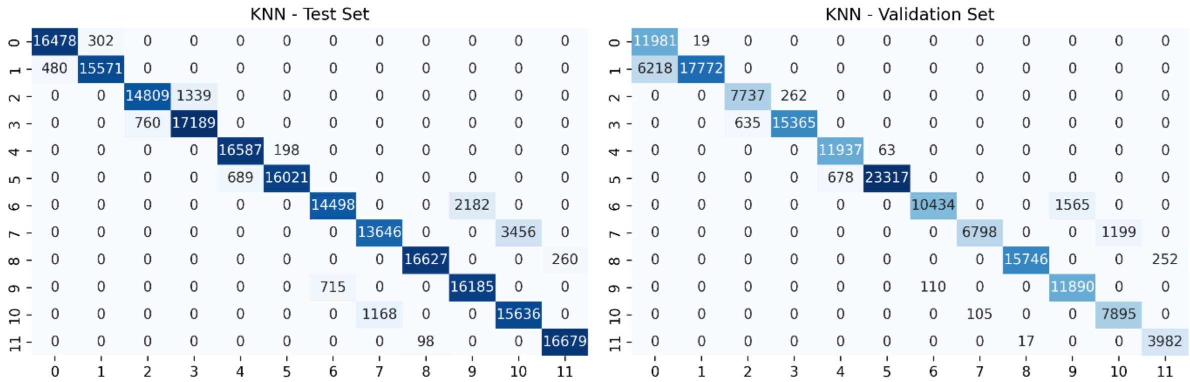

Figure 4-4 Confusion Matrices of KNN

4.2.5 Random Forest

- **Test Accuracy: 99.11%**
- **Validation Accuracy: 92.98%**

Discussion: Random Forest, an ensemble of decision trees, offers a robust performance. The almost perfect test accuracy suggests the model captures the training data's nuances. However, the dip in the validation accuracy, similar to the Decision Tree, hints at some level of overfitting. Yet, the overfitting is less pronounced than with a single Decision Tree, showcasing the power of ensemble methods in generalizing better. The confusion matrices of this classifier with test and validation dataset is given the below Figure 5-5.

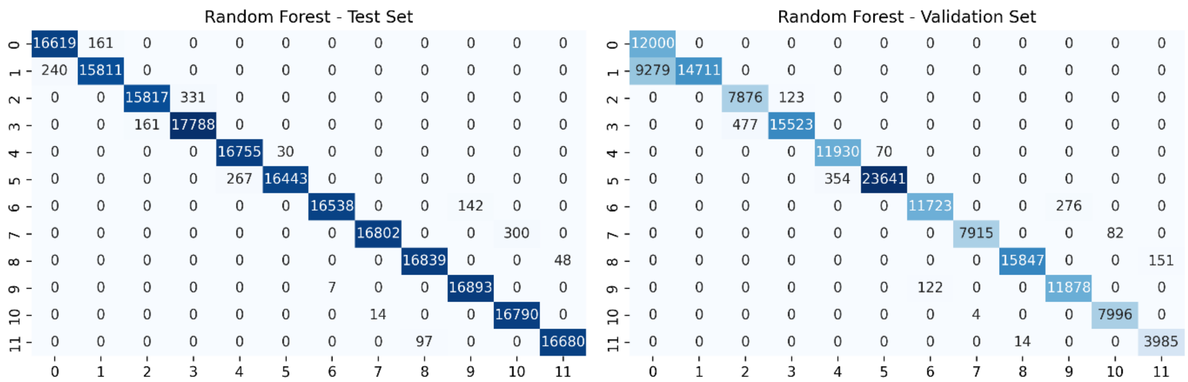

Figure 4-5 Confusion Matrices of Random Forest

4.2.6 XGBoost

- **Test Accuracy: 98.64%**
- **Validation Accuracy: 87.83%**

Discussion: XGBoost, a gradient boosting framework, builds trees sequentially, where each tree corrects its predecessor's errors. The high test accuracy suggests that the model fits the training data very well. The difference in accuracy between test and validation sets, however, indicates potential overfitting. Given that boosting algorithms like XGBoost can be sensitive to noise in the data, careful hyperparameter tuning and regularization may help bridge this gap. The confusion matrices of this classifier with test and validation dataset is given the below Figure 5-6.

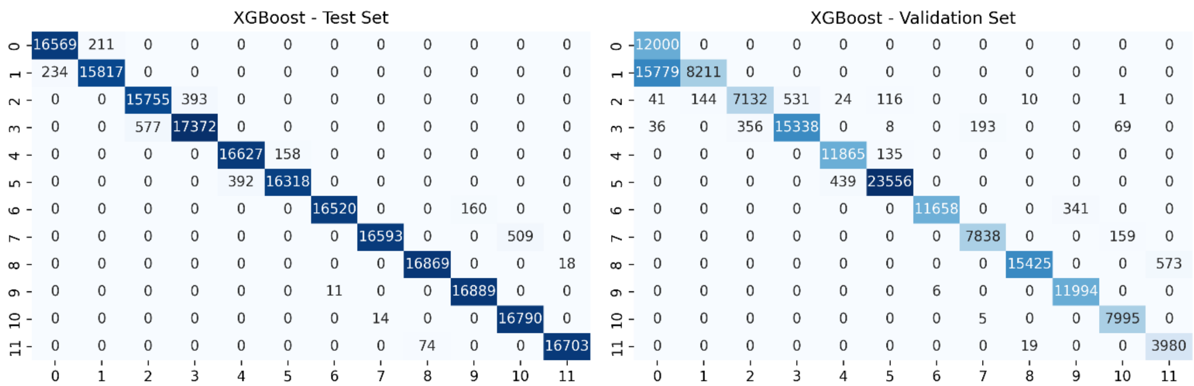

Figure 4-6 Confusion Matrices of XGBoost

The results clearly depict a wide variance in performance across models, emphasizing the importance of model selection tailored to the specific nature of the data. While ensemble methods like Random Forest and XGBoost exhibit impressive accuracies on the training set, care must be taken to prevent overfitting, ensuring that these models remain practical and reliable in real-world applications. Regularization, pruning, and hyperparameter tuning are essential tools in achieving this balance. On the other hand, simpler models like Logistic Regression, though not as accurate, provide a valuable baseline and can be more interpretable. The choice of classifier should ultimately align with the specific goals of the research, considering factors like interpretability, computational efficiency, and the need for real-time predictions.

4.3 Secondary Imputation Model Results

The secondary imputation model serves a pivotal role in the overarching two-tier system. As a reminder, its primary responsibility is to contend with instances where sensor data might be faulty, uncalibrated, or missing. The importance and effectiveness of such a model can't be overstated. understanding the full scope of our innovative two-tier model's importance requires a comparative analysis of the system's performance with and without the secondary imputation model.

Table 4-2 Performance comparison

Algorithm	**Test Accuracy Without Imputation**	**Validation Accuracy Without Imputation**	**Test Accuracy with Imputation**	**Validation Accuracy with Imputation**
Logistic Regression	66.53%	64.24%	72.82%	72.09%
Polynomial Regression	78.65%	79.52%	83.91%	86.72%
Decision Tree	95.88%	91.45%	98.76%	93.93%
KNN	90.57%	89.02%	94.22%	92.87%
Random Forest	96.51%	91.36%	99.11%	92.98%
XGBoost	96.05%	85.92%	98.64%	87.83%

By examining the accuracy metrics of the primary classification algorithms both with and without the imputation layer as depicted in the table 4-2 above, a clear enhancement in performance can be observed:

- **Logistic Regression**: Improved from a test accuracy of 66.53% to 72.82% and a validation accuracy of 64.24% to 72.09%.

- **Polynomial Regression**: Showcased betterment from a test accuracy of 78.65% to 83.91% and a validation accuracy rise from 79.52% to 86.72%.
- **Decision Tree**: Notably increased from a test accuracy of 95.88% to 98.76%, with the validation accuracy improving from 91.45% to 93.93%.
- And similar significant improvements can be observed across other classifiers like KNN, Random Forest, and XGBoost.

This improvement in accuracy metrics underscores the transformative impact of the secondary imputation model on the primary classifiers.

4.3.1 Holistic Data Management

The effectiveness of the secondary imputation model is evident through the improvement in accuracy metrics across all classifiers. It substantiates the assertion that handling missing or faulty data intelligently, rather than merely discarding or using rudimentary techniques, drastically optimizes the system's performance.

4.3.2 Significance in Real-World Scenarios

As mentioned earlier, in real-world scenarios, especially in aircraft landing gear simulations, data discrepancies due to sensor malfunctions or glitches are frequent. The secondary imputation model's data-driven approach successfully navigates these challenges, contributing to the impressive results.

4.3.3 Feature Relationships & System Dynamics

Our innovative two-tier system doesn't work in isolation. The interconnectedness between main and redundant features, reflecting the underlying physics and engineering of the aircraft systems, makes the secondary imputation model's success possible. It's not just about mathematical modeling but understanding the very essence of the system's design and operation.

4.3.4 Enhancing Reliability and Confidence

The tangible improvement in performance metrics with the secondary imputation model provides a solid foundation for stakeholders to base their decisions on. It offers an additional layer of reliability, ensuring that the derived insights are not just accurate but also dependable.

With the combined strengths of the primary classification models and the secondary imputation model, this research presents a robust and innovative approach to aircraft landing gear system analysis. Future research directions could further refine the imputation strategies, integrating more advanced algorithms or even exploring deep learning techniques for data imputation.

5 Integration of Explainable AI Techniques

5.1 Case in Point: Landing Gear Health Prediction of the 36536th Instance

Consider the proof of the concept, we have randomly picked the 36536th instance in our dataset with corresponding 5 features data and examine how our model came to the conclusion of the health status of that point. The model predicted a specific health state for the landing gear as “pumpFail_veryHighTemp” failure category. While such a prediction is valuable; without context, it remains an isolated data point. Enter XAI. Through the force plot as shown in the figure 6-1, we could discern the most influential factors driving this prediction.

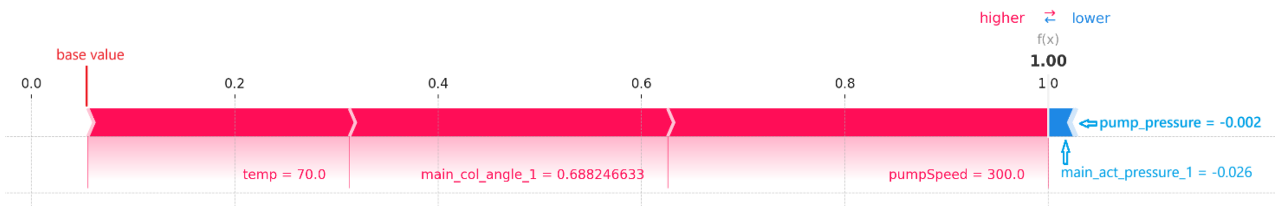

Figure 5-1 Forced SHAP plot for 36536th health status classification

5.1.1 Understanding the Force Plot

The **base value** is the starting point of the plot. It represents the average prediction of the model over the entire dataset. In this case, it is the average likelihood of the “pumpFail_veryHighTemp” class, which is 0.0065. The **output value** (f (x)) is the model's prediction for this specific instance. The difference between the base value and the output value is explained by the contributions of each feature. The **features** are displayed as colored arrows in the figure 6-1. The direction of the arrow (left or right) indicates whether the feature is pushing the prediction to decrease (left) or increase (right). The color represents the feature value (blue for low and red for high), and the width of the arrow represents the magnitude of the feature's impact.

5.1.2 Interpreting the plot for the 36536th instance

Here are the three factors that positively influenced the prediction:

1. **pumpSpeed**: The speed of the pump had the most substantial positive influence on the prediction with a SHAP value of approximately 0.374. pumpSpeed zero is the main character in “pumpFail_veryHighTemp” failure categorisation.

2. **main_col_angle_1**: The angle of the main column 1 also played a significant role in pushing the prediction positively with a SHAP value of about 0.313. It shouted saying that there is no movement in the landing gear.

3. **temp**: Temperature had a positive SHAP value of approximately 0.257, indicating that the temperature readings for this instance were higher than average.

Now, regarding the features that lowered or pushed back the prediction:

4. **main_act_pressure_1**: This feature had a significant negative influence on the prediction with a SHAP value of approximately −0.026. This means that the pressure in the main actuator 1 for this instance was a strong piece of evidence that pushed the prediction away from the average prediction of the model. A lower value of this pressure than the average could indicate some other issues in the landing gear health.

5. **pump_pressure**: This feature also had a negative impact on the prediction, albeit to a lesser extent, with a SHAP value of approximately −0.002. This means the pump pressure for this instance was another piece of evidence that slightly pushed the prediction away from the average

prediction. Again, a lower pump pressure than average might be indicative of certain conditions related to the aircraft's landing gear.

6 Significance of Key Sensors and Their Economic Implications

In our extensive study on aircraft landing gear health assessment, the temperature sensor and the pump speed measurement sensor emerged as pivotal elements. These sensors are not mere tools for data collection; they are the linchpins ensuring optimal aircraft performance and, most importantly, passenger safety. These two specific sensors are absent in traditional ATA 32 landing gear architecture. This study signifies the need of capturing the temperature and speed of hydraulic pump plays a crucial role, hence the focus has to be given by the aviation industry for the integration of the above-mentioned sensor along with their redundant sensors.

From an economic standpoint, integrating these sensors into existing aerospace models presents a transformative opportunity. Accurate readings can preempt potential issues, translating to significant savings by preventing prolonged aircraft downtimes and expensive repairs. An upfront investment in retrofitting existing aircraft with advanced sensors can lead to long-term benefits, including reduced maintenance costs and a prolonged aircraft lifespan. Additionally, in an industry where reputation is paramount, airlines equipped with state-of-the-art sensors stand out, promoting passenger trust and brand loyalty.

7 Conclusion

This research ventured into the intricate realm of aircraft landing gear systems, aiming to develop a robust model for health assessment through sensor data analysis. Central to our methodological approach was the two-tier system: a primary classification model strengthened by a secondary imputation model. In the vast expanse of sensor data, inconsistencies and anomalies are inevitable. The introduction of our imputation model, tailored to fill these data gaps intelligently, proved to be a game-changer, ensuring data integrity and consistency.

Our comprehensive evaluation of various classifiers highlighted the nuanced nature of our dataset, with results indicating clear variances in performance. Notably, ensemble methods like Random Forest and XGBoost showcased impressive accuracies on the training set. Still, the overarching narrative emphasized the necessity of a balance between achieving high accuracy and preventing overfitting.

Furthermore, the pivotal role of temperature and pump speed measurement sensors emerged as a cornerstone for accurate predictions. Their importance transcends mere technical functionalities, extending to significant economic ramifications for the aerospace industry. Proactive investments in these sensors can lead to substantial long-term operational savings and heightened safety standards.

In essence, this study sheds light on the profound impact of strategic data handling and the role of specific sensors in the ever-evolving domain of aerospace systems. The insights garnered not only propel the aerospace sector toward enhanced safety protocols but also underline the symbiotic relationship between technology and economic efficiency. Future endeavours in this field would do well to remember that in the delicate dance of machinery, every data point, every sensor, holds the potential to shape the future of air travel.

8 Future Works and suggestions

Building on the foundation of our research, there are several avenues for exploration and improvement. The unique approach we've undertaken, which emphasizes redundancy and categorizes features for system health monitoring, indeed offers a novel solution to challenges in aircraft landing gear systems. However, every study, including ours, presents certain limitations and assumptions, which, when addressed, can further strengthen the research. Here are some recommendations for future endeavors:

- **Integration with Real-world Data:** While our research heavily depended on simulations, future studies could seek to incorporate real-world flight data. By doing so, researchers could gain insights into more diverse and complex scenarios, ensuring that the model's robustness is tested in real-world environments.
- **Expanded Sensor Technologies:** The rapid advancements in sensor technologies present a prime opportunity. Future work should explore the integration of state-of-the-art sensors, understanding their capabilities and constraints. This could improve the model's sensitivity and predictive accuracy.
- **Deepening Redundancy Approaches:** Our approach acknowledged the real-world importance of redundancy. Building on this, future research can delve deeper into creating more layers of redundancy. This could involve exploring the possibility of utilizing tertiary sensors or even creating synthetic data through advanced algorithms when both primary and secondary sensors fail.
- **Enhanced Machine Learning Models:** As machine learning and artificial intelligence evolve, there's potential to explore more advanced algorithms that can improve prediction accuracy and robustness. Techniques like neural networks or ensemble models, adapted specifically for aviation requirements, could be studied.
- **Holistic Aircraft Health Monitoring:** While our focus was on the landing gear system, future research could look at integrating our system into a

more comprehensive aircraft health monitoring system. This would involve understanding interactions between different aircraft components and how they affect overall health predictions.

- **Considering Environmental Factors:** In real-world scenarios, external factors like weather conditions, air traffic, and even pilot behavior can play crucial roles in the health of aircraft systems. Future studies should seek to integrate these parameters, ensuring a holistic view of the system's health.

In conclusion, while our research lays a significant foundation for aircraft landing gear system monitoring, it also opens the doors to numerous possibilities. Each of these recommendations, when pursued, holds the promise of elevating our solution, making it even more valuable in the ever-evolving domain of aviation safety.

9 Reference

[1] ICAO, "Accident Statistics," https://www.icao.int/safety/iStars/Pages/Accident-Statistics.aspx, 2023.

[2] H. S., "Incident: Finnair Saudi Airlines A330-200, main wheel tyre damage leads to hydraulics leak and steering failure," http://avherald.com/h?article=41b9c d1f, Online, 2009, accessed Aug 2023.

[3] SKYbrary, "A343, Helsinki Finland, 2009," https://www. skybrary.aero/index.Php/A343,_Helsinki_Finland_2009, Article accessed Aug 2023.

[4] Yang, S., Crisp, M., Penty, R.V. and White, I.H. (2018). RFID Enabled Health Monitoring System for Aircraft Landing Gear. *IEEE Journal of Radio Frequency Identification*, 2(3), pp.159–169. doi:https://doi.org/10.1109/jrfid.2018.2822770.

[5] Phillips, P., Diston, D. and Starr, A. (2011). Perspectives on the commercial development of landing gear health monitoring systems. *Transportation*

Research Part C: Emerging Technologies, 19(6), pp.1339–1352. doi:https://doi.org/10.1016/j.trc.2011.03.005.

[6] Boniol, Frédéric and Wiels, Virginie and Ait Ameur, Yamine and Schewe, Klaus-Dieter The landing gear case study: challenges and experiments. (2016) International Journal on Software Tools for Technology Transfer, 19 (2). 133-140. ISSN 1433-2779

[7] Kang, Byung-Hyuk, Bang-Hyun Jo, Bo-Gyu Kim, Jai-Hyuk Hwang, and Seung-Bok Choi. 2023. "Linear and Nonlinear Models for Drop Simulation of an Aircraft Landing Gear System with MR Dampers" *Actuators* 12, no. 7: 287. https://doi.org/10.3390/act12070287

[8] Chen, J., Chen, S., Liu, Z., Luo, C., Jing, Z. and Xu, Q. (2020). Health Monitoring of Landing Gear Retraction/Extension System Based on Optimized Fuzzy C-Means Algorithm. *IEEE Access*, 8(2), pp.219611–219621. doi:https://doi.org/10.1109/access.2020.3042888.

[9] David Grewell, Khumprom, Phattara, and Nita Yodo. 2020. "Deep Neural Network Feature Selection Approaches for Data-Driven Prognostic Model of Aircraft Engines" *Aerospace* 7, no. 9: 132. https://doi.org/10.3390/aerospace7090132

[10] Dangut, M.D., Jennions, I.K., King, S. *et al.* A rare failure detection model for aircraft predictive maintenance using a deep hybrid learning approach. *Neural Comput & Applic* **35**, 2991–3009 (2023). https://doi.org/10.1007/s00521-022-07167-8

[11] Zhao, Y., Zhao, H., Ai, J. and Dong, Y. (2022). Robust Data-Driven Fault Detection: An Application to Aircraft Air Data Sensors. *International Journal of Aerospace Engineering*, 2022, pp.1–17. doi:https://doi.org/10.1155/2022/2918458.

[12] Jacazio, G., Ritter, O., Wende, G., Gentile, R., Antonio Carlo Bertolino, Marino, F. and Raviola, A. (2018). Intelligent Diagnostics for Aircraft Hydraulic Equipment. *EUROPEAN CONFERENCE OF THE*

PROGNOSTICS AND HEALTH MANAGEMENT SOCIETY 2018, 4(1). doi:https://doi.org/10.36001/phme.2018.v4i1.410.

[13] Shen, Kenan, and Dongbiao Zhao. 2023. "An EMD-LSTM Deep Learning Method for Aircraft Hydraulic System Fault Diagnosis under Different Environmental Noises" *Aerospace* 10, no. 1: 55. https://doi.org/10.3390/aerospace10010055

[14] Swischuk, R. and Allaire, D. (2019). A Machine Learning Approach to Aircraft Sensor Error Detection and Correction. *Journal of Computing and Information Science in Engineering*, 19(4). doi:https://doi.org/10.1115/1.4043567.

[15] Shukla, B., Fan, I.-S. and Jennions, I.K. (2020). Opportunities for Explainable Artificial Intelligence in Aerospace Predictive Maintenance. *EUROPEAN CONFERENCE OF THE PROGNOSTICS AND HEALTH MANAGEMENT SOCIETY 2020*, 5(1), pp.11–11. doi:https://doi.org/10.36001/phme.2020.v5i1.1231.

Appendices

Appendix A : Ethical approval letter

28 April 2023

Dear Mr Kunchitigara Palya Narasimhaiah ,

Reference: CURES/18258/2023
Project ID: 21100
Title: Machine learning-based Health Monitoring of Landing Gear Actuation Systems

Thank you for your application to the Cranfield University Research Ethics System (CURES).

We are pleased to inform you your CURES application, reference CURES/18258/2023 **has been reviewed. You may now proceed with the research activities you have sought approval for.**

If you have any queries, please contact CURES Support.

We wish you every success with your project.

Regards,

CURES Team

Appendix B : Risk analysis

In the journey of any scientific endeavor, understanding and accounting for potential risks is paramount. This not only adds credibility to the research process but also prepares the researcher for unforeseen challenges. Our research, though rigorous and comprehensive, is not exempt from potential pitfalls. To ensure a holistic and transparent understanding, we present a dedicated risk analysis that evaluates potential threats to the validity and applicability of our findings. This analysis is coupled with mitigation strategies, showcasing proactive measures taken to address and minimize these risks.

Table 9-1 Risk Assessment and Mitigation Strategies for the Research Project

Risk	Description	Mitigation Strategy	Level of Risk
Inaccurate Simulation Data	Simulated data might not accurately reflect real-world conditions, leading to models that don't generalize well in practical scenarios.	Used robust and industry-accepted simulation tools. Cross-validated simulated results with accepted historical simulation data to ensure consistency and reliability.	Medium
Biased Classifier Performance	Some classifiers might seem effective based on training data but may perform poorly with unseen data.	Leveraged a diverse set of classifiers and tested their performance against both validation and test datasets. Validation datasets are isolated and completely unseen by the trained model. This approach ensured a holistic evaluation and minimized the risk of selection bias.	High
Misinterpretation of Explainable AI (XAI) Outputs	Stakeholders might misinterpret or misrepresent XAI outputs, leading to incorrect decisions.	Ensured rigorous documentation and provided explanatory guidelines for the XAI outputs to ensure clarity and correct interpretation of results.	Medium
Model Scalability and Evolution	As aircraft technology evolves, the model might become obsolete or may not scale effectively.	Designed the model architecture to be modular, allowing for easy updates and integrations. Kept abreast of industry advancements to ensure periodic model recalibrations and updates.	Low

Appendix C : Project Timeline Overview

The progression of this research project, spanning from April to August 2023, followed a meticulously planned schedule to ensure methodical and systematic execution. The timeline details the core research activities and delineates the exact weeks dedicated to each pivotal task.

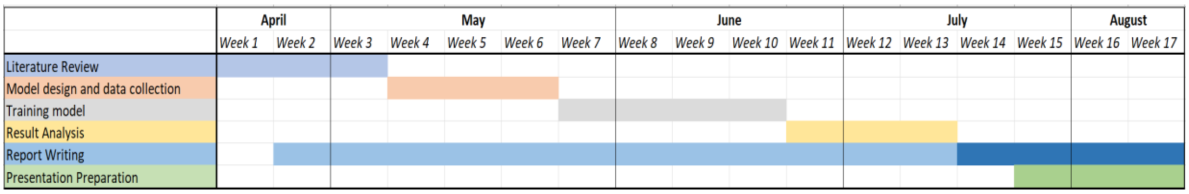

Figure Appendix-1 Project timeline

- **Literature Review**: The project commenced with a profound exploration of existing literature. Weeks 1 to 3 (April) were dedicated to immersing ourselves in prior research, identifying gaps, and establishing a foundation for the upcoming stages.

- **Model Design and Data Collection**: With the theoretical foundation established, the next phase entailed designing our unique two-tier ML model and collecting pertinent data. This stage consumed weeks 4 to 6, stretching from late April to mid-May.

- **Training Model**: June was an intensive period with the primary focus on training our ML model. By harnessing the accumulated data, we dedicated weeks 7 to 10 to perfect the model, ensuring its accuracy and efficiency.

- **Result Analysis**: With the trained model in hand, July saw the onset of result analysis. Weeks 11 to 13 were reserved for evaluating and interpreting the model's predictions, ensuring they align with real-world scenarios.

- **Report Writing**: The process of documenting our findings was spread across the project, ensuring real-time recording and reflections. Spanning from week 2 all the way to week 17, this ongoing task culminated in August, with a comprehensive report detailing every phase of the research.

- **Presentation Preparation**: Concurrently, in the concluding phase of the project, weeks 15 to 17 were set aside for crafting an impactful presentation. This entailed synthesizing our findings and designing visuals that would best communicate the significance and outcomes of our research to our audience.

Mapping this structured timeline not only facilitated effective project management but also ensured that each research phase received the requisite focus and diligence. The ensuing Gantt chart provides a visual representation of this timeline, capturing the sequence and overlap of these core activities.

Appendix D : Codes

Classifier Evaluation using Two-Tier Preprocessing

In the code presented, we employ a two-tier preprocessing methodology to clean and normalize data before it is input to various classifiers.

1. **Data Loading**: The datasets **MasterData.csv** and **MasterDataValidator.csv** are loaded, serving as training/test and validation datasets, respectively.
2. **Two-Tier Preprocessing**:
    - **Normalization**: Features are scaled using the **RobustScaler**, which is less prone to outliers.
    - **Outlier Handling**: Certain features that exceed a given threshold are adjusted using redundant or correlated features, improving the quality of data fed into the classifiers.
3. **Classifier Training and Evaluation**: Six classifiers (Logistic Regression, Polynomial Regression, Decision Tree, K-Nearest Neighbors, Random Forest, and XGBoost) are trained on the preprocessed training dataset. Their performance is subsequently evaluated on both the preprocessed test and validation sets.

```
import pandas as pd
from sklearn.preprocessing import RobustScaler, PolynomialFeatures,
LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
import os

# Check if the results directory exists, if not, create it
if not os.path.exists("results"):
    os.makedirs("results")

# Load the datasets
data = pd.read_csv("MasterData.csv")
data_validator = pd.read_csv("MasterDataValidator.csv")

# Features and Target Variable
features = ['pumpSpeed', 'temp', 'pump_pressure',
'main_act_pressure_1', 'main_col_angle_1']
X = data[features]
y = data['health']

# Splitting the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

# Load validation data
X_valid = data_validator[features]
y_valid = data_validator['health']

# Convert string labels to integers for XGBoost compatibility
label_encoder = LabelEncoder()
y_train = label_encoder.fit_transform(y_train)
y_test = label_encoder.transform(y_test)
y_valid = label_encoder.transform(y_valid)

# Preprocessing with RobustScaler
scaler = RobustScaler().fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)
X_valid_scaled = scaler.transform(X_valid)

# Polynomial features for Polynomial Regression
poly = PolynomialFeatures(degree=2)
X_train_poly = poly.fit_transform(X_train_scaled)
X_test_poly = poly.transform(X_test_scaled)
X_valid_poly = poly.transform(X_valid_scaled)

# Training the models
models = {
    "Logistic Regression": LogisticRegression(solver='lbfgs',
max_iter=5000),
    "Polynomial Regression": LogisticRegression(solver='lbfgs',
```

```
max_iter=5000),
    "Decision Tree": DecisionTreeClassifier(random_state=42),
    "KNN": KNeighborsClassifier(),
    "Random Forest": RandomForestClassifier(random_state=42),
    "XGBoost": XGBClassifier(use_label_encoder=False,
eval_metric="logloss", random_state=42)
}

# Train the models
for name, model in models.items():
    print(f"Training {name} Classifier...")
    if name == "Polynomial Regression":
        model.fit(X_train_poly, y_train)
    else:
        model.fit(X_train_scaled, y_train)

# Evaluate the models on test and validation datasets, and plot the
results
for name, model in models.items():
    print(f"Evaluating {name} Classifier...")

    if name == "Polynomial Regression":
        y_pred_test = model.predict(X_test_poly)
        y_pred_valid = model.predict(X_valid_poly)
    else:
        y_pred_test = model.predict(X_test_scaled)
        y_pred_valid = model.predict(X_valid_scaled)

    # Accuracy
    test_accuracy = accuracy_score(y_test, y_pred_test)
    valid_accuracy = accuracy_score(y_valid, y_pred_valid)
    print(f"{name} Classifier Test Accuracy: {test_accuracy *
100:.2f}%")
    print(f"{name} Classifier Validation Accuracy: {valid_accuracy *
100:.2f}%\n")

    # Confusion matrices visualization
    fig, ax = plt.subplots(1, 2, figsize=(12, 4))
    sns.heatmap(confusion_matrix(y_test, y_pred_test), annot=True,
fmt="d", cmap="Blues", ax=ax[0], cbar=False)
    ax[0].set_title(f"{name} - Test Set")
    sns.heatmap(confusion_matrix(y_valid, y_pred_valid), annot=True,
fmt="d", cmap="Blues", ax=ax[1], cbar=False)
    ax[1].set_title(f"{name} - Validation Set")
    plt.tight_layout()
    plt.savefig(f"results/{name}_confusion_matrices.png", dpi=300)
    plt.show()
```

Deployment code: Interactive Health Prediction System with Explainable AI (XAI) Insights

The provided code is designed for a two-step machine learning prediction system that integrates user interactivity and explainability. The purpose is to predict the health of a system based on five selected features. Here's a breakdown of the steps and functionality:

1. **Data Loading**: The code begins by loading a dataset named **MasterData.csv**.
2. **Feature Selection**: Five primary features are chosen for modeling, including **pumpSpeed**, **temp**, **pump_pressure**, **main_act_pressure_1**, and **main_col_angle_1**.
3. **Primary Model**:
   - The data is normalized using the **RobustScaler** from scikit-learn.
   - A Decision Tree classifier is trained on this normalized data.
   - The trained primary model is then saved as a **pkl** file for future use.
   - An SHAP explainer is initialized for the Decision Tree model to provide insights on the impact of each feature on the predictions.
4. **Secondary Model (Imputation Model) Function**: This function handles user interactivity and imputes missing data based on some redundant features.
   - The user can either input new values or use an existing instance from the dataset.
   - The function handles missing data and scales the provided instance.
   - Predictions are made using the primary model, and the instance data is adjusted using some heuristic corrections.
   - The adjusted instance is then fed again to the primary model for a secondary prediction.

- Finally, the SHAP force plot is generated and saved to visualize the contribution of each feature to the model's prediction for the given instance.

5. **SHAP Explanation**: A separate function (**show_shap_values**) is provided to generate and visualize the SHAP values using a force plot.

6. **Testing**: The **secondary_model_preprocessing** function is tested at the end of the script.

```
import pandas as pd
import shap
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import RobustScaler
import joblib


"""
It intakes the instance value and give prediction it is user
interavtive and XAI
"""

# Load the dataset
data = pd.read_csv("MasterData.csv")

# Features and target variable
features = ['pumpSpeed', 'temp', 'pump_pressure',
'main_act_pressure_1', 'main_col_angle_1']
target = data['health']

# 1. Training the Primary Model
# Fit the Robust Scaler on the training data for only the five
selected features
scaler_primary = RobustScaler().fit(data[features])

# Normalize the data using the fitted Robust Scaler
data_normalized = scaler_primary.transform(data[features])

# Train the Decision Tree Classifier for the primary model using
normalized data
clf_primary_normalized = DecisionTreeClassifier(random_state=42)
clf_primary_normalized.fit(data_normalized, target)

# Save the primary model
joblib.dump(clf_primary_normalized,
"primary_model_normalized.pkl")

# Initialize the explainer for SHAP
explainer = shap.TreeExplainer(clf_primary_normalized)

# 2. Secondary Model (Imputation Model) Function
def secondary_model_preprocessing(data, primary_model, scaler):
```

```
    # Ask the user to enter new values or use an existing
instance
    choice = input("Do you want to enter new values or use an
existing instance? Enter 'new' or 'existing': ").lower()

    if choice == "new":
        # Get new values from the user
        instance = {}
        for feature in features:
            instance[feature] = float(input(f"Enter value for
{feature}: "))
        instance = pd.Series(instance)
    else:
        # Use existing instance
        instance_num = int(input("Enter the instance number: "))
        instance = data.iloc[instance_num].copy()
        print(f"\nExisting data for instance {instance_num}:\n")
        print(instance)

        # Ask the user if they want to modify any values
        modify_choice = input("\nDo you want to modify any
values? Enter 'yes' or 'no': ").lower()
        if modify_choice == "yes":
            for feature in features:
                new_value = input(f"Current value for {feature}
is {instance[feature]}. Enter a new value or press enter to keep
the current value: ")
                if new_value:
                    instance[feature] = float(new_value)

    print(f"\nActual Health: {instance['health']}")

    # Handle Missing Values
    if pd.isnull(instance['pump_pressure']):
        instance['pump_pressure'] = instance['pump_torque']
    if pd.isnull(instance['main_act_pressure_1']):
        instance['main_act_pressure_1'] =
instance['main_act_pressure_2']
    if pd.isnull(instance['main_col_angle_1']):
        instance['main_col_angle_1'] = -
instance['main_act_position_2']

    # Scale the instance using Robust Scaler
    scaled_instance = scaler.transform([instance[features]])[0]

    # Primary Model Prediction
    primary_model_prediction =
primary_model.predict([scaled_instance])[0]
    print(f"Primary Model Predicted Health:
{primary_model_prediction}")

    # Correct outliers using the Redundant Features
    threshold = 2
    if abs(scaled_instance[features.index('pump_pressure')]) >
threshold:
        scaled_instance[features.index('pump_pressure')] =
scaled_instance[features.index('pump_torque')]
    if
abs(scaled_instance[features.index('main_act_pressure_1')]) >
```

```
threshold:
        scaled_instance[features.index('main_act_pressure_1')] =
scaled_instance[features.index('main_act_pressure_2')]
    if abs(scaled_instance[features.index('main_col_angle_1')])
> threshold:
        scaled_instance[features.index('main_col_angle_1')] = -
scaled_instance[features.index('main_act_position_2')]

    # Secondary (Imputation) Model Prediction
    secondary_model_prediction =
primary_model.predict([scaled_instance])[0]
    print(f"Secondary Model (Imputer) Predicted Health:
{secondary_model_prediction}")


    # Display the SHAP force plot for explanation
    shap_value = show_shap_values(scaled_instance.reshape(1, -
1), primary_model, explainer)
    shap.save_html("shap_force_plot.html", shap_value)

# Function to show SHAP values using force plot
def show_shap_values(instance, model, explainer):
    shap_values = explainer.shap_values(instance)
    shap.initjs()
    return shap.force_plot(explainer.expected_value[0],
shap_values[0], instance)

# Test the function
secondary_model_preprocessing(data, clf_primary_normalized,
scaler_primary)
```