AN ABSTRACT OF THE THESIS OF

Nathan Algarra for the degree of Master of Science in Mechanical Engineering presented on 1 December 2023.

Title: Functional Analysis to Inform Early Design Decisions for Novel Systems with Substantial Operations and Maintenance Costs

Abstract approved:

Andy Dong

Long field-life systems are those for which operational, as opposed to capital, expenditures dominate the life time cost of the system. In established product categories, operational experience and historical maintenance data allows engineers to make informed decisions influencing maintenance during the early design stages when design changes are at their cheapest and few costs have been committed. Design engineers of novel products lack the same maintenance assessment resources afforded to designers working on established products. There may be little to no operational experience and historical maintenance data may not exist for components and assemblies used in novel designs. Prior research has shown that knowledge of functions can assist in making informed design decision early in the process. This research examines the circumstances under which functions can be used to predict the maintenance intervals of long field-life systems. A model to predict maintenance intervals was built using machine learning over maintenance data mined from an established engineered system, a civil aviation jet and its engines. Through functional analysis, the essential functions of the system's parts were determined and associated with the parts' maintenance intervals. A supervised machine learning model was used to learn the required maintenance, resulting in a function-maintenance model. This function-maintenance model was evaluated in three ways. First, the model was calibrated using classical machine learning metrics. Then, through grounding by applying the model to a large design repository containing known devices and their associated functions. Finally, to

evaluate the model's application to novel systems, it was tested on a prototype renewable energy system and compared against the assessment of an expert designer with experience in the field. Through these evaluations, I conclude that there is an identifiable relationship between functionality and maintenance intervals. Furthermore, this relationship can be leveraged to inform early design decisions influencing the maintenance of novel systems. In developing a model to assess maintenance intervals of possible designs and the understanding to inform design decisions, this research contributes to methods to reduce design costs through earlier decision making and lifetime costs through improved system maintainability.

Functional Analysis to Inform Early Design Decisions for Novel Systems with Substantial Operations and Maintenance Costs

by

Nathan Algarra

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented 1 December 2023
Commencement June 2024

Master of Science thesis of Nathan Algarra presented on 1 December 2023.

APPROVED:

Major Professor, representing Mechanical Engineering

Head of the School of Mechanical, Industrial, and Manufacturing Engineering

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Nathan Algarra, Author

TABLE OF CONTENTS

Page

1 Introduction 1

1.1 Problem and Motivation . . . 1

1.2 Contribution . . . 2

1.3 Definitions . . . 3

2 Background 5

2.1 The Engineering Design Process . . . 5

2.1.1 Overview . . . 5

2.1.2 Early Design Decisions . . . 6

2.1.3 Early Design Tools . . . 7

2.2 Maintenance . . . 9

2.2.1 Overview . . . 9

2.2.2 Current Methods . . . 10

2.3 Functional Modeling . . . 12

2.3.1 Overview . . . 12

2.3.2 Applications . . . 12

2.3.3 Functional Modeling . . . 13

2.4 Machine Learning . . . 17

2.4.1 Overview . . . 17

2.4.2 Supervised Classification Models . . . 19

2.4.3 Assessing Machine Learning Models . . . 21

3 Methods 24

3.1 Functional Modeling Jargon . . . 24

3.2 Data Mining . . . 26

3.2.1 Maintenance Documentation . . . 26

3.2.2 Data Collection . . . 27

3.3 Functional Modeling . . . 32

3.3.1 Overview . . . 32

3.3.2 Functional Decomposition . . . 33

3.4 Machine Learning . . . 35

3.4.1 Overview . . . 35

3.4.2 Data Preparation . . . 36

3.4.3 Training the Model . . . 40

3.5 Evaluating the Model . . . 41

TABLE OF CONTENTS (Continued)

Page

3.5.1 Calibration: Performance Metrics 41
3.5.2 Grounding: Design Repository 43
3.5.3 Verification: Laboratory Upgrade Point Absorber Application . . . 44

4 Results 47

4.1 Function-Maintenance Model 47
4.1.1 Overview 47
4.1.2 Using the Model 50

4.2 Model Evaluation 57
4.2.1 Calibration: Performance Metrics 57
4.2.2 Grounding: Design Repository Application 62
4.2.3 Verification: LUPA Application 65

4.3 Applying the Model 66
4.3.1 Carriage Rail Securement 66

4.4 Discussion 69
4.4.1 Function-Maintenance Relationships 69
4.4.2 Function-Maintenance Model 70
4.4.3 Applying the Function-Maintenance Model 71

4.5 Limitations 72
4.5.1 Training Data 72
4.5.2 Artifacts of Training Data 73
4.5.3 Evaluation 74

4.6 Improving the Work 75
4.6.1 Intra-Coder Reliability 75
4.6.2 Documentation of Maintenance Intervals 76
4.6.3 Documentation of Preventative Maintenance Type 76
4.6.4 Function Through Action Versus Inherent Characteristics 77

5 Conclusion 78

5.1 Overview 78

5.2 Key Findings 78

5.3 Future Work 79
5.3.1 More Training Data 79
5.3.2 Machine Learning Algorithms 80
5.3.3 Additional Features 81

5.4 Parting 81

LIST OF FIGURES

Figure | Page

2.1 Costs committed versus costs spent throughout the engineering design process. 7

2.2 The vertical stabilizer and rudder of a civil aircraft. The vertical stabilizer is found at the rear of the fuselage on all civil aircraft. The rudder is flight control surface found on the trailing edge of the vertical stabilizer. The rudder rotates about a vertical axis, at the front of the rudder, that causes its trailing edge to swing port and starboard to induce yaw. 16

3.1 Functional decomposition example of the Rudder Trim Tab and Pitot-Static System that ends with the functional descriptions of *Change Mechanical Energy* and *Sense Pneumatic Energy*, respectively. 34

4.1 Decision tree generated using level-1 and level-2 functions, flows and function-flow pairs. As well as component or assembly and function-flow count. . . . 48

4.2 Decision tree generated using level-1 and level-2 flows, component or assembly, and function-flow count. 49

4.3 3D model of LUPA showing the carriage, carriage rail, and spar. The model has been simplified to improve visibility. Note the three visible screw holes that secure the carriage rail to the spar. 67

LIST OF TABLES

Table Page

2.1 The 6 steps of the Engineering Design Process as described by Ullman. Steps 1 through 4 are often called 'early design.' 5

2.2 The level-1 function (*Control Magnitude*) and flow (*Energy*) with a few of their corresponding level-2 and level-3 subfunctions/subflows. This is not an exhaustive list of all subfunctions/subflows for *Control Magnitude* and *Energy* but serves to show the tree structure of the functional basis. 15

2.3 Example confusion matrix comparing the known category of 45 data points versus the predicted category of those data points. 21

3.1 Chapters from which maintenance data was primarily mined 27

3.2 Example of data mined from Learjet and CJ610 maintenance manuals. . . . 29

3.3 Data synthesized from parts described in Table 3.2. Only two sets of function-flow pairs are shown for brevity. 30

3.4 Excerpt from the "Relative Aircraft Age and Utilization Table" found in the Learjet MM. All maintenance frequency conversions were completed using expected utilization of a moderate age, mid utilization aircraft. 31

3.5 Converted values of a month, year, landing and engine cycle into flight hours for a standard case aircraft. 32

3.6 Half order of magnitude buckets used for the training data. The three columns under the heading Bucket Range are maintenance intervals, measured in flight hours. All flight hour values are rounded to the nearest integer. 38

3.7 Data formatted to train the model. As there are 357 unique features, only 11 are shown. Component or assembly, function-flow count, three level-1 functions, three level-1 flows, and three level-1 function-flow pairs. 39

3.8 Table of decreasing complexity parameter values and their corresponding calibration metrics. 42

4.1 Five sample parts chosen to highlight the process through which maintenance predictions are made. Only two function and flow features are shown as none of the sample parts perform more than two functions. 51

LIST OF TABLES (Continued)

Table | Page

4.2 Complexity parameter table calculated using the Full Model decision tree using decreasing complexity parameter values. 58

4.3 Complexity parameter table calculated using the Flow Model decision tree using decreasing complexity parameter values. 59

4.4 Confusion matrix relating the known maintenance frequency with the Full Model's predictions. 60

4.5 Confusion matrix relating the known maintenance frequency with the Flow Model's predictions. 60

4.6 Per class metrics for the Full and Flow Model, given by class. 61

4.7 Per model metrics for the Full and Flow Model. 62

4.8 9 sample parts of the 131 assessed from the Datsun truck. 64

4.9 Confusion matrix of model predicted versus expert design engineer assessed maintenance values for 30 LUPA parts 66

Maintainability is a design parameter. Maintenance is required as a consequence of design [6].

Chapter 1: Introduction

1.1 Problem and Motivation

In some long field-life systems, Operations and Maintenance (O&M) can account for a 60-80% of lifetime costs [1]. Mature technologies are able to draw on operational experience and historical data to better understand the performance characteristics of the components and assemblies that comprise the device in real-world use and environments. By leveraging this knowledge, designers are able to predict maintenance requirements and make informed decisions on aspects that may impact O&M early in the design process before significant design decisions have been committed. This design process breaks down when the affordances of mature technologies are removed, as is the case for novel long field-life systems. Several maintenance-related questions arise in this situation: How can maintenance intervals be predicted when the system is the first of its kind? Do reliability databases apply when components and systems are used in a novel context? How can the maintenance demands of innovative new concepts be evaluated without expending significant time and resources to detail out the concepts?

To explore these questions and to better understand the impact of novelty on the design of systems with significant O&M costs, Marine Rewewables Energy systems (MREs) are examined. MREs are a paragon of the type of novel systems that this research seeks to support. Characteristics of MREs create challenges in reducing O&M costs during their design. Offshore wind projects – the most mature MRE – are anticipated to pay up to one-third of the total project cost in O&M [2–4]. Even at such a high O&M cost, offshore wind fares the best of all MREs (wave, tidal, salinity gradient, etc.). Incorporating lessons learned from their terrestrial counterparts lends significant experience and data, a unique advantage amongst all MREs. Neary [5] uses 6 inputs to predict O&M costs for MREs, four of which are directly related to maintenance and one indirectly.

- Marine operations → Performing onsite *maintenance*
- Shore-side operations → Towing the device shore-side to perform *maintenance*
- Replacement parts → Recoverable parts used during *maintenance*
- Consumables → Parts consumed during *maintenance*

- Insurance → Impacted by many risks, one of which is the lack of operational experience performing *maintenance*
- Post-installation environmental monitoring

Despite O&M's status as a major cost driver, it is nearly impossible to estimate their costs until the latter design stages when the detailed design, environmental factors, and location are determined. At this stage of a project, O&M is a cost to be mitigated as opposed to a design variable to be optimized. Current techniques allow for maintainability considerations during early design but require an extant, completed design with materials and components specified [6–8]; these techniques have limited application to novel systems that lack historic data. As a result, the design changes necessary for MREs to improve maintainability and therefore O&M, are not able to be considered until far later in the design process. Designers are left mitigating costs or implementing changes that can be 10- to 100-fold [9] more costly compared to the same changes made during early design. Work in the design for maintainability field has shown that maintenance is not a set of actions that occur to a device once designed and manufactured. Instead, maintenance is an aspect of the device that is inherent to the culmination of design decisions made throughout the development of the device [10, 11].

The question is neither on the impact of early design for maintainability considerations nor on the efficacy of those considerations on reducing lifetime O&M costs, but instead on how these design practices can be implemented despite the limitations inherent to novel technologies. This research posits that functions, the elementary description of *WHAT* a part does as opposed to *HOW* it does it, has an inherent relationship with the maintenance demands of the part. Furthermore, this function-maintenance relationship can be leveraged during early design in conjunction with functional modeling to assess potential concepts and inform design decisions.

1.2 Contribution

My research contributes to the fields of design theory and design for maintainability in two ways. This research shows that there are consistent relationships between the maintenance intervals and the parts used in a design. This relation is based on the abstract functions embodied by that part rather than through the physical characteristics of the part.

This contribution is seated in the time-value of information; the earlier information is available the more impactful it can be. Information on the functions of a system is available long before the detailed information that is required for physical component testing is available. A function-maintenance relationship model would enable designers to make more informed decisions regarding maintenance before it would otherwise be possible. Second, this function-maintenance model can be utilized during conceptual design to assess functional models and conceptual/high-level designs for their relative maintenance requirements. The current state of the design for maintainability field provides guidelines to improve maintainability but lacks assessment tools, especially during early design; this model works to close that gap.

1.3 Definitions

To ensure clear and consistent communication, key terms used throughout this thesis are defined below and given additional context as needed.

Part is a general term used for brevity to mean any component, assembly, or system.

Maintainability refers to "...the aspects of a product that increase its serviceability and repairability, increase the cost-effectiveness of maintenance, and ensure that the product meets the requirements for its intended use" [12]. A key aspect of maintainability are the non-technical considerations, including operations costs associated with maintenance, training of maintenance personnel, and impacts of downtime due to maintenance or logistical support, etc.

Maintenance is "...all actions necessary for retaining a system or product in or restoring it to, a desired operational state" [6]. This chapter's epigraph best summarizes the difference between maintainability and maintenance. Maintenance is an action that is performed, whereas maintainability is a result of the summation of design decisions – intentional or not – made throughout the part's development. The term maintenance action used throughout this research is a singular maintenance act.

Component is any part that is repaired, replaced or serviced wholesale, without any further disassembly performed. An example to clarify the distinction between this and the next term (assembly), is a hydraulic actuator. Though the actuator is comprised of a variety of parts – including the piston, valves and seals – those parts would not be replaced individually during routine maintenance. The entire actuator would be removed and replaced, making it a component.

Assembly is anything excluded by the definition of a component. The purpose of this maintenance-centered definition is to capture the context in which the part is used.

Mean time between maintenance (MTBM), for the purposes of this research, is the average time between preventative maintenance actions. All types of preventative maintenance are included (described in Section 2.2.1).

Chapter 2: Background

2.1 The Engineering Design Process

2.1.1 Overview

The engineering design process is the iterative process that design teams embark upon that ends, not when the final bolt is fastened during manufacturing, but when the final bolt is unfastened and the device decommissioned at the end of its lifetime. Various formulations exists between codified design processes [9, 13, 14], but a suitable example is provided by Ullman [1].

Step	**Title**	**Description**
1	Product Discovery	The market opportunity for a product is identified. The three primary sources for products are: technology push, market pull and product change.
2	Project Planning	The finite pool of money, personnel, and equipment are allocated against uncertain schedules and cost estimates.
3	Product Definition	Customers are identified and their requirements for the product are determined. Engineering specifications and measures of success are established.
4	Conceptual Design	Concepts are generated and assessed against the information determined in the previous step. Conceptual design is the final step of what is called 'early design' and ends upon selection of a concept.
5	Product Development	The concept selected in the previous step is developed into a completed product and is released for production at the end of this step.
6	Product Support	The final design is manufactured, begins its service life and, depending on the type of product, is decommissioned or disposed of.

Table 2.1: The 6 steps of the Engineering Design Process as described by Ullman. Steps 1 through 4 are often called 'early design.'

For a clear understanding of the design process as formulated by Ullman, each step along with a brief description is displayed in Table 2.1. This research is primarily concerned with conceptual design. Specifically, considerations made during conceptual design as opposed to during product development as well as the impact and timing of those considerations on product support.

2.1.2 Early Design Decisions

The early design phase of a project is captured by two statements: "early design is characterized by a lack of information" [6] and "product cost is committed early in the design process and spent late in the process" [1]. Taken together, the early design phase can be seen as a race to gather as much information as possible to reduce uncertainty and make the most informed decisions possible [9, 14, 15]. Regardless of the reduction in uncertainty and the decisions made – informed and intentional...or otherwise – upon selecting a concept and moving forward into product development, a significant portion of the project's costs have been committed. The commonly cited heuristic is that ∼70% (sources range from 70% to 90% [14]) of a product's cost is committed during early design, but this claim is so poorly defined as to be unverifiable [16, 17]. Thematically similar, but verifiable results conclude that decisions made during early design account for 86% of the cost of all design changes [18] or that the average manufacturing cost can change by 47% based on the design of the product [17]. A more measured claim would be that any product's final cost will have an inherent lower limit. Subsequent design decisions can – at best – maintain this lower limit, but – at worst – will raise the lower limit [16]. Despite the disagreement in the value and measurement of costs committed during early design, the design community is unanimous on early design's impact; decisions made during early design have a significant impact on costs, performance, and ultimately, the success of a product.

Fig. 2.1 shows the relationship between committed and spent costs. The purpose of a thorough early design process and the 'race' to gather information is to ensure that the decisions made, concept selected, and the costs committed represent the best case scenario.

The decisions made during early design can be delayed or changed, but they come at exponential increase in cost. A well-researched heuristic is that if a design change costs $1 during early design, that same design change will cost $10 during product development and $100 during production [14]. Within the aerospace industry, an exemplar O&M dominated system, discussed further in Section 3.2, it was found that decisions made during early design are 13 times more expensive than decisions made later in the design process [18].

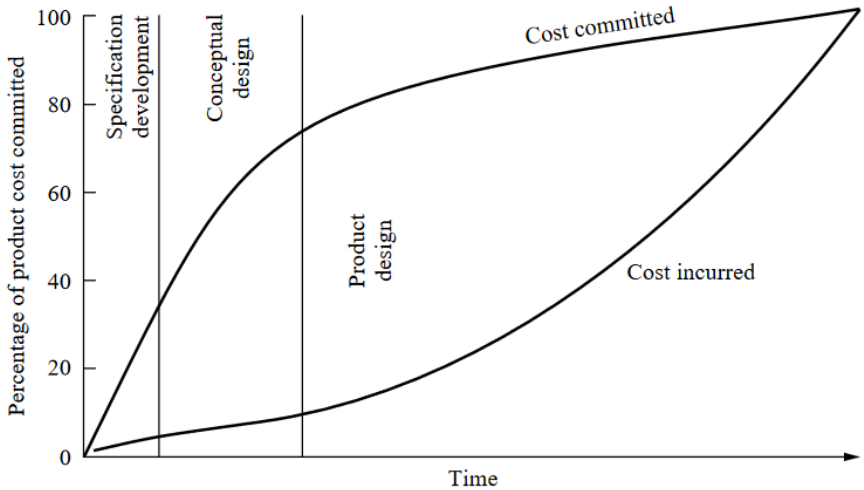


Figure 2.1: Costs committed versus costs spent throughout the engineering design process.

This highlights the need for early design decisions to be made after reducing uncertainty as much as possible and utilizing all information available. Bypassing early design doesn't bypass this problem; it simply removes the active decision from the designer and replaces it with passive assumptions. Despite the clear case for a thorough early design process, it remains "the least managed, the least documented, and the least understood" [14].

2.1.3 Early Design Tools

Two major avenues exist to reduce uncertainty and inform early decisions. A serious effort devoted towards the generation and evaluation of concepts; and design aids and tools, including the collection of tools called Design for X (DFX).

The purpose of a devoted effort towards the generation and evaluation of concepts is two-fold. First, a single concept thrust into development is statistically unlikely to be the best possible concept. The more concepts that are conceived of, the more likely that one of them will be a good concept [14]. The second aspect relates to the the definitions of the seemingly ambiguous descriptors "good", "better", and "best". The "best" concept is the concept that has the highest potential for success as defined during product definition (a "good" concept's definition mirrors this). Without the evaluation of a variety of potential concepts, a developer isn't able to reduce the uncertainty in a project's success as they don't know that their concept is "better" than other feasible concepts. The reduction in uncertainty has been quantified; projects that seriously consider multiple concepts had a 22% failure rate, with failure defined by "... could not meet the project goals and were canceled." Projects that focused on a single concept had a 75% failure rate [19]. Huge reductions in uncertainty such as this gain even more significance when the relatively minuscule costs spent generating concepts is factored in.

The second avenue to reduce uncertainty and inform early decisions are design aids and tools, including DFX. This avenue straddles the line between conceptual design and product development. Depending on the specific requirements of the tool, it may be used to produced "better" concepts or may be more appropriate to improve a concept once selected. Regardless, these tools fill a similar role. By evaluating previous experience and historical data, relationships and insights were identified and then refined into easily leveraged design heuristics. Design for assembly was the first design tool developed [20] that provided all designers with the knowledge and guidance that may otherwise only be available to expert designers. Shortly after, manufacturing expertise was refined to develop design for manufacture resources [21]. Following the widespread implementation and resultant benefits of the then combined design for manufacture and assembly (DFMA) [22, 23], the DFX field expanded. DFX refers to the variety of design tools including design for: life-cycle, recyclability, environment, quality, maintainability and reliability [24, 25] (this list is not exhaustive and includes only some of the most well known).

Design for maintainability and reliability are expanded upon in the following section, Section 2.2.2, but can quickly be classified into the "improving a concept once selected" category. Of the DFX and other design tools that are more firmly rooted in the concept improvement category, they can be leveraged as a concept evaluation tool as well. Using DFMA for reference, if the fundamental attributes that make a system more easily assembled and manufactured are known, and the list of potential concepts are of sufficient fidelity, DFMA can be used to evaluate and determine the "best" concept with regard to DFMA.

Despite the significant impact on development and lifetime costs, uncertainty reduction, and probability of project success, early design methods fail to accommodate O&M dominated systems. Instead, O&M considerations are largely addressed during product development through design for maintainability/reliability and maintenance/reliability prediction, negating the potential benefits of earlier consideration.

2.2 Maintenance

2.2.1 Overview

In the most extreme cases, estimates attribute 60-80% of lifetime costs to O&M. For offshore wind, the most developed MRE, O&M costs are 30% of lifetime costs [2, 3] or of cost per kWh [26]. When considering lifetime O&M costs, designers are left in a situation similar to that of manufacturing costs; good design for maintainability decisions can limit O&M cost increases, but bad design decisions stand to significantly increase lifetime O&M costs.

The tools available to designers to influence the O&M costs committed during early design will be described broadly as design for maintainability. Included in this umbrella term are design for reliability, inspectablitity, and serviceability. When taken together, these areas encompass how the design of a device influences it's required maintenance frequency, the ease of inspections necessary to evaluate when maintenance is necessary, and the ease of and time required to perform all activities required to restore a device to a desired operational state. Related to maintainability is supportability, the ease of logistical and administrative support for a device. Together, supportability and maintainability closely align with the "O" and "M" of O&M. This research focuses primarily on maintenance, as such supportability will not be explored in this research.

All maintenance falls into one of two categories, preventative or corrective. As the names imply, the difference is in when the maintenance task occurs. Preventative maintenance occurs before a system/part is no longer in the desired operational state. Corrective maintenance occurs after a system/part is no longer able to operate as intended. Within each category are subcategories that further specify the objective or the driver of maintenance [6].

Corrective maintenance includes deferred and emergency maintenance. Emergency maintenance is the least favorable of any maintenance task and is conducted when immediate intervention is required. It is considered the least safe, least effective and is estimated to incur costs three to five times higher than preventative maintenance. Deferred maintenance is also unplanned, but allows for time to plan and schedule the maintenance task. Run to failure strategies on non-essential aspects of a design are often maintained using deferred maintenance.

Preventative maintenance can be broken down into five subcategories, in ascending order of difficulty to implement they are: time based, failure finding, risk based, condition based, and predictive. Time based maintenance is the simplest strategy but has the largest potential for unnecessary costs. "Replace every 1,000,000 cycles" or "Lubricate every 1,000 operational hours" are time based maintenance tasks. Failure finding maintenance generally relates to parts that are in a passive state until required, like safety valves or emergency stop systems. Failure finding maintenance is conducted to ensure the part will operate as desired when needed. Risk based maintenance is a refined method of time based maintenance; where the risk (likelihood times consequence) of a failure is considered and used to adjust maintenance intervals. Condition based maintenance is the strategy of inspecting parts for signs of early stage failure evidence and planning maintenance based on a part's condition. Predictive maintenance is similar to condition based maintenance, but instead uses machine learning and internet of things to inform maintainers on when maintenance may be required.

2.2.2 Current Methods

The traditional approach to maintainability has largely been to ignore it or only give a cursory examination until well into product development [27, 28]. Even in aerospace and military design, where maintainability is a key concern as 30+ year expected life cycles are not uncommon, design teams are often not allocated appropriate time and resources to give maintainability the consideration it requires. More readily visible problems, like manufacturing or performance are of greater concern and as a result receive the most design considerations [27].

The traditional approach leaves maintenance to be addressed in three ways [27]. For systems with high maintenance demand but low risk (relative to the next example), like those found in automotive engineering, reliability and maintenance databases built on years of experience and historical data are leveraged; familiar conditions and applications of parts allow designers to make decisions informed by history without significant design effort. During product development, these systems are them optimized for maintainability. In contrast, systems with high maintenance demand and high risk, e.g. military applications, similar historical data informs design decisions but a tendency to over-design exists when faced with novel situations. Though the previous two design approaches have limitations, maintainability is a consideration. The last method to address maintenance is to wait until product development begins, implement maintainability axioms and build a maintenance plan around the emerging design. Without previous experience, historical data or the resources to over-design systems this is the situation most developers of novel systems find themselves in. Maintainability axioms have proven effective but they ensure a design, once selected, follows best practices, they do not guide design decisions or assess maintainability prior to concept selection. Tjiparuro and Thompson review 22 maintainability considerations (e.g. safety, diagnosability, testability, etc.) and build a comparison matrix to assess which considerations play the most significant and encompassing roles. Through this, five design for maintainability axioms are provided [10]:

- Simplicity, in design of the system and in parts used. Simplicity impacts accessibility, modularization, identification among others.
- Parts/Components, the weights of components, their shape, surface texture, space for tool access and so on are key factors to achieve good maintainability.
- System environment, the conditions under which maintenance is to be carried out should be controlled to improve serviceability.
- Assembly/Disassembly, the sequence and combination of actions to disassemble and reassemble components should be as simple as possible.
- Identification, parts and connections should be clearly identified and adjustments facilitated.

Maintainability prediction techniques are available [6–8], but focus on known or learned data to estimate reliability, failure intervals, maintenance downtime, etc. While these are ideal for known systems with large verified data sets, they are intended primarily for maintenance strategy planning, not conceptual design work.

Despite the current techniques in design for maintainability and maintenance prediction, there is a distinct lack of maintainability assessment tools targeted at low fidelity conceptual designs. This gap exists for good reason, without component information, techniques leveraging historical data are not able to be implemented. This is further compounded for novel systems as not only is component information unknown, but a lack of design convergence means that the fundamental structure of a system is unknown. This can lead to functionally diverse concepts or prototype iterations, making it difficult to quickly and consistently apply previous understanding to new designs; which is something current methods are not equipped, or intended, to perform.

2.3 Functional Modeling

2.3.1 Overview

Functions are the elementary descriptions of *WHAT* something does. All devices perform at least one function; a wave energy converter may *Generate Electricity.* This is the function of the device, a simple description of *what* the device does, without any reference to *how* the device accomplishes it. By expanding this singular function into multiple sub-functions that make up a functional chain, we can say that a wave energy converter may *Capture Hydrodynamic Energy*, *Convert Hydrodynamic Energy* into *Electrical Energy* and *Export Electrical Energy.* This process can be repeated until the functions performed by every system – or all the way down to individual components – are described in simple abstract terms. This method of decomposing the physical architecture of a device into a series of functions is called Functional Modeling.

2.3.2 Applications

Developing a robust understanding of a proposed device in terms of function is essential to meeting the one measure of success that all designs are judged against, does the device's function satisfy the customer requirements [13, 14, 29]? This is a fundamental aspect of a functional approach to design, it focuses and centers the design process on the customer requirements [30]. The second key aspect of a functional approach is aptly described by the maxim "Form follows function" [14]. When applied to the realm of mechanical design, the maxim means that the arrangement and location of parts, the selected components, the physical processes performed, the geometry of systems, and the device as a whole follows from the simple description of the device's function. By describing the function and sub-functions of a device, the designer is creating a blueprint that will guide the development of the physical device. The idea that a functional approach provides designers with a useful understanding of a system is supported by its many successful applications.

By abstracting away from the physical, designers experience reduced component/design fixation and improved creativity. In exploring a wider breadth of the design space, an increased quantity and variety of concepts are generated, increasing the likelihood of a unique and innovative design [14, 29, 31]. Tying functions to specific customer requirements improves design focus, improving the probability of adequately satisfying customer requirements and limiting non-value added design work [29]. Methods based on functional modeling not only improve concept generation but allow for concept evaluation methods to ensure design efforts are focused on the most promising concepts. By relating the frequency with which certain components solve specific functions, O'Halloran [8] provides a method to reduce reliability uncertainty in concepts. In a similar vein, functions can serve as the basis for failure analysis of: complex systems [32–34], rapidly integrated features [35], and modes of component failure [36]. The key aspect that unites these applications of functional design is that they are available to designers prior to the specification and selection of any physical components.

2.3.3 Functional Modeling

To realize the benefits of functional modeling, two problems were necessary to address, that of a consistent functional language and a more nuances definition of a function [37]. Without a solution, both problems impact inter- and intra-coder reliability. If one designer doesn't create a consistent functional model of a system when decomposed at two different instances, the results of any application of the model would be dependent on the particulars of that instance of functional decomposition. These effects would be exacerbated by the differences in modeling of two different designers.

To address this, structured functional modeling methods and languages have been developed. Collins codified the first functional language, using 105 unique keywords and antecedent adjectives; but the scope of functional description was limited to rotary-winged aircraft [38]. Future work included the distinction between functions and flows, with the verb-object form following from this distinction [39]. Functions were restricted to verbs and flows restricted to nouns (objects), that when combined described the operation as well as what the operation was performed on. Using a simple description of a wave energy converter, the function *Capture* is performed on the flow of *Hydrodynamic Energy*. To combat the varying levels of abstraction possible to describe a single function or flow (*Hydrodynamic Energy* versus *Energy*), a hierarchy was established that allowed for consistent levels of specificity to be used [40].

Through reconciling two previous functional language efforts [41, 42], Hirtz developed the 'functional basis' [40]. The functional basis has 52 functions and 82 flows that are combined in the standard verb-object pairing to create function-flow pairs. Functions are organized into a hierarchy of three levels and flows into four levels, though level-4 will not be used for this research as they are too descriptive. Level-1 functions and flows are the most abstract terms and increase in detail in ascending levels. The 8 level-1 functions are: *Branch*, *Channel*, *Connect*, *Control Magnitude*, *Convert*, *Signal* and *Support*. The three level-1 flows are: *Material*, *Energy* and *Signal*. As *Signal* is both a function and a flow, *SignalFcn* and *SignalFlow* are used to differentiate. Table 2.2 shows the level-1 function *Control Magnitude* and flow *Energy* along with some of their level-2 and level-3 functions and flows that serve as more descriptive versions of the terms. The middle and right columns are from "Appendix A: Flow Definitions" and "Appendix B: Function Definitions" [40]. By providing definitions and examples for all terms, Hirtz made a significant progress in increasing inter-coder reliability by decreasing verbiage. This research utilizes the functional basis and the method of implementation is described in Section 3.3.2.

The second development necessary for consistent modeling was in a more nuanced view of a function. Most views on functions can be classified into what Deng describes as the two types of functions, action-functions and purpose-functions [43–45]. Those definitions that align with the term action-function are generally interested in describing the physical processes in component-agnostic terms; describing them without reference to the particular components that execute that solution. Action-functions are generally more descriptive and align more closely with functional modeling efforts taken later in the design process when there is a more clear picture of the design. The term purpose-function encapsulates the views that functions describe the physical world in component- and solution-agnostic terms; by capturing the intention of the designer and the end goal of the process. Purpose-functions are generally more abstract and are more appropriate for early design efforts when there is more uncertainty in the design.

Utilizing a functional decomposition performed later in this research (seen in Fig. 3.1 found in Section 3.3.2) the example of an aircraft rudder is shown in Fig. 2.2 to be examined to clarify the action- versus purpose-function distinction. The rudder is controlled by the pilot and swings port or starboard to induce yaw (rotation of the aircraft about the vertical axis). Only the level-2 function-flow pair is described for brevity.

The action-function of the rudder is to *Guide Gas*. As the rudder moves, the flow of air (*Gas*) past the rudder is redirected (*Guide*) port or starboard. This describes the physical operation of the part without describing the components performing the operation. However, *Guide Gas* does not describe – or does so only in the most shallow sense – the intention of the designer.

The purpose-function *Change Mechanical Energy* better captures the intention of the

Function/Flow			Definition	Example
Level-1	Level-2	Level-3		
Control Magnitude			To alter or govern the size or amplitude of a flow (*Material*, *Energy*, *Signal*).	-
	Change		To adjust the flow of *Energy*, *Signal*, or *Material* in a predetermined and fixed manner.	In a hand-held drill, a variable resistor *Changes* the *Electrical Energy* flow to the motor, this changing the speed at which the drill turns.
		Increment	To enlarge a flow in a predetermined and fixed manner.	A magnifying glass *Increments* the visual signal (i.e., the print) from a paper document.
		Decrement	To reduce a flow in a predetermined and fixed manner.	The gear train of a power screwdriver *Decrements* the flow of *Rotational Energy*.
	Stop		To cease, or prevent, the transfer of a flow (*Material*, *Energy*, *Signal*).	A reflective coating on a window *Stops* the transmission of UV radiation through a window.
		Prevent	To keep a flow from happening.	A submerged gate on a dam wall *Prevents* water from flowing to the other side.
Energy			-	-
	Hydraulic Energy		Work that results from the movement and force of a liquid, including hydrostatic forces.	Hydroelectric dams generate electricity by harnessing the *Hydraulic Energy* in the water that passes through the turbines.
		Hydraulic Pressure	The pressure field exerted by a compressed liquid.	A hydraulic jack uses the flow *Hydraulic Pressure* to lift heavy objects.
		Volumetric Flow	The movement of fluid molecules.	A water meter measures the *Volumetric Flow* of water without a significant pressure drop in the line.

Table 2.2: The level-1 function (*Control Magnitude*) and flow (*Energy*) with a few of their corresponding level-2 and level-3 subfunctions/subflows. This is not an exhaustive list of all subfunctions/subflows for *Control Magnitude* and *Energy* but serves to show the tree structure of the functional basis.

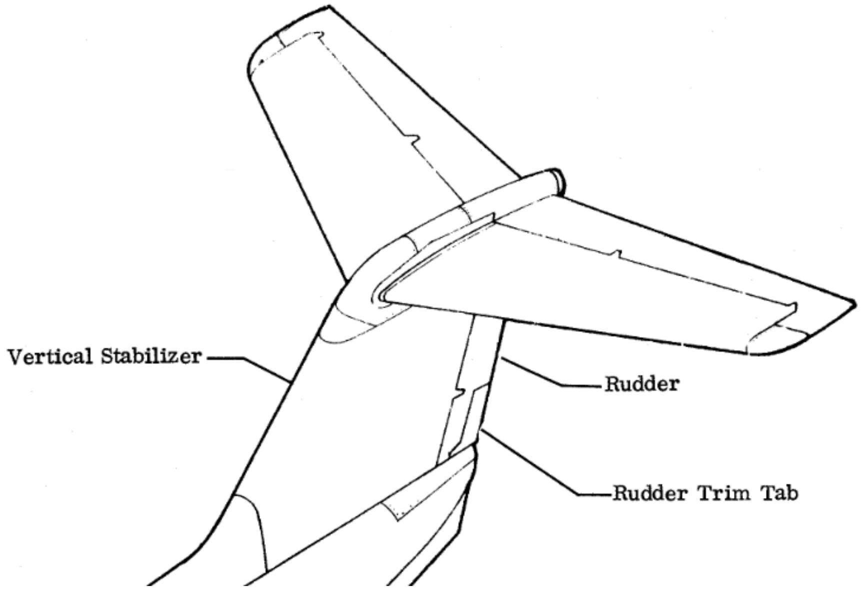


Figure 2.2: The vertical stabilizer and rudder of a civil aircraft. The vertical stabilizer is found at the rear of the fuselage on all civil aircraft. The rudder is flight control surface found on the trailing edge of the vertical stabilizer. The rudder rotates about a vertical axis, at the front of the rudder, that causes its trailing edge to swing port and starboard to induce yaw.

designer. It describes the physical process – again without reference to the components performing the operation – but additionally without reference to the solution implemented to perform the operation. During the earliest stages of design, the aircraft designer identified the need for the aircraft to exert (*Change*) a force (*Mechanical Energy*) that would result in yaw. At some future point, it was decided that a traditionally implemented rudder was the preferred manner to accomplish that function.

The reason for this distinction is thus, *Guide Gas* incorporates an inherent solution to the problem of inducing yaw, that is to redirect the flow of air. The components required to perform that operation are not specified, but they have been greatly down-selected by nature of the implied solution. The purpose-function *Change Mechanical Energy* provides no information on the solution to the problem of inducing yaw, let alone the components required to implement a solution. Though in reality, the purpose-function may down-select the types of solutions to the problem of inducing yaw; but the restriction is significantly reduced. Unfortunately, the distinction is not always well defined as either one can imply some of the other; "nevertheless, a distinction between these two functions is still necessary and important" [43].

While the purpose/action distinction may seem pedantic, it is at the core of the benefits of functional modelings during early design. In aircraft design, where significant design convergence has occurred; it is unanimous that yaw is best accomplished by a flap that runs vertically and rotates about a forward vertical axis (a rudder). The design convergence towards a rudder causes the action- and purpose-function to appear equivalent, but a difference can be seen in the fact that *Change Mechanical Energy* could be implemented by a wider variety of processes. For example, in no-atmosphere conditions, yaw can be induced by a thruster placed off-center of a spaceship's vertical axis. In novel fields where design convergence hasn't occurred, like with MREs, this distinction allows the earlier examination of a wider breadth of the design space.

Functional analysis has proven useful during early design but a connection between function and preventative maintenance intervals has yet to be explored. Furthermore, no application that relates function to a maintenance adjacent consideration (reliability, failure, etc.) has provided a simple and comprehensible design aid that is sufficiently transparent so as to build confidence with a designer who would be using it.

2.4 Machine Learning

2.4.1 Overview

Machine learning is the use of algorithms to build a model, that can process and interpret data for a useful purpose. Two aspects of machine learning make it distinct from traditional human or computer based data-analysis. Machine learning differs from traditional data-analysis because it doesn't present a static picture of the data. Instead, it interprets data and can provide predictions, insights or learn dynamically from new data [46]. The second difference is in scale. The human brain has a limited capacity to interpret the vast amounts of data available in the modern world. While computers can be limited by computational resources or complexity of the data, a computer's capacity eclipse a human's capacity. Machine learning models have proven effective at extracting valuable insights from data sets that are too large or complex for a human – regardless of expertise – to comprehend, let alone derive useful information from [46].

Within engineering design, machine learning has been heavily explored for it's capacity to support modeling complex optimization efforts, synthesize new designs, and capture implicit customer or designer knowledge [47]. Focused on early design efforts, machine learning has been implemented to provide guidance on design methods most appropriate for a situation. This highlights machine learning's capacity to leverage the documented expertise of designers for novel design situations and for less experienced designers [48]. Data-driven methods, which includes those utilizing design repositories, have proven to be an effective base of information to draw from for applications in additive manufacturing [49], operational strategies of renewable energy grids [50], and understanding large sets of customer needs [51, 52]. These methods are able not only bring known design experience into play, but allow for insights that are otherwise obscured by the size or complexity of data to be brought to the surface to guide design decisions. The rapid exploration of potential design alternatives within a broad design space has also been explored as models can explore, generate, combine, etc. far faster than human designers [53].

Despite the broad application of machine learning in engineering design, many remain complex and out of reach for designers in industry. Especially for those without the expertise, time, or company support to invest into machine learning applications.

Within the umbrella term of machine learning, there are two general categories of models: supervised and unsupervised. The differences in categories relate to what information is provided to the model and the intended output of the model. Across and within the two categories, there is no one model that works for everything. An understanding of the underlying mathematics of the model to be employed, the nuances of the data set, the end goal and the associated assessment performance metrics are required to inform the best choice of model for that particular situation.

2.4.1.1 Supervised Models

Data sets to be used with a supervised model are first split between training and testing subsets – generally 70%/30%, respectively. The algorithm is first applied to the inputs (features) of the training data and provided with the known outputs, creating a model of the data. The features of the testing data are then fed into the model, with the known outputs withheld. The model predicts the outputs of the testing data based on its experience with the training data. The predicted and known (or reference) outputs are then compared to calibrate and evaluate the model.

Supervised models can further be distinguished by the desired output of the model, classification or regression. Regression models yield a continuous numeric value. A linear regression model common in statistics or algerbra is a simple supervised regression model. Classification models make prediction on the category or class that a data point belongs to based on the input features. The spam filter utilized for emails is a supervised classification model. Based on the features of an email, the model makes a prediction if the email belongs to the "spam" or "not spam" category. Characteristics that distinguish supervised models include:

- Once a model is trained and calibrated, prediction are computationally simple
- The results are generally more accurate and reliable than unsupervised models as they are trained using known, validated data
- The models are more interpretable, allowing closer inspection of the model

Due to the nature of the data used and the desired outputs, this research uses supervised machine learning models. This section will focus primarily on the pertinent background information.

2.4.1.2 Unsupervised Models

Unsupervised machine learning models receive input features that are unlabeled, meaning the model does not know what the values of each feature in a data point represent. Additionally, no reference output values are provided, preventing any evaluation of the model. Unique characteristics of unsupervised machine learning models include:

- Unsupervised models are generally more difficult to implement successfully
- Unlabeled features of large data sets don't need to be manually labelled
- Allows users to develop an understanding of the data set that may inform further, more reliable analysis

2.4.2 Supervised Classification Models

Decision trees are the primary machine learning model utilized in this research, as such, this section will focus on them above other supervised classification models. Decision Trees are a popular machine learning model whose most significant attribute is their simplicity to explain and follow the logic of. Decision trees are represented as a tree that starts at a single node and branches out into additional nodes that can each branch out further. Each node represents a feature of the system (such as its color); depending upon the attribute of that feature (e.g., the color is red), the tree is traversed in the 'red' direction rather than, say the 'green' direction. Once a node fails to reduce information entropy (or purity) by a certain preset parameter or is manually pruned, that branch ceases to split and becomes a 'leaf' node. Each leaf node has a categorical determination for any data points following the branches down to that node.

Decision trees' other distinct attributes, apart from their ease of comprehension, are the ability to handle poor performing features, handle missing data, computational speed, and their propensity towards overfitting. Addressed in turn, feature selection is a major part of most machine learning endeavors, due to the method implemented by decision trees to select the feature most appropriate for a given node, they will almost always select the best option. The method is based on entropy and gain, where *Entropy* $[H(S)]$ is a measure of the purity of a node. Purity refers to how many data points at a node are correct versus incorrect.

$$H(S) = -p_{+} * log_2(p_{+}) - p_{-} * log_2(p_{-})$$

where p_{+} and p_{+} are data points at that node correctly sorted and incorrectly sorted, respectively. *Gain* is a measure of the relevance of a feature, measured from 0 (no use) to 1 (most useful).

$$Gain = 1 - H(S_f)$$

where $H(S_f)$ is the *entropy* of each possible feature that node can assume. Because each node is selected based on the best possible increase in purity, poor performing features are never used and can remain in the data set without fear of hurting performance. Decision trees can handle missing data when presented with a null value for a feature by stopping the data point at that node and classifying it there. Once a decision tree is calibrated, predictions are made based on a series of binary choices, allowing for rapid predictions on large data sets.

Lastly, decision tree are prone to overfitting since they are sensitive to initial conditions and the particulars of a data set. To combat this, they must be pruned manually or algorithmically to avoid overfitting. Complexity parameter is a computed term that can be used to determined the optimal tree size. The implementation varies but generally, complexity parameter corresponds to the risk of missclassification at that size of a decision tree based on cross-validation.

		Reference		
		Category 1	**Category 2**	**Category 3**
Predicted	**Category 1**	1	2	3
	Category 2	4	5	6
	Category 3	7	8	9

Table 2.3: Example confusion matrix comparing the known category of 45 data points versus the predicted category of those data points.

2.4.3 Assessing Machine Learning Models

As supervised methods utilize known outputs in constructing the model and are then compared against 'new' reference outputs, the metrics used to assess them are based on this comparison. Metrics discussed are the confusion matrix, accuracy, positive predictive value, negative predictive value, sensitivity, and specificity.

A confusion matrix is a visual representation of the reference versus predicted categories. Table 2.3 shows the results from a multi-class (3 categories) prediction. The matrix displays the **Reference** categories along each column and the model **Predicted** categories along each row. Interpreting a confusion matrix is done column by column from left to right. The first column shows the quantity of data points that are known to be in 'Category 1'. The top cell in the column represents the number of data points (1) known to be in 'Category 1' that were predicted to be in 'Category 1'. The cell below, shows that 4 data points were known to be in 'Category 1' but were predicted to be in 'Category 2'. The final cell in the first column shows that 7 'Category 1' data points wrongly predicted as 'Category 3'. This trend continues with the remaining columns. The summation of values in a column gives the total number of data points known to be in that category. The summation of a row gives the total number of data points predicted to be in that category.

The grey boxes that run diagonally from top-left to bottom-right show the data points that were correctly predicted, these values are known as true positives (TP). The related terms: true negative (TN), false negative (FN) and false postive (FP) describe variations of correctly/incorrectly predicted positive or negative results. Because these values are dependent on a reference category, they will be addressed in reference to 'Category 2'. The TP result for 'Category 2' is the center cell containing the number 5. The cells above (2) and below (8) are FN results, they are known to be in 'Category 2' but were falsely predicted to be in ‘Category 1’ for the cell containing 2 and ‘Category 3’ for the cell containing 8. The cells left (4) and right (6) of the TP cell are the FP results. FP results are predictions that incorrectly placed 4 data points into ‘Category 2’ which should have been in ‘Category 1’ and 6 data points into ‘Category 2’ which should have been in ‘Category 3’. TN results are all cell not orthogonal to the TP results, these are correct negative predictions.

The terms TP, TN, FP and FN are used to define the remaining metrics, listed below with their equation and a brief description.

Accuracy is the most commonly used metric; it describes the total number of correct predictions over the total number of predictions. Accuracy is the only metric described that is calculated for the model as a whole as opposed to on a per category basis.

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Positive Predictive Value (PPV), also known as precision, gives the probability that a prediction into a given category is correct.

$$\frac{TP}{TP + FP}$$

Negative Predictive Value (NPV) gives the probability that a negative prediction for a category (that a result is NOT in that category) is correct. NPV gives no information on the correct category, only that it is not in that category.

$$\frac{TN}{TN + FN}$$

Sensitivity, also known as recall, gives the probability that all of data points of a given category are correctly predicted.

$$\frac{TP}{TP + FN}$$

Specificity gives the probability that all of data points NOT in a given category are NOT predicted into that given category. Specificity does not say if the data points are predicted into their own correct category, just that they are not falsely predicted into the reference category.

$$\frac{TN}{TN + FP}$$

Macro-average, henceforth referred to as average, gives the mean value of a class specific metric across all instances of that metric. Below gives the macro-average PPV for a multi-class (3 categories) model.

$$\frac{PPV_1 + PPV_2 + PPV_3}{3}$$

Chapter 3: Methods

Described in this chapter is a reference of functional modeling language followed by the four major phases of this research. The methods used to mine the relevant maintenance data are presented first. Then the functional modeling process used to derive functions from the mined maintenance data is described. This is followed by the machine learning techniques used to build the function-maintenance model. Lastly, the three methods used to evaluate the constructed model are discussed.

3.1 Functional Modeling Jargon

Functional modeling relies heavily on semantics in order to convey complex physical processes as abstract concepts; this means that the clear and consistent use of functional language is required to perform – and to convey – this research. In support of this, common terms and clarifications on functional information are provided below along with an example of it's usage.

Function is used in two manners: as the general description of *WHAT* a part does and as a technical description of the part using the functional language of Hirtz et al. [40] comprising the verb portion of the verb-object format.

Function (technical descriptor) is a capitalized verb(s), written in italics. "[The impact bar] *Distributes* the force of a crash..."

Flow is the object portion of the verb-object format and is a capitalized noun(s), written in italics. "All of these parts serve to restrict the flow of some *Material*, whether it be *Particulate* for filters..."

Function-flow pair is a unique combination of a function (technical descriptor) and a flow, written in italics following the verb-object format. "the function of the pitot-static system is to *Sense Pneumatic Energy*." *Sense* is the function (technical descriptor) and *Pneumatic Energy* is the flow, together they make the function-flow pair.

Level-1, Level-2, and Level-3 are the levels of increasing detail that all functions and flows can be described at. Referencing Table 2.2, the level-1 function of *Control Magnitude* can be further described by the level-2 functions of *Change* or *Stop*. Similarly, the level-

2 function of *Change* can be further described by the level-3 functions of *Increment* or *Decrement*. A part can be described at all three levels using the following convention "the part's function is *Control Magnitude/Change/Increment*." In the same manner, a part's flow can be described at all three levels by "the part's flow is *Energy/Hydraulic Energy/Hydraulic Pressure*." Not all functions/flows have corresponding verbiage for all three levels. In these cases the more detailed descriptor is repeated follow by the number of the next level. For example, the function *Convert* does not have a corresponding level-2 or level-3 verb. Written in the form described previously, the three levels of this function are *Convert/Convert2/Convert3*.

The three levels of a part's function and flow are often combined to fully describe a its function and flow pair, "...the rudder assembly that performs the function and flow *Control Magnitude/Change/Condition* and *Energy/Mechanical Energy/Rotational Energy*." For the sake of brevity, often a single level of both function and flow are described rather than all three levels of both function and flow. The previous sentence about the rudder assembly could be rewritten as "...the rudder assembly performs the function-flow pair of *Change Mechanical Energy*." Though three function-flow pairs could be created based on each level of function and flow (i.e. level-1 = *Control Magnitude Energy*, level-2 = *Change Mechanical Energy*, and level-3 = *Condition Rotational Energy*), this is still considered to be one function-flow pair as they describe the same thing, just at different levels of detail.

Signal(Function) and *Signal*(Flow) are a function and flow that both use the word "Signal." *Signal*, the function, is the act of conveying information where *Signal*, the flow, is the information that is conveyed. To differentiate between the two *SignalFcn* is used to indicate the function and *SignalFlow* is used to indicate the flow.

FcnL3*Decrement*, FlowL2*ME*, and FFL1*CMM* are example, short form descriptors that are primarily used while discussing the machine learning model produced in this work. FcnL3*Decrement* can be expanded as "The level-3 function *Decrement*", FlowL2*ME* as "The level-2 flow *Mechanical Energy*, and FFL1*CMM* as "The level-1 function-flow pair of *Control Magnitude Material*." This short form is used as it conveys three pieces of information in a succinct way. Whether the information presented is a function, flow, or function-flow pair; the level of detail used; and the actual function, flow, or function-flow pair information.

Abbreviations are used mostly in tables or in the short form descriptor, described above. Common abbreviations are listed here for ease of reference.

- *Change Magnitude* = *Change Mag.* = *CM*

- *Energy* = *E*
- *Mechanical Energy* = *Mech. Energy* = *ME*
- *Translational Energy* = *Transl. Energy* = *TE*
- *Rotational Energy* = *Rot. Energy* = *RE*
- *Hydraulic Energy* = *Hyd. Energy* = *HydE*
- *Material* = *M*

3.2 Data Mining

In order to identify any function-maintenance relationships and build a model to describe it, high-quality maintenance data was required. It needed to be thorough and contain sufficient detail to extract the function and flow of the component or system of interest. Additionally, the maintenance data needed to be validated and and come from a product where maintenance and reliability was a key factor in the design and operation of the product. One such long field-life product was the Learjet 25 series, a commercial business-jet. With production ceasing in 1982, the newest 25 series Learjets have been in service for more than 40 years. Between the significant operational experience, strict regulatory requirements on O&M and the need to balance maintainability versus performance, the civil aviation sector serves as an exemplary wealth of knowledge and experience from which to draw upon for this research.

3.2.1 Maintenance Documentation

The author had revision number 23 of the Gates Learjet 25b/c/d/f maintenance manual (MM) [54] as well as the maintenance documentation for the CJ610 Turbojet Engine, the standard engine on all Learjet 25 series. The standard engine is revision 11 of the CJ610 Turbojet Engines MM [55] and the accompanying Illustrated Parts Catalog revision number 22 [56]. Between these three documents, all maintenance tasks and requirements necessary to comply with Federal Aviation Administration's (FAA) safety and maintenance regulations were available, including references and documentation required to develop a robust understanding of the intent and impacts of said maintenance task or requirement. Table 3.1 shows the primary chapters of the maintenance documents utilized in this research. Other chapters, from both manuals, as well as the CJ610's illustrated parts catalog were used as references to contextualize component-system interactions and assess functions.

Document	Chapter	Title	Contents used
Learjet MM	4	Airworthiness Limitations	"Life limited" items
Learjet MM	5	Time Limits and Maintenance Checks	General and special inspection, replacement and overhaul schedule
Learjet MM	12	Servicing	System and component services
CJ610 MM	72	Engine General	Services, inspection, checks, cleaning and preservation

Table 3.1: Chapters from which maintenance data was primarily mined

3.2.2 Data Collection

To establish a link between functionality and preventative maintenance considerations, the essential data of function, flow, and maintenance interval were identified as the primary target of this data mining effort. The remaining data served to develop a complete understanding of those three elements or for organizational purposes.

All mined and synthesized data were captured in a spreadsheet, an excerpt of which is shown in Table 3.2 and Table 3.3. Data captured verbatim from the maintenance documents, shown in Table 3.2, include:

- Part #: A number assigned to each data point, used for organizational purposes.
- Chapter, Section, and Subject: The Chapter, Section, and Subject of the maintenance action.
- Additional Details: Any additional information provided by the MM. The Learjet MM gives more refined location information, the major system addressed, the type of inspection, or conditions required for the maintenance action. The CJ610 MM gives the broader maintenance task.
- Maintenance Task: The maintenance task as described in the respective MM.
- Frequency: The interval and measure of frequency given for that maintenance action. Written as Freq. in the table.

Synthesized data, shown in Table 3.3, consist of:

- Part #: A number assigned to each data point, used for organizational purposes.
- Part of Concern: The part that the maintenance task seeks to test, verify or maintain the functionality of. A single maintenance task may spawn multiple parts of concern,

each given a new part #.

- Part Type: An unofficial, broad classification of the part used primarily for quick-reference.
- Component versus Assembly: Whether the part of concern is a component or an assembly. Written as Comp. vs Assy. in the table.
- Function: The level-1, 2, and 3 function performed by the part of concern, identified using the techniques described in Section 3.3.2. There are four function columns in the master spreadsheet to accommodate parts that perform multiple functions, only two function columns are shown in Table 3.3 for legibility.
- Flow: The level-1, 2, and 3 flow of material, energy, or signal through the part of concern, identified using the techniques described in Section 3.3.2. There are four flow columns in the master spreadsheet to accommodate parts with multiple flows, only two flow columns are shown in Table 3.3 for legibility.
- Function-Flow Count: The total number of unique function-flow pairs embodied by the part of concern. As each function and flow has three levels that are describing the same function/flow, just at increasing levels of detail, the three levels of function-flow pairs are considered one unique pair. Written as FF count in the table.
- Frequency (Hours): The maintenance frequency used to build the model after being converted to flight hours, if it did not start in flight hours and if possible to do so. Written as Freq. (Hours) in the table.
- Notes: Additional notes added to clarify information, used only as needed.

As a majority of maintenance actions fell under a regularly schedule inspection or service (e.g. 200/400 hour inspection, 600 landing/12 year inspection, 24 month inspection, life limit-cycles, etc.) that used a variety of measures of frequency, a common unit of measurement was required. Flight hours was the most commonly used measure, it was chosen as the standard unit for maintenance intervals. For maintenance actions that fell into inspections that had both flight hours as well as another measure of frequency (e.g. 7200 hour/12 year inspection), the flight hours interval was used as the recorded frequency. Maintenance actions that fell into inspections that did not use flight hours as a measure of frequency (e.g. 3000 landing inspection, 6000 landing/12 year inspection, etc.) had to be converted into hours.

Table 3.4 is an excerpt from the “Relative Aircraft Age and Utilization Table” from the *Special Inspection* subject of the *Time Limits and Maintenance Checks* chapter of the Learjet MM. Table 3.4 sorts aircraft into three groups based on age of the aircraft, which

Part #	Chapter Section Subject	Additional Details	Maintenance Task	Freq.
6	Servicing Lubrication Landing Gear	Main gear	Lubricate piston rod felt wiper with hydraulic fluid (MIL-H-5606) with oil can	600 hour
26	Servicing Lubrication Engine Accessory	N/A	Engine accessory lubrication consists of lubricating the EPR connections, starter-generator, hydraulic pump, and throttle cable ball joint connectors. Lube type CSP, applied by hand	On installation and as required
126	Time Limits/Maintenance Checks Inspection Program 200/400 Hour Inspection	Tailcone and pylon sections	Throttle cables for condition, proper routing, and security of clamps	200 hour
295	Time Limits/Maintenance Checks Special Inspections 3,000 Landings Inspection	X-ray and eddy current inspection	Forward and aft corner of upper main entry door - frame/skin splice at stringer 6	3,000 landings
864	Time Limits/Maintenance Checks Special Inspections 6,000 Landing/12 Year Inspection	Landing gear	Completely disassemble main and nose gear actuators (Refer to Chapter 32). Conduct visual, dye penetrant, and magnaflux inspection of all components as appropriate. Determine extent of wear by making required dimensional checks	6,000 landing/12 year
919	Time Limits/Maintenance Checks Special Inspections 7,200 Hour/12 Year Inspection	Flight controls	With all moveable flight control surfaces removed, (flaps, ailerons, rudder, elevators, and spoilers), inspect all bearings, bushings, and rollers for security, roughness, seizure, rust or corrosion, and wear. Preventative maintenance replacement at this time is recommended. Applies to parent structure element as well as those on the moveable surfaces	7,200 hour/12 year
1,999	Airworthiness Limitations Description and Operation Replacement Items	Landing gear	Main gear strut (with cylinder assembly P/N 2341101) and actuator, hard surface landings only	9,000 landing

Table 3.2: Example of data mined from Learjet and CJ610 maintenance manuals.

Part #	Part of Concern	Part Type	Comp. vs Assy.	Function 1	Flow 1	Function 2	Flow 2	FF Count	Freq. (Hours)	Notes
6	Piston rod felt wiper	Wiper seal	Comp.	*Change Mag.* *Stop* *Prevent*	*Material* *Solid* *Particulate*	N/A	N/A	2	600	None
26	EPR connection to engine	Connector	Comp.	*Connect* *Couple* *Link*	*Material* *Solid* *Object*	*Support* *Stabilize* *Stabilize3*	*Material* *Gas* *Gas3*	3	N/A	Not used for model due to frequency given
126	Elevator cables	Cable	Comp.	*Channel* *Transfer* *Transmit*	*Energy* *Mech. Energy* *Transl. Energy*	N/A	N/A	1	200	N/A
295	Fwd corner of upper main entry door skin splice at stringer 6	Exterior	Comp.	*Channel* *Guide* *Translate*	*Material* *Gas* *Gas3*	N/A	N/A	1	4,500	N/A
864	Main landing gear actuator subsystem	Exterior	Assy.	*Convert* *Convert2* *Convert3*	*Energy* *Hyd. Energy* *Pressure*	*Convert* *Convert2* *Convert3*	*Energy* *Mech. Energy* *Transl. Energy*	2	5,760	12 years used as it is more restrictive
1,999	Main gear strut	Strut	Comp.	*Change Mag.* *Change* *Decrement*	*Energy* *Mech. Energy* *Transl. Energy*	*Support* *Stabilize* *Stabilize3*	*Material* *Solid* *Object*	2	13,500	N/A

Table 3.3: Data synthesized from parts described in Table 3.2. Only two sets of function-flow pairs are shown for brevity.

#	Inspection Parameter	Age / Utilization	Older Low	Older Mid	Older High	Moderate Low	Moderate Mid	Moderate High	Newer Low	Newer Mid	Newer High
1	Present Aircraft Age	(Years)	18	16	14	12	10	8	6	4	2
2	Present Total Time	(Flt. Hrs.)	4320	7680	10080	2880	4800	5760	1440	1920	1440
3	Avg. Monthly Use	(Flt. Hrs.)	20	40	60	20	40	60	20	40	60
4	Avg. Flight Length	(Flt. Hrs.)	1.0	1.5	2.0	1.0	1.5	2.0	1.0	1.5	2.0
5	Present Total Landings	(Approx.)	4320	5120	5040	2880	3200	2880	1440	1280	720

Table 3.4: Excerpt from the "Relative Aircraft Age and Utilization Table" found in the Learjet MM. All maintenance frequency conversions were completed using expected utilization of a moderate age, mid utilization aircraft.

are further subdivided by utilization. An age independent, mid-utilization aircraft was selected as the standard case and all frequency measures were converted using this case. The reason for choosing a mid-utilization aircraft was based on the assumption that the utilization rates of these aircraft would follow a normal distribution, with the majority of aircraft falling into the mid-utilization group, providing the most general depiction of the maintenance requirements. The age of the aircraft did not need to be considered as within each utilization level, the conversions between measures of frequency were consistent. Table 3.5 shows a sample of all conversions in their original units of frequency and in flight hours.

Maintenance actions that fell into inspections that didn't use flight hours (e.g. 12 month inspection, 3000 landing inspection, 6000 landing/12 year inspection, etc.), were converted using Table 3.4. Whenever a maintenance interval was given with two different measures of frequency (i.e. 6000 landing/12 year inspection) that when converted, did not yield the same flight hours, the more frequent interval was used. This is in line with common maintenance practices of "...or whichever comes first." The only maintenance actions for the CJ610 measured in a frequency other than flight hours was measured in engine cycles. An engine cycle is defined as, "...a flight consisting of a start, takeoff, landing and shutdown" [55]. Based on Table 3.4, an engine cycle would most appropriately take on the value of an average flight length, 1.5 flight hours.

Some maintenance actions were incident based as opposed to time based (e.g. "following overweight landings" and "landings on non-paved runways"). As these incident based maintenance actions could not be related back to a measure of time they were not utilized

Original frequency	**Flight hours**
Month	40
Year	480
Landing	1.5
Engine cycle	1.5

Table 3.5: Converted values of a month, year, landing and engine cycle into flight hours for a standard case aircraft.

in the model.

3.3 Functional Modeling

3.3.1 Overview

As discussed in Section 2.3, functional modeling can suffer from subjective interpretation. As the functional information derived in this section was to serve as the attributes that the machine learning algorithm would learn on, a consistent functional decomposition method was vital. For example, whenever a data point was described using the feature "the part performs the function '*Change Magnitude* = Yes' ", it had to convey the same information about the elementary functions of the part.

To improve consistency throughout the functional decomposition process, three guidelines were established that were continuously reassessed to ensure they were consistently applicable and captured the functions of the part of concern. Whenever the modeling techniques were adjusted, the location in the master spreadsheet (Table 3.2 and Table 3.3) was noted and all previous decompositions were reassessed using the adjusted technique. The three guidelines consisted of: a set functional language, a clear definition of a function, and a consistent decomposition process.

The functional basis was used as the modeling language. The two major reasons for this choice were its common usage and clarity. The functional basis as described by Hirtz et al. [40] is one of the most cited works in the field of functional modeling. This is not a comment on the techniques or justifications used, but on the increased likelihood that any particular design engineer would be familiar with the techniques and justifications of this particular functional language.

In regards to clarity, the functional basis is significant amongst functional languages due to the authors' reconciliation of multiple previous efforts and the reference material included in their research. Two such references were "Appendix A: Flow Definitions" and "Appendix B: Function Definitions" [40]. Table 2.2 shows a few sample functions/flows, their level of specificity, their subfunctions/subflows, along with the accompanied text found in their respective appendix. As each function and flow has a definition and example, I was able to refer to the appendix to apply a consistent judgement.

Reiterating from Section 2.3.3, a part's function can be described through two different lenses. An action-function describes the physical operation executed by the part, whereas a purpose-function describes the intention of the designer that the part is embodying. To maintain a consistent approach to functional decomposition, I chose to decompose all parts to their purpose-function.

Though mining data from detailed maintenance manuals allowed for the descriptive action-function, the more abstract purpose-function better coincides with the intent of this research: as a early design technique. During the design process, a designer will first identify the purpose-functions that need to be performed. At a future point in the design process, the designer will refine the abstract purpose-functions into more detailed action-functions (if the early design process is skipped, as is often the case, there may not ever be a distinction made between purpose- and action-functions).

3.3.2 Functional Decomposition

Functional decomposition began once the data from Table 3.2 had been documented. Fig. 3.1 shows the decomposition process for two parts, the rudder trim tab and the pitot-static system. The first row shows the text as it appears in the maintenance manual. The second row shows the specific part of concern for the data point and the portion of the overall maintenance task that pertains to the part of concern. The Contextualize part of concern as needed row includes the text and image references found in the manual that, along with the maintenance task, helped to develop a thorough understanding of the part as it is used in the specific context. *Marks' Standard Handbook for Mechanical Engineers*[57] was referenced as needed. Contextualizing the part was not a formal process, once I felt confident in my understanding of the part, I continued. With a clear picture of the part and the related maintenance task, I determined if the part

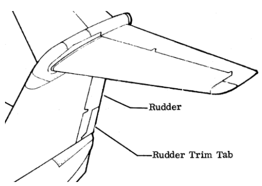
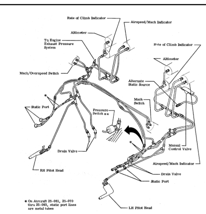
	6,000 Landing/12-Year Inspection Rudder trim tab	24-Month Inspection Pitot-Static System
Maintenance task described in manual	"Visually inspect all moveable surface assemblies for corrosion, fatigue cracks, wear, evidence of interference and general condition. Inspect all balance weight locations for evidence of vibration, fastener looseness and general security. On the aileron and ruder assemblies, inspect the trim tab installations for general condition and security. Noting any hinge or push rod looseness that may contribute to excessive tab 'free play.' "	"Perform pitot-static system leak check."
Identify part of concern and relevant maintenance action	Rudder trim tab "...inspect the [rudder] trim tab installations for general condition and security, noting any hinge or push rod loosensess that may contribute to excessive tab 'free play.' "	Pitot-static system "...pitot-static system leak check."
Contextualize part of concern as needed	 "Rudder trim tab provides yaw trim for the airplane. Rudder trim is obtained through the Yaw Trim Switch on the cockpit center pedestal. A rudder trim indicator is located on the cockpit center pedestal. A rudder trim tab system is an integral part of the rudder and further increases directional stability of the aircraft."	 "The flight environment data system consists of the pitot and static system which senses and supplies air pressure to the air data instruments."
Assess component vs. assembly and determine action-function	Component. The rudder trim tab can't be disassembled into consituent parts. *Guide Gas* As the trim tab is adjusted, it *Guides* the flow of air (*Gas*) along its surface.	Assembly. Individual parts within the pitot-static system can be replaced. *Import Gas* *Convert Pneumatic to Hydraulic Energy* *Sense Hydraulic Energy* The pitot and static tubes *Import* dynamic and static air (*Gas*) that each exert (*Converts*) pressure (*Pneumatic Energy*) on a liquid. The pressure (*Hydraulic Energy*) of the 2 liquids are measured (*Sensed*) relative to one another to determine the air speed, elevation, etc.
Abstract action-function into the purpose-function	*Change Mechanical Energy* The purpose of *Guiding Gas* along the surface of the rudder trim tab is to elecit a *Change* in the force (*Mechancial Energy*) exerted on the flight control surface. This force causes the plane to yaw.	*Sense Pneumatic Energy* The purpose of the pitot-static system as a whole is to measure (*Sense*) air pressure (*Pneumatic Energy*).

Figure 3.1: Functional decomposition example of the Rudder Trim Tab and Pitot-Static System that ends with the functional descriptions of *Change Mechanical Energy* and *Sense Pneumatic Energy*, respectively.

was a component or an assembly. The simultaneous step, the Determine action-function row, served as a thought exercise for the final step. "Appendix A: Flow Definitions" and "Appendix B: Function Definitions" [40] were referenced and the function-flow pair was determined using the action-function lense. Then I used the context acquired, the description of the physical operation of the part, and self-questioning to make a judgement on the designer's intent (purpose-function) behind the part. By posing the below questions (regarding the pitot-static system), the purpose-function was more easily illuminated.

- Is *Import Gas*, *Convert Pneumatic to Hydraulic Energy* and *Sense Hydraulic Energy* a means to an end? Or is it the end goal?

- If these functions are a means to an end, what is the end goal?
- What is the design problem that these functions solve?
- Can that problem be generalized or solved in another manner?

In answering these questions, I determined the purpose-function of the pitot-static system to be *Sense Pneumatic Energy*. By first confirming to myself what the physical operation of the part was, I was more easily able to make and justify the distinction between the action- and purpose-function.

This process was performed until the purpose-functions of all parts were identified. For approximately 20% of parts that served multiple uses, the functions and flows were recorded and the multiple embodied functions were taken into account through the FF count column in Table 3.3. This column served to capture the possibility that the quantity or combination of functions/flows embodied in a part influences its maintenance intervals.

3.4 Machine Learning

3.4.1 Overview

The first step in constructing a function-maintenance model was the choice to use a supervised or an unsupervised algorithm. After considering the strengths and use cases of each, I determined that a supervised algorithm was the appropriate choice for the following reasons:

- A supervised model can be trained using the known output values. Unsupervised models are not trained using known output values, loosing a major benefit of this data set as models incorporating known output values are consistently more accurate.
- The data set can be partitioned into a training and testing set to evaluate the quality of a supervised learning model, unsupervised models have no such evaluation method.
- Due to the nature of the mined data, the features were already known and labelled. This bypassed the burdensome requirement of identifying and labelling the data that are necessary to use a supervised algorithm.

After deciding to use a supervised algorithm, further down-selecting by type of output was still necessary. Supervised algorithms are either regression – outputs are continuous values – or categorical – outputs are categories. A regression model would output a continuous range of MTBM values based on the range of values from the training data.

Any future predictions would result in outputs within the 1 to 30,000 hours range, making the model useless for systems with possible MTBM intervals either measured in a different unit (i.e. cycles) or with a different range of values. Additionally, considering the early design intent of the model, a prediction specified down to a single digit is unlikely to be accurate to that level and would make comparing prediction results more cumbersome. A categorical model was selected as predictions given within a bucket (category) are more appropriate for the uncertain nature of early design and as well as easier to comprehend the differences in maintenance demands of a large set of predictions.

Lastly, prior to implementing any type of algorithm, the mined/synthesized data needed to be prepared by establishing a consistent set of features that all parts were to be assessed on.

3.4.2 Data Preparation

The MTBM values needed to be sorted in buckets to allow for categorical predictions. The buckets needed to be generalizable so as to accommodate any possible measure and value of maintenance frequency. A caveat was that the buckets needed to be grounded in a comprehensible manner so as to not seem arbitrary and difficult to grasp for a designer.

Different techniques were used to categorize the MTBM intervals into coherent groups, including: k-means clustering, jenks natural breaks optimization, and various mathematical transformations. K-means clustering (an unsupervised machine learning algorithm) seeks to find k number of clusters such that inter-cluster variance is maximized while intra-cluster variance is minimized. As k-means clustering performance suffers in lower dimensions, the 1D array of MTBM values was not appropriate for this method. Jenks natural breaks optimization attempts to identify the optimal number of breaks, and their location, in order to reduce intra-class variation. Though the jenks natural breaks method was successful in determining the optimal number of breaks (categories) and their location, these categories were only optimal for the particular distribution of MTBM values of the training data.

Any mathematical transformations used to identify optimal categories as well as the two methods listed above resulted in buckets that were non-generalizable or arbitrary. Non-generalizable because the resultant categories were fit to the particular distribution of MTBM values of the Learjet. No guarantee could be given that the maintenance intervals of another system would fit into the buckets created around a specific distribution of values. Arbitrary because ‘Bucket 1’ to ‘Bucket 2’ versus ‘Bucket 3’ to ‘Bucket 4’ had no consistent relationship or real world meaning. If two parts were predicted to have a difference in maintenance frequency of one bucket, no information could be gained from that other than ‘a difference of one bucket’. How far apart are the centers of those particular buckets? What is the range of those two particular buckets?

In the pursuit of a generalizable category system, the buckets were constructed to be relative to one another. While this removed the possibility of any absolute assessment of a single component or assembly, it allowed assessments to be made between parts within the same system, regardless of the particular distribution of that system’s maintence intervals. This addressed the issue of the wide range and measurements of possible maintenance intervals. The training data contained MTBM values ranging from 1 to 30,000 flight hours, whereas another possible system may measure maintenance intervals in cycles and have a range in the hundreds of thousands, if not millions of cycles. By creating buckets that are measured relative to each other, maintenance predictions are given in dimensionless terms that maintain the integrity of predictions regardless of the measurement and range of a system’s maintenance interval. If two parts had a relative difference in maintenance frequency of one bucket, that difference remains consistent regardless of which two buckets or the range and measurement of MTBM for that system.

The basis chosen to organize relative maintenance demand was orders of magnitude (OOM), specifically half-orders of magnitude. OOMs are defined by the equation

$$n \sim N = a * 10^b,$$

where b is a value, known as the OOM, that along with a – the initial value – results in an N that that is closest to the true value, n. Traditionally, the OOM of a value is restricted to integers. Half-OOMs allow the value of b to include multiples of 0.5. As exponent multiples of 0.5 equate to taking the square root of a number – in this case always 10 raised to some exponent – a half-OOM gives the equation

$$n \sim N = a * 10^{\frac{b}{2}} = a * (\sqrt{10})^b = a * 3.16^b,$$

or approximately a three-fold increase. Half-OOMs provide a convenient way to break down the significant range possible in a full OOM difference. Additionally, they are intuitive to understand and apply to the real world [58]. When looking at maintenance intervals, once a month and once a year are about an OOM difference, but there is clearly a wide range of

OOM Bucket	**Bucket Range**			**# of Parts in Bucket**
	Lower	**Center**	**Upper**	
0	0	1	6	15
1	7	10	55	7
2	56	100	208	172
2.5	209	316	658	405
3	659	1,000	2,081	193
3.5	2,082	3,162	6,581	591
4	6,582	10,000	20,811	610
4.5	20,812	31,623	65,811	2

Table 3.6: Half order of magnitude buckets used for the training data. The three columns under the heading Bucket Range are maintenance intervals, measured in flight hours. All flight hour values are rounded to the nearest integer.

possible intervals that are not appropriately described by monthly or annually. By applying half-OOMs to familiar measures of time, we can capture the trend of: weekly to monthly, monthly to quarterly, and quarterly to annually; with each being ∼3.16 times more frequent than the previous. As all predictions from the model will give results that are relative to one another, this breakdown lends itself well to contextualizing maintenance intervals into familiar units of time. Using buckets that are easy to grasp and mentally manipulate comes with the bonus of a reduced mental load on the designer, a known problem with design aids that can limit or prevent their usage.

Table 3.6 shows the breakdown of OOM buckets, the range and center of each bucket as well as the total number of parts sorted into each bucket. The 0.5 and 1.5 OOM buckets were not used as they over-crowded the sparsely populated 0 to 50 hours intervals and led to extremely restrictive ranges. As discussed in Section 2.4.2, supervised classification models base their predictions on their exposure to a sufficiently large and diverse training data set. Table 3.6 shows that of the 1995 unique data points, a total of 24 parts required maintenance in the 0, 1, and 4.5 OOM buckets. As the model has minimal training on parts in these buckets – therefore minimal exposure to the features that indicate a part should be in these buckets – it is expected that the model would rarely, if ever, make a prediction

Part #	OOM Bucket	Comp. vs Assy.	FF Count	FcnL1 *Channel*	FcnL1 *Connect*	FcnL1 *Convert*	FlowL1 *Material*	FlowL1 *Energy*	FlowL1 *SignalFlow*	FFL1 *Channel Material*	FFL1 *Convert Energy*	FFL1 *Connect Material*
86	2.5	Comp.	3	1	0	2	1	2	0	1	2	0
376	4	Comp.	2	1	1	0	2	0	0	1	0	1
566	4	Assy.	2	0	0	2	0	2	0	0	2	0

Table 3.7: Data formatted to train the model. As there are 357 unique features, only 11 are shown. Component or assembly, function-flow count, three level-1 functions, three level-1 flows, and three level-1 function-flow pairs.

into them. This leaves five buckets with sufficient training data to potentially allow for predictions. These buckets span two OOMs, for the training data this covers maintenance intervals from 56 to 20,811 flight hours.

After all MTBM values were converted into one of 8 buckets, the remaining data – as formatted in Table 3.3 – was addressed. While mining data, I recorded a part's function in the Function 1 column then in ascending column order for parts with multiple functions. As most parts had only one function, the latter function columns had few entries. 14 parts served four functions, as a result the Function 4 column only has 14 entries. Each column is considered a unique feature, therefore the contents of feature Function 4 would be disregarded when building the model as it doesn't contain enough information for any predictive value. The algorithm does not capture that the Function 1, Function 2, etc. features are the same feature and should be judged the same. The same applies for all flow and function-flow pair features. To address this, each possible function, flow, and function-flow pair was converted from the value of a feature to the feature itself.

Table 3.7 shows an excerpt of the converted data, where each possible function, flow, or function-flow pair is a feature. The value of the feature represents the number of occurrences of that function, flow, or function-flow pair in that part. Table 3.7 shows Part 86, a component in the 2.5 OOM Bucket (209 to 658 flight hours) that performs three functions. The level-1 function of *Channel* once and *Convert* twice on the flows *Material* once and *Energy* twice; resulting in one occurrence of the function-flow pair of *Channel Material* and two occurrences of *Convert Energy*. Table 3.7 only shows level-1 functions, flows, and function-flow pairs but each of these has a corresponding level-2 and level-3 counterpart that is not shown for legibility.

Every data point has 357 features, that inform the algorithm and are the basis of the model's predictions. Of those features, only the component or assembly feature is categorical, with the two possible responses found in the name of the feature. The continuous features, with values ranging from 0 to 4 based on the number of occurrences in the part, are:

- 1 function-flow count
- 8 level-1 functions
- 19 level-2 functions
- 29 level-3 functions
- 3 level-1 flows
- 14 level-2 flows
- 26 level-2 flows
- 20 level-1 function-flow pairs
- 102 level-2 function-flow pairs
- 134 level-3 function-flow pairs

3.4.3 Training the Model

Various statistical and machine learning methods were investigated for their potential to illuminate insights into possible function-maintenance relationships. Techniques included whisker plots, k-nearest neighbors, and neural networks. Exploration of whisker plots was halted as they provided a distribution of maintenance intervals by feature but had no predictive capability or externally valid insights. K-nearest neighbors proved to be an accessible machine learning algorithm but struggles computationally and with over-fitting when presented with a large numbers of features. Other supervised classification methods proved equally accessible while allowing easier testing with resultant models that were less prone to over-fitting. Though neural networks could perform well given the data set, their black-box nature runs against the intent of the model, to inform and aid designers. A designer is unlikely to place confidence in the model if they are unable to investigate the inner workings and processes of the model itself.

The two machine learning models that were explored extensively using R, a programming language commonly used for statistical analysis, were decision trees and random forests. Random forest modeling was performed using the *randomForest* package[59] and the decision tree modeling utilized the *rpart* [60] and *rpart.plot* [61] packages.

As random forests are the aggregation of a large number of decision trees (the *randomForest* default is 500), each trained on different subsets of the total feature list, they are generally more accurate than an individual decision tree. After calibrating and predicting against the known maintenance intervals, the random forest models did not prove to be significantly more accurate than a decision tree; compounded with their lack of human intelligibility, I discontinued their exploration.

To create the decision tree model, the data was randomly split into training and testing sets, containing 70% and 30% of the data points, respectively. The model was then built, or trained, on the training data set and calibrated on the testing data set, described in Section 3.5.1. A 'full' decision tree was made utilizing all available features. Additionally, more focused decision trees were made as well. The focused decision trees were trained by excluding various groups of features as a psuedo-dimensionality reduction experiment. The purpose of these decision trees was twofold: to assess if any retained a reasonable level of predictive power using fewer features and if so, which ones. Additionally, decision trees with fewer types of features or excluding the more descriptive levels of functional information simulate a design environment with limited clarity on the necessary functionality of a design.

3.5 Evaluating the Model

In order to evaluate the model, three evaluation methods were used. First, the model was calibrated using classical machine learning techniques. Then, the model was grounded by predicting against unseen data and assessing if the results correspond with a reasonable assessment. Finally, the model was evaluated for external validity by comparing its prediction results on the maintenance intervals of a novel engineered system against an expert designer's analysis.

3.5.1 Calibration: Performance Metrics

To calibrate the model towards the best balance of predictive capability while avoiding overfitting, an appropriate complexity parameter had to be determined. First, the *printcp* function of the *rpart* package was used to generate a table containing decreasing complexity parameters values and their corresponding number of splits, relative error, cross-validation error (called xerror by the function), and x-standard deviation. Table 3.8 shows a sample output of the *printcp* function. This table show all possible complexity parameter values down to -1, which represents a maximally overfit decision tree. In order to avoid overfitting, the largest possible complexity parameter was selected whose cross-validation error was within one standard deviation of the minimum cross-validation error. That is to say. All

#	Complexity Parameter	# of Splits	Cross-Validation Error	Standard Deviation
1	0.06570397	0	1.01444	0.014718
2	0.01046931	1	0.93430	0.015396
3	0.00938628	4	0.92202	0.015479
4	0.00722022	5	0.90542	0.015582
5	0.00433213	6	0.89892	0.015620
6	0.00288809	7	0.90397	0.015591
7	0.00216606	9	0.90108	0.015608
8	0.00072202	10	0.91552	0.015520
9	0.00036101	12	0.91986	0.015493
10	0.00000000	16	0.91697	0.015511
11	−1.00000000	34	0.91697	0.015511

Table 3.8: Table of decreasing complexity parameter values and their corresponding calibration metrics.

values within one standard deviation of the minimum cross-validation error are considered to be at the minimum. By selecting the largest complexity parameter that corresponds to a minimum cross validation error, the decision tree will result in the fewest number of splits that still yields the minimum cross-validation error. This is due to the definition of the complexity parameter as used in the *rpart* package. Complexity parameter is the minimum decrease in relative error required for a split to occur. A larger complexity parameter equates to a higher threshold required for a split to be allowed, it therefore mitigates a decision tree's inclination towards overfitting. If a split is allowed to propagate, it can be said with confidence that the split improved the performance of the model. As the complexity parameter and the trained features were adjusted, accuracy and PPV were the primary metrics used to assess the quality of the model. Results of this calibration are presented in Section 4.2.1.

Using Table 3.8 as an example; the global minimum cross-validation error is 0.89892, in row #5. By adding the x-standard deviation of each row to 0.89892, the largest cross-validation error that is said to be at the minimum can be found, 0.87220, in row #4. In this example, the decision tree with the best predictive capability while minimizing the risk of overfitting has a complexity parameter of 0.00722022 and will result in five splits.

3.5.2 Grounding: Design Repository

The Oregon State Design Repository is database of products compiled over the course of multiple efforts [62–66]. The primary ‘part’ table connects to various design-related tables, including: failure data, material, manufacturing process, etc. The part table was the only table queried for the grounding effort. The attributes queried from the table, a brief description of their content and an example part are listed below:

- Part ID: Unique ID number assigned to each part in the repository
  - 12,194
- Name: Part (this corresponds to the term ‘part of concern’)
  - Gasket
- Description: Description of the part
  - Gasket for sealing the manifold and the engine connection
- Child of Part: Points to the Part ID of the ‘parent’ part
  - 12,177 (exhaust assembly)
- System: Points to the System ID
  - 319 (Datsun Truck)

After reviewing all systems included in the repository, system 319 – a 1979 Nissan Datsun 620 truck – met all three criteria necessary to use the system for grounding the model. The criteria were:

1. A long field-life system
2. A sufficient number and variety of parts
3. Readily available information on the parts in the system

A majority of the design repository failed the first criterion as most system were household or consumer products that were not designed to be maintained. Of the long field-life systems, the Datsun had the most unique parts at 131. Additionally, due to the common nature of the system (an internal combustion engine truck), information about each part and their operation was readily available.

Though the repository included tables on functions and flows, I chose not to query those tables as I could not guarantee the same functional decomposition process was followed. The same concern excluded the use of the Assembly attribute in the part table.

After querying the repository for all of the Datsun's parts, the process described in Section 3.3.2 and Fig. 3.1 was repeated to identify the functions, flows, function-flow pairs, function-flow count, and component versus assembly for each part in the system. The data preparation techniques described in Section 3.4.2 were conducted prior to feeding the Datsun's data into the function-maintenance model.

A codified process was necessary in order to apply a consistent assessment of a vague term like "reasonable" onto predictions intended for use in early design situations with a large degree of uncertainty. In addition, the assessment needed to align with the buckets predicted by the model (the model used is found in Fig. 4.1, described in Section 4.1.1). The assessment process was as follows:

1. Does this part need to be maintained, in any capacity, more often than the average part on the truck?
    - If yes, the part should reasonably be in the 2 or 2.5 OOM bucket
    - If no, the part should reasonably be in the 3.5 or 4 OOM bucket
2. Within the above average frequency buckets (2 and 2.5) or below average frequency buckets (3.5 and 4), does a more conservative frequency seem necessary or does a more conservative frequency seem excessive?
    - If a more conservative frequency seems necessary, the part should reasonably be in the 2 or 3.5 OOM bucket
    - If a more conservative frequency seems excessive, the part should reasonably be in the 2.5 or 4 OOM bucket
    - If there was no clear answer, either bucket was deemed reasonable

The face validity assessment and commentary are described in Section 4.2.2.

3.5.3 Verification: Laboratory Upgrade Point Absorber Application

To investigate the external validity of the model (the model used is found in Fig. 4.1, described in Section 4.1.1), evaluation of the model's predictions against "real" prediction values was required [67]. As stated throughout this research, a major problem in novel fields is the lack of "real" data to verify against. In place of "real" data, I verified the model against the best case scenario available to a potential developer in a novel field, an expert designer with experience in the field. Dr. Paasch is a Professor Emeritus of Mechanical Engineering at Oregon State University in the Design Engineering Lab. His research background is in design theory, applied design processes, reliability and survivability in the marine renewable energy field, and automotive engineering fields. Additionally, Dr. Paasch previously served as the director of the Northwest National Marine Renewable Energy Center, now known as the Pacific Marine Energy Center.With experience in design processes, maintainability, and marine renewable energy systems, Dr. Paasch's assessment represent the best case scenario for predicting the maintenance intervals for this work's case study, a wave energy converter.

The system chosen to verify the model on was the Laboratory Upgrade Point Absorber (LUPA) developed at Oregon State University [68]. LUPA is a tank-scale, open-source wave energy converter meant for researchers and developers to help offset the high costs of tank-scale testing and to encourage the open sharing of data and lessons learned. Specifically designed as an open-source platform, LUPA is the only fully designed and manufactured wave energy converter with publicly available design documentation – highlighting the difficulty of accessing detailed maintenance information in novel sectors. I reviewed the 3D models, bill of materials, and device documentation to determine the functions of the 162 unique parts in LUPA. All features used to train the full model were documented. The features were then formatted, input into the model, and the predicted maintenance intervals were returned.

Of the 162 parts in LUPA, 30 unique parts and their corresponding predictions were selected for verification. The model predicted ∼23.3% of parts into the 2 and 3.5 OOM buckets, 40% of parts into the 2.5 OOM bucket and ∼13.3% of parts into the 4 OOM bucket. Following this same breakdown, 7 parts from the 2 and 3.5 OOM buckets, 12 from the 2.5 OOM bucket and 4 parts from the 4 OOM bucket were randomly selected from their respective buckets. All 3D models, bill of materials, and design documents used to assess the system were provided to Dr. Paasch along with the following prompt:

> A list of 30 unique mechanical parts is included along with all 3D modeled parts and assemblies, the bill of materials, and all available design documents. Please review these parts and assess the relative maintenance demand of the parts. For the purposes of this work, "maintenance" is defined as any time an operator or maintenance personnel must inspect, repair, replace or service a component or assembly. "Maintenance" includes all types of preventative maintenance – time, failure finding, risk, condition, and predictive. Please assign all parts to the 4 relative maintenance buckets; the buckets are based on order of magnitude (OOM) relative differences. The 4 buckets are listed below with a brief description.

- 2 OOM bucket - parts in this bucket require the most frequent maintenance
- 2.5 OOM bucket - parts in this bucket require maintenance approximately 3.16 times, a half OOM, less frequently than parts in the 2 OOM bucket
- 3.5 OOM bucket - parts in this bucket require maintenance approximately 10 times, 1 OOM, less frequently than parts in the 2.5 OOM bucket and 13.16 times, 1.5 OOMs, less frequently than parts in the 2 OOM bucket
- 4 OOM bucket - parts in this bucket require the least frequent maintenance; 2 OOMs, 1.5 OOMs and a half OOM less frequent maintenance than the 2 OOM bucket, 2.5 OOM bucket and 3.5 OOM bucket, respectively

The model's predictions and Dr Passch's assessments were then compared using a confusion matrix, presented in Section 4.2.3.

All models are wrong, but some are useful.

George Box

Chapter 4: Results

This chapter summarizes the results of, and discusses the model generated. First, two different models generated in this work are examined and explored through sample parts. Then, an evaluation of the model is presented through the calibration, grounding, and verification efforts. An application of the model used during redesign is then presented. Finally, a discussion on the results and limitations of this research concludes the chapter.

4.1 Function-Maintenance Model

4.1.1 Overview

Fig. 4.1 and Fig. 4.2 are two decision trees generated during this work. Each tree was trained on different subsets of features. The complete list of features, described below, was used to train Fig. 4.1 whereas the features in **bold** comprise the subset used to train Fig. 4.2. The features, organized by type are:

- **1 Component or assembly feature**
- **1 Function-flow count feature**
- 8 Level-1 function features
- 19 Level-2 function features
- **3 Level-1 flow features**
- **14 Level-2 flow features**
- 20 Level-1 function-flow pair features
- 102 Level-2 function-flow pair features

In the interest of clarity, Fig. 4.1 will be referred to as the "Full Model" and Fig. 4.2 as the "Flow Model." The Full Model contains the full range of available features, with the exception of the level-3 features. Level-3 features were excluded from the model as their level of detail is unlikely to be known during early design. The Flow Model excludes functions and instead, the flows are the focus of this model. The same reason given above applies for the exclusion of level-3 flows.

In total, the Full Model was trained using 168 features and resulted in a decision tree

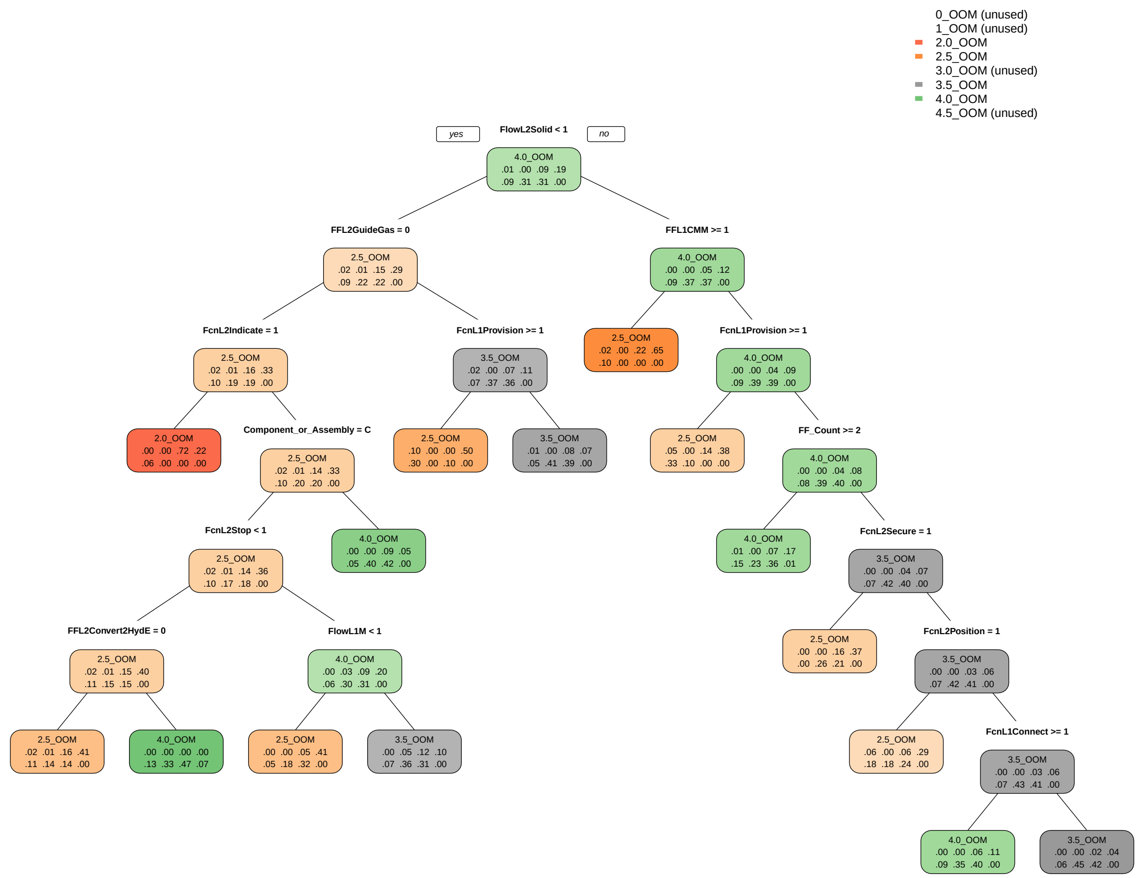


Figure 4.1: Decision tree generated using level-1 and level-2 functions, flows and function-flow pairs. As well as component or assembly and function-flow count.

8 layers deep with 15 possible leaf nodes. The Flow Model was trained using 19 features and resulted in five layers of 8 possible leaf nodes. Across both models, all nodes predict parts into one of the four following OOM buckets: 2, 2.5, 3.5, and 4. This is in line with expectations based on the number of data points in those buckets. Due to the depth of each tree and the features used to train each model, the amount and type of information required to most effectively use each model is different. The Full Model requires an understanding of the flows of energy, material and signal as well as the functions acting on them. In contrast, the Flow Model only requires information on the system's proposed flows. Flow information may be known earlier in the process, allowing the Flow Model to be used earlier; this could be related to regulations restricting the use of pressurized gases or customer requirements

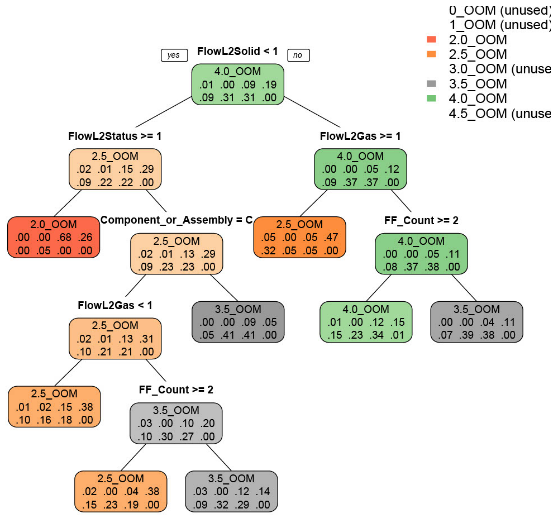


Figure 4.2: Decision tree generated using level-1 and level-2 flows, component or assembly, and function-flow count.

of high force output only possible through hydraulic energy, etc. If the information needed to respond to a prompt is not available, the prediction at that node can be taken and the implication of possible responses can be explored.

The use of the Flow Model is intended to demonstrate the different manners of utilizing the explored function-maintenance relationship. The variety of realizations must each be evaluated to understand their uses and limitations, this is explored in Section 4.2.

The prediction, whether at a leaf or non-leaf node, is given by the value in the top-center, inside the node. The percentage of parts predicted into each bucket is given by the $2x4$ grid below the prediction. When used to predict maintenance intervals, this can be taken as the probability of a part to be in each maintenance interval. Examining the top node of either tree (the top node is the same as it represents the distribution of all the training data), from top left towards bottom right, the values refer to each bucket in ascending order. The top left value of 0.01 (1%) is the probability that a part in this node will be in the 0 OOM bucket. Following this pattern, the probability per bucket at this node is: 0 OOM = 1%, 1 OOM = 0%, 2 OOM = 9%, 2.5 OOM = 19%, 3 OOM = 9%, 3.5 OOM = 31%, 4 OOM = 31%, and 4.5 OOM = 0%. From this grid, the final prediction of each node is the bucket with the highest probability. Due to rounding, the 3.5 and 4 OOM bucket appear the same, but the 4 OOM bucket has a higher probability of containing a part in that node, so it is the final prediction.

By comparing the two models, some consistencies become apparent. The top node – which can be read as "Does the level-2 flow of *Solid* occur less than one time in this part?" – is the same in both models. Recall that decision trees test all available features to determine which one reduces informational entropy the most, the feature that does is then used as the prompt for that node. As both trees share the same top node, this tells us that the occurrence of the level-2 flow of *Solid*, is the most informative single question that can be asked about a part. By responding "Yes, the part has less than one occurrence (zero occurrences) of the level-2 flow of *Solid*", both models predict that the new part's maintenance interval is likely in the 2.5 OOM bucket. If no answer is available or the response is in the negative, then the predicted maintenance interval remains in the 4 OOM bucket. By answering a single question about a part, a 1.5 OOM difference in maintenance frequency is assessed. Using the maintenance intervals of the training data, this is the difference between a part with a maintenance interval of 209 to 658 flight hours (2.5 OOM bucket) compared to 6,582 to 20,811 flight hours (4 OOM bucket); in general terms this is a difference of 31.6 times more/less frequent. When applied to familiar measures of time, a part with a 'yes' response requires weekly maintenance and a part with a 'no' response requires annual maintenance.

Another similarity is the second layer nodes of "FFL2$GuideGas$ = 0" from the Full Model and "FlowL2Gas≥1" from the Flow Model. Though they are not asking the same question, the level-2 flow *Gas* comprises half of the function-flow pair of *Guide Gas*. The repeat appearance of the feature (or of features related to) *Gas* indicates that the knowledge regarding its use in a design is highly informative, this further explored in Section 4.1.2.4.

4.1.2 Using the Model

Five parts have been selected from the training data to be manually passed through each model to facilitate the exploration and understanding of the models. The five selected parts are shown in Table 4.1 along with contextual information and features pertinent to their maintenance prediction. First, each part will be passed through both models and the resultant prediction will be given and compared to the known prediction, found in the <u>MTBM Bucket</u> column of Table 4.1. Then the features that inform and impact the outcome of each prediction are explored. Finally, the insights uncovered through the exploration of the models from the perspective of that part and the other parts that share

Part #	Additional details	Part of concern	Part type	MTBM bucket	Full Model Pred.	Flow Model Pred.	Comp. vs Assy.	Function 1	Flow 1	Function 2	Flow 2	FF Count
42	Electrical functional checks	Visual warning system	Part type	2	2	2	Assy.	*SignalFcn Indicate Display*	*SignalFlow Status Visual*	N/A	N/A	1
6	Nose gear	Piston rod felt wiper	Wiper seal	2.5	2.5	3.5	Comp.	*Change Mag. Stop Prevent*	*Material Solid Particulate*	N/A	N/A	1
1,236	Flight controls - rudder (removed from vertical stabilizer)	Rudder assembly	Flight control surface	3.5	4	3.5	Assy.	*Change Mag. Change Condition*	*Energy Mech. Energy Rot. Energy*	N/A	N/A	1
300	Summary of X-ray and eddy current inspection	Crown skin circumferential splice at frame 19	Exterior surface	3.5	3.5	3.5	Comp.	*Channel Guide Translate*	*Material Gas Gas3*	N/A	N/A	1
1,826	Combustion section check	Inner combustion liner shell	Combustion chamber shell	3	2.5	2.5	Comp.	*Provision Store Contain*	*Material Gas Gas3*	*Channel Guide Translate*	*Material Gas Gas3*	2

Table 4.1: Five sample parts chosen to highlight the process through which maintenance predictions are made. Only two function and flow features are shown as none of the sample parts perform more than two functions.

its maintenance bucket are explored. Insights into the nature and understanding of the hypothesized function-maintenance relationship are examined. Additionally, insights into how this relationship can be captured and modeled, as well as their impacts are examined.

4.1.2.1 Sample Part #42

The first part to be passed through the models is part #42, the Visual Warning System, which provides visual alerts in the cockpit for a variety of safety and aviation information. The part's function and flow are *SignalFcn/Indicate/Display* and *SignalFlow/Status/Display*. Starting with the Full Model, the part's path down the decision tree is (a description of each prompt and the path taken is provided for the first example):

1. FlowL2*Solid* < 1 or "Does the level-2 flow of *Solid* occur less than one time in this part?" → Yes, the left branch is taken
2. FFL2*GuideGas* $= 0$ or "Does the level-2 function-flow pair of *Guide Gas* occur 0 times in this part?" → Yes, the left branch is taken
3. FcnL2*Indicate* $= 1$ or "Does the level-2 function of *Indicate* occur 1 time in this part?" → Yes, the left branch is taken
4. Part #42 is predicted to be in the 2 OOM bucket with a probability of 72%

Using the Flow Model:

1. FlowL2*Solid* $< 1 \rightarrow$ Yes
2. FlowL2*Status* $\geq 1 \rightarrow$ Yes
3. Part #42 is predicted to be in the 2 OOM bucket with a probability of 68%

Both models correctly predicted this part to be in the same bucket. The final prompt on each path, the level-2 function *Indicate* (Full Model) and the level-2 flow *Status* (Flow Model) are the prompts that provide the highest fidelity information on the part's bucket. The definitions of each term provides the context that explains this, *Indicate*: "to make something known to the user about a flow" and *Status*: "a condition of some system, as in information about the state of the system." The predictions in the most frequent maintenance bucket correspond to the reasonable assumption that, if information is important enough to be collected and displayed to a user, it is important enough to demand frequent maintenance.

4.1.2.2 Sample Part #6

Part #6 is a piston rod felt wiper found on the shock strut of the nose landing gear. The part's function and flow are *Change Magnitude/Stop/Prevent* and *Material/Solid/Particulate*. Using the Full Model:

1. FlowL2*Solid* $< 1 \rightarrow$ No
2. FFL1*CMM* $\geq 1 \rightarrow$ Yes
3. Part #6 is predicted to be in the 2.5 OOM bucket with a probability of 65%

Using the Flow Model:

1. FlowL2*Solid* $< 1 \rightarrow$ No
2. FlowL2*Gas* $\geq 1 \rightarrow$ No
3. FF_Count $\geq 2 \rightarrow$ No
4. Part #6 is predicted to be in the 3.5 OOM bucket with a probability of 39%

The Full Model correctly predicted the maintenance bucket and the Flow Model was off by an order of magnitude. Upon examining the other parts sorted into the same leaf node by the Full Model, it is clear that the features containing function information were vital in classifying this part. Of the 58 parts in the leaf node, 47 of them were either a wiper seal, filter, or boot. All of these parts serve to restrict (*Change Magnitude*) the flow of some *Material*, whether it be *Particulate* for filters and wiper seals or ice (*Object*) build up in the case of boots.

The parts sorted into the leaf node by the Flow Model are not nearly as homogeneous as they vary from wiper seals to structural members. Considering the only requirement of parts in this leaf node is that they have only one flow and it must be the level-2 flow of *Solid*, very little information about these parts is actually captured by the model. The final prompt of FF_Count ≥ 2 says that this part has only one flow which makes the second prompt (FlowL2*Gas* ≥ 1) void of information because the part can only have one flow and it was already determined by the first prompt. If the top node – recall, contains all parts and therefore matches the distribution of the entire data set – is referenced, this leaf node appears to closely match the distribution of the entire data set. The resemblance between this node and the top node indicates that this leaf node serves as a 'catch-all' for parts that the model has insufficient data to make a prediction more meaningful than a null hypothesis.

4.1.2.3 Sample Part #1,236

Part #1,236 is the rudder assembly that performs the function and flow *Change Magnitude/Change/Condition* and *Energy/Mechanical Energy/Rotational Energy*. Using the Full Model:

1. FlowL2*Solid* $< 1 \rightarrow$ Yes
2. FFL2*GuideGas* $= 0 \rightarrow$ Yes
3. FcnL2*Indicate* $= 1 \rightarrow$ No
4. Component_or_Assembly = Component $\rightarrow$ No
5. Part #1,236 is predicted to be in the 4 OOM bucket with a probability of 42%

Using the Flow Model:

1. FlowL2*Solid* $< 1 \rightarrow$ Yes
2. FlowL2*Status* $\geq 1 \rightarrow$ No
3. Component_or_Assembly = Component $\rightarrow$ No
4. Part #1,236 is predicted to be in the 3.5 OOM bucket with a probability of 41%

The Flow Model correctly predicted the rudder assembly whereas the Full Model's prediction was 0.5 OOMs less frequent. Upon closer inspection of the leaf nodes from each model, the differences are realized to be negligible. The leaf node in the Full Model has a probability of 40% for the 3.5 bucket and 42% for the 4 bucket. The leaf node in the Flow Model has a probability of 41% for both the 3.5 and 4 bucket. Despite the final predictions being different, a superior interpretation of the results would be that the rudder assembly has a probability of 82% to be in the 3.5 or 4 bucket.

Relevant to understanding this prediction is clarifying what maintenance of an assembly entails. For this example, the rudder assembly as a whole requires maintenance, on average, in the 3.5/4 OOM frequency. The components that comprise the rudder assembly may, and certainly do, require more frequent maintenance though. The maintenance frequency of an assembly is determined by the frequency of the maintenance tasks that refer to the assembly as a **whole**, and is not influenced by the maintenance frequency of its individual components.

With this distinction in mind, the component or assembly feature may be thought of as informing the possible intensity or type of maintenance action to be expected. The maintenance action related to part #1,236 is "Conduct a general visual inspection of the rudder assembly for permanent deformation, corrosion, fatigue cracks, wear or chafing, and general condition." Another assembly (selected at random) is part #577, the nose landing gear subsystem. Its related maintenance task is "Completely disassemble nose landing gear. Conduct visual, dye penetrant, and Magnaflux inspection of all components as appropriate. Determine extent of wear by making dimensional checks." The maintenance tasks required for these two assemblies can serve to better educate the designer's understanding of the maintenance prediction of an assembly. A maintenance task for an assembly, as opposed to the individual components that comprise it, is more likely to involve a broad inspection of concerns not addressed during individual component maintenance or entail a complete overhaul of the assembly.

4.1.2.4 Sample Part #300

Part #300 is crown skin circumferential splice at frame 19, a section of the exterior skin of the aircraft. Part #300 performs the function and flow *Channel/Guide/Translate* and *Material/Gas/Gas3*. Using the Full Model:

1. FlowL2$Solid < 1 \rightarrow$ Yes
2. FFL2$GuideGas = 0 \rightarrow$ No
3. FcnL1$Provision \geq 1 \rightarrow$ No
4. Part #300 is predicted to be in the 3.5 OOM bucket with a probability of 41%

Using the Flow Model:

1. FlowL2$Solid < 1 \rightarrow$ Yes
2. FlowL2$Status \geq 1 \rightarrow$ No
3. Component_or_Assembly = Component $\rightarrow$ Yes

4. FlowL2$Gas < 1 \rightarrow$ No
5. FF_Count $\geq 2 \rightarrow$ No
6. Part #300 is predicted to be in the 3.5 OOM bucket with a probability of 32%

Part #300 is a good example of a part that pertains to the two most informative features for both models (discussed above in Section 4.1.1), the second layer node of FFL2*GuideGas* = 0 from the Full Model and FlowL2$Gas \geq 1$ from the Flow Model. Exploring this part – and the large number of parts it represents – is essential to understand any model developed from this work. Similar to any engineering tool, an informed understanding of the inputs, processes, and outputs of this model must be applied to understand its capabilities and limitations.

The models generated in this work were all trained on maintenance data mined from a civilian aircraft and its engines. Despite attempts to generalize all concepts applied, artifacts from the mined data undoubtedly exist. One such artifact explains the prominence of the two previously described features. Apart from the flight control surfaces and wings (that induce/adjust movement in all 6 degrees of freedom), the very form of the components that make up all external surfaces of the aircraft serve a singular purpose: to perform the function and flow of *Channel/Guide/Gas* and *Material/Gas/Gas3*. More simply, all external surfaces are aerodynamic. This function is not embodied by an operation of the component, but by an intrinsic property of the component. Since drag caused by poor aerodynamics represents a potential performance loss – and therefore revenue loss – maintenance of the aircraft's external surface components is of significant concern.

An additional group of parts, the so-called "structural members" similarly explain why the first node of both models is FlowL2$Solid < 1$. Structural members were described with the function and flow *Support/Stabilize/Stabilize3* and *Material/Solid/Object* and includes parts such as: keel beam, brackets, tires, support structures, wing spars, wing spar caps, etc. Between the large number of structural members and the exhaustive maintenance that requires an inspection of each structural part individually, they represent a large portion of the parts that fit their functional description. This is the same case for external aerodynamic parts.

The structural members are examined further but the same concept applies to the external surfaces. Of the 1,995 data points used to build this family of decision trees, 1,169 respond "No" to the top node of FlowL2$Solid < 1$. Of that portion, 416 of the parts match the functional description of structural members (*Support/Stabilize/Stabilize3* and *Material/Solid/Object*), most of which are in fact structural members. 77% of the 416 parts that are functionally similar to structural members fall into the maintenance frequency buckets that align with inspections that occur at the following frequencies: 6,000, 7,200, and 12,000 hour; 3,000 and 6,000 landing; and 12 year. The inspections at those intervals include the exhaustive inspection required of all structural members in the aircraft. The awareness of this information regarding the input data explains the significant difference in prediction (2.5 versus 4 OOM bucket) capable after responding to top node of the each model. These artifacts and their implications concerning functional design and the models external validity are discussed throughout Chapter 5. For the current exploration of these models, an understanding of these artifacts facilitates a more useful understanding of the models' results. For example, if a concept is generated that includes the requirement to *Stabilize* a *Solid,* the model will make a prediction that aligns with a structural member, i.e. a beam, that **on average** does not require frequent maintenance. If the designer's intent is more akin to a tire or a wheel – which may share functional similarities – the maintenance considerations attributed to that function may be grossly underestimated. The model would still be useful, but additional consideration with an understanding of these artifacts would need to be applied.

4.1.2.5 Sample Part #1,826

Part #1,826 is an inner combustion liner shell, found near the combustion chamber in the engines. It performs the functions and flows *Provision/Store/Contain* and *Material/-Gas/Gas3* as well as *Channel/Guide/Translate* and *Material/Gas/Gas3.* Using the Full Model:

1. FlowL2*Solid* $< 1 \rightarrow$ Yes
2. FFL2*GuideGas* $= 0 \rightarrow$ No
3. FcnL1*Provision* $\geq 1 \rightarrow$ Yes
4. Part #1,826 is predicted to be in the 2.5 OOM bucket with a probability of 50%

Using the Flow Model:

1. FlowL2*Solid* $< 1 \rightarrow$ Yes
2. FlowL2*Status* $\geq 1 \rightarrow$ No
3. Component_or_Assembly = Component $\rightarrow$ Yes
4. FlowL2*Gas* $< 1 \rightarrow$ No
5. FF_Count $\geq 2 \rightarrow$ Yes
6. Part #1,826 is predicted to be in the 2.5 OOM bucket with a probability of 38%

Part #1,826 is functionally similar to part #300 but shows how the models are able to differentiate between the two. The Full Model made the distinction through the prompt FcnL1*Provision* ≥ 1, whereas the distinction was captured with FF_Count ≥ 2 by the Flow Model. Both models predicted a maintenance 0.5 OOMs more frequent than the actual frequency; however, for a part experiencing frequent cycles of high temperature and pressure, a more conservative prediction is appropriate. In fact, this part has another data point that entails a maintenance task that does fall into the 2.5 OOM bucket.

4.2 Model Evaluation

This section describes the evaluation techniques explored and the results of those explorations. First, the calibration results of the Full and Flow Models are described. Then the grounding of the Full Model against unseen data to assess a qualitative degree of "reasonableness" is reported. Lastly, "verification" results are described. The model was applied to a prototype wave energy converter and the results were "verified" against the predictions of the best case scenario for maintenance prediction, an expert designer engineer with experience in maintenance as well as wave energy converter design.

4.2.1 Calibration: Performance Metrics

After using the printcp function of the rpart package to calculate the cross-validation error at all levels, from a single layer tree to a maximally overfit tree, the optimal complexity parameter was determined. The relevant portions of the printcp output for the Full Model are shown in Table 4.2. Row #11 gives the complexity parameter (0.00144404) that corresponds to the minimum cross-validation error (0.87220). This complexity parameter equates to the decision tree with the smallest cross-validation error as determined by resampling 10 random subsets of the training data. Though row #11 gives the most accurate model, any row within one standard deviation of row #11's cross-validation error is also said to be at the minimum and therefore just as accurate. The highest row within one standard deviation is the optimal choice, as the larger complexity parameter results in a more general model. Row #9 is the highest minimum row and equates to a complexity parameter of 0.00361011 for the Full Model.

The same process was followed for the Flow Model using Table 4.3. Row #5 corresponds to the smallest cross-validation and row #4 contains the optimal complexity parameter,

#	Complexity Parameter	# of Splits	Cross-Validation Error	Standard Deviation
1	0.06570397	0	1.00000	0.014858
2	0.02671480	1	0.93430	0.015396
3	0.02238267	2	0.91336	0.015534
4	0.00866426	3	0.89675	0.015633
5	0.00758123	4	0.90108	0.015608
6	0.00722022	6	0.90830	0.015565
7	0.00649819	7	0.90253	0.015600
8	0.00433213	10	0.89170	0.015661
9	0.00361011	11	0.88520	0.015696
10	0.00216606	12	0.87437	0.015751
11	0.00144404	14	0.87220	0.015762
...	.....	...	.....	.....
16	−1.00000000	56	0.89386	0.015649

Table 4.2: Complexity parameter table calculated using the Full Model decision tree using decreasing complexity parameter values.

0.00722022. As the optimal complexity parameter is determined by minimizing cross-validation error then using the largest complexity parameter at that minimum, there is no correct value. Assuming best practices have been followed to determine the optimal value, the litmus test of a "good" complexity parameter are the external evaluations. If the predictions aren't externally valid or reasonable, the model may be overfit to the training data and aren't generalizable. If the predictions don't provide any useful information (e.g. predicting every single part into the 3.5 or 4 OOM bucket would be correct 62% of the time but that doesn't provide any useful insights), the complexity parameter may be too large and over-simplifying the model. Section 4.2.2 explores this topic by evaluating the reasonableness of predictions and the usefulness of prediction were previously explored in Section 4.1.2.

The new complexity parameters were then used to build the Full and Flow Model (Fig. 4.1 and Fig. 4.2) using the training data set. This resulted in variations of their

#	Complexity Parameter	# of Splits	Cross-Validation Error	Standard Deviation
1	0.06570397	0	1.01444	0.014718
2	0.01046931	1	0.93430	0.015396
3	0.00938628	4	0.92202	0.015479
4	0.00722022	5	0.90542	0.015582
5	0.00433213	6	0.89892	0.015620
...	.....	...	.....	.....
11	−1.00000000	34	0.91697	0.015511

Table 4.3: Complexity parameter table calculated using the Flow Model decision tree using decreasing complexity parameter values.

first iteration that were pruned to reduce overfitting while conserving as much predictive capability as possible. The testing data was then predicted on and compared to their known buckets.

The confusion matrices of the known versus predicted buckets are shown in Table 4.4 and Table 4.5. The columns of the confusion matrices are the known maintenance buckets from the test data, while the rows are the predicted maintenance buckets for the same test data. Diagonally from the top left to bottom right are correctly predicted data points (highlighted in gray). Because the buckets sequentially increase in value, the location of a cell within the matrix allows for additional insights. Values above the grey diagonal were predicted to require more frequent maintenance than the actual maintenance frequency. Values below the diagonal were predicted to require less frequent maintenance than the actual maintenance frequency. The farther from the grey diagonal that a cell is, the larger the difference between known and predicted value. The top right and bottom left cells represent these extremes. The top right cell contains parts that were predicted to be in the 0 OOM bucket but were actually in the 4.5 OOM bucket. This would be the most conservative prediction possible. The bottom left cell represents parts predicted to be in the 4.5 OOM bucket but are known to be in the 0 OOM bucket, this would be the most optimistic prediction possible.

Comparing the two matrices, the Full Model made more correct predictions into the

Full Model Predicted Bucket	Known Bucket							
	0 OOM	1 OOM	2 OOM	2.5 OOM	3 OOM	3.5 OOM	4 OOM	4.5 OOM
0 OOM	0	0	0	0	0	0	0	0
1 OOM	0	0	0	0	0	0	0	0
2 OOM	0	0	3	3	0	0	0	0
2.5 OOM	0	1	30	108	38	31	31	0
3 OOM	0	0	0	0	0	0	0	0
3.5 OOM	0	0	8	22	21	94	106	0
4 OOM	2	0	5	14	10	35	40	0
4.5 OOM	0	0	0	0	0	0	0	0

Table 4.4: Confusion matrix relating the known maintenance frequency with the Full Model's predictions.

Flow Model Predicted Bucket	Known Bucket							
	0 OOM	1 OOM	2 OOM	2.5 OOM	3 OOM	3.5 OOM	4 OOM	4.5 OOM
0 OOM	0	0	0	0	0	0	0	0
1 OOM	0	0	0	0	0	0	0	0
2 OOM	0	0	4	2	0	0	0	0
2.5 OOM	0	1	24	85	29	32	25	0
3 OOM	0	0	0	0	0	0	0	0
3.5 OOM	2	0	16	48	35	118	135	0
4 OOM	0	0	2	7	5	10	17	0
4.5 OOM	0	0	0	0	0	0	0	0

Table 4.5: Confusion matrix relating the known maintenance frequency with the Flow Model's predictions.

2.5 and 4 OOM buckets, with 23 correct predictions more for each bucket compared to the Flow Model. The Flow Model made one correct prediction more than the Full Model in the 2 OOM bucket and 24 more in the 3.5 OOM bucket. Beyond looking solely at correct predictions, the location and distribution of values across the matrices brings us to other insights. Assuming a more conservative prediction is preferred to an optimistic prediction, we can inspect the columns (skipping the 0, 1, and 4.5 buckets due to lack of data) from left-to-right and compare the prediction distribution.

- 2 OOM - Though the Flow Model makes one additional correct prediction, the remaining known 2 OOM parts are not equally close to the grey "correct" cell. The Full Model has a smaller standard deviation from the known value.
- 2.5 OOM - The Full Models performs significantly better than the Flow Model for this bucket.

OOM Bucket	Full Model		Flow Model	
	PPV	**Expanded PPV**	**PPV**	**Expanded PPV**
2	50.0%	100%	66.7%	100%
2.5	44.0%	73.6%	43.3%	70.4%
3.5	37.5%	88.0%	33.3%	81.4%
4	37.7%	70.8%	41.5%	65.9%

Table 4.6: Per class metrics for the Full and Flow Model, given by class.

- 3 OOM - As neither model included predictions into this bucket, only the distribution of incorrect predictions can be examined. The standard deviation of both models from the known value are similar, but the Full Model is more conservative.
- 3.5 OOM - This is the first column where the Flow Model made significantly more correct predictions. Not only that, but the difference in correct predictions are accounted for by the Full Model's higher number of overly optimistic predictions.
- 4 OOM - The Full Model made more correct predictions but this is at least partially offset by the Flow Model's lower standard deviation from the known value.

From the confusion matrices, the other metrics used to evaluate supervised classification models were calculated. All per class metrics are shown in Table 4.6. The two metrics that apply to the model overall are the accuracy and the average PPV (seen in Table 4.7). The Full Model had an accuracy of 40.2% and the Flow Model had an accuracy of 37.5%. As accuracy is the total number of correct predictions over the total number of predictions, it gives a general sense of the models' performance. The other model level metric used, average PPV, gives the probability that any given prediction, regardless of class, is correct. When evaluating a machine learning model, there are no target values for metrics. Acceptable values are entirely context and judgement dependent. The only "true" judgement possible is in comparison to verified information, this is explored in Section 4.2.3 where the model's results are judged against the best proxy for "verified" data possible in a novel field.

Traditionally, the classes of a multi-classification model are not sequential (i.e. Colors: Red, Blue, etc.) so a prediction is black or white, either correct or incorrect, with no shades of gray. This work is not intended to provide designers with "hard" and exact values of maintenance intervals (this research wouldn't be need if that was possible), but instead to

Model	Accuracy	Average PPV	Expanded Accuracy	Expanded Average PPV
Full	40.2%	42.3%	79.4%	76.9%
Flow	37.5%	46.2%	83.1%	79.4%

Table 4.7: Per model metrics for the Full and Flow Model.

inform early maintainability considerations.

In pursuit of metrics that capture the shades of grey that still allow for more informed design choices, PPV, accuracy, and average PPV were modified to create what will be called "Expanded PPV", "Expanded Accuracy", and "Expanded Average PPV." Expanded PPV is seen in Table 4.6 alongside PPV. Recall, PPV is calculated on a per class basis and is defined as the number of correct prediction for a class over the total number of predictions for that class. Phrased another way, PPV gives the probability that any given prediction into a class is accurate. To account for the "gray" area of correctness, the definition of a "correct prediction" is expanded to include predictions ± 0.5 OOMs from the true value. With this change, if a part is known to be in the 2.5 OOM bucket but is predicted to be in the 2 OOM bucket, this prediction would be deemed correct. Expanded accuracy and expanded average PPV follow the same logic and are seen in Table 4.7. The expanded PPV, accuracy, and expanded PPV for both models increased significantly, approximately doubling all values.

4.2.2 Grounding: Design Repository Application

The first foray into external validation was by grounding the model. The intent of grounding is to evaluate if the model results in reasonable predictions when exposed to unseen data. The 131 data points of the 1979 Nissan Datsun 620 truck were predicted using the Full Model. My personal judgement was applied following the process described in Section 3.5.1 for a "Yes" or "No" answer if I found the prediction to be reasonable. To better convey the results of this endeavor, 9 parts are shown in Table 4.8 along with their necessary feature for prediction. Also shown are the predicted maintenance bucket, the "Yes" or "No" assessment of reasonableness, and commentary on the prediction or assessment. Of the 131 predicted parts, 81 were assessed to be reasonable. Three recurring themes arose from unreasonable predictions.

4.2.2.1 Theme #1: Mechanical Simplicity

Part #12,127, the impact bar shows the first theme. The model fails to capture the relative maintenance requirement of functions that are essential, but are performed by a part that is mechanically simple (e.g. does not move) and as a result, is more likely to require condition based preventative maintenance as opposed to time, failure, or risk based preventative maintenance. This is likely caused by a distinction not captured in the functional modeling process or as a feature: is the function performed by an action of the part or by an inherent characteristic of the part? A clear distinction may be difficult to make, but the impact bar of the car exemplifies the difference. Does the impact bar *Distribute Mechanical Energy* by an action it performs? Or by an inherent characteristic? It *Distributes* the force (*Mechanical Energy*) of a crash by its nature of being connected to the frame of the car. A part that inherently performs a function may be more likely to require condition based maintenance which, by definition, does not lend itself well to a consistent maintenance interval. The impact bar is likely to fail due to one of two reasons, a crash or rust. A car crash, which would entail unscheduled maintenance is outside of the scope of this research. Rust is a prime example of the difficulty in assigning a maintenance interval to a mechanically simple part. The context and usage of the part heavily influence the impact of rust, two considerations that novel systems are not able to account for without historical precedent.

4.2.2.2 Theme #2: Broad Design Space

Part #12,041, the muffler, embodies the second theme. It relates to the abstract nature of functional modeling and the wide design space possible within the execution of a particular function. The muffler's function and flow are *Change Magnitude/Change/Decrement* and *Energy/Acoustic Energy/Velocity*. The function-flow pair of *Change Magnitude Energy* could describe a wide variety of maintenance intensive parts, like a gear box, a transformer, or the hydraulic shocks of the Learjet.

Part #	Part	Description	Parent Part	Comp. vs Assy.	Function 1	Flow 1	Function 2	Flow 2	FF Count	OOM Bucket Prediction	Reasonable?	Notes on Prediction
12,056	AM-FM antenna	Gets AM-FM reception	Cabin interior	Comp.	*Channel Import Import3*	*Energy EM Energy EM Energy3*	*SignalFcn Sense Detect*	*Energy EM Energy EM Energy3*	2	2.5	Yes	An antenna in a personal vehicle is used primarily to listen to the radio, but does serve the purpose of receiving important weather and traffic information.
12,127	Impact bar	None	Bumper	Comp.	*Branch Distribute Distribute3*	*Energy Mech. Energy Transl. Energy*	N/A	N/A	1	2.5	No	Theme #1, discussed below.
12,142	IC engine	Combustion engine that runs the car	Engine compartment	Assy.	*Convert Convert2 Convert3*	*Energy Chem. Energy Affinity*	*Convert Convert2 Convert3*	*Energy Mech. Energy Transl. Energy*	2	4	Yes	Referring to the commentary on maintenance of assemblies in Section 4.1.2.3, this is reasonable.
12,034	Side mirror	Indicates status around vehicle	Exterior	Comp.	*Channel Import Import3*	*SignalFlow Status Visual*	*Channel Guide Translate*	*SignalFlow Status Visual*	2	2.5	Yes	The user inspecting, and adjusting as needed, their rear and side view mirrors is supposed to be performed every time before operating a the vehicle. Though this is likely not the case, this does fall within the definition of a maintenance action "all actions necessary for retaining...operational state."
12,111	Spark plugs	Provide spark to the engine	Engine compartment	Comp.	*Provision Supply Supply3*	*Energy Therm. Energy Temp.*	N/A	N/A	1	2.5	Yes	None
12,128	Radiator	Takes in hot coolant from engine and cools it through forced convection	HVAC	Comp.	*Channel Guide Translate*	*Energy Therm. Energy Temp.*	*Change Mag. Change Increase*	*Energy Therm. Energy Heat Rate*	2	2.5	Yes	None
12,194	Gasket	Seals the manifold and engine connection	Exhaust assembly	Comp.	*Change Mag. Stop Prevent*	*Material Mixture Gas-Gas*	N/A	N/A	1	3.5	Yes	None
12,041	Muffler	Reduces the amount of noise emitted by the exhaust system	Exhaust assembly	Comp.	*Change Mag. Change Decrease*	*Energy Acoustic Energy Particle Velocity*	N/A	N/A	1	2.5	No	Theme #2, discussed below.
12,038	Cup holder	Holds cup	Cabin interior	Comp.	*Support Secure Secure3*	*Material Solid Object*	N/A	N/A	1	2.5	No	Theme #3, discussed below.

Table 4.8: 9 sample parts of the 131 assessed from the Datsun truck.

4.2.2.3 Theme #3: Inappropriate Data

Part #12,038 exemplifies the last theme. This part is a cup holder, obviously not something that would be considered in a maintenance analysis. In the interest of thoroughness, all parts in the Datsun truck recorded in the design repository were passed through both models regardless if they fell into this theme. 13 parts fell into this theme including the ash tray, floor mats, speaker covers, etc.

4.2.3 Verification: LUPA Application

The verification results of the Full Model as measured against the expert designer's assessment are shown in Table 4.9. The grey diagonal indicates the model and expert assessing a part into the same maintenance bucket. Above the diagonal represents the model assessing a part requiring more frequent maintenance than the expert assessed and vice versa for below the diagonal.

From Table 4.9, the model predicted the same relative maintenance bucket as the expert design engineer 6 times, more conservatively 17 times, and less conservatively the remaining 7 times. The model's inaccuracy at higher OOM buckets is not surprising due to the nature of functional decomposition. The cell found in the 4 OOM column and 2 OOM row can be examined to understand this issue better. The damper plate is one of the three parts referred to by this cell. The damper plate is a solid circular piece of 6061 aluminum at the bottom of the spar that has lead ballast secured to it. When viewed in terms of function, the damper plate performs two function-flow pairs (listed at the 2nd level), *Change Mechanical Energy* and *Stabilize Solid*. These function-flow pairs describe the plate as *Stabilizing* the weights (*Solid*) secured to it and increasing (*Change*) the force of hydrodynamic drag (*Mechanical Energy*) of the spar. Both of these functions maximize the hull's relative motion to the spar and are essential to the operation of the device. Despite its importance, the damper plate is just a solid piece of metal with no moving parts and should not warrant maintenance actions at the highest frequency level, this result falls in line with Theme #1: Mechanical Simplicity, found during the Grounding Evaluation in the previous section. Similar to any tool or software used in engineering, this model still requires an engineer to review the results and assert their judgment, but the model can reduce the number of parts an engineer needs to assess.

Model Predicted Bucket \ Expert Predicted Bucket	0 OOM	1 OOM	2 OOM	2.5 OOM	3 OOM	3.5 OOM	4 OOM	4.5 OOM
0 OOM	0	0	0	0	0	0	0	0
1 OOM	0	0	0	0	0	0	0	0
2 OOM	0	0	1	1	0	2	3	0
2.5 OOM	0	0	3	3	0	1	5	0
3 OOM	0	0	0	0	0	0	0	0
3.5 OOM	0	0	0	1	0	1	5	0
4 OOM	0	0	2	0	0	1	1	0
4.5 OOM	0	0	0	0	0	0	0	0

Table 4.9: Confusion matrix of model predicted versus expert design engineer assessed maintenance values for 30 LUPA parts

4.3 Applying the Model

This section describes an application of the model to redesign LUPA, a wave energy converter, to reduce its required maintenance frequency. LUPA is a piece of laboratory equipment. It was not designed or intended for the activities and environmental conditions that a commercial wave energy converter will be expected to withstand. As such, design considerations that would be paramount to a commercial wave energy converter, like maintainability, took a lower priority than design considerations related to LUPA's role as an open-source platform for analyzing concepts, validating numerical models, and innovating control schemes.

The results from the LUPA application come from a possible physical or functional redesign. The redesign is given from a maintenance-centric point of view and will not account for other possible design considerations. The redesign presented is intended as an example of a modification to the physical or functional architecture of the device that could be stimulated by examining the functional drivers of maintenance more closely and from an earlier stage in development.

4.3.1 Carriage Rail Securement

The redesign considered is in regard to the system that allows for vertical translation of the hull along the spar. Fig. 4.3 shows a simplified version of the system. The carriage rail is secured to the spar via 10 screws extending through the carriage rail into the spar, three of the holes for those screws are visible in Fig. 4.3. The carriage at the top of Fig. 4.3 is the

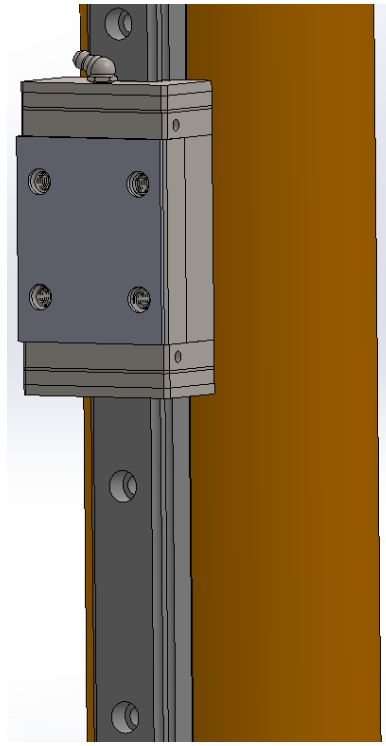


Figure 4.3: 3D model of LUPA showing the carriage, carriage rail, and spar. The model has been simplified to improve visibility. Note the three visible screw holes that secure the carriage rail to the spar.

top carriage rail that the hull is secured to.

Upon analysis, it was found that the screws perform the function and flow of *Support-/Secure/Secure3* and *Material/Solid/Object*. Referencing the Full Model:

1. FlowL2*Solid* $< 1 \rightarrow$ No
2. FFL1*CMM* $\geq 1 \rightarrow$ No
3. FcnL1*Provision* $\geq 1 \rightarrow$ No
4. FF_Count $\geq 2 \rightarrow$ No
5. FcnL2*Secure* $= 1 \rightarrow$ Yes
6. The carriage rail screws are predicted to be in the 2.5 OOM bucket with a probability of 37%

Regarding the physical configuration, two considerations related to maintenance are concerning, inspection and access. The screws are set into a hole, meaning they need to be viewed straight on where the carriage, and therefore the entire hull, is located and moving. To safely perform one possible maintenance task, say, verifying that the torque of the screws are within an acceptable range, the carriage and hull would need to be locked in place to allow maintenance personnel to access the screws. Suppose this is deemed unsafe to perform during *in situ* maintenance; in that case, the device will need to be removed from the water onto a maintenance vessel or towed to shore, increasing the operations cost to perform this relatively frequent maintenance task. If the engineers decide the maintenance task is to perform a visual inspection of the screws, possibly to verify a mark that indicates rotation of the screw, the hull would still need to be locked into place in multiple positions to allow for visual inspection of all screws.

Noting the potential maintenance difficulties, possible options to explore include: repositioning the screw to a more accessible location, removing the need for the carriage rail system – and therefore the screws – by changing how the spar and hull move relative to one another, or securing the rail to the spar through other means. Expanding on the last two options, is there an alternative way to secure the spar to the hull that doesn't necessitate screws and are there better options than screws if the rail design is kept?

The carriage rail itself is described by the function *Channel/Guide/Translate* and the flow *Energy/Mechanical Energy/Translational Energy*, and is predicted by the Full Model to be:

1. FlowL2$Solid < 1 \rightarrow$ Yes
2. FFL2$GuideGas = 0 \rightarrow$ Yes
3. FcnL2$Indicate = 1 \rightarrow$ No
4. Component_or_Assembly = Component $\rightarrow$ Yes
5. FcnL2$Stop < 1 \rightarrow$ Yes
6. FFL2$Convert2HydE = 0 \rightarrow$ Yes
7. The carriage rail is predicted to be in the 2.5 OOM bucket with a probability of 41%

If the rail is removed entirely and a linear sleeve bushing or rollers directly onto the spar itself are used instead, what would be the impact? Unfortunately, both of these options are functionally the same as the original method so the model gives no indication that there would be any difference.

The second option could lead to stitch welding the rail to the spar. This would be functionally described by *Connect/Couple/Link* and *Material/Solid/Object*. When passed through the Full Model this would result in:

1. FlowL2$Solid < 1 \rightarrow$ No

2. FFL1$CMM \geq 1 \rightarrow$ No
3. FcnL1$Provision \geq 1 \rightarrow$ No
4. FF_Count $\geq 2 \rightarrow$ No
5. FcnL2$Secure = 1 \rightarrow$ No
6. FcnL2$Position = 1 \rightarrow$ No
7. FcnL1$Connect \geq 1 \rightarrow$ Yes
8. The stitch welding is predicted to be in the 4 OOM bucket with a probability of 40%

This redesign exploration using the Full Model highlights two insights into the system as designed. First, the design decision to use the spar (via the rail, bushing, etc.) to guide the movement of the hull has locked in some maintenance requirements. An explanation of this function-maintenance relationship may be that the function *Guide Mechanical Energy*, regardless of how its implemented, often requires maintenance in the 2.5 OOM bucket. This maintenance is likely to be related to the greasing of bearings or condition inspections for early signs of wear. The second insight allowed for design effort to be focused on the aspects that could easily be improved. Screws versus stitch welding are functionally different and this difference results in a 1.5 OOM difference in preventative maintenance frequency. Not only is the frequency different, the maintenance action itself is likely to be different. A series of stitch welds are easier to inspect for signs of failure, as once one stitch fails it is easily recognizable but still gives time before the adjacent welds fail.

4.4 Discussion

This section includes a discussion of the results, broken down into three sections, each concerning one of the key findings. First, the existence of function-maintenance relationships are discussed. Then the modeling and generalization of those relationships is addressed. Lastly, the usefulness derived from modeling and understanding function-maintenance relationships is discussed.

4.4.1 Function-Maintenance Relationships

Through the data mining and modeling effort, it was evident that certain functional information had significant and consistent implications in the preventative maintenance demands of the physical parts embodying that functional description. Sample part #42 explored in Section 4.1.2.1, shows one such significant and consistent relationship. The

Visual Warning System assembly and the functional information used to describe it (*SignalFcn/Indicate/Display* and *SignalFlow/Status/Display*), was consistently predicted to require preventative maintenance in the most frequent maintenance bucket (2 OOM) by both the Full and Flow Models. The level-2 function *Indicate* led to a 72% chance (in the Full Model) and the level-2 flow *Status* led to a 68% chance (in the Flow Model) of accuracy in the 2 OOM prediction.

Similarly, sample part #1,826 from Section 4.1.2.5 was predicted by both models to be in the 2.5 OOM bucket. While the actual maintenance interval for this data point was in the 3 OOM bucket, a consistent and reasonable prediction is shown. Additionally, it's worth noting that part #1,826 has a second maintenance task that *IS* performed with a frequency that falls into the 2.5 OOM bucket. As for significance, the Full Model provides a probability of 50% that the part is in the 2.5 OOM bucket and a 30% chance that the part is in the 3 OOM bucket.

Though some function-maintenance relationships are captured by this model, many functions are not represented in the model. The unrepresented function-maintenance relationships may or may not be captured in future work. It may be the case that only some functions have a consistent and significant maintenance relationship, whereas the rest may have large uncertainty bands. Despite this possibility, knowledge of the captured function-maintenance relationships is still useful. As uncertainty reduction is a major goal during early design, understanding which functions carry large ranges of uncertainty versus those whose function-maintenance relationships are better understood can guide design decisions.

4.4.2 Function-Maintenance Model

The choice to utilize a decision tree model should not obscure that a whole family of supervised classification algorithms could be implemented to illuminate the function-maintenance relationships. Functional analysis has proven capable of capturing and distilling design information that can then be utilized as features to inform a model. Combined with the consistent and significant relationships uncovered, this points to the validity of supervised classification algorithms in the modeling of these relationships. Through the grounding effort using a Datsun truck as well as through the verification effort using LUPA, the model was shown to contain a degree of external validity. The model produced reasonable predictions over 60% of the time on the Datsun truck, and over 50% of the predictions made on the LUPA data was within ± 0.5 OOMs of the expert designer's predictions.

With more diverse training data incorporated into the model, discussed in Section 4.5.1, a updated supervised classification model is likely to improve on this model's external validity. However, this model has shown that some relationships captured are externally valid and can be leveraged outside the training data's sector.

4.4.3 Applying the Function-Maintenance Model

By passing sample parts through the Full and Flow Model and evaluating the results, we come to the aspect that makes this work useful for design engineers of novel systems. The function-maintenance relationships identified and modeled using decision trees, are comprehensible and can be used with a sufficient degree of confidence so as to support and inform early design decisions. This cumulative finding is described by a few key understandings developed thus far:

- The structure of the decision tree model and how predictions are made.
- The factors that influence a prediction.
- How to assess consistency and significance of a relationship.
- The limitations of the model, when they arise, and how they influence predictions.

An understanding of the structure of a decision tree and the series of prompts that result in a prediction are necessary for any usage of the model. The factors that influence a prediction provide the designer with insight into the "thought process" of the model. If the designer knows that some aspect of a functional requirement is relevant to its maintenance demand, but they can see that the model doesn't account for it, they know to regard that prediction more cautiously. Sample part #42 (Section 4.1.2.1) exemplifies the value in understanding consistency; the Full and Flow Model predict the part to be in the 2 OOM bucket with a probability of 72% and 68%. This should tell the designer that this prediction is likely to be accurate. Sample part #300 (Section 4.1.2.4) shows the value in understanding significance of the function-maintenance relationships. Following the Full Model, the part comes to its third prompt, "Does the level-1 function of *Provision* occur greater than or equal to one time?" The negative response gives a prediction of 3.5 OOM with a probability of 41% and a probability of 39% for the 4 OOM bucket. If the response was instead in the affirmative – as is the case for part #300 – then the prediction if 2.5 OOM with a probability of 50%. Understanding this can inform the designer on the implications of various functional alternatives. The final understanding required to effectively utilize this model is knowing when not to use this model. Sample part #6's Flow Model prediction is one such case (Section 4.1.2.2). The series of prompts from the Flow Model captured very little positive information about the part and the predicted leaf node is a "catch-all" node that is equivalent to a null hypothesis. Theme #1 and #2, discussed in Section 4.2.2, and the artifacts discussed in more detail in Section 4.5.2 provide other examples of the importance of understanding the model's limitations.

There are three primary drivers of uncertainty and unknowns that exist in the environment that this model seeks to operate within. First, this research is in the initial stages of identifying and modeling function-maintenance relationships. The extent and depth of function-maintenance relationships are not fully captured, modeled, or understood in this research and more work is required. Second, the early design phase of the engineering design process is notable for its high degree of uncertainty. Third, compounding with the general lack of information in early design, novel technologies have an even higher degree of uncertainty as they lack comparisons and design convergence. As a result, a robust understanding of the model is essential for its practical application.

4.5 Limitations

4.5.1 Training Data

The first and most obvious limitation that I believe has the greatest potential to improve the model is a lack of training data. The model has insufficient data at the extreme ranges of maintenance frequency. This could be indicative of the nature of long field-life systems and the ranges of maintenance intervals that they occupy. It is possible that scheduled maintenance intervals rarely extend past a 2 OOM range. The "missing" parts could be sufficiently reliable so as to not need maintenance within the service life of the system. The remaining parts may also be hard to access or technically difficult to assess the degradation of and are not worth the reduced operational availability of scheduled maintenance. Operational availability or service costs may necessitate a run-to-failure strategy with corrective maintenance performed as needed, if this is the case for parts outside of the 2 OOM range, this model would not capture them.

Though the lack of training data above the bulk 2 OOM range is interesting, the extreme infrequency of maintenance makes them of lower concern than the data points, or lack thereof, below the majority of the data. Using the training data for reference, the 0 and 1

OOM buckets represent the 0 to 6 and 7 to 55 flight hour maintenance intervals. Generally, accepted logic would tell us that we should design systems to require maintenance as infrequently as possible whilst ensuring the system remains in the required operational state. Assuming the designers of the aircraft and its engines followed this logic during design, we may conclude that the items in the highest frequency buckets perform functions that are so essential to the operation of the system, that they must be assessed at that frequency. If this is the case, the lack of data points in the most frequent buckets constitute cause for concern. The model produced may not be capturing the functions that experienced and validated designers have deemed so vital to continued operation that they must be inspected as frequently as every 1.5 flight hours.

Conversely, this logic may be an incorrect assumption in this situation. It is conceivable that the functions with the shortest maintenance intervals may not be the most vital, but are functions that require the least time and resources required to maintain, similar to the side mirrors seen in Table 4.8 of the grounding evaluation. Regardless of the reason, additional training data stands to clear any confusion.

4.5.2 Artifacts of Training Data

Two artifacts are known to pervade the model: the emphasis on *Guide Gas* rooted in the aerodynamic requirement of aircraft design and the treatment of parts that contain the level-2 flow *Solid* being heavily influenced by the significant number of structural members found in the aircraft. The influence of these artifacts is that functionally similar parts are likely to be sorted into the buckets that correspond to the exhaustive inspections undertaken to certify external aerodynamic surfaces or structural members.

The addition of more diverse training data would clarify the artifact that stems from the quantity and specific timing of structural member inspections. Two results are possible, the new training data will either support or contradict the relationship as seen from the current training data. If the additional data supports the relationship between parts that serve a function similar to that of structural members (*Support/Stabilize/Stabilize3* and *Material/Solid/Object*) and the 4 OOM bucket, then this “artifact” is miss-classified and is capturing a consistent and significant relationship. If the additional data contradicts this relationship, then the new model would de-emphasize the consistency of this relationship and indicate that this relationship was not externally valid. Until a new model can be built using more diverse training data, it’s impossible to tell if this “artifact” is unique to aircraft design or if it's generalizable to other long field-life systems.

The artifact caused by the aerodynamic requirements of aircraft design seems less likely to be externally valid. Using a MRE as an example, the external surfaces of a wave energy converter may be better described as *Stop Liquid*, leaving *Guide Gas* to explain pneumatic or HVAC systems. This would result in the model giving unjustified or unreasonable predictions. Additional training data would help to reduce the influence of this artifact by diluting the particular design requirements (aerodynamics) of aircraft design on parts that perform functions related to *Guide Gas*.

4.5.3 Evaluation

The evaluation efforts uncovered three themes that highlighted the model's limitations. The easiest to address is theme #3, discussed in Section 4.2.2.3. This relates to misusing the model; when parts are predicted on that would not be a maintenance concern the model's predictions will not be reasonable. The example given was a cup holder. This is not the type of part this model is intended for.

Theme #1 is explored in Section 4.2.2.1 and is seen again in Section 4.2.3; it relates to the model's inability to capture mechanical simplicity and the nuances of a function performed by an operation of the part (a gear box that *Changes Mechanical Energy*) versus a function performed by the inherent characteristics of a part (a damper plate *Changes Mechanical Energy* by its high drag coefficient and mass). Theme #2 is similar but pertains to the breadth of parts that could be used to accomplish a function.

The limitations related to theme #1 and #2 is that the model's use of functional information does not always capture important factors that impact a part's maintenance demand. This is both a feature and bug. By restricting the model's input to functional information, it's able to be used earlier in the design process but fails to capture some aspects of design that may impact O&M considerations (e.g. the size and weight of a part that performs a function could vary greatly, thus the operational requirements to perform its maintenance could vary greatly as well).

Through machine learning calibration, explored in Section 4.2.1, the limitations of the model's accuracy were seen. The Full Model – which requires both function and flow information – has an accuracy of 40.2% and an expanded accuracy of 79.4%. The Flow Model – potentially useful earlier as it relies primarily on flow information – has an accuracy of 37.5% and an expanded accuracy of 76.9%. The macro-average PPV of the Full Model was 42.3% and 83.1% for the average expanded PPV. The Flow Model's average PPV was 46.2% and the average expanded PPV was 79.4%. These metrics highlight the superiority of the Full Model, but at the expense of requiring more information on the system. As there is no threshold for machine learning metrics that define a "good" model versus a "bad" model, the goodness of these models must be taken in the context of their application and the impacts of their results.

Considering the high degree of uncertainty present in the application context – early design of a novel system – a judgement by the designer is necessary to say if an average PPV or expanded average PPV of 42.3% and 83.1% is acceptable. Is a 42.3% chance for a prediction to be in the bucket useful during early design of a novel system? What about a 83.1% chance of a prediction to be within ± 0.5 OOMs of its predicted bucket? That information may be helpful despite the values, but if the risk of an incorrect prediction is significant, that would reduce the goodness of the model. So what is the impact of an incorrect prediction? As the intent of the model is to guide and inform design decisions as opposed to making design decisions, the potential risk associated with an incorrect prediction is mitigated by an informed understanding of the model and the designer's own judgement.

This is not to say the model cannot be improved. Increasing overall accuracy and the PPV of each class would significantly improve the potential value this model could provide. Additional training data could result in an increase in these metrics or highlight an upper limit of the strength of connection between functions and maintenance.

4.6 Improving the Work

As is often the case, hindsight is 20/20. This section addresses improvements to the current work that, had I been wiser, I would have implemented. Future work that falls outside the scope of this project is presented in Section 5.3.

4.6.1 Intra-Coder Reliability

Following the methods described in Section 3.3, all functional decomposition was performed at least once using the final guidelines. Though a consistent process was applied, without a separate decomposition effort, there was no verifiable degree of intra-coder reliability. Ideally, two functional decomposition efforts would have been undertaken during the course of this work. By doing so, the difference between the first and second modeling effort could have been captured and quantified, then used to prove a degree of internal consistency.

4.6.2 Documentation of Maintenance Intervals

As currently documented, each data point is defined by a maintenance action at a certain frequency, not by the part of concern. This most closely aligned with the structure and presentation of information of the maintenance documents. Organizing the mined data by part may have allowed additional insights to be captured without requiring significant work. For example, data point #1,225 and #1,456 are both the aileron balance tab hinge, the maintenance interval of these two data points were 6,000 and 12,000 hours. As is currently recorded, each of these data points are included in the training data, as if to say that this part needs maintenance every 6,000 hours as well as every 12,000 hours. It would have been more accurate to say that this part had a maintenance interval of every 6,000 hours, as 12,000 would be captured by that as well. That example is fairly clear cut, but others would have proven more difficult. Data points #645 and #897 are both the aileron bearings, that have a maintenance interval of every 5,760 hours and every 7,200 hours. These intervals correspond to the 6,000-landing/12-year (when both intervals are converted to hours using Table 3.4, 12 years is more restrictive at 5,760 hours) and the 7,200-hour/12-year (7,200 hours is used as it was given directly in hours) inspections. What interval should be used for the aileron bearings in this case? Only muddying the waters further is the fact the the aileron bearings are addressed two more times, except this time when explicitly removed from the wing, at the 6,000 and 12,000 hour intervals. There are some clear improvements to be made in how the data was documented – combining 6,000 and 12,000 hour parts – but other improvements would require additional consideration. The following section describes one such consideration which could address the less clear cut cases.

4.6.3 Documentation of Preventative Maintenance Type

When documenting data in Table 3.2, the maintenance task served two purposes. The part of concern was extracted from it and it provided additional context to understand the part of concern's function. A missed opportunity was to identify the type of preventative maintenance action to be performed. All five types of preventative maintenance may not have been decipherable, but at the very least, the distinction between time, failure finding, and condition based maintenance is possible. If the maintenance action involves a replacement (e.g. replace the seal) or performing a specific task (e.g. lubricate) they are easily identifiable as a time based task (part #6 in Table 3.2). Failure finding tasks would include tests to verify the proper operation of a part or system (part #42 in Table 4.1). If the maintenance action involves an inspection (e.g. inspect for signs of corrosion, cracks...) it aligns with a condition based task (part #1,999 in Table 3.2). This distinction may aid in the aileron bearing dilemma discussed above. The function of the bearing (*Guide Solid*) may have a defined time based maintenance interval (e.g. packing the bearing with lubricant) and a distinct condition based maintenance interval (e.g. inspecting for signs of fretting). This additional information may address theme #1 (discussed in Section 4.2.2.1), as functions performed by the inherent nature of a part may be distinguishable by a tendency towards condition based maintenance tasks. Additionally, the predictions of the model could include not only a frequency, but the type of maintenance to be performed as well. A prediction similar to "This part has a probability of 75% to be in the 2.5 OOM bucket and the maintenance action has a 50% chance to be an inspection/condition check" would be extremely useful. In the case of an ocean deployed MRE system, the difference between a functional check and the removal/replacement of a part could have vastly different implications. A designer with knowledge of this could implement systems to perform the functional check via remote sensors whereas the latter may require a specialized vessel to be hired and entail a costly and time consuming operation. Having a general knowledge of the intervals when major maintenance actions will be required can inform design decisions. For example, if it is known that a major maintenance action is required at a certain interval, the positioning and accessibility of parts requiring maintenance at that same interval may be of lower concern. If the device is onshore for maintenance already, all other maintenance actions become substantially easier.

4.6.4 Function Through Action Versus Inherent Characteristics

The previous section described a possible improvement to this work that could address theme #1. A more direct method to address theme #1 would be to document whether a function is performed by an action of the part or an inherent characteristic of the part. The artifact caused by the aerodynamic requirements of aircraft design, discussed in Section 4.5.2, could be distinguished from functionally similar parts by the addition of this feature into the model.

Better to make a poor guess than to assume it's not relevant, because that's a big guess.

Jochem Weber

Chapter 5: Conclusion

5.1 Overview

In the design of products for which lifetime costs will be dominated by O&M, early maintenance consideration is essential to the quality of design, number and impacts of late stage design changes, and commercial success. Many developers of novel long field-life products fail to inform themselves and implement known best practices to estimate O&M costs. This is not entirely their fault though. Significant barriers exist including the business environment, lack of historic data, and minimal operational experience. To hurdle these barriers, support is required to justify the business case of a thorough early design process and to leverage the success of mature sectors through resources that are accessible to designers and can accommodate the uncertainty inherent when exploring the unknown.

To provide that support, functional analysis was proposed as the means to distill early design decisions into their elementary tasks, or functions. Consistent and generalizable relationships between the elementary functions a part performs and its preventative maintenance requirements were then hypothesized. Machine learning was proposed as a method to capture this relationship and allow for its application to guide and inform maintenance considerations during early design. The function-maintenance model developed in this research is best understood as a family of decisions trees that predicts the maintenance frequency of parts based on the functions they perform. The creation and exploration of these models has resulted in three key findings, discussed below.

5.2 Key Findings

The first two findings regard the hypothesized function-maintenance relationships and the final finding pertains to the model itself. Each finding can be summarized by one word and when combined, gives: **EXIST**, **GENERAL**, and **USEFUL**.

The first finding is that function-maintenance relationships **EXIST**; that there are relationships between the description of *WHAT* a part does and its relative maintenance demand. This finding is seen in the significant and consistent relationships, explored in

Section 4.1, that the models are able to capture.

Second, function-maintenance relationships can be captured, modeled and **GENERLIZED** to other long field-life systems. The grounding and verification efforts, discussed in Section 4.2, showed that the model gives externally valid, reasonable predictions when confronted with new data.

The last finding stems from an understanding of the first two findings and is the most important to the practical application of this research. The methods and models used in this research result in a function-maintenance model that is comprehensible and **USEFUL** in building confidence with its users and supporting early design decisions. A sample use case was described to show how a design engineer could incorporate the model into their design processes to inform maintenance considerations. The verification effort (Section 4.2.3) pitted the model against the best-case scenario for maintenance predictions for a novel long field-life system and gave reasonable results. When applied to a redesign effort (Section 4.3), the model gave reasonable predictions that allowed the focusing of design effort. Furthermore, the model provided useful and actionable information on the impacts of the functional design and guided informed design changes that resulted in a 1.5 OOM (Section 4.3.1) decrease in maintenance frequency of that part.

5.3 Future Work

This section addresses possible avenues of future work that fall outside the scope of this project. Possible improvements to this work are explored in Section 4.6.

5.3.1 More Training Data

More training data, both in terms of quantity and diversity is a clear step towards better understanding function-maintenance relationships. More training data generally leads to improvements in machine learning models as the sample (the training data) is more likely to be representative of the population. The diversity of additional training data is important. Training data from other long field-life systems would improve the model in a few ways.

More diverse training data would improve the model's external validity. It may be the case that parts that perform the function *Indicate* are not generally maintenance intensive as this model predicts. This relationship may only be valid within the aviation field. If function-maintenance data from another device in another field (say a piece of heavy equipment, like a crane) does support this relationship, that would improve the external validity of that relationship.

The artifacts identified in the current model could be clarified and be mitigated. The artifact related to structural members may be address similarly to the point above; but it could be the case that it and the aerodynamics artifact are in fact by-products from the aviation training data. Data from another field would dilute the impact of these artifacts, allowing the model to more accurately capture the nature of long field-life systems in general and not just the specific nature of aircraft.

5.3.2 Machine Learning Algorithms

More advanced machine learning techniques could be better codify function-maintenance relationships. Random forest models are less susceptible to overfitting as they make each prediction based on the most common answer given by a large number of decision trees each trained on a different subset of features. The benefit of this is that if many decision trees (a random forest) each come to the same maintenance prediction by asking different questions it is more likely that the conclusion is valid. A neural network is better able to handle the large number of features recorded from the maintenance data. The Full Model makes predictions based on, at most, 7 features of the 168 features provided. Neural networks could take advantage of more features to refine its predictions.

This avenue of work is likely to result in establishing function-maintenance relationships with more confidence, but they come at the cost of intelligibility. Sample part #1,236, examined in Section 4.1.2.3, exemplifies what is lost with a less intelligible, albeit more accurate, model. If this prediction was made using a neural network, it may have been made with a higher confidence but the knowledge that the model considered the question "Is this part a component or assembly?" would be lost. The Full and Flow Model shows us that this was considered, because of that we know that the prediction is likely to be referencing the interval required for a full inspection or breakdown of the assembly. With a neural network based function-maintenance model, the designer would not be aware of that information and may take the prediction as the components that comprise the rudder assembly only require preventative maintenance in the 4 OOM bucket. Due to the high degree of uncertainty that exists when utilizing this model, the ability for a designer to pass judgement on the model's predictions is essential.

5.3.3 Additional Features

Additional training features could be explored without significantly delaying the early design usage of the model. Input and output flows of each function could be modeled to capture dependencies, a known factor that influences maintenance and reliability. Additionally, the impact of adjacent functions could be modeled through additional features. Environmental conditions and their impact on functions may improve the fidelity of predictions.

5.4 Parting

Without early design tools intended for use with novel systems, developers will continue to be limited in their ability to make informed decisions during the earliest stages of design. As a result, costly, late stage design changes are more likely and the final products developed are not likely to be the best the developer is capable of. Though tools intended for use during early design of novel systems fight an uphill battle against the unknown and a lack of experience, their development and implementation stand to significantly benefit the developers who fight the same uphill battle. The use of function-maintenance relationships to inform early design decisions is one such tool. The initial steps that this research has taken in characterizing these relationships and establishing external validity is essential if function-maintenance relationships are ever to be useful in the design of novel systems.

Bibliography

[1] D Ullman, *The mechanical design process 4^{th} edition* (McGraw-Hill, New York, NY, 2010).

[2] J Carroll, A McDonald, and D McMillan, "Failure rate, repair time and unscheduled o&m cost analysis of offshore wind turbines", Wind Energy **19**, 996–1177 (2016) `10.1002/we.1887`.

[3] I Dinwoodie, D McMillan, M Revie, I Lazakis, and Y Dalgie, "Development of a Combined Operational and Strategic Decision Support Model for Offshore Wind", Energy Procedia **35**, edited by JOG Tande, T Kvamsdal, and M Muskulus, 157–166 (2013) `10.1016/j.egypro.2013.07.169`.

[4] R Tiusanen, E Heikkilä, M Räikkönen, and T Välisalo, "System approach to reliability engineering - case: wave energy converter", in Eproceedings of esrel2020 psam15, eProceedings of ESREL2020 PSAM15 (2020), `10.3850/978-981-14-8593-0`.

[5] V Neary, P Kobos, S Jenne, and Y Yu, *Levelized cost of energy for marine energy conversion (mec) technologies*, tech. rep. SAND2016-8298C (Sandia National Laboratories and National Renewable Energy Laboratory, Albuquerque, NM, 2016).

[6] B Blanchard, D Verma, and E Peterson, *Maintainability: a key to effective serviceability and maintenance management* (Wiley-interscience, New York, NY, 1995).

[7] *Engineering design handbook: maintainability engineering theory and practice*, tech. rep. AMCP 706-133 (U.S. Army Material Command, Alexandria, VA, Jan. 1976).

[8] B O'Halloran, C Hoyle, I Tumer, and R Stone, "The early design reliability prediction method", Research in Engineering Design **30**, 489–508 (2019) `10.1007/s00163-019-00314-8`.

[9] G Dieter and L Schmidt, *Engineering design, 4^{th} edition* (McGraw-Hill, New York, NY, 2009).

[10] Z Tjiparur and G Thompson, "Review of maintainability design principles and their application to conceptual design", Journal of Process Mechanical Engineering **218**, 103–113 (2004) `10.1115/1.1862678`.

[11] A Desai and A Mital, “Improving maintainability of products through the adoption of a comprehensive design for maintainability method”, International Journal of Industrial Engineering **17**, 103–114 (2010) `10.23055/ijietap.2010.17.2.168`.

[12] B Dhillon, *Engineering maintainability* (Gulf publishing company, Houston, TX, 1999).

[13] G Pahl, W Beitz, J Feldhusen, and K Grote, *Engineering design: a systemic approach* (Springer London, London, UK, 2007).

[14] D Ullman, *The mechanical design process 6th edition* (David Ullman Sixth Edition, 2017).

[15] F Phillips and R Srivastava, *Committed costs vs. uncertainty in new product development*, tech. rep. WP-1993-02-01 (IC2 Institute, The University of Texas at Austin, Austin, TX, Jan. 1993).

[16] J Barton, D Love, and G Taylor, “Design determines 70% of cost? a review of implications for design evaluation”, Journal of Engineering Design **12**, 47–58 (2001) `10.1080/09544820010031553`.

[17] K Ulrich and S Pearson, “Does product design really determine 80% of manufacturing cost?”, 1993.

[18] T James, O Kevin, and W Kristin, “A comparison of design decisions made early and late in develop,emt”, in Proceedings of the 21st international conference on engineering design, Design Organisation and Management (2017).

[19] *Prototyping has benefited aquisition programs, but more can be done to support innovation initiatives*, tech. rep. GAO-17-309 (U.S. Goverrnment Accountability Office, Washington D.C. US, 2017).

[20] G Boothroyd, “Product design for manufacture and assembly”, Computer-Aided Design **26**, 505–520 (1993) `10.1016/0010-4485(94)90082-5`.

[21] H Stoll, “Design for manufacture”, Manufacturing Engineering **100**, 67–73 (1988).

[22] M Andreason, S Kahler, and T Lund, *Design for assembly* (Springer, New York, NY, 1983).

[23] G Kobe, “Dfma: design for manufacture and assembly”, Automotive Industries, 34–38 (1990).

[24] T Kuo, S Huang, and H Zhang, “Design for manufacture and design for ‘x’: concepts, applications, and perspectives”, Computers and Industrial Engineering **41**, 241–260 (2001) 10.1016/S0360-8352(01)00045-6.

[25] G Huang, ed., *Design for x: concurrent engineering imperatives* (Chapman & Hall, London, UK, 1996).

[26] G van Bussel and C Schöntag, “Operation and maintenance sspects of large offshore windfarms”, in Proceedings of ewec (1997), pp. 272–275.

[27] S Gardner and D Sheldon, “Maintainability as an issue for design?”, Journal of Engineering Design **6**, 75 (1995) 10.1080/09544829508907905.

[28] R Billinton and J Krasnodebski, “Practical application of reliability and maintainability concepts to generating station design”, IEEE Transations on Power Apparatus and Systems **PAS-92** (1973).

[29] K Otto and K Wood, *Product design: techniques in reverse engineering and new product development* (Prentice Hall, Upper Saddle River, NJ, 2001).

[30] K Otto and K Wood, “Product evolution: a reverse engineering and redesign methodology”, Research in engineering design **10**, 226–243 (1998).

[31] N Crilly, “Fixation and creativity in concept development: the attitudes and practices of expert designers”, Design Studies **38**, 54–91 (2015) 0.1016/j.destud.2015.01.002.

[32] T Kurtoglu, I Tumer, and D Jensen, “A functional failure reasoning methodology for evaluation of conceptual system architectures”, Research in Engineering Design **21**, 209–234 (2010) 10.1007/s00163-010-0086-1.

[33] D Jensen, O Bello, C Hoyle, and I Tumer, “Reasoning about system-level failure behavior from large sets of function-based simulations”, Artificial Intelligence for Engineering Design **28**, 385–398 (2014) 10.1017/S0890060414000547.

[34] Y Guillouët and F Torres, “Identifying critical failure-propogation in function models of complex systems”, in Proceedings of the 31st european safety and reliability conference (2021).

[35] R Wichmann, K Gericke, and B Eisenbart, “Analysing risk to function fulfilment: applying the function integrity diagnosis and documentation method”, Journal of Engineering Design **32**, 346–373 (2021) 10.1080/09544828.2021.1913485.

[36] I Tumer and R Stone, “Mapping function to failure mode during component development”, Research in Engineering Design **14**, 25–33 (2003) 10.1007/s00163-002-0024-y.

[37] J Fahl, T Hirschter, S Haag, T Staiger, and A Albers, “Functions in the early phase of product development: a systemic literature review”, in Ieee international symposium on systems engineering (2020).

[38] J Collins, B Hagan, and H Bratt, “The failure-experience matrix: a useful design tool”, in Proceedings of design engineering technical conference (1975).

[39] G Pahl and W Beitz, *Engineering design: a systemic approach* (Springer Londong, London, UK, 1984).

[40] J Hirtz, R Stone, A McAdams, S Szykman, and K Wood, “A functional basis for engineering design: reconciling and evolving previous efforts”, Research in engineering design **13**, 65–82 (2002) 10.1007/s00163-001-0008-3.

[41] S Szykman, J Racz, and R Sriram, “The representation of function in computer-based design”, in Proceedings of the 1999 asme design engineering technical conferences (1999).

[42] A Little, K Wood, and D McAdams, “Functional analysis: a fundamental empirical study for reverse engineering, benchmarking, and redesign”, in Proceedings of the asme design theory and methodology conference (1997).

[43] Y Deng, “Function and behavior representation in conceptual mechanical design”, Artificial intelligence for engineering design, analysis and manufacturing **16**, 343–362 (2002) 10.10170S0890060402165024.

[44] A Chakrabarti, “Supporting two views of function in mechanical desings”, in Workshop on functional modelling and teleological reasoning, 15$^{\text{th}}$ aaai national conference on artificial intelligence (1998).

[45] L Chittaro and A Kumar, “Reasoning about function and its applications to engineering”, Artificial Intelligence in Engineering **12**, 331–336 (1998) 0954-1810/98/$19.00.

[46] S Burger, *Introduction to machine learning with r* (O’Reilly Media, Inc., Sepastopol, CA, 2018).

[47] J Panchal, M Fuge, Y Liu, S Missoum, and C Tucket, “Special issue: machine learning for engineering design”, Journal of Mechanical Design **141** (2019).

[48] M Fuge, B Peters, and A Agogino, "Machine learning algorithms for recommending design methods", Journal of Mechanical Design **136**, `10.1115/1.4028102` (2014) `10.1115/1.4028102`.

[49] Y Xiong, P Duong, D Wang, S Park, Q Ge, R N., and D Rosen, "Data-driven design space exploration and exploitation for design for additive manufacturing", Journal of Mechanical Design **141**, `10.1115/1.4043587` (2019) `10.1115/1.4043587`.

[50] Y Gao, Y Matsunami, S Miyata, and Y Akashi, "Operational optimization for off-grid renewable building energy system using deep reinforcement learning", Applied Energy **325**, `10.1016/j.apenergy.2022.119783` (2022) `10.1016/j.apenergy.2022.119783`.

[51] N Dehaibi, N Goodman, and E MacDonald, "Extracting customer perceptions of product sustainability from online reviews", Journal of Mechanical Design **141**, `10.1115/1.4044522` (2019) `10.1115/1.4044522`.

[52] D Suryadi and H Kim, "A data-driven methodology to construct customer choice sets using online data and customer reviews", Journal of Mechanical Design **141**, `10.1115/1.4044198` (2019) `10.1115/1.4044198`.

[53] A Raina, C McComb, and J Cagan, "Learning to design from humans: imitating human designers through deep learning", Journal of Mechanical Design **141**, `10.1115/1.4044256` (2019) `10.1115/1.4044256`.

[54] *Gates learjet 25b/c/d/f maintenance manual*, version 23, Gates Learjet Corporation (Wichita, KS, 1987).

[55] *Cj610 turbojet engines maintenance manual*, version 11, General Electric Aircraft Engine Group (Evendale, OH, 1976).

[56] *Cj610 turbojet engines cf700 turbofan engunes illustrated parts catalog*, version 22, General Electric Aircraft Engine Group (Evendale, OH, 1976).

[57] E Avallone and T Baumeister III, eds., *Marks' standard handbook for mechanical engineers, 10th edition* (McGraw-Hill, New York, NY, 1996).

[58] J Hobbs and V Kreinovich, "Optimal choice of granularity in commonsense estimation: why half-orders of magnitude?", International Journal of Intelligent Systems **21**, 801–873 (2006) `10.1002/int.20164`.

[59] A Liaw and M Wiener, "Classification and regression by randomforest", R News **2**, 18–22 (2002).

[60] T Therneau and B Atkinson, *Rpart: recursive partitioning and regression trees*, R package version 4.1.19 (2022).

[61] S Milborrow, *Rpart.plot: plot 'rpart' models: an enhanced version of 'plot.rpart'*, R package version 3.1.1 (2022).

[62] M Bohm and R Stone, "Product design support: exploring a design repository system", in Proceedings of asme international mechanical engineering congress 2004 (2004).

[63] M Bohm, R Stone, T Simpson, and E Steva, "Introduction of a data schema to support a design repository", Computer-Aided Design **40**, 801–811 (2008) `10.1016/j.cad.2007.09.003`.

[64] M Bohm, R Stone, and S Szykman, "Enhancing virtual product representations for advanced design repository systems", Journal of Computing and Information Science in Engineering **5**, 360–372 (2005) `10.1115/1.1884618`.

[65] S Szykman, "Architecture and implementation of a design respository system", in Proceedings of design engineering technical conference 2002 (2002).

[66] J Murdock, S Szykman, and R Sriram, "An information modeling framework to support design databases and repositories", in Proceedings of asme 1997 design engineering technical conferences (1997).

[67] K Carley, *Validating computational models*, tech. rep. CMU-ISR-17-105 (Institute for Software Research, Pittsburgh, PA, Apr. 2017).

[68] B Bosma, C Beringer, M Leary, and B Robertson, "Design and modeling of a laboratory scale wec point absorber", in Proceedings of the 14^{th} european wave and tidal energy conference (2021).