ON THE APPLICATION AND COMPARISON OF CONDITION-BASED MAINTENANCE PROGNOSTICS APPROACHES: LOGICAL ANALYSIS OF DATA, ARTIFICIAL NEURAL NETWORKS, AND PROPORTIONAL HAZARDS MODELS

By

Hanna Lo

Submitted in partial fulfillment of the requirements
for the degree of Master of Applied Science

At

Dalhousie University
Halifax, Nova Scotia
August 2016

TABLE OF CONTENTS

LIST OF TABLES iv

LIST OF FIGURES v

ABSTRACT vi

LIST OF ABBREVIATIONS AND SYMBOLS USED vii

ACKNOWLEDGEMENTS xii

CHAPTER 1 INTRODUCTION 1

1.1 MOTIVATION AND OBJECTIVES 2

1.2 THESIS LAYOUT 2

CHAPTER 2 LITERATURE REVIEW AND BACKGROUND IN CBM 3

2.1 CONDITION-BASED MAINTENANCE 3

2.1.1 General Concepts 3

2.1.2 Methods and Techniques 4

2.2 LOGICAL ANALYSIS OF DATA 6

2.2.1 Basic Stages 6

2.2.2 Application 10

2.3 NEURAL NETWORKS 10

2.3.1 Basic Components and Architecture 11

2.3.2 Application 14

2.4 PROPORTIONAL HAZARDS MODEL 15

2.4.1 Definitions and Concepts 15

2.4.2 Application 17

CHAPTER 3 METHODOLOGY 18

3.1 DATA PREPROCESSING 18

3.2 LAD FORMULATION 22

3.3 NEURAL NETWORK DESIGN 28

3.4 PHM TESTING 32

CHAPTER 4 EXPERIMENTAL RESULTS AND ANALYSIS 34

4.1 RESULTS 34

4.1.1 LAD 36

4.1.2 Neural Networks 37

4.1.3 PHM 42

4.2 COMPARISON OF EXPERIMENTAL RESULTS 43

4.2.1 Evaluation over Error 43

4.2.2 Evaluation over Half-Life Error 48

4.2.3 Evaluation over Cost Score 51

4.2.4 Discussion of Results 55

4.3 INDUSTRIAL APPLICATION 58

4.3.1 The MacKay Bridge Problem and Background 58

4.3.2 Laboratory vs. Field Data 59

4.3.3 Experimental Analysis 62

4.3.4 Discussion and Comparison of Results 68

CHAPTER 5 CONCLUSIONS 72

REFERENCES 74

APPENDIX A Neural Network Matlab Design Script 80

APPENDIX B Matlab Training Algorithms 84

APPENDIX C Model Results 85

APPENDIX D Neural Network Design Taguchi Results 86

APPENDIX E Bridge Field Results Comparison Graphs 87

LIST OF TABLES

Table 1: Sample Set of Monitored Data (Esmaeili, 2012) 23
Table 2: Sorted Condition Attribute and its Corresponding Class (Esmaeili, 2012) 24
Table 3: Binary Transformation of Condition Indicator Values (Esmaeili, 2012) 25
Table 4: Design Parameters and Corresponding Levels for Feedforward ANN Design 31
Table 5: Design Parameters and Corresponding Levels for Feedback ANN Design 31
Table 6: Eigenvalues, Variability, and Cumulative Variability 34
Table 7: Set of Condition Data for a Test Equipment 35
Table 8: Set of CMAPSS Prognostics Results (from LAD) for a Test Equipment 36
Table 9: S/N Ratios for Optimal FFNN Architecture Taguchi Experiment 38
Table 10: S/N Ratios for Optimal FFNN Architecture Taguchi Experiment 39
Table 11: CMAPSS Error Summary 44
Table 12: CMPASS Percent Error Summary 46
Table 13: CMAPSS Error Summary by Actual RL 46
Table 14: CMAPSS Half-Life Error Summary 49
Table 15: CMAPSS Cost Scores 54
Table 16: Run Time (in Minutes) for Each Approach 56
Table 17: MacKay Bridge Laboratory Data Summary 60
Table 18: MacKay Bridge Field Data Summary 62
Table 19: Inputs Experiment Results 64
Table 20: S/N Ratios for Optimal FFNN Architecture Taguchi Experiment with MacKay Bridge Data 65
Table 21: Bridge Error Summary 66
Table 22: Residual Life Predictions of Bridge Field Components 68

LIST OF FIGURES

Figure 1: Standard ANN Configuration 11

Figure 2: Flowchart for Research Methodology 18

Figure 3: Flowchart for Step 1: Data Preprocessing of Research Methodology 19

Figure 4: Scree Plot for CMAPSS Data 21

Figure 5: Flowchart for Step 2a: LAD Formulation of Research Methodology 22

Figure 6: Flowchart for Step 2b: Neural Network Design of Research Methodology 29

Figure 7: Error as a function of the Actual RL for LAD CMAPSS Results 37

Figure 8: Optimal Feedforward ANN Architecture for CMAPSS Data 38

Figure 9: Optimal Feedback ANN Architecture for CMAPSS Data 40

Figure 10: Error as a function of the Actual RL for Feedforward ANN CMAPSS Results 41

Figure 11: Error as a function of the Actual RL for Feedback ANN CMAPSS Results 42

Figure 12: Error as a function of the Actual RL for PHM CMAPSS Results 43

Figure 13: Box Plot Comparison of CMAPSS Errors between Models 45

Figure 14: Box Plot Comparison of CMAPSS Half-Life Errors between Models 50

Figure 15: Box Plot Comparison of CMAPSS Interval Errors between Models 51

Figure 16: Cost Score Example Illustrated (Not to Scale) 53

Figure 17: Comparison of Predicted MRL between the Models for a Test Equipment 56

Figure 18: Fatigue Locations for the MacKay Bridge: (a) rib-to-deck joint, (b) rib-to-floor beam joint, (c) former stiffener between adjacent stiffening ribs (Clarke, 2014) 59

Figure 19: Gauge Positions for Type A Fatigue Locations (Clarke, 2014) 60

Figure 20: Sample Rainflow Histogram Output for MacKay Bridge Field Data (Clarke, 2014) .. 61

Figure 21: Sample Data of Laboratory Data Rainflow Histogram 63

Figure 22: Optimal Feedforward ANN Architecture for Bridge Data 66

Figure 23: Box Plot Comparison of Bridge Errors between the Models 67

Figure 24: Comparison of Prognostics Results for Component C14 (modified from Clarke, 2014) 70

ABSTRACT

Condition-based maintenance (CBM) is becoming increasingly important because it performs more efficient diagnoses and prognoses based on equipment condition compared to time-based methods. CBM models greatly inform maintenance decisions. This thesis examines three CBM fault prognostics models: artificial neural networks (ANNs), logical analysis of data (LAD), and Proportional Hazard Models (PHM). ANNs are layered models that imitate the human brain. LAD is a non-statistical pattern recognition technique. PHM statistically relates equipment age, condition, and hazard rate. A methodology is developed to apply and compare these models, and used on NASA's Turbofan Engine Degradation Simulation Dataset and the structural health monitoring dataset from Halifax's A. Murray MacKay Bridge. Results are evaluated using three metrics: error, half-life error, and a cost score. This thesis concludes that the LAD and feedforward ANN models compares favourably to the PHM. However, the feedback ANN has lower performance because of large variability in predictions.

LIST OF ABBREVIATIONS AND SYMBOLS USED

AI	Artificial Intelligence
ANN	Artificial Neural Network
ART2	Adaptive Resonance Theory
BPNN	Backpropagation Neural Network
CBM	Condition-Based Maintenance
C-MAPSS	Commercial Modular Aero Propulsion System Simulation
CM	Condition Monitoring
CMAPSS	C-MAPSS dataset provided by NASA
DAL	Dalhousie University
FB1	Single-Delay Feedback ANN
FB2	Two-Delay Feedback ANN
FBNN	Feedback (or Recurrent) Neural Network
FF	Single-Layer Feedforward ANN
FFNN	Feedforward Neural Network
HBB	Halifax Harbour Bridges
HMM	Hidden Markov Model
LAD	Logical Analysis of Data
MILP	Mixed Integer Linear Program
MRL	Mean Residual Life
MSE	Mean-Squared Error
NASA	National Aeronautics and Space Administration
PC	Principal Component

PCA	Principal Component Analysis
PHM	Proportional Hazards Model
RL	Residual Life
S/N	Signal-to-Noise
S-N	Stress-Life
SHM	Structural Health Monitoring
SOM	Self-Organizing Map
SPC	Statistical Process Control
TBM	Time-Based Maintenance
a	arbitrary weighting value for positive error used in the cost score function
a_{ij}	binary decision variable for whether attribute j exists in observation i
$b_{a, c}$	level attribute binary variable for the original variable with value a and a cut-point at c
$b_{a, c', c''}$	interval attribute binary variable for the original variable with value a and two consecutive cut-points at c' and c''
C	equipment replacement cost
D	measure of damage calculated using Miner's rule
d	degree or number of literals of a pattern
d_{ij}	error for observation i of equipment j
E	set of equipment in the training dataset
$f(t)$	equipment's lifetime probability density function as a function of time

$h(t, Z(t))$	PHM hazard rate function, which is a function of time and a vector of covariates
$h_0(t)$	PHM's baseline hazard rate, which is caused by aging
$i\Delta$	number of extra periods a piece of equipment can survive up to given that is has already survived up to T_0
K	equipment failure cost
$L(\theta_k)$	likelihood function for a probability density function with unknown parameters θ_k
MRL	mean residual life, which is the expected residual life of a piece of equipment given that it has already survived up to T_0
N_i	expected failure age under stress range i
N_l	negative generated pattern l
P	pattern generated according to the logical analysis of data method
P_k	positive generated pattern k
q	number of binarized values
$R(t)$	equipment's reliability as a function of time
$r_{i,j}$	additional constraint variable representing a previously generated pattern
S	set CM observations belonging to a certain class
S^-	set of CM observations belonging to the class opposite of class S
$S(x)$	sigmoid activation function as a function of x
S/N	signal-to-noice ration calculated in the Taguchi method
SP_b	baseline condition survival probability used in LAD

SP_{obs} conditional survival probability of the CM observation

SP_P pattern conditional survival probability used in LAD

T_0 time a piece of equipment has already survived up to

T_j survival time of equipment *j*

w_{ij} weight between node *i* and node *j* in an ANN

w_j binarized decision variable for whether attribute *j* is present in a given pattern

w_k^+ non-negative weight for the positive pattern *k* (used in the discriminant function)

w_l^- non-negative weight for the negative pattern *l* (used in the discriminant function)

w_{q+j} binarized decision variable for whether negation *q+j* is present in a given pattern

$x_i^{(k)}$ output of node *i* residing in layer *k*

y_i binary decision variable for the coverage of observation *i* (also the performance metric measured from the Taguchi experiments for experiment *i*)

$Z(t)$ vector describing the covariates at time *t*

β slope/shape parameter (used in the sigmoid activation function; used in the Weibull distribution)

γ_i coefficient of the covariate $Z_i(t)$

Δ discriminant function value used for classifying new observations; when this value is positive, the new observation is of the positive class, and vice versa

η scale parameter for the Weibull distribution

$\psi(Z(t))$	PHM’s covariate hazard rate, which is caused by covariates

ACKNOWLEDGEMENTS

First, I would like to acknowledge Nova Scotia Power Incorporated for initiating this research. I would also like to recognize the NASA Ames Prognostics Data Repository from which I retrieved the set of condition monitoring data used in this thesis' first case study. The dataset for the second case study in this thesis is the MacKay Bridge Condition Monitoring dataset provided by Justin Clarke. As such, I would like to extend my thanks Justin Clarke for allowing me to use it. In addition, Sasan Esmaeili's work with Logical Analysis of Data and with Proportional Hazards Model, especially his model codes, have been a great help in this thesis.

I would like to thank my supervisors, Dr. Alireza Ghasemi and Dr. Claver Diallo, for their advice, assistance, and continued encouragement in the duration of this thesis. Without them, this achievement would not have been possible. Thanks are also extended towards Dr. Peter VanBerkel and Dr. John Newhook for their time in reviewing this project as members of the Supervisory Committee.

I would also like to acknowledge and thank my family for their caring support and encouragement that made it easier to complete this thesis.

CHAPTER 1 INTRODUCTION

A long history of catastrophic failures demonstrates the importance of reliability in industry. For example, in 1946, a Lockheed Constellation aircraft crashed after generator faults triggered a fire during its training flight. Five of its six crew members were killed and the entire fleet was grounded. In 1979, improper maintenance procedures caused the engine of a DC-10 aircraft to break away during take-off. A total of 273 people were killed in the subsequent crash. In 1986, NASA's space shuttle, Challenger, exploded at take-off when O-ring failures led to burning gas leakages from the booster rockets. The accident killed the shuttle's entire crew and cost NASA over $4 billion (Ebeling, 1997). A range of impacts are evident from these examples. Maintenance is one method for ensuring acceptable reliability and efficiently managing both failures and their impacts over an equipment's lifetime. Maintenance can be defined as a set of activities necessary for preserving and restoring system function, in a cost effective manner, so that a certain operational level can be maintained (Ebeling, 1997; Tsang, 1995).

Traditionally, industries utilize run-to-failure maintenance policies (Li, Tse, Tu, & Yam, 2001; Tsang, 1995), a reactive management strategy in which an equipment is operated until it fails and then is repaired or replaced. This means that failures can still cause debilitating consequences by obstructing or disrupting the system. As a result, many industries now use time-based maintenance (TBM) policies (Banjevic, Jardine, Ennis, & Makis, 2001). TBM policies schedule maintenance actions at a certain age or at fixed time intervals, determined by the minimal expected cost per unit time. However, TBM has a limited ability to optimize the equipment's costs and utilization because little is known about the equipment's actual condition. In other words, maintenance can be completed too early or too late and in doing so, cost the company and sometimes causing the unintended degradation of the equipment's remaining useful life (Abu-Elanien & Salama, 2007).

These limits and higher reliability needs have led many industries (Bunks, McCarthy, & Al-Ani, 2000; Li, et al., 2001), such as aviation, manufacturing, and automobile, to consider other maintenance options. Condition-based maintenance (CBM) is one of these increasingly popular solutions (Baruah & Chinnam, 2005). CBM is a maintenance program that makes diagnoses or prognoses using the equipment's age, current health, and past condition information (Jardine, Lin, & Banjevic, 2006; Tsang, 1995). In CBM, a model is constructed based on known information and its quality is tested on additional data. In this way, system reliability is evaluated and assessed in an efficient manner.

1.1 Motivation and Objectives

A critical part of CBM's application and use is model building for diagnostics and prognostics. Many different approaches can be used for constructing these models, including, but not limited to, statistical approaches, artificial intelligence approaches, and model-based approaches (Jardine, et al., 2006). With this abundance of options open to the industry, the question arises of which model is best.

The objective of this research is to assess the application and performance of three CBM prognostic models. These models are the logical analysis of data (LAD), artificial neural networks (ANN), and proportional hazards model (PHM). The methodology for comparing the models, in this thesis, will be presented in detail and applied to two industrial problems.

1.2 Thesis Layout

This thesis begins with an introduction to the key concepts and literature relating to the CBM prognostics models used in this work. In Chapter 3, the methodology for the application and comparison of the three model is outlined, which includes information on data preprocessing and experimental design for each model. Chapter 4 discusses the results, obtained from the experiments, for two datasets. The first dataset is the Turbofan Engine Degradation Simulation Data Set, provided by the NASA Ames Prognostics Data Repository (Saxena & Goebel, 2008). To illustrate the methodology's applicability in industry, experimental results using data from the structural health monitoring of the A. Murray MacKay Bridge (MacKay Bridge), in Halifax, is used as the second dataset. This dataset is provided by Clarke (2014). Finally, Chapter 5 presents the conclusions and recommendations of this research.

CHAPTER 2 LITERATURE REVIEW AND BACKGROUND IN CBM

Over the last few decades, the growths in demand, cost, and amount of aging equipment in various industries have led to an increased need for reliability in industrial systems (Jardine, et al., 2006; Li, et al., 2001; Lin, Banjevic, & Jardine, 2006). As a result, common maintenance policies have evolved from run-to-failure maintenance to time-based preventive maintenance (TBM) to condition-based maintenance (CBM).

This chapter will outline the basic elements, applications, and literature relating to CBM and the key techniques examined in this thesis.

2.1 Condition-Based Maintenance

CBM uses both the equipment's past operating history and current condition to make optimized maintenance decisions for specific criteria, such as risk, cost, reliability, and availability (Banjevic, et al., 2001; Bunks, et al., 2000; Jardine, et al., 2006). CBM's ability to help make optimized maintenance decisions benefits industries because it can reduce maintenance costs, limit probabilities of catastrophic failure, decrease the severity of failure, identify root cause of failures, and provide information on the equipment life cycle to improve asset management (Abu-Elanien & Salama, 2007; Baruah & Chinnam, 2005; Yacout, 2010).

2.1.1 General Concepts

In order to apply CBM, a system for observing the equipment's current condition must exist. This is called condition monitoring (CM). CM can be defined as the extraction and integration of data into useful information for fault diagnostics and prognostics, maintenance decision making, and optimization support (Ghasemi, Yacout, & Ouali, 2010; Li, et al., 2001). A system of sensors, inspections, and other methods are used by CM to collect the equipment's condition information so that CBM can make maintenance decisions.

The need to integrate all CM information into decision making means that the proper implementation of a CBM policy is not simple. An effective CBM policy consists of three key steps (Jardine, et al., 2006; Lee, Abujamra, Jardine, Lin, & Banjevic, 2004):

1. Data acquisition; which collects and stores information about the equipment,
2. Data processing; which manages and analyzes the collected information, and
3. Maintenance decision making; which provides diagnostic and prognostic information for maintenance decision making.

For this thesis, interest lies in the third step. Information provided in this step can be divided into two categories: diagnostic and prognostic. Diagnostics is the process of detecting, isolating, and identifying faults that have already occurred in the equipment (Baruah & Chinnam, 2005; Bunks, et al., 2000; Jardine, et al., 2006; Mortada, Yacout, & Lakis, 2014). Prognostics, on the other hand, looks to the future and aims to predict the occurrence of incipient faults (Baruah & Chinnam, 2005; Bunks, et al., 2000; Jardine, et al., 2006). Diagnostic and prognostic information allow decision makers to make informed decisions, such as when to best schedule repairs and replacements. In this way, maintenance can be completed as close as possible to when it is needed, which is when the current equipment condition indicates that it is nearing failure. The questions then are: what types of models are best for providing diagnostic and prognostic information? And how are these applied to real world cases?

2.1.2 Methods and Techniques

Approaches to diagnostic and prognostic modeling can generally be divided into three categories (Jardine, et al., 2006): model-based approaches, statistical approaches, and artificial intelligence (AI) approaches.

Model-based approaches are typically made up of two components (Esmaeili, 2012): a mathematical model and a residual generation method. The mathematical model imitates and describes the equipment's underlying mechanics and physics. While the residual generation method obtains residual signals that indicate the presence of faults. Common modeling techniques (Jardine, et al., 2006) include mathematical modeling and computer simulation and common residual generation methods include Kalman filtering and parameter estimation. Various industries have applied model-based approaches to their diagnostic and prognostic problems. For example, electric utilities have found model-based approaches extremely useful for predicting equipment's internal temperature (Tallam, et al., 2007). In certain large electrical assets, this is important because the internal temperature, although very difficult to measure directly, is indicative of abnormal operation. In civil engineering, structures and materials can be modeled using finite element analysis. Particularly, a computerized finite element analysis program is used to observe the failure behaviour of a bridge deck and aid in fault diagnosis (Clarke, 2014). These and other applications show that model-based approaches are usually more effective than other approaches when the constructed model is extremely accurate and the modeled system is not extremely complex (Jardine, et al., 2006; Stanek, Morari, & Frohlich, 2001).

Statistical approaches, as its name implies, use statistical concepts to identify and predict faults. Common statistical approaches, for CBM diagnostics and prognostics, include hypothesis testing, statistical process control (SPC) methods, proportional hazards models (PHM) and hidden Markov models (HMM). Hypothesis testing (Ma & Li, 1995) is used to detect the presence of a fault in a bearing, with a null hypothesis testing the difference between the probability density function of the bearing vibration signal when faults are present and when faults are absent (normal working condition). Similar to this hypothesis, null hypotheses for CBM typically include the presence of a fault or the equipment’s ability to survive up to a certain period (Jardine, et al., 2006). SPC (Jardine, 2002; Jardine, et al., 2006; Liu, Ma, Yu, & Tu, 2014) diagnose faults by studying the deviation of the current condition value from a baseline ‘normal’ value. For example, an X-bar control chart (Liu, et al., 2014), constructed for the production output, is used to predict the failure state of a two-unit production system. When the production output deviates from the norm, one or both of the units in the production system is near failure. The PHM (Banjevic, et al., 2001; Jardine, et al., 2006) is a model that relates an underlying hazard (failure) rate function with age and condition variables. HMMs, using a finite-state Markov chain to model the equipment’s observed and hidden states, are widely applied in the diagnosis and prognosis of equipment (Baruah & Chinnam, 2005; Bunks, et al., 2000; Jardine, et al., 2006; Xu & Ge, 2004), such as gearboxes, pumps, and manufacturing machinery.

Artificial intelligence (AI) approaches are models which enable a computer system to imitate intelligent decision making. Two common AI approaches include artificial neural networks (ANN) and expert systems. ANNs (Jardine, et al., 2006) are based off the human brain and its learning capabilities. Using processing elements, which are connected in a layered network, and adjustable weights, the ANN learns the trends in the data. Expert systems (Jardine, 2002; Jardine, et al., 2006; Stanek, et al., 2001) perform diagnosis and prognosis by combining an expert's knowledge with a computer program that automates different reasoning and problem solving methods. In electric utilities, expert systems have been applied to diagnose equipment faults in large assets such as circuit breakers (Stanek, et al., 2001). Another AI technique is logical analysis of data (LAD) (Boros, et al., 2000), which extracts fault-characterizing patterns directly from the condition data. Jardine et al. (2006) performs an in-depth comparison and review of multiple independent studies on CBM techniques. From the results of these independent studies, the paper reaches the conclusion that it is typically harder to implement AI models, because of difficulties in data procurement, but these approaches offer improved performance as compared to the more traditional model-based and statistical approaches.

2.2 Logical Analysis of Data

The first type of AI approach that will be examined in this thesis is logical analysis of data (LAD). LAD studies the condition indicators and finds information that reveals equipment failure and survival signs (Abramson, Alexe, Hammer, & Kohn, 2005). In studying this hidden knowledge, LAD extracts patterns from the data. For this reason, LAD can be used as a pattern classifier. LAD, as a classifier, has been shown to outperform other classification methods (Yacout, 2010). In addition, LAD modeling does not require statistical analysis (Mortada, Yacout, & Lakis, 2011; Yacout, 2010). These two advantages are among some of the reasons that make LAD a good prognostics solution.

2.2.1 Basic Stages

First introduced in 1988 (Crama, Hammer, & Ibaraki), LAD is a combinatorics, optimization, and Boolean logic approach that can be used in diagnostic and prognostic problems (Abramson, et al., 2005; Alexe, et al., 2004; Gubskaya, et al., 2011). One of the most important concepts behind LAD is the identification of a pattern. A pattern is a set of literals that discriminates between a certain set of effects or a class of effects (Boros, et al., 2000; Crama, et al., 1988). Literals are either attributes or their negations. Attributes can be defined as the binarized values of the causes that lead to those certain effects (Boros, et al., 2000). In CBM, attributes are the condition indicators that can differentiate between the equipment’s failure and survival. Negations are the absence or the complement of the attributes (Boros, et al., 2000). In other words, if a certain binary attribute has the value of 1, then its negation will have the value of 0. Sets of literals that are reflected in one or more observations of failure and in little or no observations of survival are known as positive patterns. Sets of literals that have the opposite behaviour are known as negative patterns (Boros, et al., 2000; Ryoo & Jang, 2009). Using these patterns, LAD can classify new observations.

LAD models are constructed following four basic steps: data binarization, pattern generation, theory formation, survival function estimation. For CBM application, this means that:

- Data binarization transforms condition monitoring data into binary values, representing the failure and survival of the equipment
- Pattern generation generates condition-characterizing patterns through mathematical models
- Theory formation defines a discriminant function for weighting patterns and classifying observations, and
- Survival function estimation provides information, such as survival probabilities and mean residual life (MRL), for equipment prognosis.

Originally designed to process only Boolean inputs, LAD has since seen development to process non-Boolean data because many condition monitoring variables (and similar data) are numeric (Boros, et al., 2000). A method, called data binarization, has been developed to deal with this pitfall. Binarization changes numeric value data into several binary values by comparing data values to thresholds, called cut-points (Boros, et al., 2000). An example of this process is given later in Section 3.2. Two types of binary variables can be generated based on the cut-points: level attribute variables and interval attribute variables.

Level attribute variables generate one binary variable for every cut-point, as defined by Equation 1 (Boros, et al., 2000; Mortada, et al., 2014),

$$b_{a,c} = \begin{cases} 1 & ; if\ a \geq c \\ 0 & ; if\ a < c \end{cases} \tag{1}$$

Where a is the variable's numeric value, b is the variable's binary value, and c is the cut-point value. Interval attribute variables associate one binary variable for pairs of cut-points, as defined by Equation 2 (Boros, et al., 2000).

$$b_{a,c',c''} = \begin{cases} 1 & ; if\ c' \le a \le c'' \\ 0 & ; otherwise \end{cases} \tag{2}$$

Where a and b are again the variable's numeric value and the variable's binary value, respectively, and *c'* and *c''* are two consecutive cut-point values.

Efficient binarization depends largely on identifying the appropriate cut-points. In his work, Esmaeili (2012) describes three methods for identifying cut-points. The most basic method is the custom definition of cut-points, where cut-points are customized and placed by users. Another method is the sensitive discriminating method, which defines cut points at the boundaries between different classes. The third method is the equipartitioning method, where cut-points are placed with equal frequency or equal width between them.

After numeric data is binarized into a new set of binary variables, patterns for characterizing each observation's state are generated. Pattern generation uses mathematical models to identify and generate patterns hidden in the historical data. One important type of pattern that can be generated is the prime pattern (Boros, et al., 2000; Crama, et al., 1988), which is a pattern that can no longer be shortened. In other words, if even one literal in the pattern is removed, the set of variables can no longer be called a pattern. Two popular methods for prime pattern generation is the set covering mixed integer linear programming model (MILP) and the hybrid greedy enumeration approach. In the set covering MILP (Crama, et al., 1988; Mortada, et al., 2011; Ryoo & Jang, 2009), a pattern is generated by minimizing the number of uncovered observations of a specific class. During optimization certain design parameters, such as the number of covered observations of the opposite class, are still respected in order to ensure that the good patterns are generated. The hybrid greedy method (Boros, et al., 2000; Crama, et al., 1988) uses a combination of the top-down and bottom-up enumeration approaches to generate patterns. The first step is a bottom-up approach, which generates short prime patterns by starting with one literal and adding more literals until the design parameters are satisfied. Design parameters are constraints such as the number of covered observations in a specific class. This step is repeated until the stopping criteria is met before it proceeds to the second step. The second step generates patterns using a top-down approach to cover any remaining uncovered observations. In this approach, literals from an existing uncovered observation are removed one at a time until it is no longer a pattern. These two approaches constitutes the hybrid greedy heuristic.

Using the generated patterns, new observations can be classified based on the historical information. Theory formation, which can be quantified by a discriminant function (Boros, et al., 2000), is necessary to classify the observation as a failure or a survival instance. If a new observation is covered only by patterns associated with failures then it is a failure observation, and vice versa (Crama, et al., 1988). Additional considerations must be made to classify observations that are covered by both failing and surviving patterns. These concerns are considered by weighting the patterns in the discriminant function in Equation 3 (Boros, et al., 2000).

$$\Delta = \sum_{k=1}^{r} w_k^{+} P_k + \sum_{l=1}^{s} w_l^{-} N_l \tag{3}$$

Where the $P_1, P_2, \ldots, P_k$ are the positive generated patterns, $N_1, N_2, \ldots, N_l$ are the negative generated patterns, and w_k^{+} and w_l^{-} are the non-negative weights for the positive patterns and non-positive weights for the negative patterns, respectively. These weights can be determined by considering (Boros, et al., 2000) the number of covered observations, the normalized inverse of the pattern's degree, or the optimal value solved with linear programming. When the discriminant value is positive, the new observation is of the positive class, and vice versa.

To perform prognostics on the equipment observations, the survival function is needed. This allows the user to make informed maintenance decisions because knowledge on the MRL and reliability performance of the equipment is provided. In 2002 (Lauer, et al.), a method is proposed using Kaplan-Meier (KM) estimators (Kaplan & Meier, 1958) to determine the conditional survival probabilities. Through those conditional survival probabilities, the survival function can also be determined because of the relationship described by Equation 4.

$$MRL = \sum_{i=1}^{\infty} i\Delta \times P(\tau > T_0 + i\Delta | \tau > T_0) \tag{4}$$

Where *iΔ* is the number of extra periods that the equipment will survive up to, given that it has already survived up to T_0.

2.2.2 Application

LAD's potential for better classification has allowed it to become widely applied in several industries and fields. Fields such as medicine and biotechnology have seen the presence of LAD (Abramson, et al., 2005; Alexe, et al., 2004; Boros, et al., 2000; Gubskaya, et al., 2011; Lauer, et al., 2002). In medicine, LAD is used to identifying the risk level of patients referred for exercise electrocardiography (Lauer, et al., 2002). To diagnose ovarian cancer patients, a LAD model applied to mass spectroscopy data inputs is used (Alexe, et al., 2004). LAD has also been applied in predicting the cell growth of polymeric biomaterials (Abramson, et al., 2005; Gubskaya, et al., 2011).

As for the field of CBM, there have been various developments. LAD models have been used for the diagnostic of various equipment such as power transformers (Mortada, et al., 2014; Yacout, 2010), aircraft (Mortada, Carroll III, Yacout, & Lakis, 2012), and bearings (Mortada, et al., 2011). In one case, LAD is used to identify 'rogue' faults in a fleet of aircraft turbo compressor components (Mortada, et al., 2012). 'Rogue' faults develop from faults that are allowed to continue growing because regular repair and maintenance bypass them. Mortada et al. (2014) uses an extended version of LAD to classify multiple power transformer fault types from dissolved gas data. Recently, Esmaeili (2012) applied LAD in a CBM prognostics problem.

2.3 Neural Networks

As previously mentioned, artificial neural networks (ANN) are one commonly used AI approach. First introduced in 1943 (McCulloch & Pitts), it is designed to make up for limitations in traditional computational approaches. Development of the concept peaks around the 1980s with Werbos' (1974) presentation of the backpropagation learning algorithm. Since then, ANNs now have the standard structure illustrated by Figure 1. ANNs layers and connects processing units, inputs and outputs in a way that mimics the human brain structure (Jardine, et al., 2006).

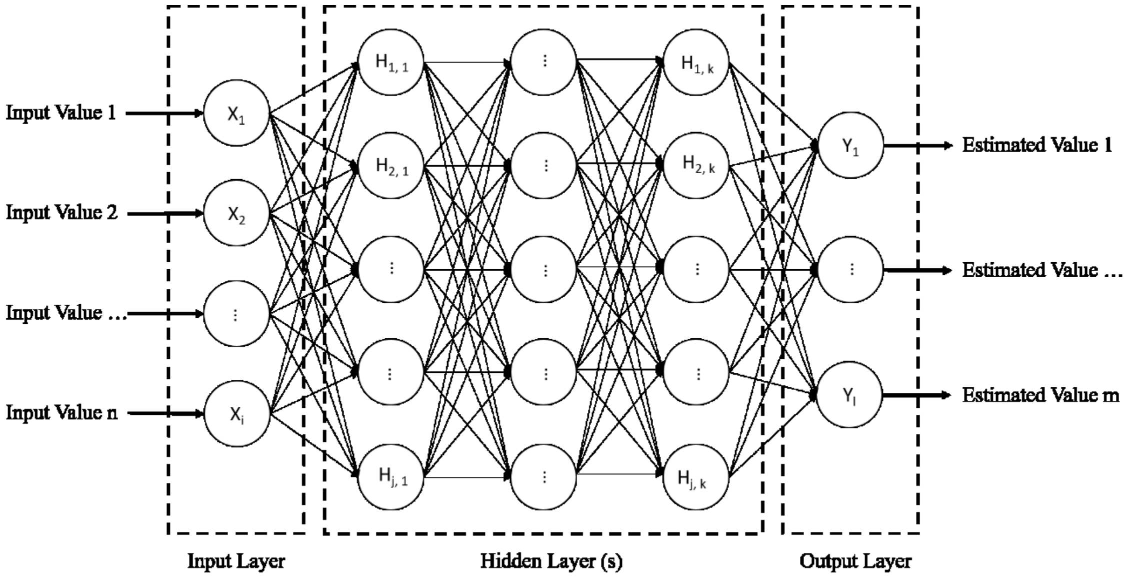

Figure 1: Standard ANN Configuration

Like the human brain, ANNs employ a learning ability to remember patterns it has observed in historical data (Jain, Mao, & Mohiuddin, 1996). Using these patterns, the ANN is able to solve problems that are otherwise difficult or impossible to solve with more traditional computing methods (Jain, et al., 1996). This advantage makes ANNs an attractive alternative to other diagnostic and prognostic approaches.

2.3.1 Basic Components and Architecture

ANNs are characterized by several basic components that can be arranged in different ways to carry out specific applications (Drew & Monson, 2000; Jain, et al., 1996). The first component is the network architecture, which refers to how the processing elements (inputs, outputs, nodes, and connections) are layered and arranged. The second component, the activation function, dictates how each node alters the input(s) into an intermediate output(s). Finally, the training algorithm defines how the weights between each node change to fit the patterns observed in the data.

The most basic unit in the ANN architecture is the node. Nodes are processing units designed to represent the human brain's biological neuron (Farrar & Worden, 2013; Jain, et al., 1996; McCulloch & Pitts, 1943). The biological neuron uses its nucleus (which contains information on its behaviour), axons and dendrites (which are channels for information transportation between neurons), and synapses (which are information signals that are passed through the axons and dendrites) to learn and adjust to the external input the body receives. In the same way, the nodes also learn input information. The activation function acts like the nucleus and determines how information is processed as it passes through the node. The connections between nodes are like the axons and dendrites. Then the training algorithm, adjusting the axons and dendrite connections, learn the data patterns like the synapses.

The basic node can be arranged in various ways. The arrangement of these nodes is known as the architecture. The most basic ANN architecture consists of an input layer, one or more hidden layers, and an output layer (Figure 1). Viewing from another perspective, ANNs can be described as weighted directed graphs (Jain, et al., 1996). Based on the direction of the connections (or edges), ANNs can be classified as one of two general architectures. The first basic architecture is the feedforward neural network (FFNN) (Jain, et al., 1996; Jardine, et al., 2006). The FFNN contains only forward-directed edges. As a result, FFNN can be considered memoryless because its outputs are independent of previous outputs. The second basic architecture is the feedback or recurrent neural network (FBNN) (Jain, et al., 1996). The FBNN is characterized by backward-directed edges or feedback connections. In most cases, the output of the previous iteration is fed back into the ANN as an input.

As a weighted graph, weights w_{ij} are associated with each connection. In a standard feedforward network, each node i is connected to each node j in the preceding layer. Therefore, weighted sums are performed at each node i for all input signals $x_j^{(k-1)}$ before it is passed through an activation function f to become the output $x_i^{(k)}$ to the next layer. In mathematical terms, this can be represented by Equation 5 (Farrar & Worden, 2013).

$$x_i^{(k)} = f\left(\sum_{j=1}^{n} w_{ij}^{(k)} x_j^{(k-1)}\right) \tag{5}$$

Where k is the layer in which node i resides and n is the number of nodes in the preceding layer k-1 to which node i is connected.

The activation function in Equation 5 defines how inputs transform within each node. Activation functions can be discrete, with binary outputs, or continuous, with continuous outputs (Sibi, Jones, & Siddarth, 2013). The most popular type of activation function is the sigmoid activation function, described by Equation 6 (Jain, et al., 1996),

$$S(x) = \frac{1}{1+e^{-\beta x}} \, , for\ 0 < S(x) < 1 \tag{6}$$

Where β is a slope parameter. Using this function as a basis, a family of activation functions has been developed that exhibit sigmoidal characteristics (Sibi, et al., 2013). Mathematical functions with sigmoidal characteristics are functions having an "*S*" shaped curve. Examples of this family of functions are the sigmoid stepwise activation function and the sigmoid symmetric activation function. Other activation functions (Jain, et al., 1996; Sibi, et al., 2013) include the threshold activation function, linear activation function, the stepwise activation function, and the Gaussian activation function.

In order to take full advantage of an ANN's brain-like structure, a training algorithm is needed to guide the network's learning. The training algorithm specifies how weights between the nodes are adjusted. Given a set of data, a subset is partitioned into a training dataset, so that the ANN can learn the historical data patterns. To better illustrate this concept, consider the backpropagation algorithm. The backpropagation algorithm (Drew & Monson, 2000; Farrar & Worden, 2013; Werbos, 1974) is one of the most popular training algorithms for ANNs. It first sets initial weight values then uses training data values to calculate the output signal for the network. After, each pass, the error between the outputs are calculated. If the error is above the set threshold, the errors are propagated backwards through the network so that the weights between the nodes can be adjusted to a new value. When the magnitude of the errors falls below the threshold, the backpropagation algorithm has successfully determined the weights of the ANN.

Training algorithms can be classified into two main categories: supervised and unsupervised. Supervised learning algorithms (Jain, et al., 1996; Jardine, et al., 2006) use a priori knowledge of the expected or targeted output to learn. Unsupervised learning algorithms (Jain, et al., 1996; Jardine, et al., 2006) do not use the direct input-output relationships to adjust the network weights. Instead, reoccurring trends and patterns in the data are used for training. Some examples of training algorithms include the backpropagation algorithm, perceptron algorithm, and the Hebbian learning rule. In the perceptron algorithm (Jain, et al., 1996), weights are adjusted for each node individually in a process that minimizes the error for each node. The Hebbian learning rule (Hebb, 1949) weights nodes that show more activity and persistent correlation.

Using these components, an ANN can be trained to directly predict the residual life (RL) of a piece of equipment. Unlike LAD, if an ANN is structured to fit prognostics needs, there is no need to estimate a survival distribution for determining the MRL or related reliability of the equipment.

2.3.2 Application

As a result of its learning abilities and its potential for solving different problems, ANNs have been applied to a variety of problems and industries. These problems include pattern recognition, clustering, categorization, function approximation, forecasting, optimization, and memory restoration (Jain, et al., 1996). Industries that have seen ANN applications include the fields of medicine (Azimi, et al., 2014), supply chain optimization (Al-Saba & El-Amin, 1999; Carbonneau, Laframboise, & Vahidov, 2008), and food science (Cimpoiu, Cristea, Hosu, Sandru, & Seserman, 2011). In supply chain optimization, ANNs are used to forecast demand in two separate occasions. Al-Saba and El-Amin (1999) design a backpropagation ANN (BPNN), a FFNN using the backpropagation algorithm, to forecast the demand of a power distribution supply chain. A FFNN and a FBNN model (Carbonneau, et al., 2008) are developed and applied to forecast the demands in a typical supply chain network. In this work, the FBNN shows better accuracy but under statistical analysis, the two techniques are not statistically different. In food sciences, a multilayer FFNN (Cimpoiu, et al., 2011) is used to quantify the antioxidant potential of tea leaves.

In CBM diagnostics and prognostics, there have been several ANN applications. The most widely used ANN architecture is the FFNN. One example of a FFNN application is in electric utilities, where a power transformer's oil insulation faults are diagnosed with a BPNN (Huang & Huang, 2002). Other types of ANNs have also seen applications in CBM. For example, a 3-layer FBNN (Li, et al., 2001) is applied to track and predict the deterioration rate of a gear train. Wang and Too (2002) developed a self-organizing map (SOM) (Kohonen, 1990), which is a network trained with an unsupervised algorithm to group nodes into neighborhoods for different outputs, to use vibration signal input data for motor fault diagnosis. Combinations of ANNs with other techniques for CBM purposes have also been used. Silva, Reuben, Baker and Wilcox (1998) use an expert system with two ANNs, an adaptive resonance theory network (ART2) and a SOM, to estimate tool wear. An ART2 network (Carpenter & Grossberg, 1987), addresses ANNs weakness in learning new things, by introducing initially unused output units for later use with new patterns. New categories of outputs are created by the network when data that do not match existing data patterns are presented to the network.

Despite the popularity of ANNs, there are several limitations. First, it is difficult to determine the optimal architecture (Jardine, et al., 2006; Yacout, 2010). Second, there is a lack of efficient procedures for obtaining the necessary data and knowledge for training the ANNs (Azimi, et al., 2014; Jardine, 2002; Jardine, et al., 2006). Third, it is often difficult to associate physical explanations to ANNs (Jardine, et al., 2006).

2.4 Proportional Hazards Model

The proportional hazards model (PHM) is one of the most popular statistical modeling approaches in CBM. PHM (Cox, 1972; Jardine, et al., 2006) can describe the equipment's underlying statistical behaviour and relate the equipment's age and condition to its failure rate.

2.4.1 Definitions and Concepts

PHM's popularity lies in its ability to relate both the age and condition information to the equipment's failure rate. In many real applications of CBM, failure and maintenance depend not only on age but also on other factors, like condition and operational settings (Johnson, 1998). These additional factors are known as covariates in reliability modelling.

The PHM method determines the equipment's hazard rate or failure rate, which is the equipment's instantaneous rate of failure at time *t*. Using this function, the conditional probability of failure and the reliability of the equipment can be determined. In PHMs, covariate hazard rates are proportional to each other. According to this rule, a general PHM's hazard function can be described with the following Equation 7 (Cox, 1972; Jardine, et al., 2006),

$$h\big(t, Z(t)\big) = h_0(t)\psi\big(Z(t)\big) \qquad (7)$$

Where h_0 is the baseline hazard function describing the failure rate caused by aging, $\psi(Z(t))$ is the hazard rate relating to the covariates, and $Z(t)$ is a vector describing the covariates at time t. By varying this baseline hazard function or the covariate hazard functions, this general PHM can be extended for different equipment behaviours. In 1972, Cox (1972) developed a PHM which assumes (1) that the condition indicators directly point to the equipment's state, (2) that the covariate hazard function is time dependent, and (3) that the baseline hazard can be modeled as a semi-parametric function (Lin, et al., 2006), which is a function that does not follow a specified distribution. Another extension is the Weibull PHM, which is often used in CBM. This model uses a Weibull baseline hazard rate and Cox's covariate hazard rate (Banjevic, et al., 2001; Jardine, et al., 2006; Lin, et al., 2006). The Weibull PHM is described by Equation 8,

$$h(t, Z(t);\ \beta, \eta, \gamma) = \left(\frac{\beta}{\eta}\right)\left(\frac{t}{\eta}\right)^{\beta-1} \exp(\textstyle\sum_i^m \gamma_i Z_i(t)\ ) \qquad (8)$$

Where β and η are the Weibull distribution's shape and scale parameters, respectively, and γ_i are the coefficients of the covariates Z*(t)*.

To apply PHM to real world problems, it is necessary to completely specify the hazard rate model. Parameter estimation can be used to make estimates of the unknown parameters of PHM's hazard rate model based on historical data. Bayes estimation (Danish & Aslam, 2012) and maximum likelihood estimation (Cox, 1972; Ding & He, 2011; Ghasemi, et al., 2010; Lin, et al., 2006) are common parameter estimation techniques that have been used in parameter estimation for PHMs. Danish and Aslam (2012) use Bayes estimation to determine the parameters for a PHM applied in the prediction of Primary Biliary Cirrhoisis patients' time of death. In a machine tool wear prediction problem, Ding and He (2011) estimate the parameters in a lognormal PHM using maximum likelihood estimation.

Using the hazard rate function, the equipment's reliability function can be calculated based on the relationship in Equation 9 (Ebeling, 1997):

$$R(t) = \int_t^\infty f(\tau)d\tau \qquad (9)$$

Where, $R(t)$ is the reliability function and $f(\tau)$ is the lifetime probability density function. Equation 9 can then be substituted into Equation 10 to determine the MRL (Ebeling, 1997):

$$MRL = \frac{1}{R(T_0)} \int_{T_0}^{\infty} R(t)dt \tag{10}$$

Where T_0 is the current survival time of the equipment. With these relationships, prognostics information can be provided to maintenance decisions makers using PHM.

2.4.2 Application

PHM's ability to map age and condition onto a hazard rate has made it popular in survival analysis. Various industries, like medicine, farming, and manufacturing, have seen PHM applied. Sankaran and Sreeja (2012) use PHM to diagnose the survival probability of AIDS patients. Not only has it been used for seriously ill patients, PHM has also been used to estimate patient survival probabilities in hospital populations (Miladinovic, et al., 2012). In farming, the survival and production life of Holstein cows (Zavadilová, Němcová, & Štípková, 2011) has been modeled using PHM, so that the best cattle management strategy could be determined. The study is also able to differentiate between characteristics associated to two different Holstein cow populations (in Canada and in Czech).

As stated before, the Weibull PHM is one of the most widely used PHM models for CBM. A CBM decision support tool (Jardine, 2002; Banjevic, et al., 2001), EXAKT^{TM}, uses the Weibull PHM in its failure rate calculations and to quantify the decision making risk. Other uses in industry include modelling of the failure rates in a multiple component bearing system (Tian & Liao, 2011) and in military vehicle engines (Lorna Wong, Jefferis, & Montgomery, 2010). CBM applications are not limited to the Weibull PHM. For example, a PHM with a lognormal baseline hazard is used to predict the wear of a lathe machine (Ding & He, 2011).

CHAPTER 3 METHODOLOGY

The following chapter will outline the steps for examining, within the context of CBM, the three models identified earlier; the artificial neural network (ANN), the logical analysis of data (LAD) and the proportional hazards model (PHM). Figure 2 summarizes these steps: step 1 is described in Section 3.1, step 2 is outlined in Section 3.2, 3.3, and 3.4, and step 3 is presented in Chapter 4.

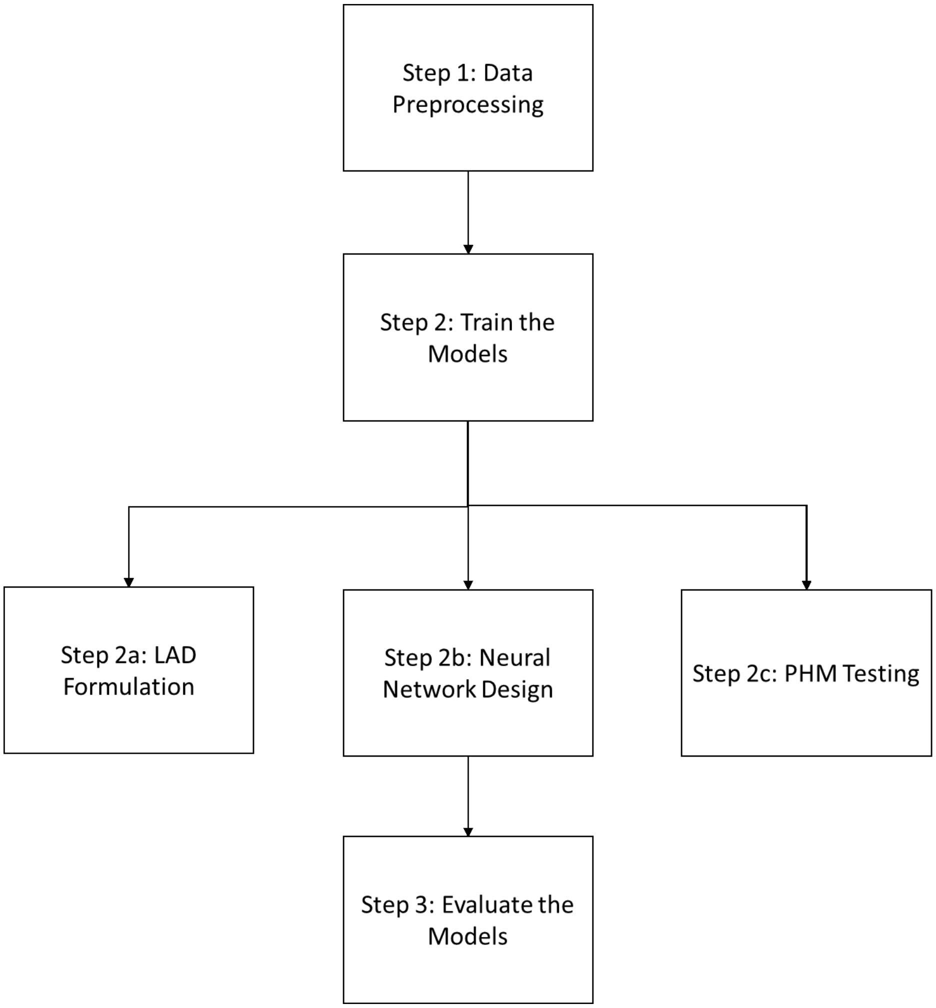
Step 1: Data Preprocessing
Step 2: Train the Models
Step 2a: LAD Formulation
Step 2b: Neural Network Design
Step 2c: PHM Testing
Step 3: Evaluate the Models

Figure 2: Flowchart for Research Methodology

3.1 DATA PREPROCESSING

In the context of CBM, the dataset under study includes information on the equipment's age and multiple condition indicators. Condition indicators are measures of the equipment's health and differ according to the equipment under observation.

Datasets of this nature are often large, which means that one challenge of CBM modeling is the dataset size. In some cases, large datasets require greater processing power and longer processing time. In order to complete the experiment in a reasonable time, it is necessary to decrease the size of the dataset under study. The data preprocessing stage is summarized in Figure 3:

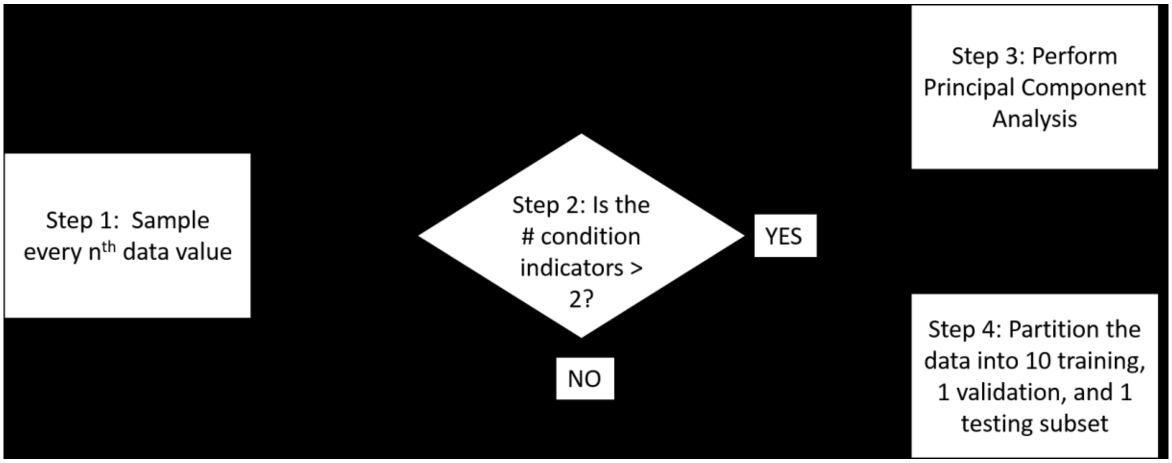
Step 1: Sample every nth data value
Step 2: Is the # condition indicators > 2?
YES
NO
Step 3: Perform Principal Component Analysis
Step 4: Partition the data into 10 training, 1 validation, and 1 testing subset

Figure 3: Flowchart for Step 1: Data Preprocessing of Research Methodology

First, every nth data value is extracted from the dataset under study. The extraction of every nth data value follows the data preprocessing step that is used by Esmaeili (2012). For example, instead of using all data values, analysis is completed on every 100th data value. In some cases, a new variable is created to describe the data located in between the extracted data value. In other words, instead of using every 100th data value, the mean and standard deviation over those 100 data values is used for analysis.

If the number of condition indicators in the dataset is greater than two, the dataset is further reduced by using principal component analysis (PCA), in this thesis. PCA (Johnson, 1998; Pearson, 1901) is a multivariate analysis method that can be used to identify outliers, create new variables, analyze clusters, and determine the true dimensionality of a dataset. In many cases, variables in a dataset are highly correlated, so using them in analysis will not provide additional information. The dataset's true dimensionality will be identified when PCA is used. Often the true dimensionality is less than the original dimensions of the dataset, which reduces the original dataset dimensions (Johnson, 1998; O'Rourke, Hatcher, & Stepanski, 2005). Many use PCA as a dimension reduction technique because of its ability to determine a dataset's true dimensionality (Jardine, et al., 2006; Lin, et al., 2006; O'Rourke, et al., 2005).

Orthogonal transforms (Pearson, 1901) are used to convert a correlated set of variables into a new set of linearly uncorrelated variables called principal components (PC). Eigen decomposition is used in orthogonal transforms. In Eigen decomposition (Johnson, 1998; Pearson, 1901), the covariance or correlation matrix is factorized into two matrices known as the eigenvalues and the eigenvectors. Each vector in the eigenvector matrix corresponds to an eigenvalue, which acts like a scaling factor during multiplication. This means that the higher the eigenvalue, the more variability is accounted for by the corresponding eigenvector. When sorted in decreasing order of importance, PCs are (Johnson, 1998; O'Rourke, et al., 2005):

1. Uncorrelated
2. Decreasing in the amount of accountable variability in the data. In other words, the first PC accounts for the most variance in the data.
3. Accounting for as much of the remaining variability as possible. This means that if the first PC accounts for 75% variance, the second PC will try to account for as much of the remaining 25% variance as possible, as will the succeeding PCs, if the second PC cannot cover the remaining 25%.

PCs, created by the product of the eigenvector and the original data value, can be used directly in subsequent analysis (Johnson, 1998; Lin, et al., 2006). However, PCA creates the same amount of PCs as there are original variables. Therefore, it is necessary to determine how many PCs should be used to truly represent the dataset. Three methods are typically used (Johnson, 1998; O'Rourke, et al., 2005): a scree plot, the amount of acceptable cumulative variance, and an eigenvalue greater than one.

In the scree plot method, PCs' eigenvalues are plotted against their PC number (Figure 4). Significant PCs occur before the point at which trend starts to level off in the scree plot and are chosen to represent the dataset. For example, Figure 4, shows eigenvalues leveling off at the 4th PC. The 3rd PC is not as high as the first and second PCs but not entirely level with PC 4 to 21. However, the 3rd PC is much closer to the eigenvalue of PC 4 than to PC

2. Therefore, PC 1 and PC 2 are chosen as the significant PCs according to the scree plot method.

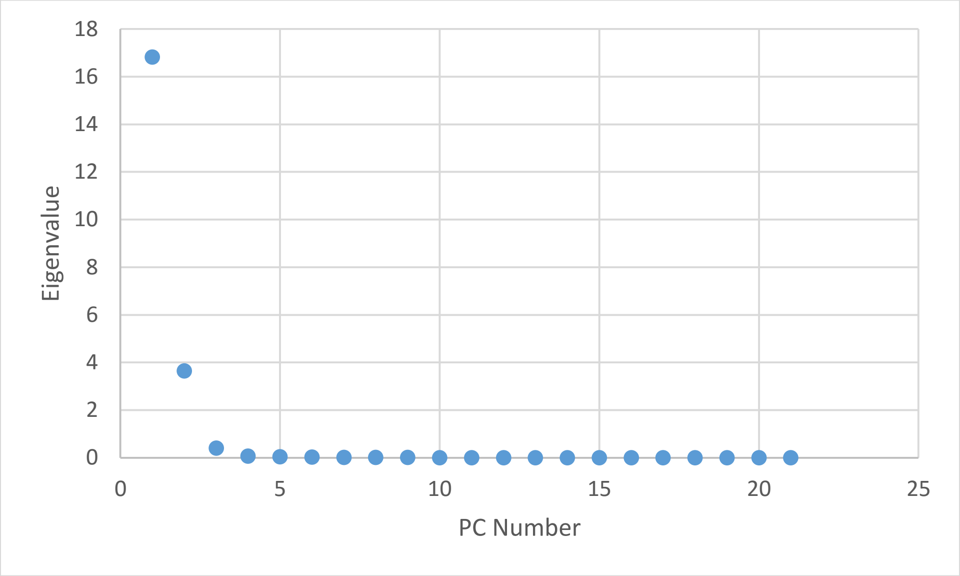

Figure 4: Scree Plot for CMAPSS Data

The amount of acceptable cumulative variance method involves choosing a value for the acceptable amount of total variance in the original dataset. PCs are chosen if the cumulative variance for that PC does not exceed this threshold. As soon as the cumulative variance in the PCs pass this value, no further PCs are kept.

The eigenvalue greater than one method applies only to PCs determined with the correlation matrix. If the PCs are determined using the correlation matrix method, PCs can be kept when their eigenvalue is greater than one.

In this thesis, the amount of acceptable cumulative variance method is used to determine the number of significant PCs. The scree plot method is discarded because of possible subjectivity in identifying the point at which the trend levels off. The eigenvalue greater than one method is not used because it is dependent on the use of the correlation matrix and cannot be easily generalized in applications. Using the amount of acceptable cumulative variance method, an acceptable cumulative variance threshold is defined at 95%.

After size reduction, the dataset is partitioned into several training sets, a validation set, and a test set. The training set allows the experimental models to learn the patterns within the data. The validation set is used only in the training of ANNs. Finally, the test set is used for evaluating the trained models and for calculating performance metrics.

3.2 LAD Formulation

The LAD formulation in this thesis is taken from the methodology described in *Development of Equipment Failure Prognostic Model Based on Logical Analysis of Data (LAD)* (Esmaeili, 2012). The LAD components deemed most effective by Esmaeili (2012) are used in this experiment. A brief outline of the LAD formulation is given by Figure 5.

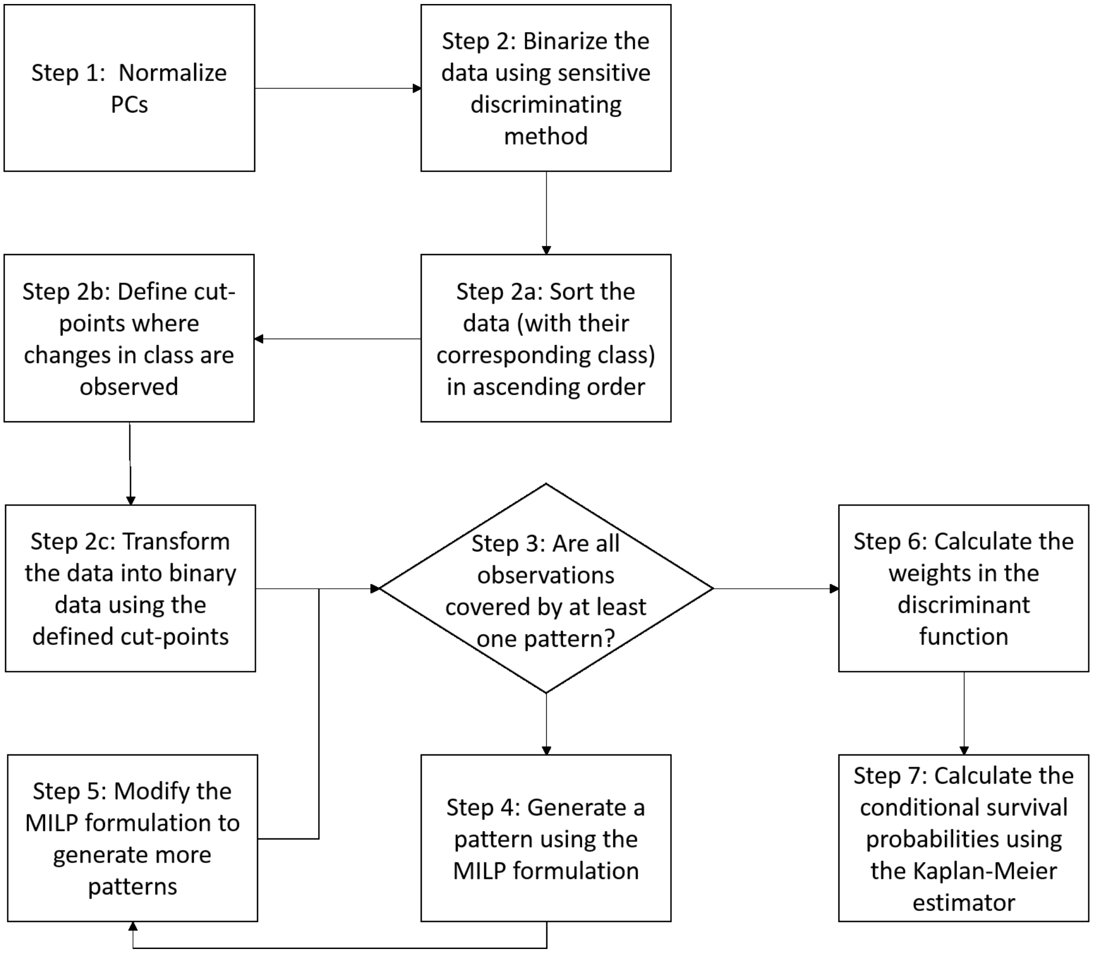
Step 1: Normalize PCs
Step 2: Binarize the data using sensitive discriminating method
Step 2a: Sort the data (with their corresponding class) in ascending order
Step 2b: Define cut-points where changes in class are observed
Step 2c: Transform the data into binary data using the defined cut-points
Step 3: Are all observations covered by at least one pattern?
Step 4: Generate a pattern using the MILP formulation
Step 5: Modify the MILP formulation to generate more patterns
Step 6: Calculate the weights in the discriminant function
Step 7: Calculate the conditional survival probabilities using the Kaplan-Meier estimator

Figure 5: Flowchart for Step 2a: LAD Formulation of Research Methodology

Before training the LAD models, the data is further processed using normalization (Kim, 1999). That is, data values are constrained between 0 and 1 in order to compress the range of values. This is necessary to reduce the number of cut-points used to binarize the data.

In the first stage of LAD, binarization of the data is completed. The sensitive discriminating method (Esmaeili, 2012) is used to define the cut-points needed for data binarization. Although, Esmaeili (2012) does not compare the performance between the sensitive discriminating method and the equipartitioning method, the sensitive discriminating method is more logically sound than the equipartitioning method. As previously described, the sensitive discriminating method defines cut-points where changes occur between classes, where differences between information are expected to occur. This means that cut-points should be defined only where it is needed and the pattern generated from the resulting binary data is at its minimal length. An introduction of more cut-points would result in longer patterns. As such, this thesis uses the sensitive discriminatory method. In this method, cut-points are defined between the values of two different classes. For example, consider a dataset (Table 1) with several pieces of equipment, their various observation moments, and the corresponding condition.

Table 1: Sample Set of Monitored Data (Esmaeili, 2012)

Observations			Attributes	
Equipment ID	Observation Time	Class	Age	Condition Indicator
1	0	-	0	14
1	1	-	1	16
1	2	-	2	20
1	3	-	3	18
1	4	+	4	20
2	0	-	0	12
2	1	-	1	18
2	2	+	2	22
3	0	-	0	16
3	1	-	1	18
3	2	-	2	20
3	3	+	3	22

In Table 1, each row belongs to one observation moment in the equipment's lifetime. The first and second columns identify the piece of equipment and the observation moment. The third column shows the class of each observation. The negative class means that the equipment is still surviving at that observation moment. The positive class, which is further highlighted by the darker background, means that the equipment has failed at that observation moment. Finally the last two columns display the condition information, which in this case are the age and one condition indicator.

The first step in the sensitive discriminating method is sorting the data and their associated classes (negative for survival and positive for failure) in ascending order. Then, cut-points are defined at places where the changes in class are observed (marked by the dark background). Table 2 shows the cut-points for the condition indicator data shown in Table 1. In the case of this example, cut-points are defined between 18 and 20, and between 20 and 22.

Table 2: Sorted Condition Attribute and its Corresponding Class (Esmaeili, 2012)

Sorted Attribute	
Condition Indicator	Class
12	-
14	-
16	-
18	-
20	-, +
22	+

Using these cut-points and Equation 1, the condition indicator data is now transformed into the binary data shown in Table 3.

Table 3: Binary Transformation of Condition Indicator Values (Esmaeili, 2012)

Observations		Condition Indicator Attributes	
Equipment ID	Observation Time	19	21
1	0	0	0
1	1	0	0
1	2	1	0
1	3	0	0
1	4	1	0
2	0	0	0
2	1	0	0
2	2	1	1
3	0	0	0
3	1	0	0
3	2	1	0
3	3	1	1

After obtaining the binarized data, an MILP optimization is used to generate patterns. According to Esmaeili (2012) both the greedy heuristic method and the MILP optimization shows similar performance. However, the solution time for the MILP is much shorter than that of the greedy heuristic (Esmaeili, 2012). Therefore, the MILP method for pattern generation is used in this thesis. The MILP model's objective is to generate a pattern that minimizes the number of uncovered observations of a certain class. Given a set of observations in a certain class *S* and a set of observations in the opposite class S^{-}, y_i is the binary decision variable for the coverage of observation *i* that can be solved with the following set covering MILP formulation, used by Esmaeili and introduced by Ryoo and Jang (2009):

$$\text{Minimize} \sum_{i \in S^*} y_i$$

subject to:

$$\sum_{j=1}^{2q} a_{ij} w_j + q y_i \geq d \qquad \forall i \in S^* \tag{11}$$

$$\sum_{j=1}^{2q} a_{ij} w_j \leq d - 1 \qquad \forall i \in S^{*-} \tag{12}$$

$$w_j + w_{q+j} \leq 1 \qquad j = 1, 2, \ldots, q \tag{13}$$

$$\textstyle\sum_{j=1}^{2q} w_j = d \tag{14}$$

$$1 \leq d \leq q \tag{15}$$

$$w_1, \ldots w_{2q} \in \{0, 1\}$$

$$y_1, \ldots y_{N^*} \in \{0, 1\}$$

Where, a_{ij} is a binary variable for whether attribute j exists in observation i, q is the number of binarized values, w_j is the binarized variable for whether attribute j is present in the pattern, w_{q+j} is the negation of attribute j, and d is the degree (or number of literals) in the pattern. In the formulation, the first constraint (Equation 11) ensures that the observations in a certain class (S) are covered by the generated pattern. The second constraint (Equation 12) ensures that the observations in the opposite class (S^-) are not covered by the generated pattern. The next constraint (Equation 13) holds true for all q attributes and states that a pattern cannot be formed by both the attribute and its negation. The last constraint (Equation 14) counts the degree of the pattern that is generated.

After a pattern is generated, the model can be modified to generate patterns one at a time until all the observations are covered by at least one pattern. The model is modified by adding an additional constraint (Equation 16) (Mortada, 2010) representing the generated pattern and then optimizing the problem without the observations that have already been covered by previously generated patterns.

$$r_{i,j} = \begin{cases} 1 & ;if \quad w_{i,j} = 1 \\ -1 & ;if \quad w_{i,j} = 0 \end{cases} \tag{16}$$

Using the generated patterns, the LAD model classifies observations based on the discriminant function in Equation 3 (Boros, et al., 2000). In this thesis, the weights for the discriminant function are determined by the relative significance of the generated patterns. This means that equal weights are given to each of the two classes of patterns and individual generated patterns within each class divide the total weight between them. For example, consider a pattern generation approach that has generated three patterns. Two of these patterns belong to the negative class and one to the positive class. The total weight for each class is 1, so the weights for the negative patterns are 0.5 and the weight for the positive pattern is 1.

Further work is done by the LAD to enable it to perform prognosis. Using the Kaplan-Meier (1958) estimator, two types of conditional survival probabilities are calculated. The first conditional survival probability is associated with the generated patterns and is defined as the ratio of observations covered by pattern *P* whose equipment survive up to *i* periods after coverage to the total observations covered by pattern *P*. This is represented by Equation 17 (Esmaeili, 2012):

$$SP_p(i) = \frac{\#(P \cap S; \tau > \tau_0 + i\Delta)}{\#(P \cap S; \tau > \tau_0)} \tag{17}$$

Where SP_p is the pattern condition survival probabilities and *S* is the set of observations covered by pattern *P*.

A second conditional survival probability is estimated in relation to the survival function based on age. According to the Kaplan-Meier estimator, this probability can be defined as the ratio of pieces of equipment that survive up to *i* periods to the total number of equipment in the training dataset. This baseline conditional survival probability is defined by Equation 18 (Esmaeili, 2012)

$$SP_b(i) = \frac{\#(E; \tau > i\Delta)}{\#(E)} \tag{18}$$

Where SP_b is the baseline conditional survival probability and *E* is the set of equipment in the training dataset. Combining these two types of condition survival probabilities, each observation's conditional survival probability (SP_{obs}) of the equipment covered by *n* number of patterns during the *i*^th period is described by the Equation 19 (Esmaeili, 2012):

$$SP_{obs}(i) = \begin{cases} \dfrac{\frac{\sum_{p=1}^{n} SP_p(i)}{n} + SP_b(i)}{2} & ; if\ t = 0 \\ \dfrac{\frac{\sum_{p=1}^{n} SP_p(i) + SP_{former}(i+1)}{n+1} + SP_b(i)}{2} & ; if\ t < 0 \end{cases} \tag{19}$$

The survival probabilities of each observation (SP_{obs}) are then transformed into the MRL using Equation 4.

The LAD model follows these steps to train using the training dataset. Once the LAD model is trained, to determine the patterns, discriminant function, and survival probabilities, the testing dataset results are calculated to compare model performance.

3.3 Neural Network Design

As previously mentioned, ANN components can be arranged in many combinations, meaning that there are many possible architectures that can be applied to a problem. In order to employ ANNs, an effective architecture for the dataset under study should be determined. Consequently, the first step in employing an ANN model is the development of an optimal ANN structure design method. The method, designed to determine the best ANN architecture, uses a combination of k-fold cross validation (Schneider, 1997) and Taguchi methods (Peace, 1993; Taguchi & Konishi, 1987).

Cross validation (Alexe, et al., 2004; Gubskaya, et al., 2011; Peace, 1993) is used to evaluate a learning method's performance on new predictions. The most basic cross validation method is called the holdout method (Schneider, 1997). In it, a dataset is divided into a training set and a testing set. The model is trained using the training set but evaluated using the test set. In k-fold cross validation (Schneider, 1997), the dataset is divided into *k* subsets and then the holdout method is applied *k* times, each time using a different subset for evaluation and the remaining subsets for training. This method eliminates the variation from specific characteristics of the samples making up the training or testing dataset. In ANN training, an additional subset of data is used for validation, which optimizes the network's ability to generalize. In this case, two different subsets are used for evaluation and validation, respectively, while the remaining subsets are used for training during the *k* times in which the holdout method is applied for k-fold cross validation.

Taguchi methods (Taguchi & Konishi, 1987) have been introduced by Dr. Genichi Taguchi to reduce the size of experiments. Orthogonal arrays (Taguchi & Konishi, 1987) are used to determine optimal experimental parameters. Orthogonal arrays are advantageous because they can relate the parameters and their levels in a balanced way without testing all combinations. Once the experiments are completed, signal-to-noise (S/N) ratios (Taguchi & Konishi, 1987) can be calculated to evaluate each parameter's performance. S/N ratios relate experimental noise with the expected signal. In ideal cases, the signal from the system is much stronger than the noise. This means that a high S/N value is preferred. Different S/N functions can be calculated depending on the experiment's objective. In this case, the objective is to minimize the mean squared error of the ANN design. The smaller-the-better S/N ratio used. This is described by Equation 20 (Peace, 1993; Taguchi & Konishi, 1987):

$$S/N = -10\log\left(\frac{\sum_{i=1}^{n} y_i^2}{n}\right) \quad (20)$$

Where y_i is the performance metric that is measured and n is the number of experiments. In this case the performance metric is the mean squared error of the testing subset in the dataset under study.

Taking the requirements of k-fold cross validation and the Taguchi method into account, the method for determining the ANN's optimal architecture is designed. The steps for this method is generally described in Figure 6.

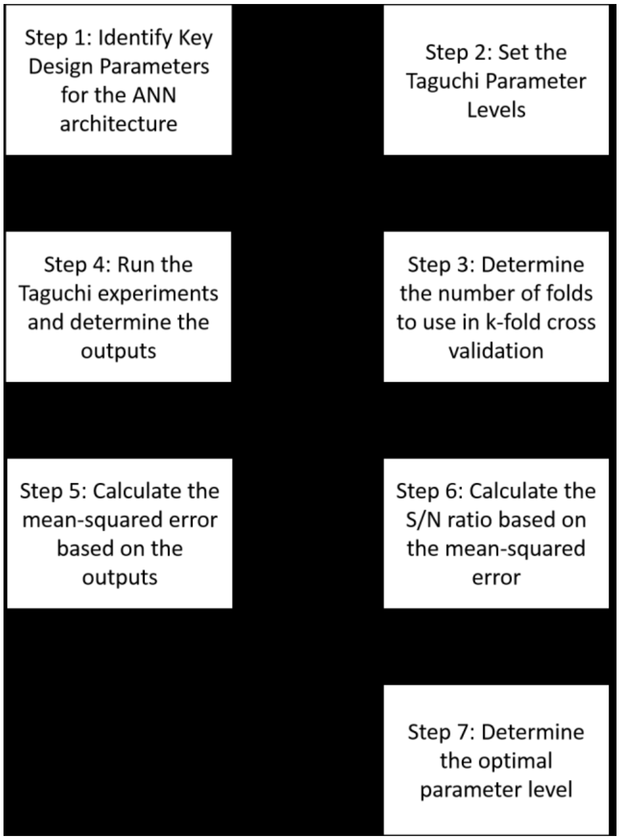
Step 1: Identify Key Design Parameters for the ANN architecture
Step 2: Set the Taguchi Parameter Levels
Step 4: Run the Taguchi experiments and determine the outputs
Step 3: Determine the number of folds to use in k-fold cross validation
Step 5: Calculate the mean-squared error based on the outputs
Step 6: Calculate the S/N ratio based on the mean-squared error
Step 7: Determine the optimal parameter level

Figure 6: Flowchart for Step 2b: Neural Network Design of Research Methodology

In the first step of the ANN design method, key design parameters are identified. Key parameters are components that can change the architecture of the ANN, such as the number of nodes, the number of hidden layers, and the training algorithm. The second step requires setting levels for each of the design parameters in a way that fits into a Taguchi orthogonal array. Levels that correspond to the design parameters are limited to sensible values for the parameters. For example, the number of nodes is limited to an integer value that is greater than zero. The next step is to determine the number of folds used in cross validation, which doubles up as the number of trials completed for the Taguchi experiment. This occurs because k-fold cross validation cycles through the partitioned data *k* times. After these decisions are made, the Taguchi experiments can be run to determine the performance metric output. In this case, the experiment is evaluated on the mean-squared error (MSE). The outputs from the Taguchi experiment are used to calculate the S/N ratio and evaluate the optimal level of each parameter. In this way, the optimal ANN is determined for the dataset.

Using the method described, two separate experiments are completed to design a feedforward ANN (FFNN) and a feedback ANN (FBNN). These are designed separately because the design parameters in these two types of ANNs are not the same and the absence of certain parameters cannot be factored into the Taguchi orthogonal matrix (like the absence of the different time delayed feedback connections in the FFNN). Matlab scripts for these two ANN designs can be found in APPENDIX A. The FFNN experiments with two 4-level parameters and one 2-level parameter (Table 4). The parameters are the training algorithm, the number of nodes, and the number of hidden layers. The L16 orthogonal matrix (Taguchi & Konishi, 1987) of the Taguchi method is used. The entries in Table 4 and Table 5 for the training algorithm are in Matlab code, where *trainlm* = Levenberg-Marquardt, *trainbfg* = BFGS Quasi-Newton, *traingdx* = Variable Learning Rate Backpropagation, and *trainrp* = Resilient Backpropagation. A further explanation for these algorithms can be found in APPENDIX B.

Table 4: Design Parameters and Corresponding Levels for Feedforward ANN Design

Feedforward ANN	Training Algorithm	Number of Nodes	Number of Hidden Layers
Level 1	trainlm	5	1
Level 2	trainbfg	10	2
Level 3	traingdx	18	N/A
Level 4	trainrp	30	N/A

The FBNN is a special class of ANNs that have feedback loops. It is often used in time-series forecasting, where the inputs can be considered the current and past conditions of the time series and the output can provide *n*-step ahead predictions based on the past values of the time-series (Jain, et al., 1996; Li, et al., 2001). This means that the input time-delays in the network are a parameter that affects the network's performance. Therefore, the FBNN experiments with the three 4-level parameters shown in Table 5. The number of hidden layers for the ANN is not a parameter in the design table because there are some difficulties involved in the implementation of multiple hidden layers for this type of ANN. That is, when creating a custom Matlab FBNN, the connections between the nodes must to manually-coded for each case. Therefore, a general code for the FBNN is extremely difficult to generate for all cases and a case-specific code is infeasible as the network gets large. Like the FFNN design, a L16 orthogonal matrix (Taguchi & Konishi, 1987) is used in the Taguchi experiment.

Table 5: Design Parameters and Corresponding Levels for Feedback ANN Design

Feedback ANN	Number of Delays	Number of Nodes	Training Algorithm
Level 1	1	5	trainlm
Level 2	2	10	trainbfg
Level 3	3	18	traingdx
Level 4	4	30	trainrp

Using the optimal architecture determined with the methodology described previously, the ANNs can be trained to directly predict the RL for prognostics purposes. Data inputs are used directly in the optimal ANN to train the ANNs and determine the results for the testing data used in evaluation.

3.4 PHM Testing

The last approach that is studied in this thesis is the PHM. This thesis uses the Weibull PHM formulation, which is commonly used in CBM applications (Banjevic, et al., 2001; Jardine, et al., 2006; Lin, et al., 2006). The Weibull distribution (Ebeling, 1997) is an extremely useful probability distribution because it can be used to model increasing or decreasing failure rates. Changes in the shape parameter, β, changes the distribution's skew and can cause it to become like other known distributions. When $\beta < 1$, the probability density function is similar to an exponential distribution with a decreasing hazard rate. When $\beta \geq 3$, the probability density function is like a symmetrical normal distribution with an increasing hazard rate. When $1< \beta < 3$, the probability density function is skewed with an increasing hazard rate. Finally, when $\beta = 1$, the probability density function takes on the characteristic of the exponential distribution with constant failure rate.

As in the case of the LAD model but not in the ANN models, the data inputs are normalized (Kim, 1999) before training. Normalization is used to reduce the size of the vector of covariates, which can also decrease the number of covariate hazard rate parameters.

The normalized data is partitioned for training and testing. The training of the Weibull PHM involves determining the unknown parameters of the Weibull baseline hazard rate and the unknown parameters of the covariate hazard rate. Maximum likelihood estimation (Ebeling, 1997) is used to completely specify the unknown parameters in this function. In this method, the unknown parameters are determined by maximizing the value of the likelihood function in Equation 21 for given values of $t_1, t_2, \ldots, t_n$ (Ebeling, 1997).

$$L(\theta_1, \ldots, \theta_k) = \prod_{i=1}^{n} f(t_i|\theta_1, \ldots, \theta_k) \tag{21}$$

Where, $f(t)$ is the probability density function with the unknown parameters θ_k. However, depending on the function $f(t)$ and number of unknown parameters, this problem is not necessarily easy to solve. Instead, the maximum of the logarithm of Equation 21 is often easier to solve. As a result, the PHM's parameters in this thesis are estimated using the log likelihood estimator.

Using the hazard rate, whose parameters are estimated using the log likelihood estimator, the MRL is calculated using Equation 9 and Equation 10. The remaining portion of data, the testing dataset, is applied to this Weibull PHM and its results are compared to the other models baseline for model comparison.

CHAPTER 4 EXPERIMENTAL RESULTS AND ANALYSIS

The methodology outlined in Chapter 3 is applied on the Turbofan Engine Degradation Simulation dataset (Saxena & Goebel, 2008). This dataset is made available by NASA and is a run-to-failure CM set created by simulating aircraft gas turbofan engines using the C-MAPSS (Commercial Modular Aero Propulsion System Simulation) tool (Saxena, Goebel, Simon, & Eklund, 2008). This dataset will be referred to as the CMAPSS dataset. Four subsets of data are provided with different combinations of operational settings and failure types. The subset with the most operational settings and failure types is selected for analysis. Information on the equipment's age, operational settings, and 21 condition variables are available.

4.1 RESULTS

The CMAPSS dataset (Saxena & Goebel, 2008) consists of 249 pieces of equipment that have run to failure. There are approximately 60,000 data points. During data preprocessing, every 10th data value is extracted resulting in approximately 6,000 data values. Every 10th data value is extracted because this thesis follows the methodology outlined by Esmaeili (2012). For further justification of this value please refer to Esmaili (2012). Since the CMAPSS dataset has more than two condition indicators (21 condition indicators), PCA is completed on the condensed dataset to reduce the dataset dimensions further. Table 6 displays the eigenvalues, variability, and cumulative variability results of the first ten PCs for the CMPASS dataset. Note that these eigenvalues are also used in Figure 4.

Table 6: Eigenvalues, Variability, and Cumulative Variability

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
Eigenvalue	16.827	3.643	0.400	0.056	0.042	0.015	0.009	0.003	0.002	0.001
Variability	0.801	0.174	0.019	0.003	0.002	0.001	0.000	0.000	0.000	0.000
Cumulative Variability	0.801	0.975	0.994	0.996	0.998	0.999	1.000	1.000	1.000	1.000

As the table demonstrates, the first and second PCs accounts for 80.1% and 17.4% of the total variance, respectively. As stated before, this thesis uses the amount of acceptable cumulative variance method (Johnson, 1998; O'Rourke, et al., 2005) for determining the number of PCs to keep. The cumulative variance threshold is 95%. The first two PCs are chosen to represent the data because they account for 97.5% of the total variance which fulfills the given cut-off criteria.

The PCs (Johnson, 1998; Lin, et al., 2006), which are the product of the eigenvector and original data value, are generated and used as inputs for the three models under study. A sample of the transformed CMAPSS data is displayed in Table 7, showing one piece of equipment with 23 consecutive observations. Each observation is described by the age and two PCs. The last observation, which is shown in a darker background, is the failure observation. This means that the last observation belongs to the failure (positive) class while the remaining observations belong to the survival (negative) class.

Table 7: Set of Condition Data for a Test Equipment

Observations		Attributes		
Equipment ID	Observation Time	Age	PC1	PC2
1	0	0	6.82	-1.40
1	1	1	1.73	0.13
1	2	2	-5.87	-3.69
1	3	3	1.72	0.13
1	…	...	…	…
1	19	19	-2.55	1.96
1	20	20	3.68	-0.15
1	21	21	3.69	-0.16
1	22	22	-2.52	1.92

From this size-reduced dataset, sets of training data, validation data, and test data are extracted for use in the training and analysis of the three models. First, 15 pieces of the 249 pieces of equipment are randomly chosen for the test dataset. Then, another 15 pieces of equipment are randomly partitioned to the validation dataset. Afterwards, 10 training datasets consisting of 70 pieces of equipment each are randomly extracted from the remaining 219 pieces of equipment. Six pieces of equipment are not partitioned to any of the 10 training datasets and are, therefore, not used in the analysis because of the random nature of the partitioning method.

Using the 10 training sets, 10 models instances are trained for each prognostics method. The predicted MRL of the test set data is used for evaluating the approaches.

4.1.1 LAD

The proposed LAD methodology is carried out using Esmaeili's (2012) Python-programmed LAD model with minor coding adjustments to the sensitive discriminating method for cut-point definition (though not to the algorithm). The data and results are interfaced through Excel and the MILP is solved using the CPLEX Python API.

Table 8 shows a sample of the LAD approach's best prognostics results that correspond to the data in Table 7. For each observation, the table includes information on the MRL prediction (column 4), the actual RL (column 3), and the difference between these two values (column 5). It is noted that although the example shows all positive errors, errors can be negative.

Table 8: Set of CMAPSS Prognostics Results (from LAD) for a Test Equipment

Observations		Residual Life		
Equipment ID	Observation Time	Actual RL	Predicted MRL	Error
1	0	22	20.41	-1.59
1	1	21	20.13	-0.87
1	2	20	19.56	-0.44
1	3	19	19.17	0.17
1	…	…	…	…
1	13	3	11.32	8.32
1	14	2	8.63	6.63
1	15	1	10.08	9.08
1	16	0	6.43	6.43

In Table 8 the MRL is calculated by the Python-LAD model following the relationship outlined by Equation 4. The actual RL is the time difference between the failure moment and the current observed moment. The difference between the two residual life values (Predicted MRL – Actual RL), or the error, is used for performance evaluation. The same prognostics information are generated for each of the modeling approaches, which can be found in APPENDIX C.

As an indicator of the model's performance, the relationship between the actual residual life and the errors are studied. This relationship can be observed in Figure 7, where the error is given on the y-axis and the actual RL, in units of 10s of cycles, is given on the x-axis. It is noted that for higher residual life values, the model makes more early predictions and underestimates the residual life (negative error). However, as the equipment approaches failure, the predictions tend to overestimate the residual life (positive error). It can also be seen that the number of predictions that have negative error is greater than the number of predictions that have positive error.

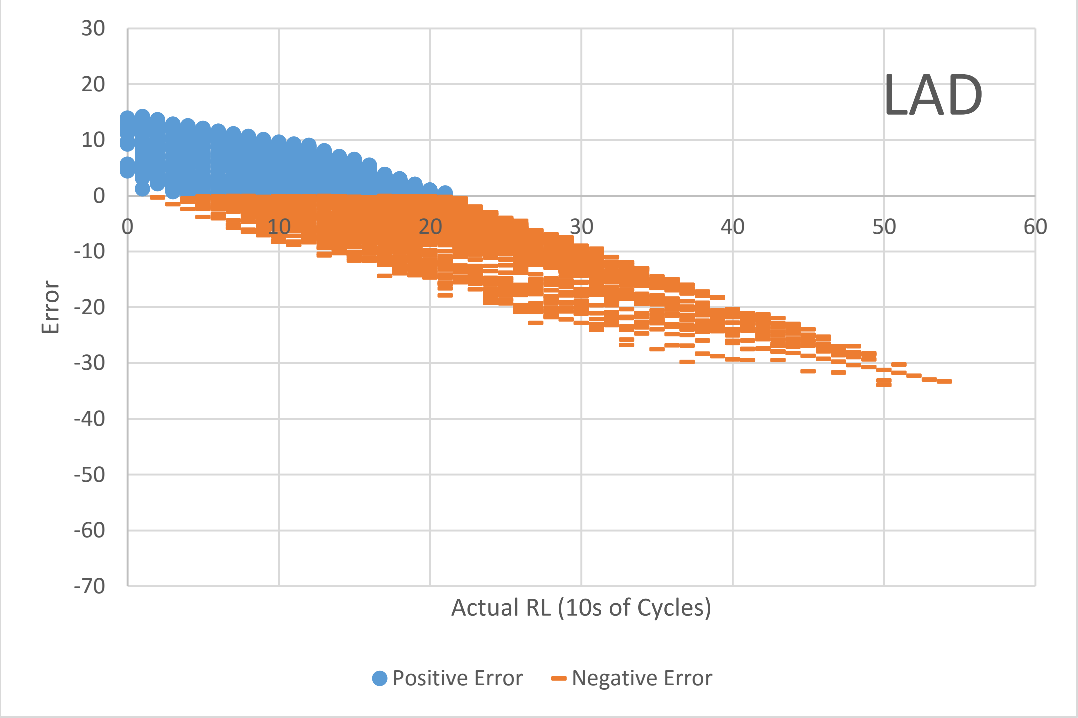

Figure 7: Error as a function of the Actual RL for LAD CMAPSS Results

4.1.2 Neural Networks

The ANN methodology described in Chapter 3 is implemented using the neural network toolbox provided in Matlab 2015b. A Matlab script (APPENDIX A) for the feedforward ANN's (FFNN) and the feedback ANN's (FBNN) model design are written to perform k-fold cross validation instead of using the neural network user interface.

Using the parameters and levels described in Table 4, a 10-fold cross validation structure, and the Matlab script for FFNN design, the Taguchi experiment is performed to determine the optimal architecture for the FFNN. S/N ratios for the first, second, third, and fourth level of the training algorithm parameter are calculated to be -32.04, -33.61, -36.45, and -34.19 respectively. The S/N ratios for the first, second, third, and fourth level of the number of nodes parameter are -34.85, -34.13, -33.61, -33.70, respectively. -34.15 and -34.00 are the S/N ratios obtained for the remaining parameter's (number of hidden layers) first and second levels. These values are presented in Table 9. APPENDIX D contains full Taguchi experiment results.

Table 9: S/N Ratios for Optimal FFNN Architecture Taguchi Experiment

Parameter	Level 1	Level 2	Level 3	Level 4
Training	-32.04	-33.61	-36.45	-34.19
Nodes	-34.85	-34.13	-33.61	-33.7
Layers	-34.15	-34.00	N/A	N/A

Comparing the S/N ratios between each parameter finds that training algorithm has the largest effect on the performance of the FFNN. Further comparing the S/N ratio values between each level finds the optimal FFNN architecture. The largest S/N ratios are associated with the best parameter levels. The optimal FFNN has 2 hidden layers and 18 nodes in each layer (Figure 8) and is trained with the Levenberg-Marquradt algorithm. The Levenberg-Marquradt algorithm (trainlm, 2016) is a supervised backpropagation algorithm that interpolates between the gradient descent method and Newton's method. It is suggested by Matlab as a first choice supervised training algorithm.

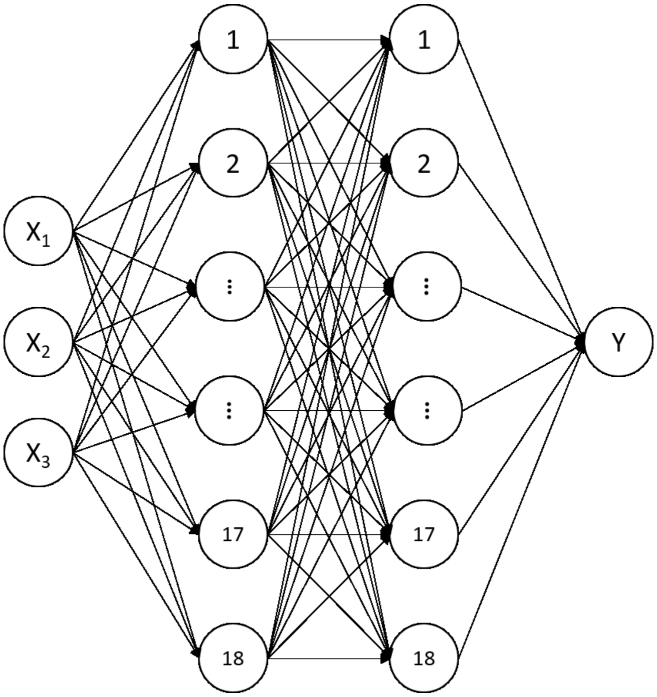

Figure 8: Optimal Feedforward ANN Architecture for CMAPSS Data

A second Taguchi experiment is done to determine the optimal architecture of a FBNN for the CMAPSS dataset. The parameters and corresponding levels described in Table 5, a 10-fold cross validation structure, and the Matlab script (APPENDIX A) for FBNN design are used to complete the experiment. Presented in ascending order of the levels, the following S/N ratio values are obtained for the parameters. The S/N ratios for the number of delays are -22.61. -20.44, -23.86, and -23.33. The S/N ratios for the number of nodes are -22.33, -21.43, -24.07, and -22.40. The S/N ratios for the final parameter, the training algorithm, are -8.53, -23.00, -32.63, and -26.06. These ratios are also presented in Table 10 (further results in APPENDIX D).

Table 10: S/N Ratios for Optimal FFNN Architecture Taguchi Experiment

Parameter	Level 1	Level 2	Level 3	Level 4
Delay	-22.61	-20.44	-23.86	-23.33
Nodes	-22.33	-21.43	-24.07	-22.40
Training	-8.53	-23.01	-32.63	-26.06

By comparing the S/N ratios between parameters, it can be concluded again that the training algorithm parameter has the most effect on the ANN performance. Comparing between parameter levels, the optimal architecture is found to feature a 2-unit time delay with 10 nodes in the hidden layer (Figure 9). The training algorithm used to train this network is the Levenberg–Marquardt algorithm (trainlm, 2016). It is of interest to notice that this is the algorithm that Matlab recommends.

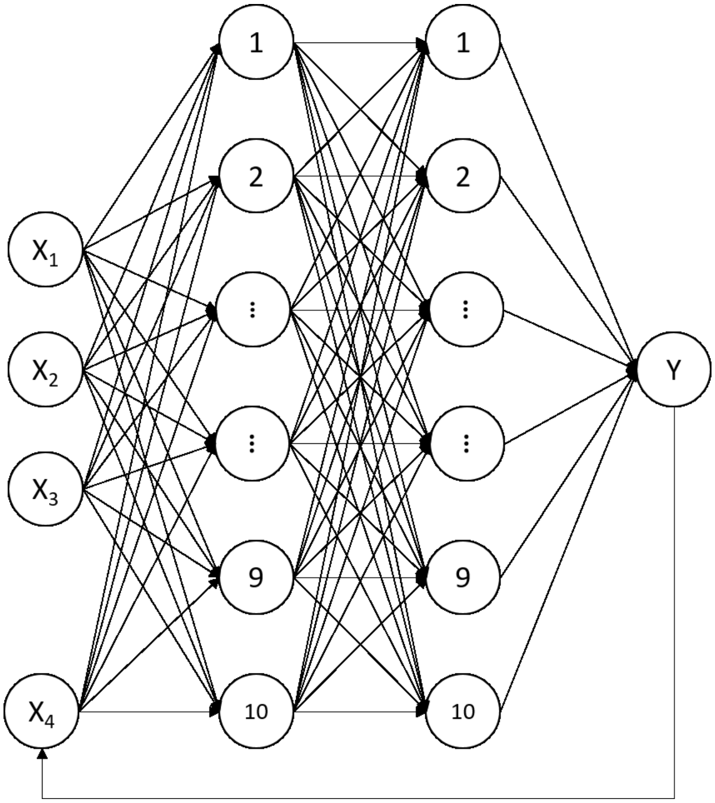

Figure 9: Optimal Feedback ANN Architecture for CMAPSS Data

Using these optimal architectures, prognostics results (found in APPENDIX C), in the same format as the sample provided in Table 8 for the LAD model, are obtained. However, unlike the LAD model, residual is obtained directly from the trained ANN rather than calculated using a MRL equation (Equation 4 and Equation 10). Plotting the actual RL and error reveals their relationship. Figure 10 shows the FFNN's prediction errors (y-axis) as a function of actual RL (x-axis), in units of 10s of cycles. The graph shows that the error increases negatively with residual life. In other words, when the equipment is still in good condition, the model tends to make early predictions of failure. On the other hand, as the residual life decreases, the error becomes closer to zero. The error near failure (zero residual life) is also consistently positive, which means that the model overestimates the failure time.

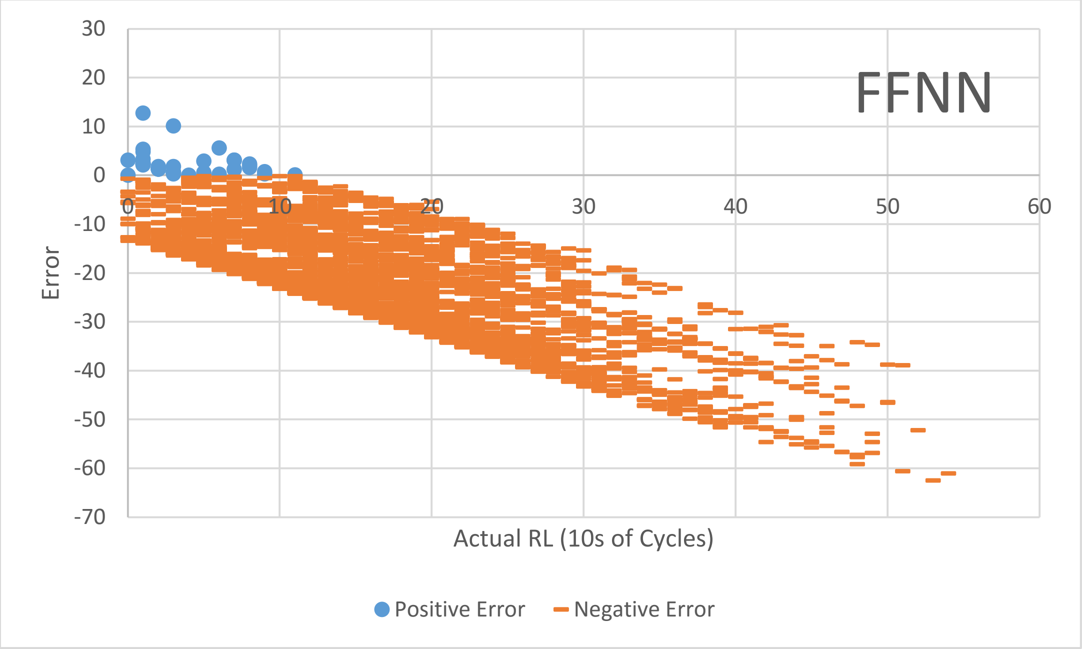

Figure 10: Error as a function of the Actual RL for Feedforward ANN CMAPSS Results

A similar relationship is observed between the actual RL and the error for the FBNN, which can be seen in Figure 11. In the figure, the FBNN's prediction errors are on the y-axis and the actual RL, in units of 10s of cycles, is on the x-axis. As the residual life increases and the equipment is farther from failure, the error is more negative. As the residual life decreases and the equipment is close to failure, the error is more positive. However, the number of negative and positive error observations are more evenly distributed. In other words, based on the Figure 11, it appears that there are almost as many positive error observations as there are negative error observations.

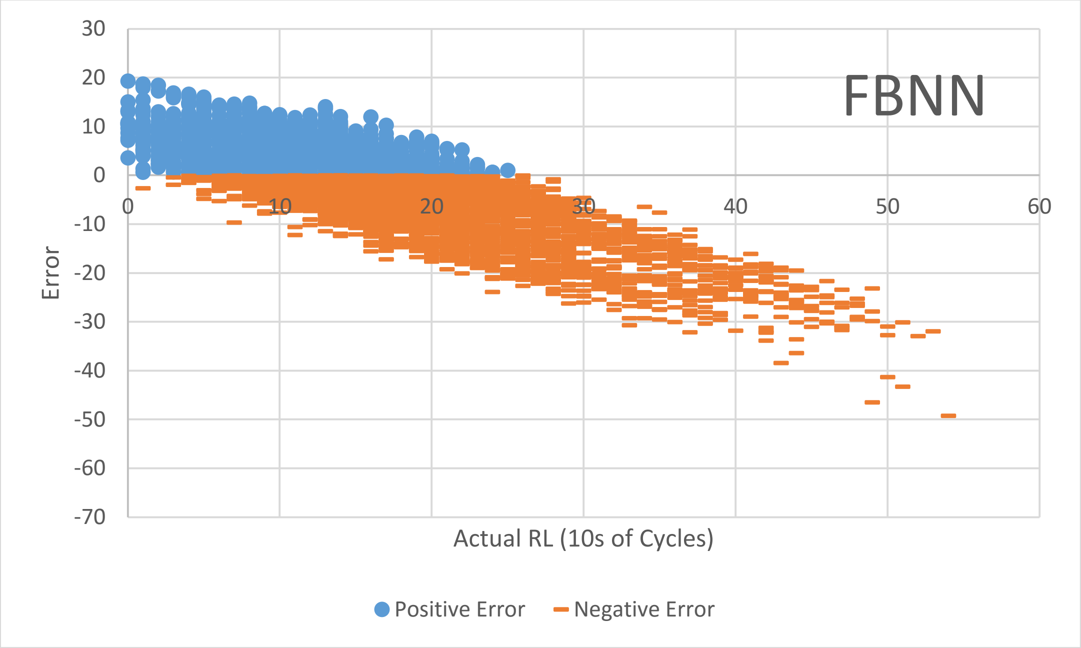

Figure 11: Error as a function of the Actual RL for Feedback ANN CMAPSS Results

4.1.3 PHM

The application of the Weibull PHM is performed using Esmaeili's (2012) Python-coded PHM. The log likelihood estimator (Ebeling, 1997) is optimized using Python's SciPy library. As with the LAD model, the data and results are processed using Excel.

Using the log likelihood estimator, the parameters for 10 model instances are determined. The specified Weibull PHM, with now known parameter values, is used to obtain prognostics results (found in APPENDIX C) in the same format as those shown in Table 8. The MRL is obtained using the hazard rate function, according to Equation 9 and Equation 10. The relationship between the actual RL (x-axis), in units of 10s of cycles, and the error for those prognostics results (y-axis) are shown in Figure 12. Negative error for a large residual life and positive error for a small residual life tendencies can again be observed in this relationship. Looking at the proportion of positive to negative error observations, it is observed that the number of positive error observations seems less than the number of negative error observations, just like the models before.

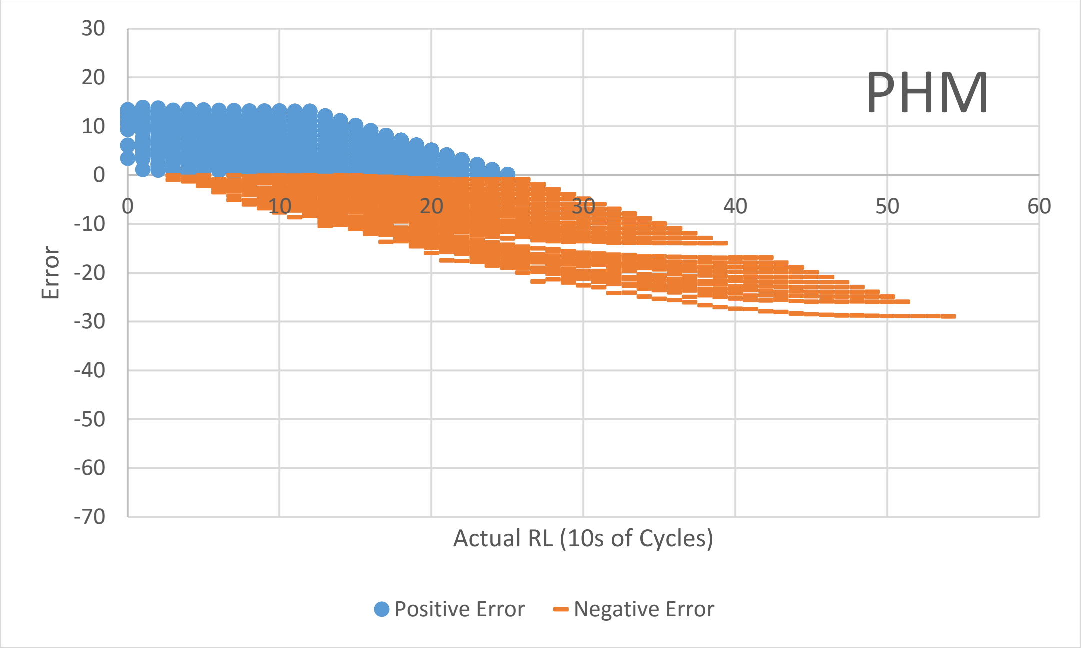

Figure 12: Error as a function of the Actual RL for PHM CMAPSS Results

4.2 Comparison of Experimental Results

The performance of these CBM approaches are evaluated based on three metrics. These metrics are the error, the half-life error, and the cost score, which are explained further in this section. The Friedman matched-pair test (Friedman, 1940) and the Wilcoxon signed rank test (Wilcoxon, 1945), from Minitab, are used to evaluate the statistical differences between the approaches. These statistics are calculated based on the testing subset.

For each model, 10 instances are trained for comparison and statistical testing. The results displayed in the following sections are results from the best performing model instance. The best performing model instance is determined according to the lowest mean squared error between instances. This means that only one model instance from each of the four models are displayed in the following results.

4.2.1 Evaluation over Error

First, performance is measured by the difference between the predicted MRL and the actual RL. The purpose of this metric is to evaluate the distance between the prediction and the actual failure time. Differences in this distance may affect maintenance decision making, which is why the direction of the distance is taken into account and absolute error is not considered. The impacts between different distances (or error) of predictions will be considered later in Section 4.2.3.

Table 11 compares the performance of the four models based on error. More information on the models' error performance is found in APPENDIX C. As observed from the table, the LAD and PHM approaches have the least variance, with a standard deviation of 9.47 and 9.57, respectively. In addition to this, these two models also have the best performance as the mean and median are closest to zero. The LAD model has the better median at -0.04 (vs. 0.17 in the PHM) but the PHM model has the better mean at -2.14 (vs. -2.70 in the LAD). Between the FFNN and the FBNN, the FFNN has the lower variance while the FBNN has mean and median closer to zero.

Table 11: CMAPSS Error Summary

	LAD	PHM	FFNN	FBNN
Minimum	-33.31	-28.90	-32.08	-47.14
25% Percentile	-8.51	-8.70	-10.00	-8.48
Median	-0.04	0.17	-0.96	-0.33
75% Percentile	4.50	5.97	4.67	5.53
Maximum	10.60	11.21	12.29	16.14
Mean	-2.70	-2.14	-3.46	-2.48
Std. Deviation	9.47	9.57	9.90	10.50
Range	43.91	40.11	44.38	63.28

Observing the comparison visually in Figure 13 some of the earlier conclusions are confirmed. LAD and PHM have a median that is closest to zero and FFNN has a median farthest from zero. It can also be seen, that there are outliers in the error for the LAD, FFNN, and FBNN approaches. Overall, the PHM and the FFNN have the least outliers. The outliers occurring for LAD and FBNN are in the negative error direction. Looking further into these outliers, it is found that these outliers occur when the residual life of the equipment is large. Negative and positive error can have different impacts on failure consequences. In the case where failure costs are greater than the replacement cost, outliers in the negative direction have less impact on the maintenance decision making than if the outliers had occurred in the positive direction. This is because it is more beneficial to have forewarning about failure than to have a late warning about failure for this situation. The impacts of negative and positive error (and by extension negative outliers) will be further investigated by the cost score later on.

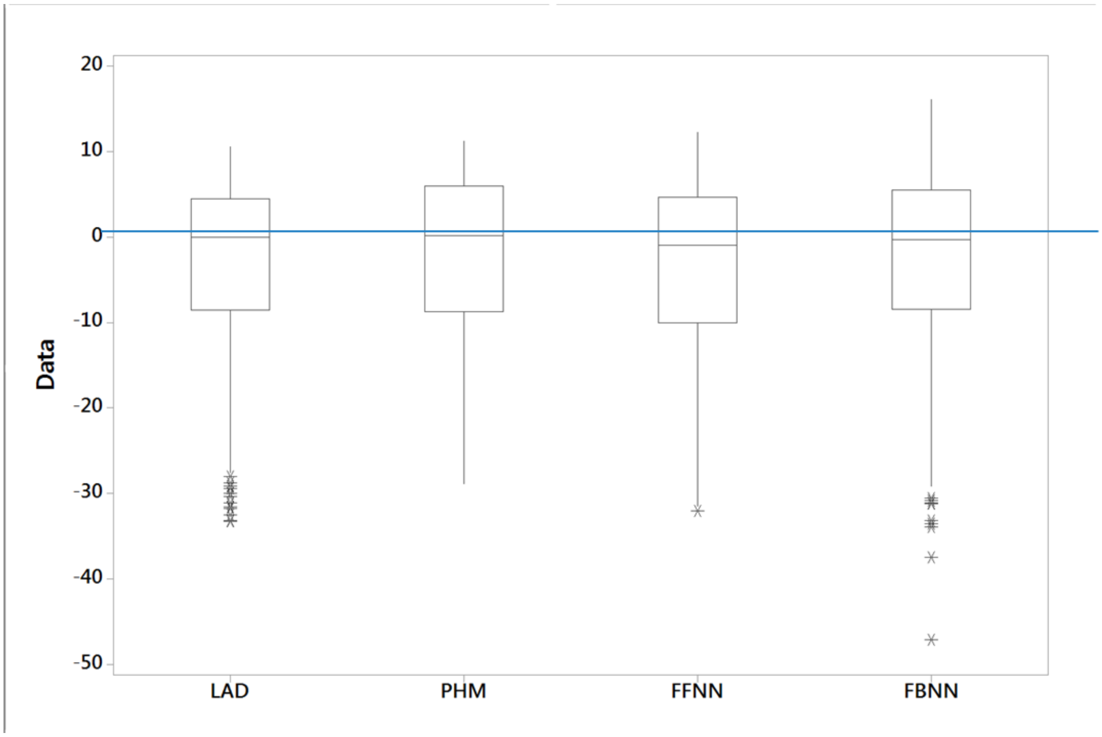

Figure 13: Box Plot Comparison of CMAPSS Errors between Models

The Friedman matched-pair test is applied to test the performance differences of these four models. The results from statistical testing for this and other metrics is provided in APPENDIX C. The resulting p-value is 0.293, which shows that the four models are not statistically different. However, as the Wilcoxon signed rank test is easily calculable, the models are further tested for differences in performance. Comparing LAD to PHM, FFNN, and FBNN obtains p-values of 0.221, 0.683, and 0.154, respectively. The p-values from comparing the PHM to FFNN and FBNN are 0.308 and 0.019, respectively. The comparison of the FFNN and the FBNN has a p-value of 0.359. The only test that shows significantly different performance is between the PHM and the FBNN, having a p-value of 0.019 which is less than the critical value of 0.05 for a 95% confidence interval.

Considering model performance based on the difference between the predicted MRL and the actual RL from another perspective, percent error can be calculated. This measure can provide further insights into the accuracy of each model. Table 12 displays the percent error summary for the four models. The table reveals that the LAD model is the best performing in terms of lowest percent error on average and that the FFNN model has the lowest standard deviation.

Table 12: CMPASS Percent Error Summary

	LAD	PHM	FFNN	FBNN
Minimum	0%	0%	0%	0%
25% Percentile	23%	29%	29%	27%
Median	42%	44%	47%	45%
75% Percentile	951%	1081%	862%	1338%
Maximum	951%	1081%	862%	1338%
Mean	81%	83%	82%	97%
Std. Deviation	137%	139%	127%	174%
Range	10%	11%	9%	13%

From earlier results, it is noted that negative errors tend to occur when the residual life is large and positive errors tend to occur when the residual life is small. The question arises of how the error metric performs for each residual life value. Table 13 shows the average (avg. error) and standard deviation (std. dev.) of the error metric for actual RLs. Only one instance is observed for actual RL values greater than 37, therefore, the table does not display these results. From the table, it is confirmed that positive error occurs when the residual life is small (positive averages) and that negative error occurs when the residual life is large (negative averages).

Table 13: CMAPSS Error Summary by Actual RL

	ACTUAL RL	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	No. of Instances	15	15	15	15	15	15	15	15
LAD	Avg. Error	5.48	6.44	5.65	4.92	4.37	4.28	4.26	3.91
	Std. Dev.	2.41	2.47	3.00	3.26	3.20	3.25	3.83	3.80
PHM	Avg. Error	6.74	6.22	5.61	5.03	4.49	3.96	3.51	3.05
	Std. Dev.	2.94	3.04	3.35	3.55	3.74	3.97	4.29	4.49
FFNN	Avg. Error	5.41	3.77	4.41	3.80	3.26	2.45	2.61	1.48
	Std. Dev.	2.80	5.39	4.31	4.23	4.10	4.89	4.04	5.47
FBNN	Avg. Error	10.31	8.11	6.35	6.11	5.72	4.85	4.24	4.02
	Std. Dev.	3.38	3.85	5.91	5.17	3.94	4.14	4.69	4.04

	ACTUAL RL	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
	No. of Instances	15	15	15	15	15	15	15	15
LAD	Avg. Error	3.31	2.78	2.31	1.62	0.95	0.57	-0.02	-0.60
	Std. Dev.	3.87	5.22	4.23	4.43	4.62	4.60	4.53	4.73
PHM	Avg. Error	2.70	2.28	1.90	1.46	1.16	0.88	0.57	0.27
	Std. Dev.	4.62	5.40	5.20	5.49	5.81	6.04	6.27	6.61
FFNN	Avg. Error	0.63	0.56	0.26	-0.43	0.34	0.02	-1.25	-0.82
	Std. Dev.	6.39	5.70	5.82	7.15	5.96	6.72	7.62	6.22
FBNN	Avg. Error	2.69	3.13	2.24	1.87	0.67	0.38	-0.62	0.42
	Std. Dev.	5.82	4.33	5.07	4.27	5.11	5.44	5.72	6.54
	ACTUAL RL	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
	No. of Instances	15	14	14	11	10	10	10	8
LAD	Avg. Error	-0.85	-1.92	-2.78	-4.61	-5.52	-6.37	-6.61	-8.53
	Std. Dev.	4.58	4.90	5.17	4.95	4.38	4.83	5.07	4.08
PHM	Avg. Error	0.06	-0.83	-1.00	-3.57	-4.78	-4.95	-5.27	-7.57
	Std. Dev.	6.83	6.86	7.02	6.62	6.34	6.45	7.12	5.80
FFNN	Avg. Error	-0.95	-1.58	-2.38	-4.80	-6.18	-6.64	-6.38	-8.82
	Std. Dev.	6.96	7.18	8.84	6.43	6.13	8.55	7.06	5.62
FBNN	Avg. Error	-1.01	-1.53	-4.55	-4.87	-5.82	-5.89	-7.42	-8.42
	Std. Dev.	6.21	7.71	5.40	5.26	4.75	3.98	5.61	6.02
	ACTUAL RL	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
	No. of Instances	8	8	6	6	6	6	5	5
LAD	Avg. Error	-8.48	-8.94	-11.44	-12.15	-12.64	-12.94	-13.93	-14.56
	Std. Dev.	3.75	4.01	4.12	4.21	4.34	4.28	3.56	3.56
PHM	Avg. Error	-7.67	-7.98	-10.94	-11.15	-11.35	-11.49	-13.21	-13.35
	Std. Dev.	6.20	6.60	5.36	5.59	5.82	6.05	5.56	5.76
FFNN	Avg. Error	-9.37	-9.08	-12.39	-12.64	-12.61	-12.97	-13.94	-14.71
	Std. Dev.	8.26	6.78	4.90	5.10	6.19	5.39	5.43	5.27
FBNN	Avg. Error	-8.65	-12.75	-11.35	-11.53	-12.65	-14.62	-13.40	-12.72
	Std. Dev.	7.36	5.04	5.31	5.12	4.10	3.47	3.09	4.23
	ACTUAL RL	**32**	**33**	**34**	**35**	**36**	**37**	**>37**	
	No. of Instances	5	5	5	4	4	3	1	
LAD	Avg. Error	-14.97	-15.64	-16.14	-18.10	-17.94	-19.27	N/A	
	Std. Dev.	3.52	4.00	3.82	5.51	4.39	4.94	N/A	
PHM	Avg. Error	-13.50	-13.52	-13.67	-15.07	-15.18	-16.75	N/A	

	Std. Dev.	6.03	6.02	6.33	7.02	7.25	8.40	N/A
FFNN	Avg. Error	-14.81	-14.34	-15.09	-16.72	-11.05	-18.08	N/A
	Std. Dev.	5.85	7.28	6.70	6.47	15.14	8.06	N/A
FBNN	Avg. Error	-13.13	-16.04	-15.82	-19.09	-13.76	-25.73	N/A
	Std. Dev.	4.17	6.08	7.72	4.27	7.08	2.90	N/A

Some general conclusions can be drawn for the maintenance decision maker based on these results. LAD and PHM are the preferred models because they have the lowest variance and average error performance. This is especially true for cases where the equipment under CM requires high reliability and the impacts of inaccurate predictions are large, LAD and PHM are preferred models because of their low variance and range of error. These models also do well in terms of percent error. On the other hand, when this is less of a concern and accuracy of the predictions has more weight, the FFNN and LAD are preferred since they have one of the lowest average percent error and variance. However, the FFNN has a lower performance in terms of error.

4.2.2 Evaluation over Half-Life Error

In addition to the error, the performance comparison also considers what this thesis terms the half-life error. This metric evaluates the model's behaviour as the engine approaches failure. In literature, the concept of half-life is used to measure the time it takes for a specimen to reach half of its initial useful life. This concept can be used to measure a specimen's rate of decay. This thesis takes advantage of this concept to evaluate model performance when the equipment has deteriorated. The half-life error is the error calculated from the half-life of the longest surviving equipment in the dataset. Consider an example where the longest survival time for a set of equipment is 10 units of time. In this case, the half-life interval is 5 units of time and the half-life error is calculated over this interval. In the CMAPSS dataset, the equipment with the longest observed lifetime survives up to 50 time units. Therefore, the half-life is defined as 25 time units.

The performance of the half-life error metric for the four models are displayed in Table 14. Further half-life error results are found in APPENDIX C. Like the results for the error metric, the LAD and PHM have the least variance, with a standard deviation of 5.74 and 6.38, respectively. However, under this metric, the FFNN which is previously worst performing under the error metric, now has the best performance with a mean and median closest to zero is the FFNN. The FFNN has a mean and median of -0.11 and 1.52, respectively. The LAD model, which performed well for the error metric, performs the second best for the half-life error metric.

Table 14: CMAPSS Half-Life Error Summary

	LAD	PHM	FFNN	FBNN
Minimum	-16.13	-18.17	-25.88	-22.28
25% Percentile	-2.91	-3.19	-4.80	-4.04
Median	1.79	2.76	1.52	1.88
75% Percentile	5.26	7.10	5.37	6.55
Maximum	10.60	11.21	12.29	16.14
Mean	0.89	1.21	-0.11	1.26
Std. Deviation	5.74	6.38	6.98	7.00
Range	26.73	29.37	38.17	38.42

Comparing the half-life error visually (Figure 14), it can be noted that the PHM has a median that is farthest from zero. The remaining models have similar medians that are closer to zero compared to the PHM. The number of outliers observed in the half-life error box plot is less than the number of outliers observed under the error metric. The outliers occur for the LAD, FFNN, and FBNN. Diving into the outliers, it is found that these are associated with observations occurring when the residual life is large. Just as with the negative outliers identified when evaluating the error metric, these negative-errored outliers have less impact than if the outliers were positive in some cases. This is because knowing about failure before it occurs is advantageous in these cases. This observation is evaluated in the cost score metric.

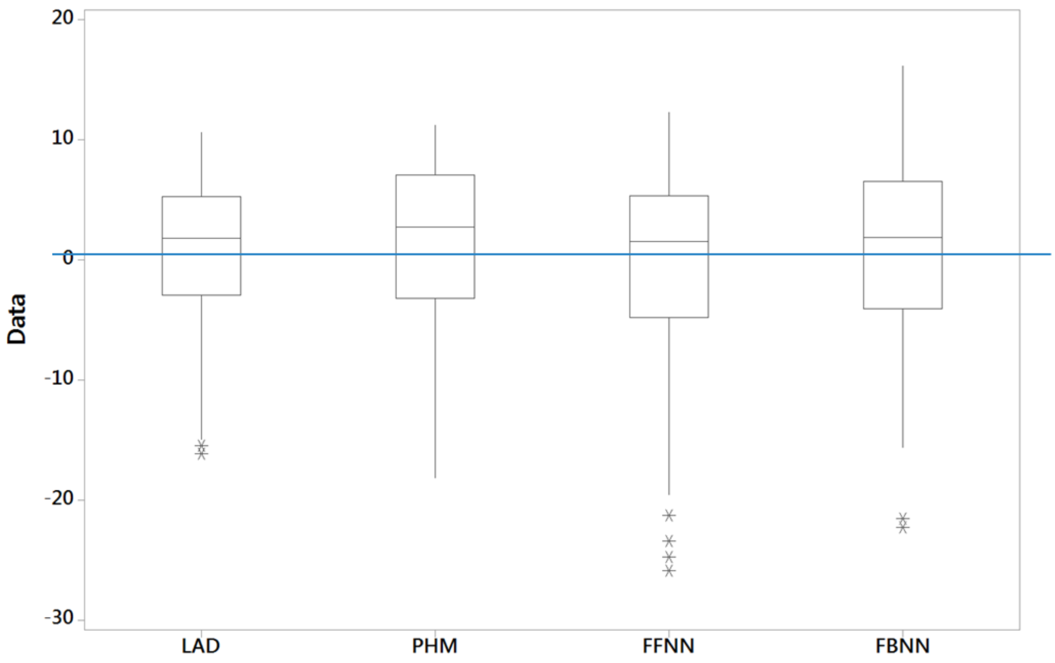

Figure 14: Box Plot Comparison of CMAPSS Half-Life Errors between Models

Testing the half-life error results with the Friedman matched-pair test (p-value = 0.01) shows that at least one of the four models is statistically different from the others. The Wilcoxon signed rank test is used to identify the model (s) that are statistically different. The p-values for the comparison of LAD to PHM, FFNN, and FBNN are 0.126, 0.541, and 0.032, respectively. The comparison of PHM to FFNN and FBNN are 0.185 and 0.006, respectively. Comparing the FFNN and the FBNN produces a p-value of 0.126. These statistical results conclude that the FBNN is statistically different from the LAD and PHM.

If the error for different decreasing intervals, like half the maximum residual life (half-life), a quarter of the maximum residual life (quarter-life), and an eighth of the maximum residual life (eighth-life) is calculated for the models, the behaviour of the models as the equipment approaches even closer to failure can be observed. Figure 15 shows the trends in the error performance at these three intervals. Studying the figure, it is noted that as the residual life interval gets smaller, the median values for all the models get farther from zero in the positive direction. This means that as the equipment approaches failure, the models tend to predict failure after it has already failed. Although it may be hard to differentiate, the median of the FFNN is actually closest to zero in all cases. However, the FFNN also has the most outliers. Again these outliers lie in the negative direction. For the other models, outliers also lie in the negative direction. Overall, the number of outliers decrease as the residual life interval becomes smaller and gets closer to failure for all approaches.

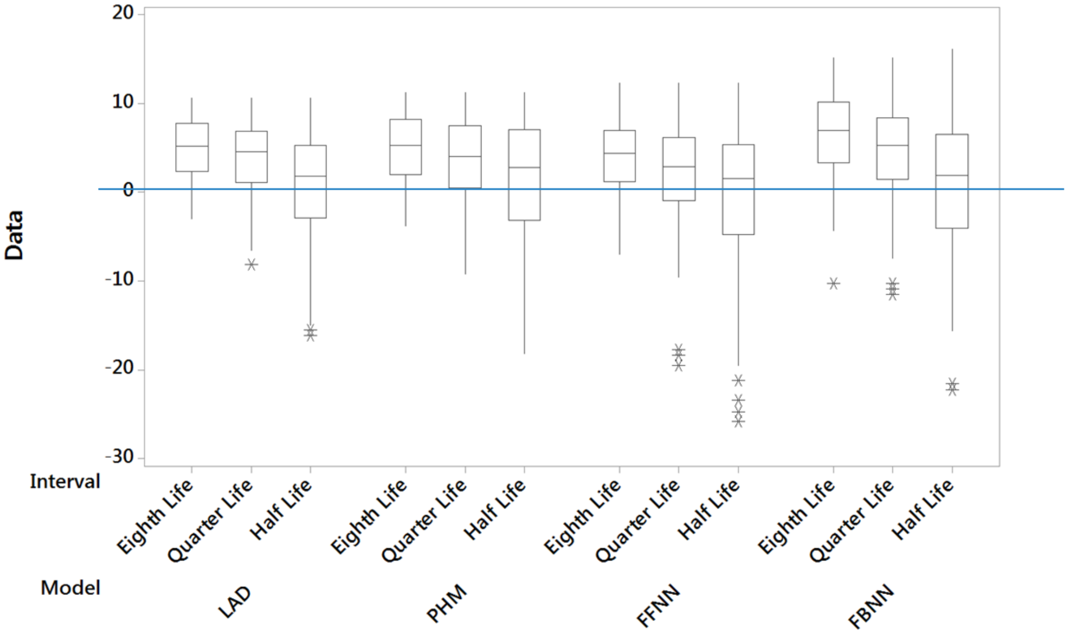

Figure 15: Box Plot Comparison of CMAPSS Interval Errors between Models

From this study of model performance as the equipment nears failure, the maintenance decision maker can refine their decision for prognostics models. In some cases, the cost of failure is much greater than the cost of replacement. For these cases, predicting failure in advance can benefit the decision maker, especially as the equipment approaches the end of its lifetime. In this instance, the FFNN is recommended as the prognostics model because it has the best average performance when the residual life interval is smaller. However, LAD can also be considered because its average half-life error performance follows closely behind the FFNN and its variance is the lowest between the models.

4.2.3 Evaluation over Cost Score

In addition to the performance closer to failure, a metric that considers the consequence of predicting failure before or after the failure actually occurs is also of interest. Consequences of inaccurate predictions of failure include, but are not limited to, monetary losses (costs), safety, and loss of life. The consequences of predicting failure before or after the failure actually occurs have different impacts. For example, if the cost of failure is much greater than the cost of replacement, then it is more beneficial to have forewarning about the failure than a late warning. This holds true especially as the equipment approaches failure. For this case, negative error, which indicates that failure is predicted before its occurrence, is more beneficial than positive error, which indicates that failure is predicted for a time point after the failure actually occurs. This means that negative and positive error can have different effects on maintenance decision making. The cost score takes into account one kind of consequence for these two types of error by assigning a monetary value to the error between the predicted MRL and the actual RL.

Whenever a failure occurs after the predicted failure time (negative error), an early replacement cost is incurred. This early replacement cost refers to the portion of the replacement cost (*C*) that remains unused because the equipment has already been replaced or repaired. For the opposite case, where the failure occurs before the predicted failure time has passed (positive error), a missed failure cost is sustained. This cost is the failure cost (*K*) that is much greater than the replacement cost (*C*). Taking these costs into account for the case with *m* pieces of equipment, each having n_j number of observations, Equation 22 is formulated and called the cost score.

$$Cost = \begin{cases} \sum_{j=1}^{m} \sum_{i=1}^{n_j} K \times \left|d_{ij}\right|^{a} & , if\ d_{ij} > 0 \\ \sum_{j=1}^{m} \sum_{i=1}^{n_j} \frac{|d_{ij}|C}{T_j} & , if\ d_{ij} \leq 0 \end{cases} \tag{22}$$

Where *K* is the failure cost, *C* is the replacement cost, T_j is the survival time of equipment *j*, *a* is an arbitrary value for weighting value, and d_{ij} is the error for observation *i* of equipment *j*. In the cost score equation, *a* is used to differentiate between the monetary impacts of positive error. For instance, a prediction that indicates failure occurring 10 periods after the actual failure will have different costs than a prediction that indicates failure occurring 2 periods after the actual failure.

The cost score's general behaviour is illustrated more clearly in Figure 16, which shows the function under different criteria for certain values of error. Note that the y-axis scales for the functions with negative and positive error are not the same, the missed failure cost is larger than the early replacement cost. On the left of the zero (blue solid line) is the early replacement cost for negative error, calculated with C = 100. On the right of the zero (various dashed lines) is the missed failure cost for positive error, calculated with K = 200. Different values of a are investigated to demonstrate how different error weights can affect the cost score's behaviour. The a values of 0.5, 1, 1.5, and 3 are shown in Figure 16.

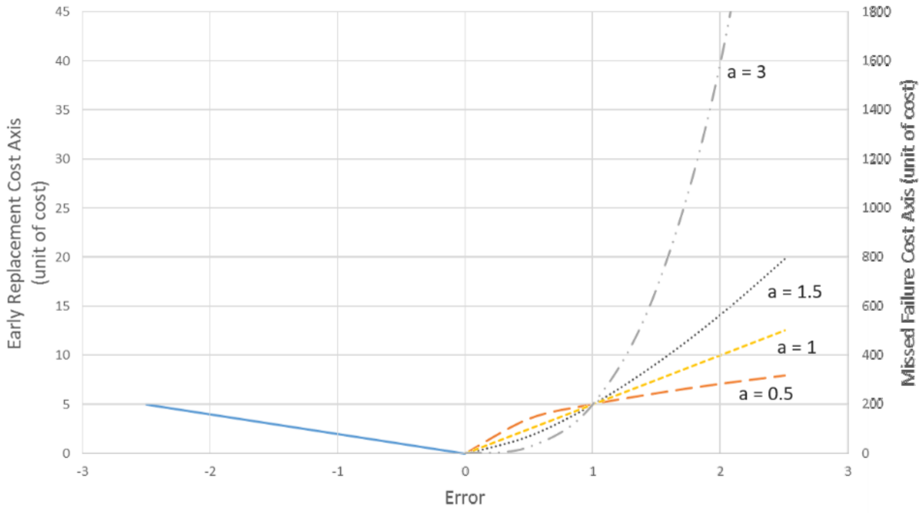

Figure 16: Cost Score Example Illustrated (Not to Scale)

As illustrated, the cost score penalizes the positive error more than the negative error. For different values of a, the cost score for the positive error behaves differently. In the case where the a value is lower than zero (a = 0.5), the cost score penalizes smaller errors more than larger errors. For a = 1, the cost score function is linear with time. For weights that are greater than one (a = 1.5$,$ a = 3), the cost score considers larger errors to have more impact while smaller errors have less impact. As the a value gets larger, the cost score's slope gets steeper. Since the cost score for the negative error does not depend on a, it remains constantly linear.

Using K = 200 units of cost, C = 100 units of cost, and comparing various values of a (0.5, 1, 1.5, 3), the cost score is calculated for data results of each model. Shown in Table 15, the cost score results reveal how the different models behave under different values of a. For a = 0.5, the FFNN has the lowest cost score, followed by LAD, PHM, and finally, FBNN. However, this holds true only until the half-life interval. After the half-life, as the equipment gets closer to failure, the PHM improves in performance and becomes better than LAD. For the cases where a = 1 and a = 1.5, LAD always outperforms PHM with the lower cost score. In addition, the FFNN has the lowest cost score and FBNN has the highest cost score. When a = 3, LAD starts out with the best performance, followed by FFNN, PHM, then FBNN. As the residual life decreases, this behaviour changes. At the equipment's half-life, the FFNN overtakes LAD in performance.

Table 15: CMAPSS Cost Scores

	a = 0.5			
Interval	LAD	PHM	FFNN	FBNN
Entire Life	92,859	100,327	88,364	101,160
Half Life	87,706	95,585	82,520	95,884
Quarter Life	70,610	68,843	59,794	76,803
Eighth Life	44,294	43,251	37,336	49,443
	a = 1			
Interval	LAD	PHM	FFNN	FBNN
Entire Life	202,073	233,678	194,981	251,514
Half Life	196,919	229,022	187,771	246,239
Quarter Life	167,802	171,461	139,733	205,592
Eighth Life	107,639	109,636	89,670	142,181
	a = 1.5			
Interval	LAD	PHM	FFNN	FBNN
Entire Life	479,450	592,378	468,553	691,021
Half Life	474,296	587,748	457,269	685,745
Quarter Life	420,937	450,252	346,523	585,432
Eighth Life	276,134	293,255	228,227	429,451
	a = 3			
Interval	LAD	PHM	FFNN	FBNN
Entire Life	8,314,361	12,036,947	8,466,354	19,473,993
Half Life	8,309,207	12,032,329	8,287,844	19,468,718
Quarter Life	7,889,769	9,670,215	6,516,788	16,932,972
Eighth Life	5,531,776	6,653,126	4,594,162	14,060,827

Statistical testing is performed for the case of a = 1.5. Results using the Friedman matched-pair test, show that there is significant difference between the model behaviour (p-value = 0). The Wilcoxon signed rank test results of this metric show more pronounced differences between model performances. The p-values for the comparison of LAD to PHM, FFNN, and FBNN are 0.541, 0.011, and 0.006, respectively. The ones for the comparison of PHM to FFNN and FBNN are 0.067 and 0.006, respectively. The final p-value, comparing the

FFNN and the FBNN is 0.103. To summarize, this statistics test found that there are significant differences between two pairs of approaches. First, LAD is significantly different from both the FFNN and the FBNN. Second, PHM is significantly different from FBNN.

Taking into consideration the results from this metric, the maintenance decision maker can add another layer to their decision making. Recall, that negative outliers are observed in both the error and half-life error metric. It is noted that in some cases, negative outliers have less impact than positive outliers. As seen from the cost score, negative errors have much less monetary impact than positive error for the case where failure cost is much greater than the replacement cost. This means that the negative errors observed previously will not have much impact on maintenance decisions and the conclusions draw before. Considering models from the cost score perspective, in situations where the failure cost is much greater than the replacement cost and where the impacts of predicting failure after it has already occurred are great, then the FFNN and LAD are preferred. In most cases of *a*, these two models perform the best with the lowest cost score, even as the residual life interval decreases. Recall that outliers in the negative direction (and large residual life) are observed in the earlier results. Since the positive error has a much larger cost than the negative error, these outliers in the negative direction have less impact on the maintenance decision making than if the outliers had occurred in the positive direction. It is important to note the cost score is not tested for the case where K is less than C (K/C < 0), therefore, no conclusions can be drawn for this case.

4.2.4 Discussion of Results

A sample of the overall prognostics results is shown in Figure 17, which illustrates the predicted MRL's general behaviour for each model. The prognostics results (APPENDIX C) corresponding to the data presented in Table 7 are plotted against the lifetime of the equipment (with units of 10s of cycles). The figure shows that the FBNN has the most variable predictions and that the LAD and PHM is generally the most stable.

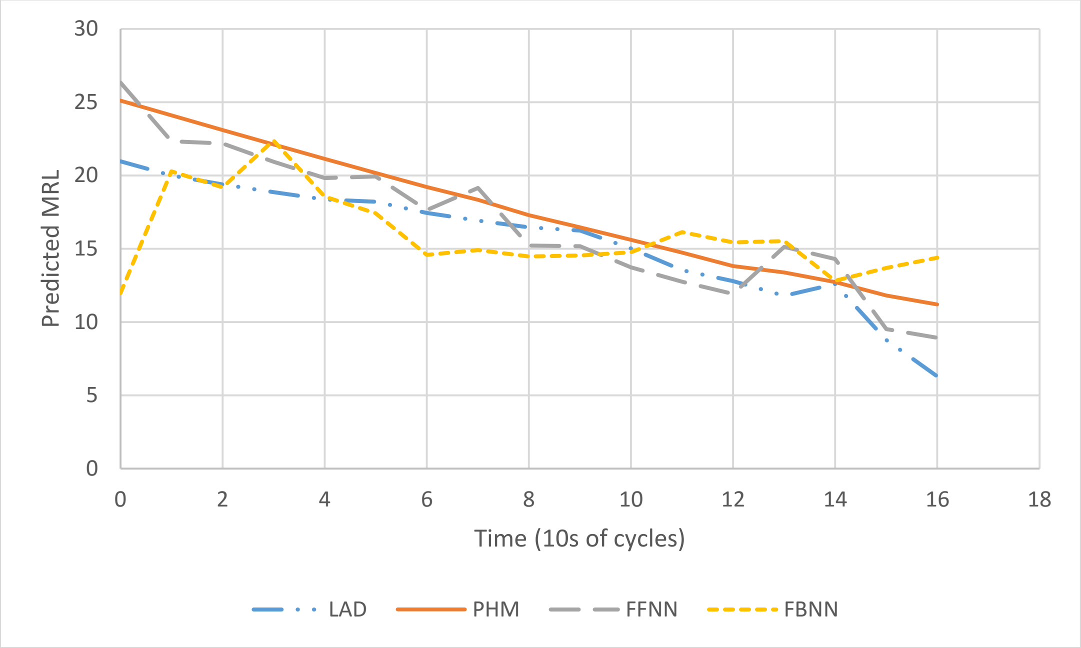

Figure 17: Comparison of Predicted MRL between the Models for a Test Equipment

In addition to the performance of the predictions, another thing that is often considered when choosing a model is the run time. A snapshot of the run time performance is given by Table 16. The run time (in minutes) during the 10-trial training for each approach is compared. The ANN methods solve much faster than the other two methods. In comparison, LAD takes the most time for training (1000 times the fastest training time). It is important to take note that the LAD and PHM modelling are completed using non-commercial programs while the ANNs are modelled in Matlab. In addition, the run time shown is for the training stage of the model. If new data is introduced to the models, different approaches to updating the model will use different run times. For models that require re-training in order to update the model, these run times still hold true. This is the case for the PHM and the ANN approaches. In the case of LAD, if the new data observation is already covered by an existing pattern, there is no need to update the model. If the new data observation is not covered by an existing pattern, the MILP is run once to obtain a new pattern. This means that LAD does not need to be retrained.

Table 16: Run Time (in Minutes) for Each Approach

LAD	PHM	FFNN	FBNN
3928.73	224.83	6.37	3.81

This additional information can further aid the maintenance decision maker. In the case where new data is continually introduced to the model by CM, updates to the model may be done often to increase the prediction ability of the model as more information is collected. This means that a short run time is needed during updating. As such, the ANN approaches or LAD is preferred because the ANN approaches have the shortest training run time between the approaches and the LAD does not require retraining of the entire model.

Taking into consideration the prediction results, the run time results, and the results discussed for the three metrics, the following points can be made to summarize the conclusions obtained from the application of the four models to the CMAPSS dataset:

- The error, for all models, between the predicted MRL and the actual residual life tends to be negative when the residual life is greater and positive when the residual life shorter.
- Based on the overall accuracy of the model, measured by error, the LAD and PHM are the best performing models.
- The models slowly become more accurate as the equipment approaches failure, with the FFNN being the best performing as this occurs.
- Considering different error weights for the case where the failure cost is much greater than the replacement cost, the FFNN and LAD show best performance in the most cases.
- In terms of training run time, the ANNs have the faster time.
- The LAD model has the least variance in results. In both the error and half-life error metrics, the LAD model has the lowest standard deviation.
- The overall statistical testing results show that the PHM is statistically different from FBNN. In two of the metrics, half-life error and cost score, the LAD is statistically different from FBNN. It can be concluded from these results that the FBNN is most statistically different.

4.3 Industrial Application

Given the interesting results obtained on the CMAPSS data, the methodology is applied to another real-world problem. For this endeavor, data is obtained from a structural health monitoring (SHM) project of the Halifax MacKay Bridge (Clarke, 2014).

4.3.1 The MacKay Bridge Problem and Background

Starting in 2015, Halifax has seen Halifax Harbour Bridges (HBB) re-deck the Angus L. Macdonald Bridge (Macdonald Bridge) in a project known as the Big Lift (Halifax Harbour Bridges, n.d.). Soon, the A. Murray MacKay Bridge (MacKay Bridge) will require a similar re-decking. First opened in 1970, the MacKay Bridge (Clarke, 2014; Halifax Harbour Bridges, n.d.) has been servicing all types of vehicles, including those over 3200kg, for the last 45 years. Currently, approximately 65,000 vehicle crossings occur on the MacKay Bridge during the average workday.

In 2010, Dalhousie University (DAL) was contracted by HHB and Buckland & Taylor Ltd. to supply and install a structural health monitoring system on the deck of the MacKay Bridge and evaluate the integrity of specimens extracted from the bridge's deck through fatigue testing. Structural health monitoring (SHM) is a non-destructive process for investigating a structure's health using sensors to provide condition information (Farrar & Worden, 2013). In other words, SHM is a structural equivalent to CBM's CM. Recently, the problem of aging infrastructure has made field applications of SHM systems more important (Clarke, 2014; Farrar & Worden, 2013) because it allows maintenance policies to incorporate real-time condition information.

A bridge's deck is one of four main components in a suspension bridges (Harazaki, Suzuki, & Okukawa, 2000), of which the MacKay Bridge is one. Other key components include the cable system (that support the deck), the pylon (or towers that support the cable system), and the anchorages (that support the cable system). The deck forms the surface that is directly in contact with traffic and directly supports the traffic load that acts on a bridge. This means that fatigue, which can be defined as the development and growth of cracks that are a result of repetitive actions and varying loads (Clarke, 2014), is likely to cause failure at the deck. Therefore, the SHM of the MacKay Bridge deck is extremely important for monitoring its condition.

The MacKay Bridge deck is constructed out of a thin steel plate and stiffening ribs. This type of deck is classified as an orthotropic deck (Mangus & Sun, 2000). Fatigue is more likely to occur in three locations. The first two locations are two of the most common fatigue locations for orthotropic decked bridges (Clarke, 2014). These are known as the rib-to-deck welded joint (Figure 18a) and the rib-to-floor beam joint (Figure 18b). The third location is a result of removing the stiffeners between adjacent ribs (Clarke, 2014), as shown in Figure 18c.

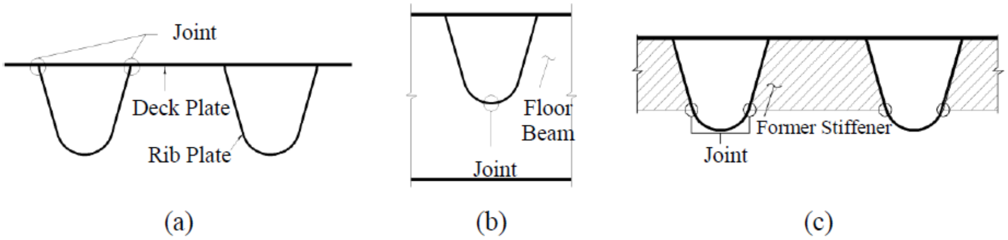

Figure 18: Fatigue Locations for the MacKay Bridge: (a) rib-to-deck joint, (b) rib-to-floor beam joint, (c) former stiffener between adjacent stiffening ribs (Clarke, 2014)

As a result of data availability, this study only considers the first type of failure location. This type of location will be referred to as Type A, which corresponds to the Type A locations in Clarke (2014).

4.3.2 Laboratory vs. Field Data

During the 2010 project, two types of data were collected. First, laboratory data was collected from fatigue testing, then, field data from the SHM system on the MacKay Bridge.

As part of the project, panels from the MacKay Bridge deck were removed and replaced. The 29 specimens, removed in 2010, were brought to the DAL laboratory for fatigue testing (Clarke, 2014). In fatigue testing (Clarke, 2014), a load is placed on the specimen to cause stress and the observed strain value is collected until the specimen fails. A total of 10 Type A specimens were tested in the laboratory. In the initial testing, one specimen was not tested and four specimens did not fail, while the remaining five specimens ran to failure. The four surviving specimens and the untested specimen were retested at a later time. At the end of the testing, all ten specimens had been observed until failure.

The laboratory data is used to train the three models, the LAD, the PHM, and the ANNs. Laboratory data that is received from this project is incomplete (insufficient data for sample A1 and A7). Therefore, only 8 Type A specimens are used in this thesis. Table 17 shows the number of cycles and available data from the laboratory-tested specimens.

Table 17: MacKay Bridge Laboratory Data Summary

ID	Load (kN)	Retest	Reported Cycles	Gathered Cycles	Average Strain	Compressed Cycles (9400)
A1	79		145,761	---	---	---
A5	83		120,677	120,906	1566.19	13
A7	74		146,472	128,512	---	---
A8	77		90,211	90,136	1522.38	10
A6	52 (63)	Y	2,019,221	1,764,352	1052.37 (1081.70)	189
A2	57 (73)	Y	1,815,820	1,815,853	938.75 (1282.11)	194
A3	41		682,160	725,868	865.73	78
A11	45 (55)	Y	3,585,617	2,003,336	1078.26 (1147.84)	214
A21	25 (29)	Y	7,631,929	6,317,027	1071.98 (1108.82)	673
A27	34	Y	1,669,980	733,197	784.57	78

In the second phase of the project (Clarke, 2014), A SHM system was installed on the MacKay Bridge deck to collect real-time data. A section of the bridge was outfitted with 82 strain gauges, 16 of which monitored Type A fatigue locations. Gauges were installed in pairs, with one on each side of the location detail. Type A gauges were installed at the locations shown in Figure 19 (Clarke, 2014).

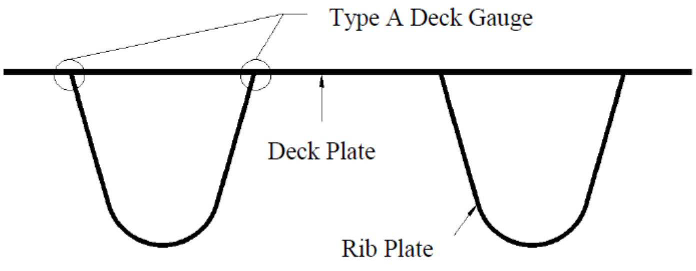

Figure 19: Gauge Positions for Type A Fatigue Locations (Clarke, 2014)

In 2012, several calibration and parameter estimation experiments were completed to set up data collection from the field. In January 2013, the SHM system started data collection for a 12 month period. It is important to note that the field data is collected for the year 2013, three years after the laboratory data is collected in 2010. Therefore, traffic adjustments are made to account for this three year difference before it is used.

Field data is collected using a standard Campbell Scientific Rainflow Counting Program (CampbellScientific, 2012). The program sorts and counts strain values into bins based on its mean and amplitude to produce an output called the Rainflow Histogram. First, strain data is reduced to a series of peaks and valleys. Then, half cycles of strain are defined and sorted based on its mean and amplitude. A histogram is created using the half cycle counts of strain falling into specific mean and amplitude ranges. An example of the Rainflow output of the SHM system is shown in Figure 20. In the figure, the mean (z-axis) and the amplitude (x-axis) defines the different bins in the Rainflow Histogram. As strain values are measured from the traffic, the number of cycles within each bin (y-axis) are counted into the columns, identified by different colors according to the mean bin number.

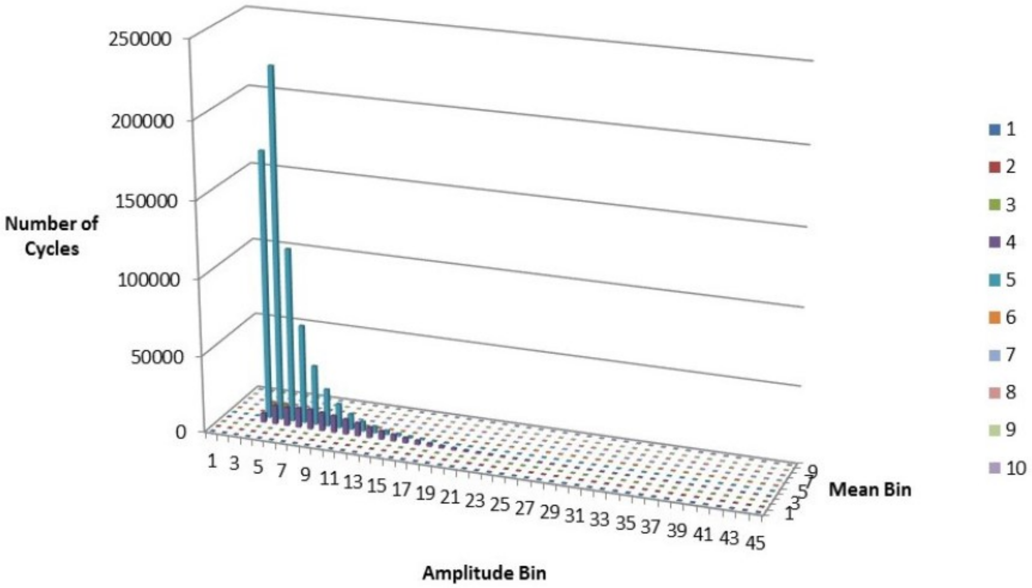

Figure 20: Sample Rainflow Histogram Output for MacKay Bridge Field Data (Clarke, 2014)

Hourly histograms were collected for each Type A channel. All Rainflow histograms have 10 mean bins and 40 amplitude bins for defining the strain cycle value. Data from these histograms, already processed with outlier detection and noise filtering (Clarke, 2014), is made available for analysis. This field data is used for prognostics predictions after the models are trained. Three Type A gauges are excluded from this thesis because of abnormalities in cycle counts and in the data distribution. The number of cycles collected from the field is shown in Table 18.

Table 18: MacKay Bridge Field Data Summary

Channel	Total cycles	Channel	Total cycles
1	833725	9	242833
2	326087	10	68544
3	93337	11	10160
4	1334456	12	1583
5	96952	13	852868
6	40606	14	1700825
7	84867	15	407
8	618611	16	1180844

4.3.3 Experimental Analysis

The differences between the laboratory data formats and the field data formats raises an obstacle with regards to the data inputs. Since, new data values must have the same input as the training dataset in order for the model to apply it for prediction, the data formats had to be the same. The laboratory data is in actual strain values while the field data is in a Rainflow histogram (as explained in Section 4.3.2). Using the same bins that is used to sort the field data, the laboratory data is converted into a Rainflow histogram. The converted data for one specimen is shown in Figure 21. It is seen, however, that the histogram shows counts only in one or two bins. The other laboratory specimens also show similar distributions. In the laboratory data, one or two bins dominate the values of the strain observed. This is notably different from the field data histogram shown in Figure 20, which has counts in a lot more bins and strain variations.

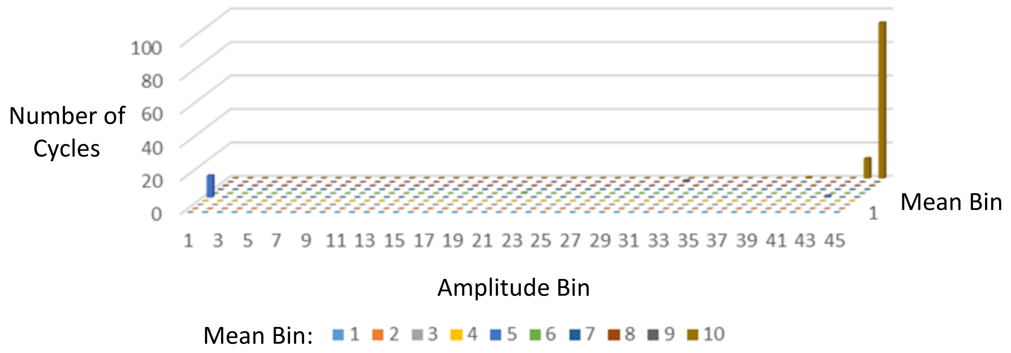

Figure 21: Sample Data of Laboratory Data Rainflow Histogram

These differences are expected because in the field, structures undergo variable amplitude loading rather than the constant amplitude loading that is observed in the laboratory environment. Based on these results, this format conversion method option is disregarded. Instead, another method to summarize the strain, as explained below, is investigated.

Since varying loads can occur in the field, some strain cycles can cause more damage than others. In literature, this difference in damage is commonly accounted for using Palmgren-Miner's rule (Miner's rule) (Miner, 1945; Palmgren, 1924). Miner's rule relates the number of cycles under various stress ranges in a linear function described by Equation 23:

$$D = \sum_{i=0}^{k} \frac{n_i}{N_i} \tag{23}$$

Where N_i is the expected failure age under stress range i, n_i is the number of observed cycles under stress range i, and k is the number of different stress ranges that the equipment has undergone. However, a lack of knowledge on the unknown parameters made this rule hard to apply.

As an alternative, this thesis experiments with several types of strain-summarizing inputs for accumulating damage. These inputs include various combinations of these measures: age in cycles, instantaneous strain (measured), average strain, cumulative strain, maximum strain, minimum strain, standard deviation of the strain, and cumulative instantaneous strain. To determine the combination of metrics that will produce the most accurate residual life predictions overall, an experiment is completed using three standard ANN architectures: the single-layer feedforward ANN (FF), the single-delay feedback ANN (FB1), and the two-delay feedback ANN (FB2). In the experiment, 10 model instances (or trials) are trained for each of the standard ANN architectures. The training, validation, and testing subsets used in training are randomly partitioned. The best overall performance between the different inputs and different architectures is determined by the average mean-squared error (MSE) over the 10 trials.

The results from experimenting with the inputs are displayed in Table 19. The overall best performing input is the cumulative strain input in the single-delay feedback ANN, which obtained a MSE value of 4.85E+08. However, the best average results (average MSE = 1.24E+10) over all three models is the input consisting of the average strain and the age. This combination of inputs is used as the input for the bridge models because it performs well in the most cases.

Table 19: Inputs Experiment Results

Combinations of Inputs	Average Mean Squared Error (over 10 trials)		
	FF	FB1	FB2
Actual and Age	4.67E+10	7.72E+08	8.56E+08
Actual	1.08E+11	6.22E+08	9.10E+08
Average	1.19E+11	8.43E+08	6.15E+08
Average and Age	3.54E+10	6.79E+08	1.03E+09
Max, Min, Std. Deviation, Average, and Age	5.47E+10	**5.95E+08**	1.01E+09
Sum	6.69E+10	1.32E+09	8.71E+08
Sum and Age	9.51E+10	1.01E+09	1.10E+09
Cumulative	1.82E+11	**4.85E+08**	1.42E+09
Cumulative and Age	1.53E+11	7.60E+08	6.24E+08

As previously mentioned, the traffic during the three years between the laboratory and field data collection periods should be considered. To account for past traffic, this work makes two assumptions: (1) the number of cycles collected during 2013 is representative of strain patterns in all previous years, and (2) the traffic increases by 2% each year (Clarke, 2014). Under these two assumptions, the number of traffic cycles observed during 2010-2012 are estimated. For example, if the number of cycles collected in period 1 of 2013 is 100, then the number of cycles for the first period of 2012 is estimated to be 98, which is 2% less than the traffic of 2013. By adding the estimated number of cycles from the previous years to the observed number of traffic cycles on the field, the total number of cumulated traffic cycles can be calculated to account for the time difference between the laboratory and field data collection periods. The field data that underwent this transformation is traffic-adjusted.

The methodology in Chapter 3 is applied to the data with average strain and age inputs that is traffic-adjusted. Laboratory data is used to train the CBM prognostics approaches. Since the residual life of the laboratory data is known, this data is also used to evaluate model performance, with results being compared over the metrics presented for the CMAPSS data. Afterwards, the trained models are used to predict the residual life of the MacKay Bridge based on the field data.

Using the parameters and levels described in Table 4, an 8-fold cross validation structure, and the Matlab script for FFNN design (APPENDIX A), the Taguchi experiment is performed to determine the optimal architecture for the FFNN using the bridge data (APPENDIX D). The Taguchi experiment for determining an optimal FBNN is not completed because the FBNN introduced before requires an initial estimate of the residual life before it can accurately perform residual life predictions. In a realistic application, as in this case, this estimate is hard to obtain. As a result, the FBNN is not tested on this dataset. S/N ratios for the training parameter in the FFNN design are -92.96, -92.01, -96.53, and -94.64 for the first, second, third, and fourth levels, respectively. The first, second, third, and fourth levels of the number of nodes parameters had S/N ratios of -92.91, -93.84, -94.63, and -94.78, respectively. The last parameter, the number of hidden layers, had S/N ratios for the first and second levels of -95.97 and -92.11, respectively. These results are summarized in Table 20.

Table 20: S/N Ratios for Optimal FFNN Architecture Taguchi Experiment with MacKay Bridge Data

Parameter	Level 1	Level 2	Level 3	Level 4
Training	-92.96	-92.01	-96.53	-94.65
Nodes	-92.91	-93.84	-94.63	-94.78
Layers	-95.97	-92.11	N/A	N/A

A comparison of the S/N ratios between parameters shows that the training algorithm parameter has the largest effect on performance just like it is for the CMAPSS dataset. Between parameter levels, the S/N ratios reveal that the optimal FFNN architecture (Figure

22) for the bridge data has 2 hidden layers, 5 nodes in each hidden layer, and is trained with the Levenberg–Marquardt backpropagation algorithm (trainlm, 2016).

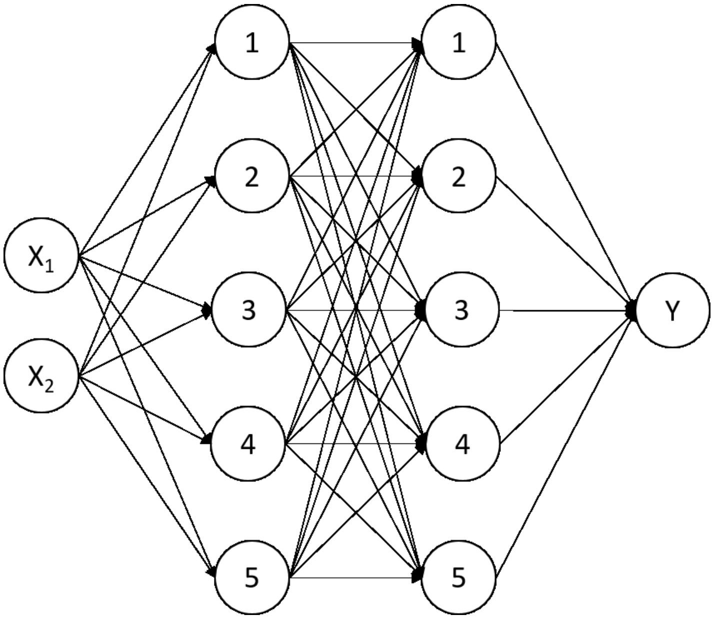

Figure 22: Optimal Feedforward ANN Architecture for Bridge Data

Firstly, the performance of the three models are compared based on error. The statistics calculated to compare the error metric is displayed in Table 21. Like with the CMAPSS data, LAD and PHM have the least variance, with a standard deviation of 49.03 and 50.36, respectively. Unlike the CMAPSS data results, PHM and FFNN have the best performance based on the error metric, with the mean (-25.78 and -1.11, respectively) and median (-16.46 and 3.24, respectively) closest to zero. Between the three models, LAD has the mean and median farthest from zero.

Table 21: Bridge Error Summary

	LAD	PHM	FFNN
Minimum	-4.22	-138.34	-212.30
25% Percentile	43.24	-60.36	-39.93
Median	111.84	-16.46	3.24
75% Percentile	135.54	8.67	34.91
Maximum	180.15	60.41	110.52
Mean	98.35	-25.78	-1.11
Std. Deviation	49.03	50.36	54.01
Range	184.37	198.75	322.82

Comparing the error metric results visually for the bridge data in Figure 23 shows that the LAD does indeed perform the worst, with the median farthest from zero. The FFNN's median is very close to zero while PHM's median leans toward the negative direction. Some outliers can be observed for the FFNN. These outliers are in the negative direction, and, as discussed before, in some cases have less impact than outliers in the positive direction.

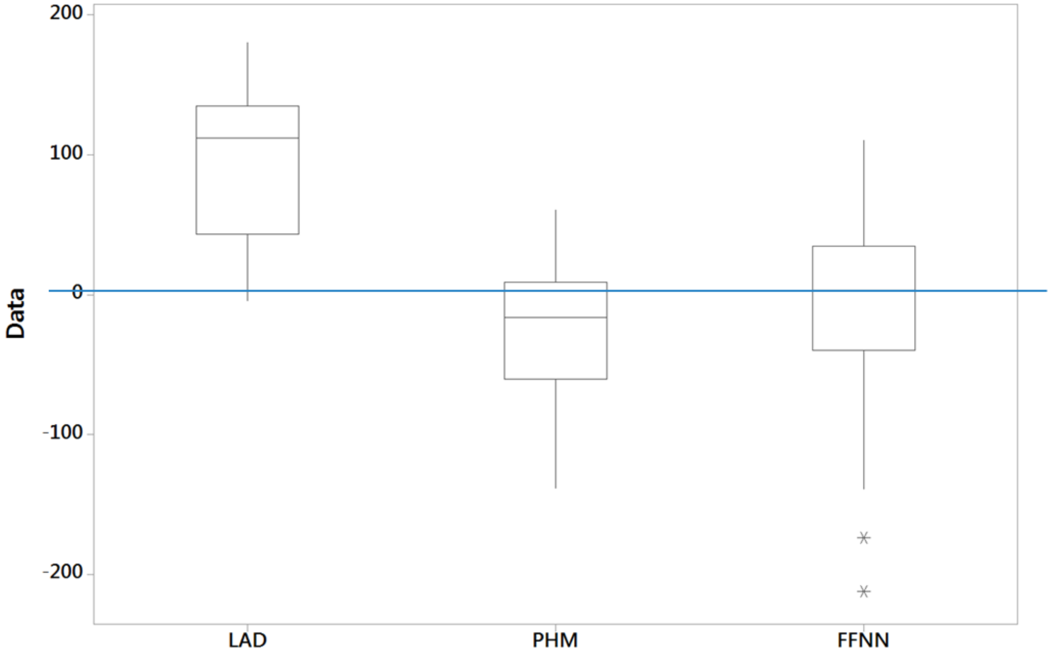

Figure 23: Box Plot Comparison of Bridge Errors between the Models

Statistical testing using the Friedman matched-pair test shows that there is statistical difference between these three models (p-value = 0.007). Further investigations into the difference uses the Wilcoxon signed rank test. Comparing the LAD to PHM and FFNN finds p-values of 0.006 and 0.126, respectively. A comparison of the PHM to FFNN obtains a p-value of 0.76. A review of these p-values concludes that the LAD and PHM are statistically different from each other.

A comparison of the model performance on the half-life error metric reveals statistics that are exactly the same as that of the error metric results discussed previously. In the bridge data, the component with the longest observed lifetime underwent approximately 660 time units, meaning that the half-life for the bridge data is 330 time units. The testing subset, which is used for statistical testing, does not have any bridge components that survive up to the half-life. In other words, the components all fail before 330 time units. As a results, no comparisons can be made based on the half-life error for the bridge data.

The cost score metric is not considered because the magnitude of the cost for the bridge is very big. As a result, the differences between each model's cost score may not be truly indicative of the consequences of predicting failure before or after it actually fails. In addition, the failure costs associated with the failure of the bridge will logically dwarf the costs of replacement.

4.3.4 Discussion and Comparison of Results

After training, the models instances are used in the prediction of MRL for actual working components represented by the field data. The best performing models instances for each approach predicts the MRLs shown in Table 22 for each of the 13 components monitored on the bridge.

Table 22: Residual Life Predictions of Bridge Field Components

	Months (9400 Cycles)			Years		
Component	LAD	PHM	FFNN	LAD	PHM	FFNN
C1	259.37	652.57	-69.82	21.61	54.38	-5.82
C2	176.71	603.18	13.66	14.73	50.27	1.14
C3	172	563.11	50.49	14.33	46.93	4.21
C4	104.12	684.58	-56.78	8.68	57.05	-4.73
C5	171	564.28	49.6	14.25	47.02	4.13
C6	183	547.76	54.88	15.25	45.65	4.57
C7	173.5	561.31	50.98	14.46	46.78	4.25
C8	265.1	634.81	-51	22.09	52.9	-4.25
C9	193.71	591.53	28.36	16.14	49.29	2.36
C10	177	556.8	55.13	14.75	46.4	4.59
C11	176.61	535.1	60.14	14.72	44.59	5.01
C14	6.5	702.94	-52.85	0.54	58.58	-4.4
C16	151.37	675.85	-58.41	12.61	56.32	-4.87
Average	170.00	605.68	5.72	14.17	50.47	0.48

These results in Table 22 reveal that the practical application of the LAD and ANN models seem to be lacking. For the FFNN, some components are projected to have failed approximately four years ago at around 2006. The PHM shows the steadiest predictions.

This thesis compares these predictions to the results of the original Civil Engineering project (Clarke, 2014). In the original project with HBB, a novel approach for assessing the MacKay Bridge fatigue reliability is presented by combining laboratory tested stress-life (S-N) curves, SHM, finite element modeling, and limit state equations (Clarke, 2014). S-N curves (CAN/CSA, 2006), which relate a certain stress range to the number of cycles in a failure curve, are often used in industry to predict the number of cycles until failure by treating all strains as elastic. Limit state equations mathematically represent the threshold of predefine limits of satisfactory behaviour in structures. Prognostics results are presented in terms of a reliability index, for the years 2012 to 2030. According to various evaluation standards, the failure threshold (Clarke, 2014) for this reliability index ranges from 0.5 (Bowman, 2012) to 1.0 (AASHTO, 2011). If the reliability index falls under this threshold, the component or structure is considered to have failed because it has passed the acceptable performance limits. After applying this threshold, it is found that Type A components should not fail until after 2030 (20 years later). A sample of how prognostics results compare between the three models studied in this thesis and the original project is shown in Figure 24. Comparisons of the remaining components are included in APPENDIX E. The component, whose results are displayed in the Figure 24, is the component with the reliability index closest to the failure threshold. In this example, the FFNN has the worst prediction and predicts failure occurring at around 2006. LAD predicts failure between 2010 and 2011 and PHM predicts failure occurring near 2060.

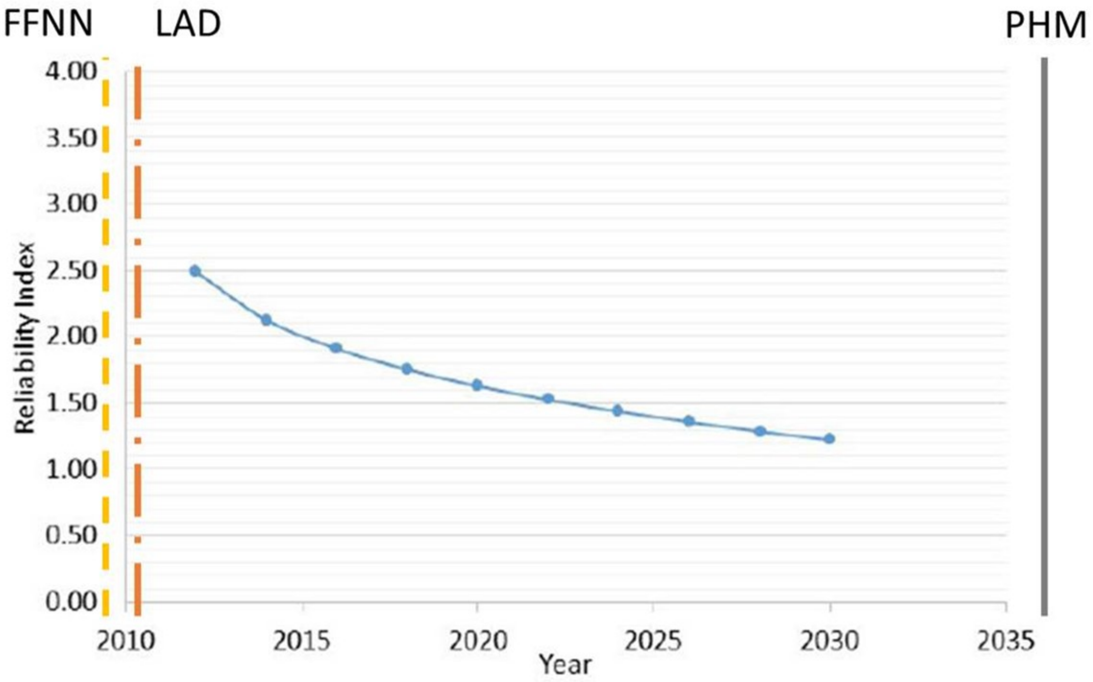

Figure 24: Comparison of Prognostics Results for Component C14 (modified from Clarke, 2014)

There are several possible reasons for this large difference in results. First, the assumptions made for considering the time difference between the laboratory and field data may not hold true. In this case, the estimated aging during the three years can cause the age parameter to be misrepresented for the field components. Second, the training set may not be representative of the population's behaviour. If the training dataset is not representative of the population, the model may actually do better or worse depending on the proportion of new and old behaviours in the new dataset. The second reason is more likely because of the large differences between the strains variations observed in the field data and in the laboratory (Figure 20 vs. Figure 21). The LAD and FFNN would, theoretically, be the most highly affected because these approaches depend on learning about historical data patterns for prediction. If certain behaviours have not been previously observed, these approaches will find it hard to make accurate predictions of that behaviour.

These results mean that the type of training data also plays a large role for the maintenance decision maker. If the training data is not representative of the population's behaviour, PHM is the preferred model, because its statistical dependencies and age-based hazard rate will make up for deficiencies in the CM data.

To summarize the prediction results, the error metric results, and the field-predicted MRLs, the following points can be made for the bridge data results:

- Like the CMAPSS results, the LAD model has the lowest variance.
- In terms of statistical difference, there is only one instance where the p-value is unacceptable. This instance, calculated based on the error metric, showed that the LAD and PHM approaches are different.
- In terms of practical application, the predictions for the field data residual life shows a lot of variability between models Compared to the original project (Clarke, 2014), the FFNN and LAD predicts failure occurring much earlier but the PHM predicts failure occurring later.

CHAPTER 5 CONCLUSIONS

In this thesis, the application of two artificial intelligence (AI) approaches and one statistical approach to condition-based maintenance (CBM) prognostics is evaluated. Logical analysis of data (LAD), artificial neural networks (ANN), and the proportional hazards model (PHM) are applied to the Turbofan Engine Degradation Simulation Data Set (CMAPSS) and to the MacKay Bridge dataset.

The following conclusions can be made from the application of the methodology, developed in this thesis:

- Overall the models predict failure after it has already occurred (negative error) when the residual life is large and predict failure before it has already occurred (positive error) when the residual life is small. In addition, the model errors get more accurate as failure is approached.
- The LAD approach shows comparable performance to the PHM and has the least variance. This behaviour is interesting because adjustments to the error can be completed to improve prediction accuracy more easily than any other model.
- The PHM, as it compares to the other three models studied in this thesis, has the most stable behaviour. In most cases, its variance and performance is very good. In addition, in cases where there is not enough representative training data, PHM still performs well.
- The FFNN also shows performance that is comparable to the PHM. In addition, it shows slight better performance, as compared to the other three approaches, when the residual life decreases and failure approaches. However, this approach has the worst performance when the training dataset is not representative of the population.
- The FBNN has the most variability and is statistically different from the other models. There is also a difficulty in applying it practically because of the need to have an initial estimate of the residual life. In real-world problems, the residual life is often hard to estimate.

Using these conclusions, the maintenance decision maker can determine a good prognostics model for CBM. Results from this thesis can be condensed into three main considerations: accuracy, run time, and data type. When accuracy is a main concern, as in the case where impacts of failure are large, LAD and FFNN are preferred models. The preferred model changes when run time is considered. If data can be easily collected and updating the model is performed often, the ANNs and LAD is preferred. On the other hand, if CM data is not easily collectable and existing data is not representative of the population's behaviour, data type comes into play. In this case, PHM is preferred.

Based on these observations, two possible research directions are recommended for improving the accuracy of the residual life predictions. The first suggested research direction is to investigate hybrid artificial intelligence models. One proposal in this direction is a LAD-ANN hybrid, in which the LAD method is used to make an initial prediction and then the ANN is used to adjust the prediction. The second research direction works with adjusting the residuals after the initial model prediction. In this case, special interest is placed on working with the LAD's residuals because the model exhibits the least variance between the models under study. This characteristic, as mentioned previously, makes the residuals more easily adjustable.

In addition, this thesis recommends that testing should be performed to confirm the results observed from applying the methodology to the MacKay Bridge dataset. This may be possible by compiling a non-presentative dataset from the CMAPSS dataset and evaluating model performance on the entire CMAPSS dataset. In other words, the question of how the models will perform when the training dataset is not representative of the population's behaviour should be answered.

REFERENCES

[1] AASHTO. (2011). *Manual for Bridge Evaluation* (2nd ed.). Washington, DC, USA: American Association of State Highway and Transportation Officials.

[2] Abramson, S. D., Alexe, G., Hammer, P. L., & Kohn, J. (2005). A computational approach to predicting cell growth on polymeric biomaterials. *Journal of Biomedical Materials Research Part A, 73A*(1), 116-124. doi:10.1002/jbm.a.30266

[3] Abu-Elanien, A. B., & Salama, M. A. (2007). Survey on the transformer condition monitoring. *IEEE Power Engineering 2007 Large Engineering Systems Conference*, 187-191. doi:http://dx.doi.org/10.1109/LESCPE.2007.4437376

[4] Alexe, G., Alexe, S., Liotta, L. A., Petricoin, E., Reiss, M., & Hammer, P. L. (2004). Ovarian cancer detection by logical analysis of proteomic data. *Proteomics, 4*(3), 766-783. doi:10.1002/pmic.200300574

[5] Al-Saba, T., & El-Amin, I. (1999). Artificial neural networks as applied to long-term demand forecasting. *Artificial Intelligence in Engineering, 13*(2), 189-197. doi:http://dx.doi.org/10.1016/S0954-1810(98)00018-1

[6] Azimi, P., Mohammadi, H. R., Benzel, E. C., Shahzadi, S., Azhari, S., & Montazeri, A. (2014). Artificial neural networks in neurosurgery. *Neurosurgery & Psychiatry Journal of Neurology*, jnnp-2014.

[7] Banjevic, D., Jardine, A. S., Ennis, M., & Makis, V. (2001). A control-limit policy and software for condition based maintenance optimization. *INFOR, 39*(1), 32-50.

[8] Baruah, P., & Chinnam, R. B. (2005). HMMs for diagnostics and prognostics in machining processes. *International Journal of Production Research, 43*(6), 1275-1293. doi:10.1080/00207540412331327727

[9] Boros, E., Hammer, P. L., Ibaraki, T., Kogan, A., Mayoraz, E., & Muchnik, I. (2000). An implementation of logical analysis of data. *IEEE Transactions on Knowledge and Data Engineering, 12*(2), 292-306.

[10]Bowman, M. D. (2012). Fatigue evaluation of steel bridges. *Transportation Research Board, 721*.

[11]Bunks, C., McCarthy, D., & Al-Ani, T. (2000). Condition-based maintenance of machines using hidden markov models. *Mechanical Systems and Signal Processing, 14*(4), 597-612. doi:http://dx.doi.org/10.1006/mssp.2000.1309

[12]CampbellScientific. (2012). *CR9000X Measurement and Control System: Instruction Manual*. Retrieved May 2016, from http://www.campbellsci.ca/manuals.

[13]CAN/CSA, S.-0. (2006). *Canadian Highway Bridge Design Code Commentary.* Mississauga, ON, CAN.: Canadian Standards Association.

[14] Carbonneau, R., Laframboise, K., & Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research, 184*(3), 1140-1154. doi:http://dx.doi.org/10.1016/j.ejor.2006.12.004

[15] Carpenter, G. A., & Grossberg, S. (1987). ART 2: Self-organization of stable category recognition codes for analog input patterns. *Applied optics, 26*(23), 4919-4930.

[16] Cimpoiu, C., Cristea, V. M., Hosu, A., Sandru, M., & Seserman, L. (2011). Antioxidant activity prediction and classification of some teas using artificial neural networks. *Food chemistry, 127*(3), 1323-1328. doi:http://dx.doi.org/10.1016/j.foodchem.2011.01.091

[17] Clarke, J. N. (2014). *Investigating the remaining fatigue reliability of an aging orthotropic steel plate deck.*

[18] Cox, D. R. (1972). Regression Models and Life-Tables. *Journal of the Royal Statistical Society Series B (methodological), 34*(2), 187–220.

[19] Crama, Y., Hammer, P. L., & Ibaraki, T. (1988). Cause-effect relationships and partially defined Boolean functions. *Annals of Operations Research, 16*(1), 299-325.

[20] Danish, M. Y., & Aslam, M. (2012). Bayesian estimation in the proportional hazards model of random censorship under asymmetric loss functions. *Data Science Journal, 11*(0), 72-88.

[21] Ding, F., & He, Z. (2011). Cutting tool wear monitoring for reliability analysis using proportional hazards model. *The International Journal of Advanced Manufacturing Technology, 57*(5-8), 565-574.

[22] Drew, P. J., & Monson, J. R. (2000). Artificial neural networks. *Surgery, 127*(1), 3-11. doi:http://dx.doi.org/10.1067/msy.2000.102173

[23] Ebeling, C. E. (1997). *An introduction to reliability and maintainability engineering* (2nd ed.). Long Grove, IL: Waveland Press, Inc.

[24] Esmaeili, S. (2012). *Development of equipment failure prognostic model based on logical analysis of data (lad).*

[25] Farrar, C. R., & Worden, K. (2013). *Structural health monitoring: a machine learning perspective.* Chichester: John Wiley & Sons.

[26] Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics, 11*(1), 86-92.

[27] Ghasemi, A., Yacout, S., & Ouali, M. S. (2010). Parameter estimation methods for condition-based maintenance with indirect observations. *IEEE Transactions on Reliability, 59*(2), 426-439. doi:10.1109/TR.2010.2048736

[28] Gubskaya, A. V., Bonates, T. O., Kholodovych, V., Hammer, P., Welsh, W. J., & Langer, R. K. (2011). Logical Analysis of Data in Structure-Activity Investigation of Polymeric Gene Delivery. *Macromolecular theory and simulations, 20*(4), 275-285. doi:10.1002/mats.201000087

[29]Halifax Harbour Bridges. (n.d.). *About the big lift*. Retrieved April 7, 2016 , from https://www.hdbc.ca/

[30]Harazaki, I., Suzuki, S., & Okukawa, A. (2000). Suspension Bridges. In W.-F. Chen, & L. Duan (Eds.), *Bridge Engineering Handbook.* Boca Raton, FL, USA: CRC Press.

[31]Hebb, D. O. (1949). *The organization of behavior: A neuropsychological approach.* John Wiley & Sons.

[32]Huang, Y. C., & Huang, C. M. (2002). Evolving wavelet networks for power transformer condition monitoring. *IEEE Transactions on Power Delivery, 17*(2), 412-416. doi:http://dx.doi.org/10.1109/61.997908

[33]Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer, 29*(3), 31-44. doi:10.1109/2.485891

[34]Jardine, A. K. (2002). Optimizing condition based maintenance decisions. *IEEE Annual Proceedings in reliability and maintainability symposium 2002*, 90-97. doi:10.1109/RAMS.2002.981625

[35]Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing, 20*(7), 1483-1510.

[36]Johnson, D. E. (1998). *Applied multivariate methods for data analysts.* Pacific Grove, California: Duxbury Press.

[37]Kaplan, E. L., & Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American statistical association, 53*(282), 457-481.

[38]Kim, D. (1999). Normalization methods for input and output vectors in backpropagation neural networks. *International journal of computer mathematics, 71*(2), 161-171.

[39]Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE, 78*(9), 1464-1480.

[40]Lauer, M. S., Alexe, S., Snader, C. E., Blackstone, E. H., Ishwaran, H., & Hammer, P. L. (2002). Use of the logical analysis of data method for assessing long-term mortality risk after exercise electrocardiography. *Circulation, 106*(6), 685-690.

[41]Lee, J., Abujamra, R., Jardine, A. K., Lin, D., & Banjevic, D. (2004). An integrated platform for diagnostics, prognostics and maintenance optimization. *Proceedings of the Intelligent Maintenance Systems*, 15-27.

[42]Li, L., Tse, P. W., Tu, P., & Yam, R. C. (2001). Intelligent predictive decision support system for condition-based maintenance. *The International Journal of Advanced Manufacturing Technology, 17*(5), 383-391. doi:http://dx.doi.org/10.1007/s001700170173

[43]Lin, D., Banjevic, D., & Jardine, A. K. (2006). Using principal components in a proportional hazards model with applications in condition-based maintenance. *Journal of the Operational Research Society, 57*(8), 910-919.

[44]Liu, L., Ma, Y., Yu, M., & Tu, Y. (2014). Economic and economic-statistic designs of an X̄ control chart for two-unit series systems with condition-based maintenance. *Quality control and applied statistics, 59*(1), 29-32. doi:http://dx.doi.org/10.1016/j.ejor.2

[45]Lorna Wong, E., Jefferis, T., & Montgomery, N. (2010). Proportional hazards modeling of engine failures in military vehicles. *Journal of Quality in Maintenance Engineering, 16*(2), 144-155.

[46]Ma, J., & Li, C. (1995). Detection of localised defects in rolling element bearings via composite hypothesis testing. *Mechanical Systems and Signal Processing, 9*(1), 63-75.

[47]Mangus, A., & Sun, S. (2000). Orthotropic Deck Bridges. In W.-F. Chen, & L. Duan (Eds.), *Bridge Engineering Handbook.* Boca Raton, FL, USA: CRC Press.

[48]McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics, 5*(4), 115-133.

[49]Miladinovic, B., Kumar, A., Mhaskar, R., Kim, S., Schonwetter, R., & Djulbegovic, B. (2012). A flexible alternative to the Cox proportional hazards model for assessing the prognostic accuracy of hospice patient survival. *PloS one, 7*(10), e47804.

[50]Miner, M. A. (1945). Cumulative damage in fatigue. *Journal of applied mechanics, 12*(3), 159-164.

[51]Mortada, M. A. (2010). Applicability and interpretability of logical analysis of data in condition based maintenance.

[52]Mortada, M. A., Carroll III, T., Yacout, S., & Lakis, A. (2012). Rogue components: their effect and control using logical analysis of data. *Journal of Intelligent Manufacturing, 23*(2), 289-302.

[53]Mortada, M. A., Yacout, S., & Lakis, A. (2011). Diagnosis of rotor bearings using logical analysis of data. *Journal of Quality in Maintenance Engineering, 17*(4), 371-397. doi:http://dx.doi.org/10.1108/13552511111180186

[54]Mortada, M. A., Yacout, S., & Lakis, A. (2014). Fault diagnosis in power transformers using multi-class logical analysis of data. *Journal of Intelligent Manufacturing, 25*(6), 1429-1439.

[55]O'Rourke, N., Hatcher, L., & Stepanski, E. J. (2005). *A step-by-step approach to using SAS for univariate & multivariate statistics.* New York: SAS Institute.

[56]Palmgren, A. (1924). Durability of ball bearings. *ZVDI, 68*(14), 339-341.

[57]Peace, G. S. (1993). *Taguchi Methods: A Hands-on Approach to Quality Engineering.* Reading, Mass: Addison-Wesley.

[58]Pearson, K. (1901). On Lines and Planes of Closest Fit to System of Points in Space. *Philiosophical Magazine, 2*, 559-572.

[59]Ryoo, H. S., & Jang, I. Y. (2009). Milp approach to pattern generation in logical analysis of data. *Discrete Applied Mathematics, 157*(4), 749-761. doi:http://dx.doi.org/10.1016/j.dam.2008.07.005

[60]Sankaran, P. G., & Sreeja, V. N. (2012). A Proportional Hazards Model for Successive Duration Times Under Informative Censoring. *Communications in Statistics-Theory and Methods, 41*(2), 262-280. doi:10.1080/03610926.2010.521286

[61]Saxena, A., & Goebel, K. (2008). Turbofan Engine Degradation Simulation Data Set. NASA Ames Prognostics Data Repository. Retrieved from http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/

[62]Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. *Prognostics and Health Management 2008 IEEE International Conference on PHM 2008*, 1-9.

[63]Schneider, J. (1997). *Cross validation*. Retrieved April 7, 2016, from https://www.cs.cmu.edu/~schneide/tut5/node42.html

[64]Sibi, P., Jones, S., & Siddarth, P. (2013). Analysis of different activation functions using back propagation neural networks. *Journal of Theoretical and Applied Information Technology, 47*(3), 1264-1268.

[65]Silva, R. G., Reuben, R. L., Baker, K. J., & Wilcox, S. J. (1998). Tool wear monitoring of turning operations by neural network and expert system classification of a feature set generated from multiple sensors. *Mechanical Systems and Signal Processing, 12*(2), 319-332. doi:http://dx.doi.org/10.1006/mssp.1997.0123

[66]Stanek, M., Morari, M., & Frohlich, K. (2001). Model-aided diagnosis: an inexpensive combination of model-based and case-based condition assessment. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews, 31*(2), 137-145. doi:http://dx.doi.org/10.1109/5326.941838

[67]Taguchi, G., & Konishi, S. (1987). Orthogonal arrays and linear graphs: tools for quality engineering. *American Supplier Institute*.

[68]Tallam, R. M., Lee, S. B., Stone, G. C., Kliman, G. B., Yoo, J., Habetler, T. G., & Harley, R. G. (2007). A survey of methods for detection of stator-related faults in induction machines. *IEEE Transactions on Industry Applications, 43*(4), 920-933. doi:http://dx.doi.org/10.1109/TIA.2007.900448

[69]Tian, Z., & Liao, H. (2011). Condition based maintenance optimization for multi-component systems using proportional hazards model. *Reliability Engineering & System Safety, 96*(5), 581-589. doi:http://dx.doi.org/10.1016/j.ress.2010.12.023.

[70]*trainlm*. (2016). Retrieved May 2016, from MathWorks Documentation: http://www.mathworks.com/help/nnet/ref/trainlm.html

[71]Tsang, A. H. (1995). Condition-based maintenance: Tools and decision making. *Journal of Quality in Maintenance Engineering, 1*(3), 3-17.

[72]Wang, C. C., & Too, G. P. (2002). Rotating machine fault detection based on HOS and artificial neural networks. *Journal of intelligent manufacturing, 13*(4), 283-293.

[73]Werbos, P. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences.

[74]Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics bulletin, 1*(6), 80-83.

[75]Xu, Y., & Ge, M. (2004). Hidden Markov model-based process monitoring system. *Journal of Intelligent Manufacturing, 15*(3), 337-350.

[76]Yacout, S. (2010). Fault detection and diagnosis for condition based maintenance using the logical analysis of data. *IEEE Computers and Industrial Engineering (CIE) 2010 40th International Conference*, 1-6.

[77]Zavadilová, L., Němcová, E., & Štípková, M. (2011). Effect of type traits on functional longevity of Czech Holstein cows estimated from a Cox proportional hazards model. *Journal of dairy science, 94*(8), 4090-4099. doi:10.3168/jds.2010-3684

APPENDIX A   Neural Network Matlab Design Script

A.1   Feedfoward ANN Matlab Design Script

```
function [t, p1, p2, p3, p4] = kfoldnnet2(fold, sets, layers, nodes,
training, Indicators, Survival)

tic

k = fold;

trainP = zeros([1, k]);
validP = zeros([1, k]);
testP = zeros([1, k]);
allR = zeros([1, k]);

% define architecture of neural network
net = fitnet(nodes, training);

net.numLayers = layers;

for i = 1:layers
    for j = 1:layers

        if i > 1
            net.biasConnect(i) = 1;
            net.layerConnect(i, i-1) = 1;
        else
            net.biasConnect(i) = 1;
            net.layerConnect(i, j) = 0;
        end
    end

    if i == layers
        net.outputConnect(i) = 1;
        net.layers{i}.transferFcn = 'purelin';
    else
        net.outputConnect(i) = 0;
        net.layers{i}.size = nodes;
        net.layers{i}.transferFcn = 'tansig';
    end

end

net = configure(net, Indicators, Survival);

for a = 1:k
% k-fold cross validation

    % create training, validating, testing indices list
    if a == 1
```

```
        trInd = find(sets <= k-2).';
        valid = find(sets == k-1).';
        test = find(sets == k).';
    elseif a == 2
        trInd = find(sets <= k-1 & sets >= a).';
        valid = find(sets == k).';
        test = find(sets == 1).';
    elseif a ==3
        trInd = find(sets <= k & sets >= a).';
        valid = find(sets == 1).';
        test = find(sets == 2).';
    else
        trInd1 = find(sets <= k & sets >= a).';
        trInd2 = find(sets <= a-3 & sets >= 1).';
        trInd = horzcat(trInd1, trInd2);
        valid = find(sets == a-2).';
        test = find(sets == a-1).';
    end

    % define network parameters
    net.divideFcn = 'divideind';
    net.divideParam.trainInd = trInd;
    net.divideParam.valInd = valid;
    net.divideParam.testInd = test;

    net = init(net);

    % run network
    [net, tr] = train(net, Indicators, Survival);
    [r] = regression(Survival, net(Indicators));


    % save performance values for run
    trainP(a) = tr.best_perf;
    validP(a) = tr.best_vperf;
    testP(a) = tr.best_tperf;
    allR(a) = r;

end

p1 = trainP;
p2 = validP;
p3 = testP;
p4 = allR;
t = toc;
```

A.2 Feedback ANN Matlab Design Script

```
function [t, p1, p2, p3, p4] = kfoldnnet5(fold, sets, delay, nodes,
training, Indicators, Survival)

tic
```

```
k = fold;

trainP = zeros([1, k]);
validP = zeros([1, k]);
testP = zeros([1, k]);
allR = zeros([1, k]);

% define architecture of neural network
net = narxnet(1:delay, 1:delay, nodes);

% prepare time series data
[Xs, Xi, Ai, Ts] = preparets(net, Indicators, {}, Survival);

net = configure(net, Xs, Ts);

for a = 1:k
% k-fold cross validation

    % create training, validating, testing indices list
    if a == 1
        trInd = find(sets <= k-2).';
        valid = find(sets == k-1).';
        test = find(sets == k).';
    elseif a == 2
        trInd = find(sets <= k-1 & sets >= a).';
        valid = find(sets == k).';
        test = find(sets == 1).';
    elseif a ==3
        trInd = find(sets <= k & sets >= a).';
        valid = find(sets == 1).';
        test = find(sets == 2).';
    else
        trInd1 = find(sets <= k & sets >= a).';
        trInd2 = find(sets <= a-3 & sets >= 1).';
        trInd = horzcat(trInd1, trInd2);
        valid = find(sets == a-2).';
        test = find(sets == a-1).';
    end

    for b = 1:delay
        trInd = trInd(trInd~=b);
        valid = valid(valid~=b);
        test = test(test~=b);
    end

    trInd = trInd - delay;
    valid = valid - delay;
    test = test - delay;

    % define network parameters
    net.divideFcn = 'divideind';
    net.divideParam.trainInd = trInd;
    net.divideParam.valInd = valid;
    net.divideParam.testInd = test;
    net.trainFcn = training;

    net = init(net);

    % run network
    [net, tr] = train(net, Xs, Ts, Xi, Ai);
    [r] = regression(Ts, net(Xs, Xi, Ai));

    % save performance values for run
    trainP(a) = tr.best_perf;
    validP(a) = tr.best_vperf;
    testP(a) = tr.best_tperf;
    allR(a) = r;

end

p1 = trainP;
p2 = validP;
p3 = testP;
p4 = allR;
t = toc;
```

APPENDIX B Matlab Training Algorithms

Four training algorithms are tested for the ANN models. In Matlab code these are: trainlm (Levenberg-Marquardt), trainbfg (BFGS Quasi-Newton), traingdx (Variable Learning Rate Backpropagation), and trainrp (Resilient Backpropagation). This appendix briefly describes these algorithms.

The Levenberg-Marquardt training algorithm is a supervised backpropagation algorithm that Matlab recommends as a first choice supervised training algorithm. To speed up its learning rate, this method is designed to approximate the Hessian matrix by interpolating between the Gradient Descent Method and Newton's Method.

The BFGS Quasi-Newton training algorithm is a supervised training algorithm that can be used as an alternative to Newton's method. Otherwise known as the Broyden-Fletcher-Goldfarb-Shanno Quasi-Newton algorithm, it is a good substitute for Newton's method when the cost of calculating the Jacobian or Hessian matrix is too expensive.

The Variable Learning Rate Backpropagation training algorithm, is a backpropagation training algorithm which uses the gradient descent momentum and an adaptive learning rate methods for updating the network's weights. The learning rate in the gradient descent with momentum equation is adjusted according to the improvements made in the weights towards the goal.

The Resilient Backpropagation training algorithm is suitable for networks which use the sigmoid activation functions and a steepest descent algorithm. In some cases, problems may arise during training because the gradient will be very small and cause small changes in the weights, although the weights are far from optimal. This algorithm tries to eliminate these effects caused by the magnitudes of the partial derivatives by basing the weight changes not only on the magnitude but also the sign of the partial derivatives.

APPENDIX C Model Results

The following electronic supplement is available at Dalspace.

C.1 Prognostic Results

Excel files containing the model outputs and metric calculations are included in this appendix for each dataset (CMAPSS and Bridge). In each file, data are divided into worksheets for each of the four modeling approaches: Logical Analysis of Data (LAD), Proportional Hazards Method (PHM), Feedfoward Neural Network (FFNN), and Feedback Neural Network (FBNN). In each worksheet, model outputs and metric calculations are included for each of the 10 training trials and for the average case.

In the case of the Bridge dataset results, the FBNN is not tested and there are no results. In addition, additional worksheets are provided with the field prediction results.

C.2 Metric Performance Results

Excel files summarizing the performance of the models on each metric are provided in this appendix. Worksheets in each file are categorized by the dataset and includes the summary and boxplots for that specific metric. Outliers are also identified.

C.3 Minitab File for Statistical Testing Results

A Minitab file is provided with the full Friedman matched-pair test and the Wilcoxon signed rank test data and results. Each worksheet in the file is for a different dataset.

APPENDIX D Neural Network Design Taguchi Results

The following electronic supplement is available at Dalspace.

Excel files summarizing the Taguchi design, Taguchi experimental model outputs, and signal-to-noise (S/N) ratios are provided in appendix. Files are differentiated by the type of neural network and the dataset. Four worksheets are available in each file, showing the different data results. The "Taguchi Design" worksheet presents the Taguchi orthogonal matrix that is used for the design experiment. The "MSE Results" worksheet has the model output results that are used to calculate the S/N ratios used to determine the optimal network design. "Other Results" provide other model outputs that can be used for further analysis of the model performance. Finally, the "Taguchi Analysis" worksheet presents the calculated S/N ratio that is used to determine the optimal levels in the network design parameters.

APPENDIX E Bridge Field Results Comparison Graphs

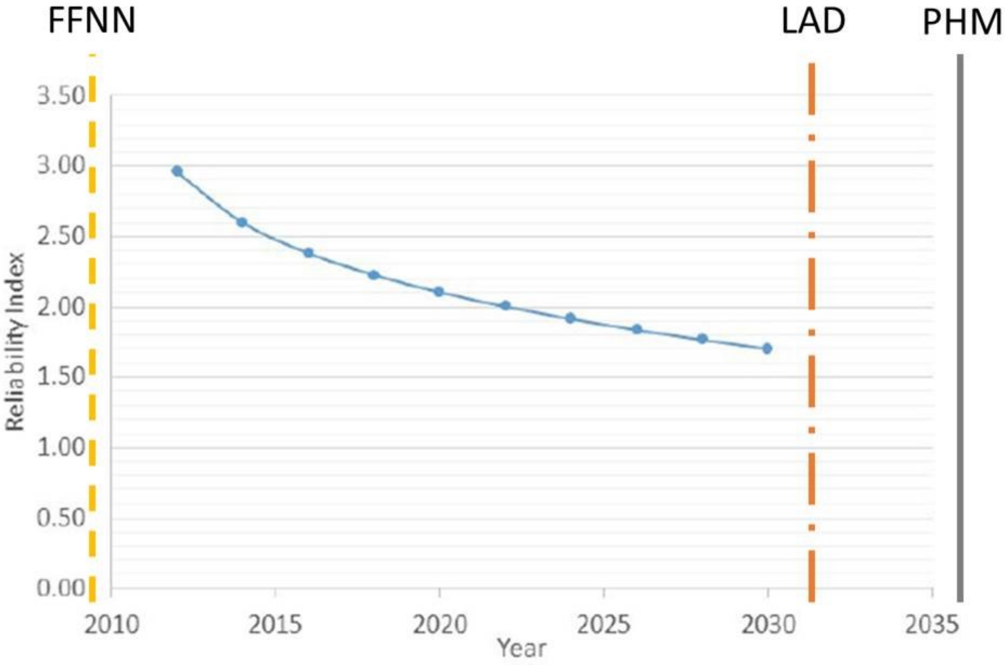

Figure E-1: Comparison of Prognostics Results for Component C1 (modified from Clarke, 2014)

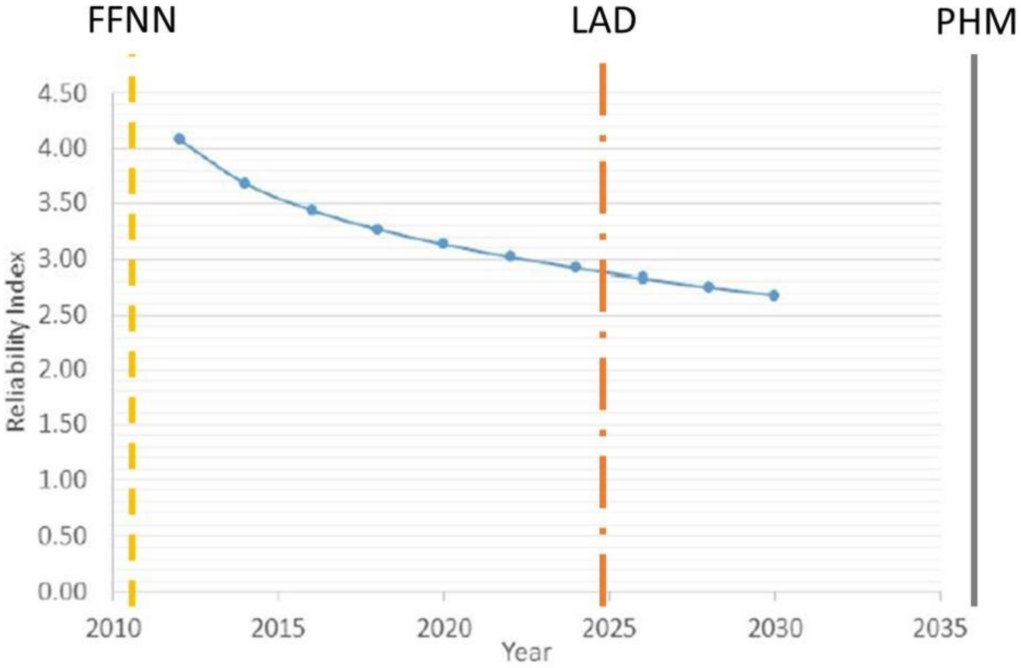

Figure E-2: Comparison of Prognostics Results for Component C2 (modified from Clarke, 2014)

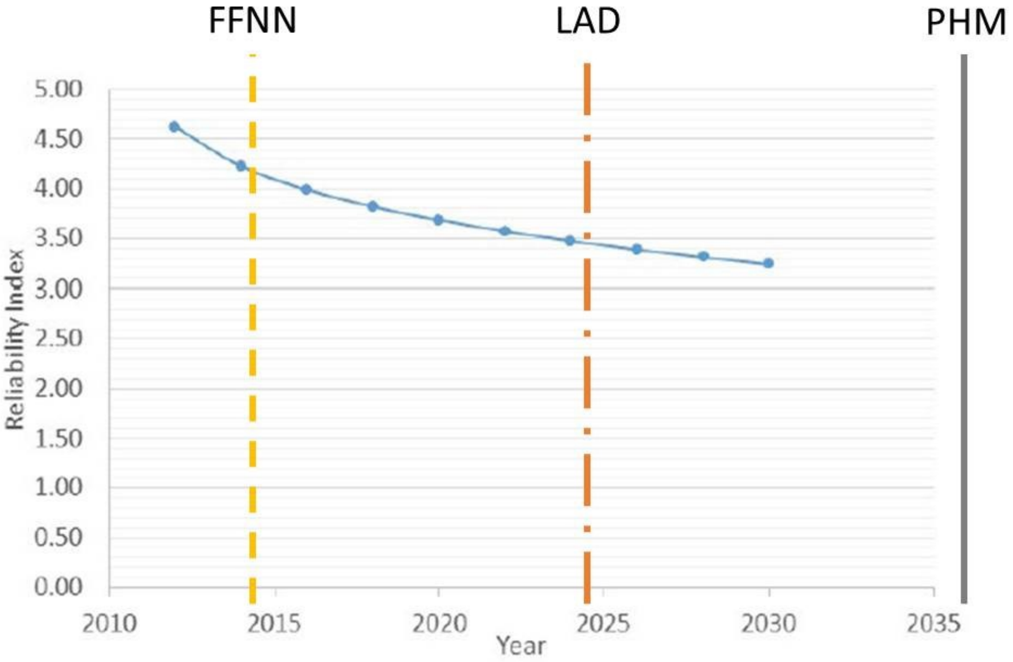

Figure E-3: Comparison of Prognostics Results for Component C3 (modified from Clarke, 2014)

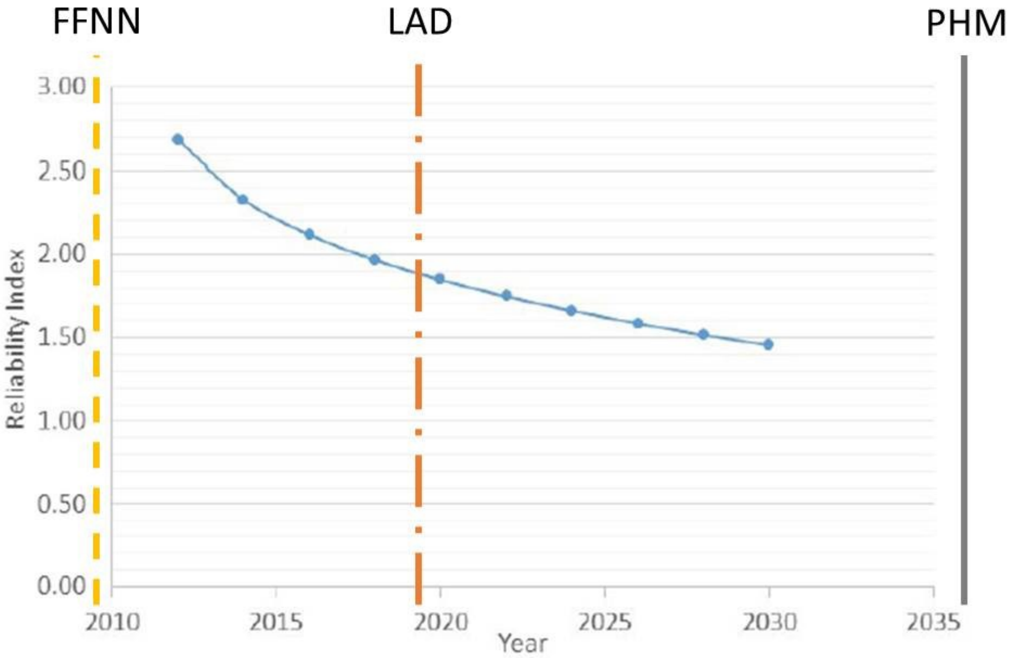

Figure E-4: Comparison of Prognostics Results for Component C4 (modified from Clarke, 2014)

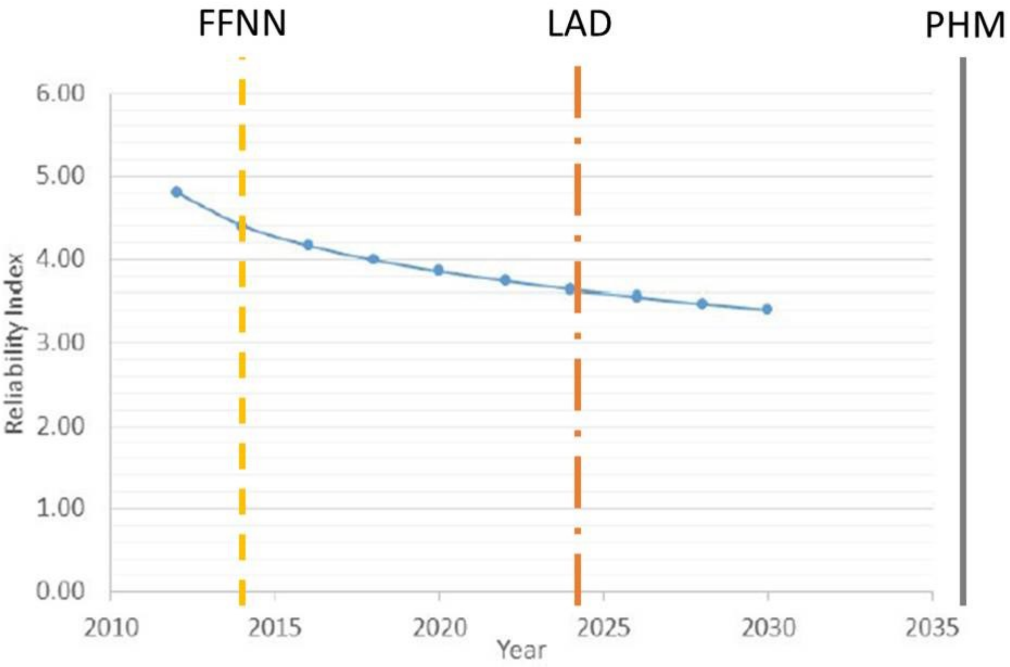

Figure E-5: Comparison of Prognostics Results for Component C5 (modified from Clarke, 2014)

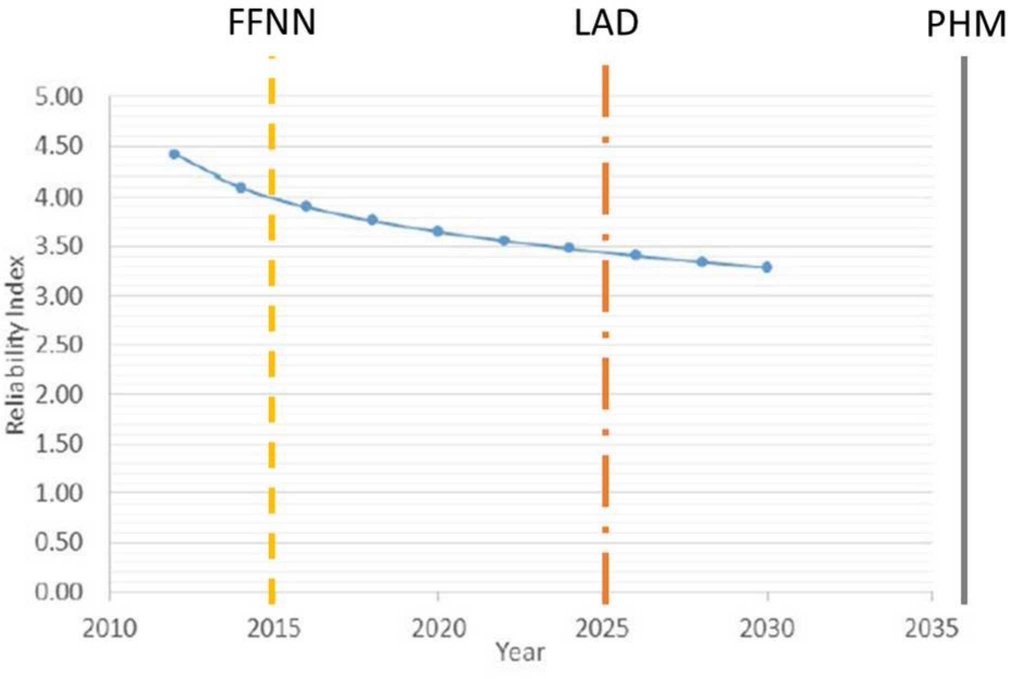

Figure E-6: Comparison of Prognostics Results for Component C6 (modified from Clarke, 2014)

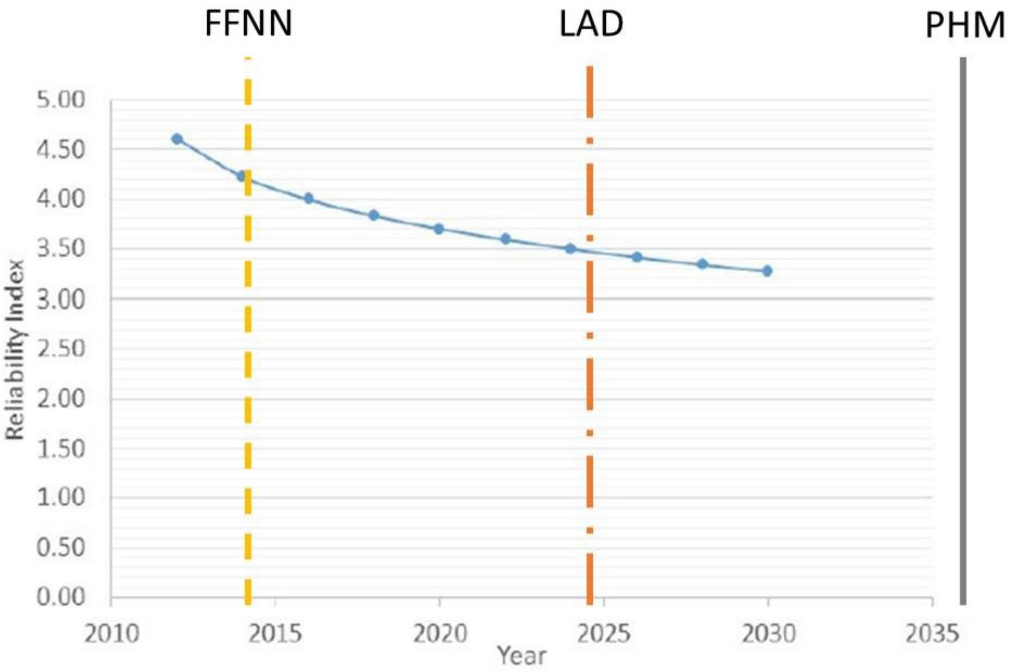

Figure E-7: Comparison of Prognostics Results for Component C7 (modified from Clarke, 2014)

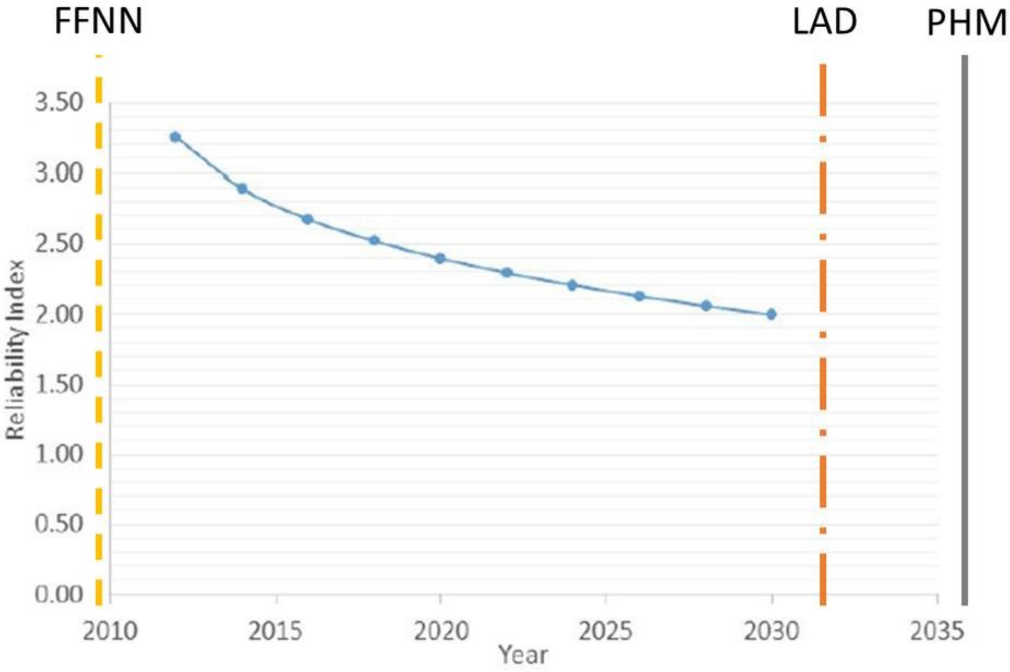

Figure E-8: Comparison of Prognostics Results for Component C8 (modified from Clarke, 2014)

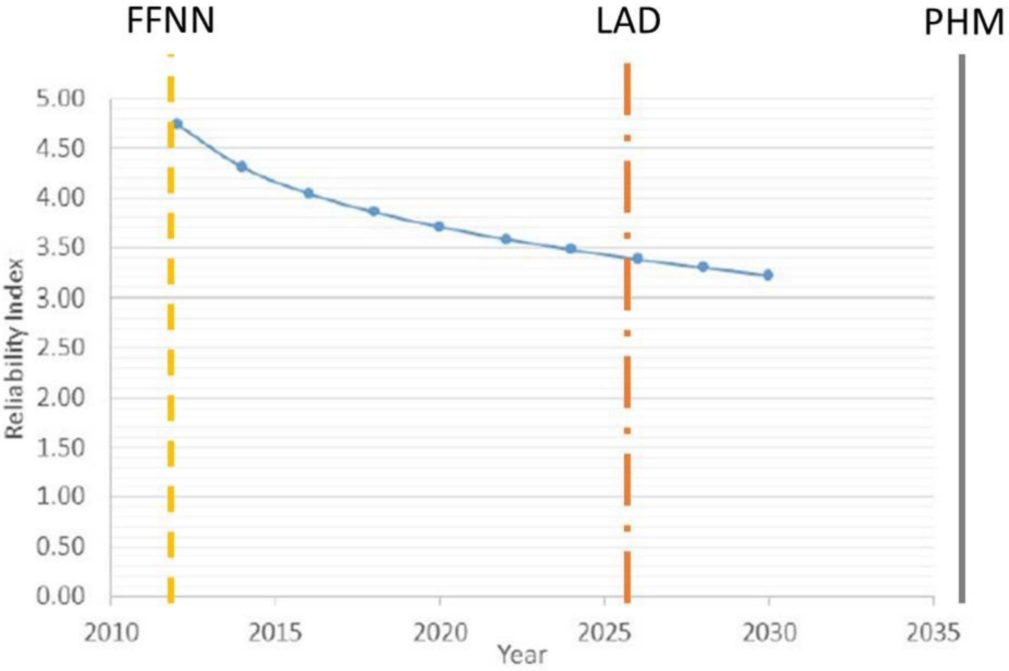

Figure E-9: Comparison of Prognostics Results for Component C9 (modified from Clarke, 2014)

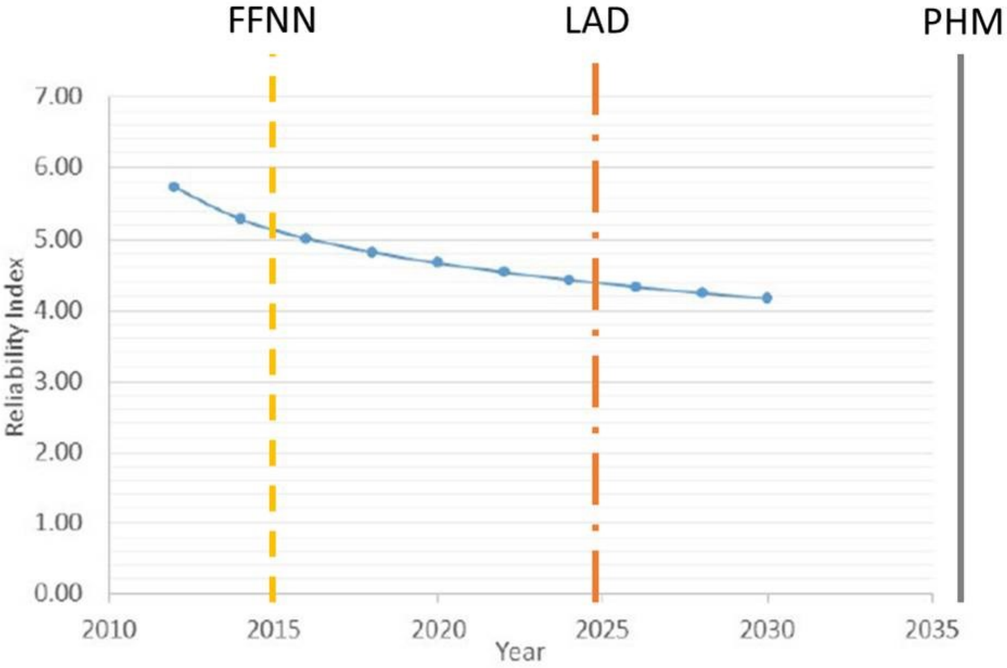

Figure E-10: Comparison of Prognostics Results for Component C10 (modified from Clarke, 2014)

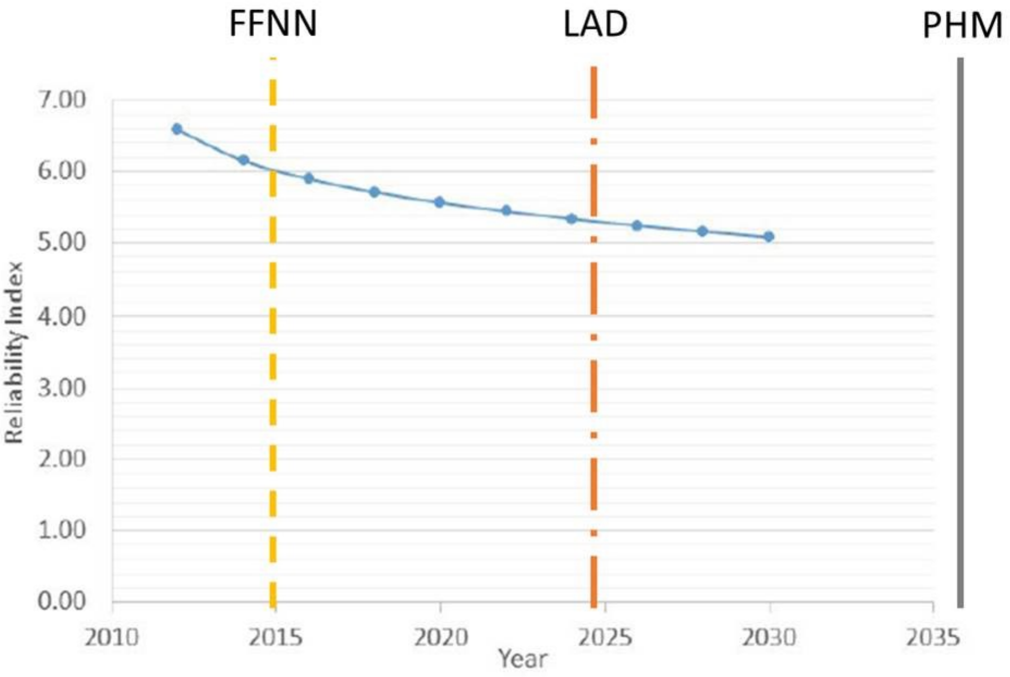

Figure E-11: Comparison of Prognostics Results for Component C11 (modified from Clarke, 2014)

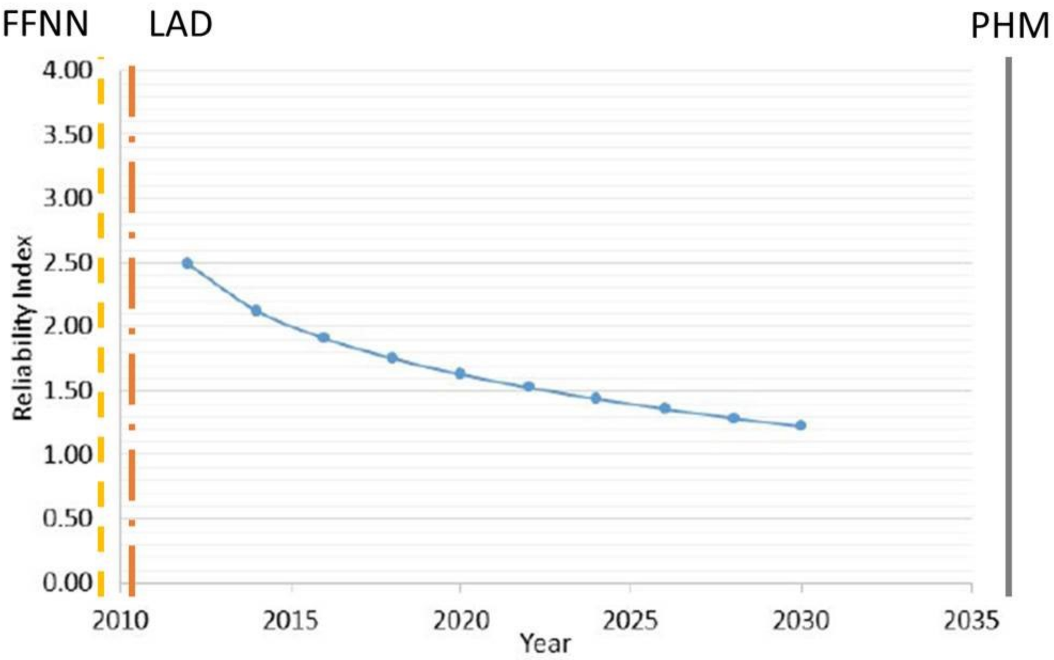

Figure E-12: Comparison of Prognostics Results for Component C14 (modified from Clarke, 2014)

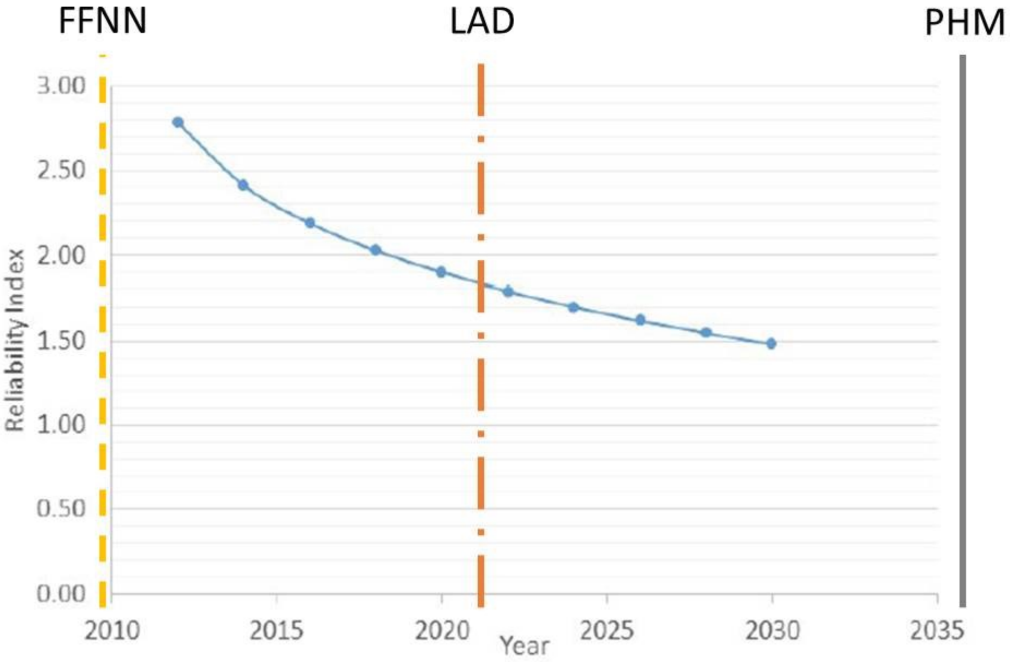

Figure E-13: Comparison of Prognostics Results for Component C16 (modified from Clarke, 2014)