12-2017

Developing Leading and Lagging Indicators to Enhance Equipment Reliability in a Lean System

Dhanush Agara Mallesh
University of Tennessee, Knoxville, dagarama@vols.utk.edu

Recommended Citation

Agara Mallesh, Dhanush, "Developing Leading and Lagging Indicators to Enhance Equipment Reliability in a Lean System. " Master's Thesis, University of Tennessee, 2017.
https://trace.tennessee.edu/utk_gradthes/4943

To the Graduate Council:

I am submitting herewith a thesis written by Dhanush Agara Mallesh entitled "Developing Leading and Lagging Indicators to Enhance Equipment Reliability in a Lean System." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Industrial Engineering.

Rapinder Sawhney, Major Professor

We have read this thesis and recommend its acceptance:

Xueping Li, Russell Zaretzki

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Developing Leading and Lagging Indicators to Enhance Equipment Reliability in a Lean System

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Dhanush Agara Mallesh

December 2017

For my parents, Appa and Amma, and Akka and Arjun,

and, my godson Leo.

Acknowledgements

I am extremely thankful and indebted to my advisor Dr. Rupy Sawhney for guidance through the course work and progress of this master's thesis. His constant support and encouragement will always be remembered. In addition to helping me academically, he showered me with many fatherly advice, and also put me in situations where I gained opportunities to improve professionally.

My appreciation also goes to Dr. Xueping Li and Dr. Russell Zaretzki for being a part of my thesis committee, and also being a constant source of motivation throughout the course of the program. The staff and faculty of the Department of Industrial and System Engineering at the University of Tennessee provided a friendly environment. In particular, my gratitude goes to Carla Arbogast for her words of encouragement and guidance during job applications, and, Enrique Macias De Anda, Dr. Ninad Pradhan, Dr. Kaveri Thakur and Vahid Ganji for their endless hours of help and dedication to all the students in Department of Industrial Engineering, and particularly to me. I am extremely thankful to Meridium, Inc. for their co-operation and willingness to work on this research. Special regards to Sarah Lukens who I could bounce ideas off and played a vital role in the completion of this research.

My colleagues and friends Bharadwaj, Guru, Dinesh, Samantha, Kapil, Abhishek, Mohit, Mozzie, Mostafa, Justin, Anju, Girish, Wolday, Julia, Bill, and other in CASRE group also deserve special mention.

Selfless affection and dedication offered by my parents Mallesh and Shantha cannot be expressed in words. My utmost indebtedness to Kavya and Arjun who helped throughout my graduate school starting from admission until I found a job.

“Do. Or do not. There is no try.”

— Jedi Master Yoda, *Star Wars*

“If you can find a path with no obstacles, it probably doesn’t lead anywhere”

— Frank A. Clark

Abstract

With increasing complexity in equipment, the failure rates are becoming a critical metric due to the unplanned maintenance in a production environment. Unplanned maintenance in manufacturing process is created issues with downtimes and decreasing the reliability of equipment. Failures in equipment have resulted in the loss of revenue to organizations encouraging maintenance practitioners to analyze ways to change unplanned to planned maintenance. Efficient failure prediction models are being developed to learn about the failures in advance. With this information, failures predicted can reduce the downtimes in the system and improve the throughput.

The goal of this thesis is to predict failure in centrifugal pumps using various machine learning models like random forest, stochastic gradient boosting, and extreme gradient boosting. For accurate prediction, historical sensor measurements were modified into leading and lagging indicators which explained the failure patterns in the equipment were developed. The best subset of indicators was selected by filtering using random forest and utilized in the developed model. Finally, the models give a probability of failure before the failure occurs. Appropriate evaluation metrics were used to obtain the accurate model. The proposed methodology was illustrated with two case studies: first, to the centrifugal pump asset performance data provided by Meridium, Inc. and second, the data collected from aircraft turbine engine provided in the NASA prognostics data repository. The automated methodology was shown to develop and identify appropriate failure leading and lagging indicators in both cases and facilitate machine learning model development.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Organizations especially manufacturing industries strive to achieve system performance excellence by providing higher quality products and services with reasonable costs and lead times. Fleischer et al. (2006) stated that the performance of a production system mainly depends on the equipment's availability and productivity. Essential elements in achieving high performance include identifying and anticipating disruptions in the delivery of products and services.

Disruptions include any unexpected event that will affect the standard performance of the system (Darmoul et al., 2013). In a manufacturing context, disruptions manifest in several ways: material unavailability, unavailability of operators, failure of equipment, production schedule overrides, etc. Reducing disruptions improves worker morale and focus (Aytug et al., 2005; Dal et al., 2000), helps equipment run smoothly, reduces raw material waste, and produces higher quality products (Ljungberg, 1998).

Identification of strategies to the improve performance of a system will depend on the critical factors causing the disruptions. Ahmad et al. (2003) listed the most important sources of disruptions in the manufacturing environments:

- Personnel

- Materials
- Schedules
- Equipment

Personnel

Personnel include operators and their skills required to work with equipment, it is considered as a critical factor as operators and equipment are in direct contact within in an environment (Miller, 1953). Personnel is influenced by human error and further affects the performance of a system. Most companies have stated that 8% of the disruption are caused by personnel (Gertman and Blackman, 1994), but the source of causes go beyond just human error. If manufacturing industry is embracing the new technology and advancements, then the technology must be practiced in a manner that imports confidence. However, the industries practice different methods, lagging behind in the skills to operate the technology results in disruptions. Some of the personnel disruptions are inevitable as they can occur without any warning during or after the operation of an equipment (Hopp and Spearman, 2011).

Materials

Materials refers to inventory comprising of raw materials, semi-finished material and finished goods at the work stations in a manufacturing line. The stock pile level of inventory correspond to the continuous operation of manufacturing line, if there were no inventory of raw materials at the start of manufacturing line, the shortage will disrupt the normal working condition leading to unscheduled downtime and reduce the performance of the system (Sawhney et al., 2010). Jeziorek (1994) suggested that the high inventory level may address the unscheduled downtime, however, they don't completely get rid of them. Multiple Japanese methodologies were testing to satisfy the inventory requirements but they were sensitive to variations resulting in a fragile process (Bennett, 2009), where the performance of the system decreased.

Schedules

Scheduling comprises of the process of planning, controlling and scheduling work or workloads in a production process. The planning process allows the organization to allocate resources like personnel and inventory to the required machinery or plant to optimize the productivity (Choi, 1997). The concept of scheduling helps improve the production efficiency by optimizing the manufacturing time and costs. On the other hand, disruptions in the operation schedule results in low levels of inventory which would lead to bad performance of the system.

Equipment

Typically manufacturing environments are subjected to one or more disruptions due to their dynamic nature. The disruptions mentioned above are referred to as real-time events, which can arbitrarily change system status and degrade its performance (Gholami et al., 2009). Equipment disruption is a key factor in the performance of the system which leads to different kinds of breakdowns, and equipment breakdown results in 80% of the downtime in manufacturing systems whether it can be controlled or not (Jabal Ameli et al., 2008). Some of the equipment breakdowns result in disruptions are power outages (Wu et al., 2008), short on equipment consumables (Hopp and Spearman, 2011), failure of equipment (Godinho Filho and Uzsoy, 2011), equipment tools goes out of adjustment (Veeger et al., 2010), tools wearing out (Hopp and Spearman, 2011), etc. This research seeks to identify which indicators contribute to system disruptions in the critical bottleneck area and therefore help organizations perform at higher levels by eliminating disruptions with improved equipment reliability and throughput.

The biggest problem equipment disruptions cause are truly unplanned equipment breakdowns. The breakdowns may occur at any point of time in the job, during or in-between them. In August 2001, a crude distillation unit malfunction in the Citgo Petroleum Co. resulted in shutdown of the refinery for 12 months, and estimated total value of loss of $230 million (Marsh, 2014). Failure of a pump in an oil refinery in West Texas led to a massive fire explosion incurring a loss of over $380 million and shut down for a year (Marsh, 2014). According to a study by Emerson (2016), unplanned outages in data centers results in a loss of nearly $9000 per minute. An analysis by Tucker et al. (2013) of 190 US Gulf of Mexico asset producers in Ziff's Energy Group revealed an opportunity to improve the production efficiency by reducing the downtime. Total (planned and unplanned) production efficiency of these assets was 88%. Out of the 12% efficiency loss, 8% was relatively caused due to unplanned maintenance. If this loss were prevented, the organization would have saved close to $600 million per year. It is challenging to analyze and repair the equipment within a short period for a human being during the unplanned maintenance since the disruptions are unpredictable resulting in long downtimes. Further leading to fragment loss, facility failures and operational upsets. Therefore, if the breakdowns were known in advance, the organizations can avoid the costly downtimes.

Equipment disruptions can cause unplanned delays in the production process. The delays result in downtime affecting the planned schedule; the process takes extra time than previously planned. Unplanned events like downtime negatively effect the intended capacity's ability to meet the demand (Melnyk, 2007). The delay in one piece of equipment affects the entire process and creates variations in the performance of the production process. Variations caused by equipment breakdowns can be minimized if breakdowns can be anticipated and corrective measures are applied in time. Therefore, when the effects of breakdowns are previously determined, it is possible to develop a methodology to predict the breakdowns and schedule a planned maintenance to reduce downtime. By doing this, high reliability and targeted performance of the system can be achieved.

Planned maintenance results in reliable and safe to operate equipment, therefore influencing the quality, manpower, material, tools and cost (Pintelon and Gelders, 1992; Ahuja and Khamba, 2008). Albino et al. (1992), Savsar et al. (1993) and Vineyard and Meredith (1992) developed simulation models to study the relationship between maintenance and production, and identified the different effect of planned versus unplanned maintenance strategies. Planned maintenance strategies resulted in optimal inventory level and satisfied demand when compared unplanned maintenance, which failed to meet demand. Mosley et al. (1998) developed a predictive model with the objective to reduce equipment downtime by scheduling maintenance and obtained 20% increase in production. Therefore, planned equipment maintenance is a vital step in the manufacturing process (Ahmad et al., 2003). The literature study shows that mathematical models have been developed to setup planned maintenance activities that mitigate the effect of downtime between the process. Such improvements can be achieved through transitioning from unplanned to planned maintenance.

The motivation behind planned and scheduled maintenance is to improve equipment health, or at least system reliability. It is a vital part of the asset management. Moreover, planned maintenance if done efficiently with proper policies may reduce equipment downtime and other undesirable effects of downtime. Maintenance evidently affects equipment components and its reliability: if little is performed, this may lead to expensive failures and higher downtime, and therefore, reliability is low; performed often, reliability will improve but will result in a linear increase in maintenance cost (Endrenyi et al., 2001).

In a system, if the maintenance is focused on the right equipment at the right time, then a significant impact is made regarding system reliability. Especially if the equipment is the bottleneck of the system, planned maintenance focuses on decreasing downtime by improving the component availability. Taking care of the bottleneck equipment improves the system reliability, production costs are cut, buffer inventories are cut, effective capacity is increased, and moreover, a significant improvement is made in the throughput.

1.2 Problem Statement

Traditionally maintenance practitioners used failure rates, mean time to failure, vibration measurements, oil analysis and a variety of other models which predict failures, but each one of them requires a particular set of equipment. This thesis examines sensors that are typically measured to monitor and evaluate the health of pumps. The sensor measurements are analyzed instead of the failure rates, and a model is developed to predict failure allowing unplanned maintenance to transition into planned maintenance. Sensor measurements themselves are not sufficient, and therefore, will be modified to obtain relevant information and then filtered bringing the leading and lagging indicators. This is used in the machine leaning models to predict failure rate in centrifugal pumps at a very high probability. The reduction in failure rates using planned maintenance will result in higher reliability of the equipment. This potentially will help eliminate the bottleneck equipment in the process leading to improvement in the throughput of the system.

1.3 Approach

The variables utilized in this thesis were obtained from an extensive database within Meridium (2017), an asset performance management organization. The database consists of a combination of equipments records and its sensor readings at hourly intervals. The equipment considered for research is centrifugal pumps with seven sensor measurements: suction pressure, discharge pressure, flow, temperature, power, rotations per minute (rpm), and vibration.

With the extensive amount of data available and limited data-driven approaches, machine learning models chose the ideal method to be used on the sensor data. Initially, a pilot model is developed using the raw sensor measurements to predict failure. However, the raw sensor data did not contain quality information to predict failure accurately. Therefore, indicators were developed which consisted of encoded information from the raw sensors that improves the machine learning methods to classify the failures (Anderson et al., 2013).

For the methods to work at peak performance and obtain high accuracy, development and selection of the indicators is an important step. The methodology to develop indicators from the sensor data, and build good prediction models using these indicators, as applied here, consists of three steps (see Figure 1.1).

The approach as shown in Figure 1.1 is briefly explained followed by a detailed explanation. The first step in the failure prediction methodology involves development of failure indicators which requires merging the different data sources. This data is used to engineer new leading and lagging indicators that relate to the working condition of

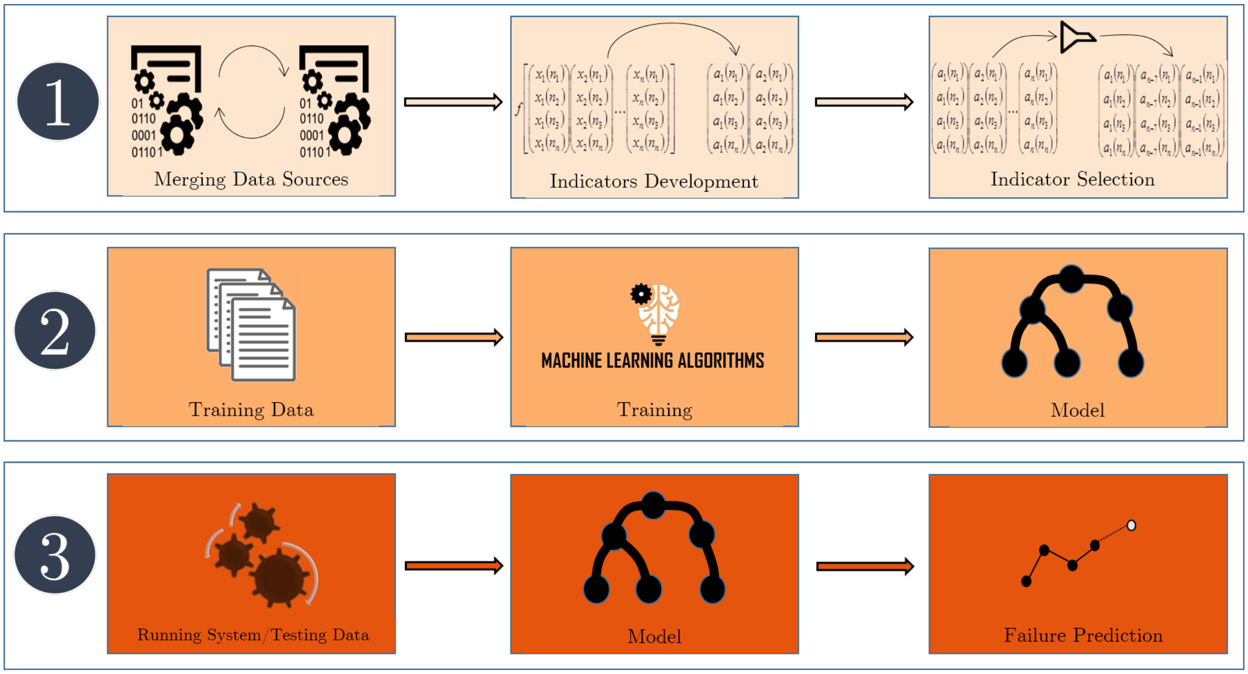


Figure 1.1: Approach: First, the indicators are developed and passed through filters. Second, the model is built using the indicators. Third, the model is used on testing data to predict the probability of failure

an equipment and the best indicators are selected for prediction using filter and wrapper indicator selection methods. In the second step, a machine learning algorithm is trained using the data obtained from the selected indicators. The third step includes the training process which optimizes the objective function and regularizes the model parameters. Since the data sources are historical, the data set is divided into training and testing dataset, where the training data will be used to choose the best model parameters while the testing dataset will be used to evaluate the prediction quality of the algorithm. The tuned and trained model will be later used to predict failures.

The first step is the most important step and the most time consuming, as domain specific knowledge is required for data acquisition, merging it, cleaning and data wrangling, and many iterations of indicator engineering go into it. Specifically, the raw data is imported from three different sources: equipment records, failure and maintenance history and sensor measurements data. The data sources include sensor reading data collected from centrifugal pumps, as well as failure-prone historical maintenance and repair records that detail failure description and type. The data sources are combined using spatial and time relationship between equipment and sensor readings. Next, the failure indicators are developed; they relate to the working condition of equipment and failure patterns. The indicators obtained from the raw data may not be directly useful or relate to the equipment. Hence, in this case, new indicators or a set of indicators holding the same information as the historical sensor measurements are developed to obtain the leading and the lagging indicators. Sensor data is tagged with a timestamp which helps to calculate the lagging indicators. Some of the indicators developed are:

- Binned indicators
- Frequency domain indicators - Fast Fourier Transform
- Time domain indicators - mean, standard deviation, range, peak, etc.
- Normalized indicators - standard core and silly pins
- Time-Frequency indicators - wavelet transforms
- Lagged indicators - lagged rolling average
- Interaction indicators

Other indicators include a non-linear combination of primary indicators and decomposition transforms. However, not all the indicators are effective enough to predict the failure at a certain stage, and the irrelevant indicators will induce higher computational time. Therefore, important indicators are selected using random forest with specified iterations by optimizing the error parameters and obtaining the importance of each indicator.

When the dataset is preprocessed, and the important indicators are obtained, the next step in the methodology is to select a machine learning algorithm and train an appropriate model. The dataset from the previous step is divided into training and test dataset, where the training data is used for model development, and the testing data is used to evaluate the model. Since the dataset is timestamped data, regular k-fold cross-validation will result in overfitting the model. Therefore, a time-dependent splitting strategy is used to get a cross-validation statistic and obtain the testing data that are subsequently compared to the training period (Arlot et al., 2010). The approach for selecting the model depends on the equipment iterating failure patterns. These failure events are triggered by the dependency of equipment on the succession of other error events, but not all these events lead to failure. After researching the different mechanisms and dependencies, the following reasons were considered in selected a failure prediction model:

- Equipment health depend on the change in error patterns
- These error patterns have innumerable conditions, where some patterns relate to the equipment leading to failure, and some are just false positives
- To learn and record those patterns which result in failure
- The machine learning techniques are used to train the model based using the recorded patterns

Decision tree learning algorithms such as boosted trees and random forest classifier have been proved to be effective in pattern recognition tasks like automatic recognition of handwritten letters (Polikar, 2006), human emotions (Horn, 2001) and fraud detection (Bishop, 2006). The above being one of the reasons to choose decision tree learners and the second reason being able to differentiate between miscued events, failures and non-failures. Miscued events go unobserved but later deteriorate and become failures which can be detected. This action can be compared to the functioning of tree classifiers in decision trees; the initial states can go unattributed but are later dynamically introduced to the lower stages of the tree structure. The probability of occurrence of failure is predicted by learning the different stages down the line of the decision tree.

The objective of decision learning algorithms is to regularize the hyper-parameters to learn the patterns that lead to failure. To treat the imbalance of classes, i.e., failure and not failures, oversampling techniques are used along with tuning the classifiers to handle the data. The indicators behave capriciously at different points of time when a failure occurs. Therefore, separate trees are developed in the decision trees to learn the various scenarios of failure sequences. In the prediction phase, the model learns all the different patterns using the training dataset.

The final step of the methodology is to select the best model. Since the failure dataset faces the imbalance problem, various classification evaluations are compared to the benchmarked metrics which are calculated at different scenarios. The comparison of the models using evaluation metrics will help obtain the better performing decision tree model resulting in the accurate prediction of failures. Also, the models are stacked together by combining the outputs using ensembling. Ensembling is relevant when there is a chance of model over-fitting or under-fitting.

1.4 Assumptions

After examining the equipment and sensor records, some characteristics have been determined which result in assumptions based on which the failure indicators and prediction model are developed. The assumptions are:

1. The number of records for failure is less due to rare occasion of failures. The period from the end of failure to the start of next failure is assumed to be non-failures and the sensor records as labeled appropriately.

2. The duration of failure is unavailable in the dataset. Therefore, the end of the maintenance period for that equipment is assumed as the end of the failure period.

3. The equipment running for an extended period and multitasking can sometimes result in sudden increase or decrease in the sensor measurements. For this reason, such scenarios are assumed to be noise in the data.

4. All the data sources are timestamped. The sensor data contains hourly interval timestamp with sensor measurements while the failure and maintenance records have

timestamp corresponding to the event. It is assumed both the timestamp and indicators contain information to predict failure.

5. Due to the complexity of equipment, it is assumed that the failure patterns at different points of time are different, i.e., different indicators can react to different types of failures resulting in various failure trends.

6. The sensed measurements contain hidden information. Therefore, multiple indicators are developed assuming that all of them are significant. In the following stage, indicator selection is used to identify important indicators.

1.5 Organization of Thesis

An outline of the thesis is shown in Figure 1.2. Chapter 2 presents the literature review to understand the indicator development and a survey of failure prediction models. Chapter 3 discusses the theoretical details of the methodology that will be used in this thesis. Chapter 4 discusses the results of the methodology applied to the Meridium's and NASA's data set. The first dataset is a technical data obtained from Meridium with centrifugal pump records, while the second dataset uses the aircraft turbine engine dataset obtained from NASA's prognostics center of excellence repository. Chapter 5 includes the conclusion remarks and an outlook to future work.

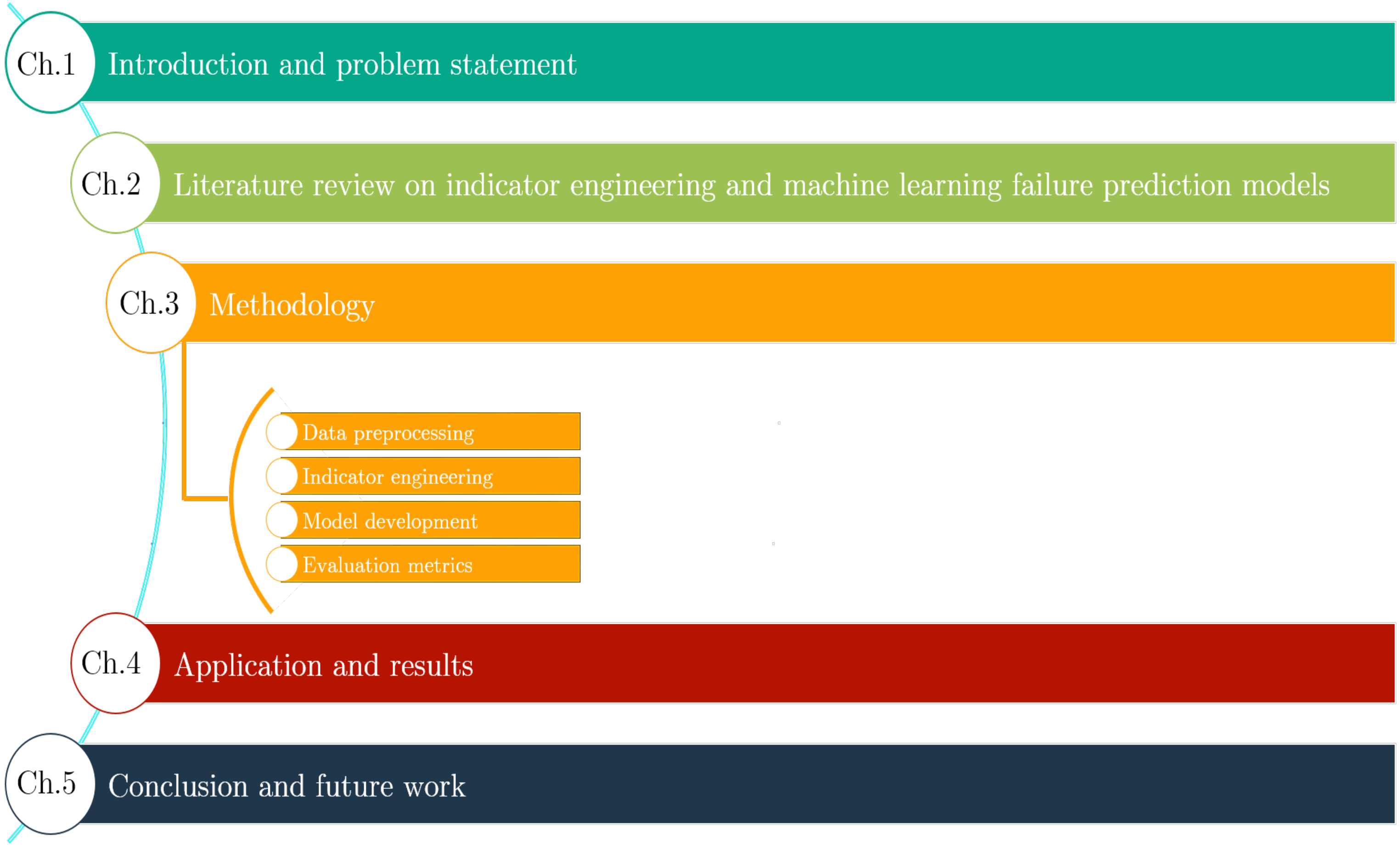

Ch.1
Introduction and problem statement
Ch.2
Literature review on indicator engineering and machine learning failure prediction models
Ch.3
Methodology
Data preprocessing
Indicator engineering
Model development
Evaluation metrics
Ch.4
Application and results
Ch.5
Conclusion and future work

Figure 1.2: Framework of the thesis

Chapter 2

Literature Review

Compared to the traditional approaches like breakdown maintenance and preventive maintenance, prognostics is still a new field being investigated. In fact, researchers studying prognostics tend to focus on specific areas instead of the field as a whole, mentioning, monitoring, fault detection and diagnostics while disregarding greater possibilities of the field (Heo, 2008; Keller et al., 2006; Liao et al., 2006). Lately, new methodologies and studies have been published for prognostics approaches, which are discussed in detail in the sections that follow.

This chapter discusses the concept of prognostics, different types of maintenance and its undesirable effects. The relationship between real-time sensor data and failure detection is also addressed. The literature relating to the various machine learning methods used for failure prediction, with an inclination towards maintenance management, are also examined in detail.

2.1 Maintenance Management Evolution

Maintenance is defined as "set of activities required to keep physical assets in the desired operating condition or to restore them to the best optimal condition" (Heo, 2008; Liao et al., 2006; Pintelon and Parodi-Herz, 2008). In other words, maintenance is a significant factor that impacts the availability and reliability of the asset to overcome the increase in failure rate which degrades the production quality and efficiency.

Early in the 1980s, maintenance costs summed up to more than $600 billion in the domestic plants in the United States, just to maintain their critical equipments. These costs doubled by the 2000's (Fluke, 2007). Moreover, in some industries like mining, petrochemical, and construction, one-third of the maintenance costs is spent on ineffective maintenance and exceeds the operational costs (Eti et al., 2005; Parida and Kumar, 2006).

Over the years, maintenance has expanded from an everyday function into a complex task. Maintenance has undergone a continuous change in the organizational level with increasing complexity of industrial technology and machine tools leading to an elevation in maintenance costs (Parida and Kumar, 2006). Initially maintenance did not receive the importance it deserved, but it evolved to become a major task to increase throughput, mainly by reducing downtime of equipment.

Maintenance evolved from a focus on breakdown and progressed to preventive maintenance. Eventually, reliability-centered maintenance was developed and advanced to a more practical approach based on condition-based maintenance. Currently, the newest maintenance is based on the prognostics.

The first type of maintenance introduced was breakdown maintenance, where actions are performed after the equipment has come to a complete stop or has become unstable. The equipment was checked for defective parts and repaired. Initially, breakdown maintenance was created only for non-repairable cases and later extended to repairable cases (Barlow et al., 1963).

Multiple types of maintenance actions were defined under corrective maintenance such as minimal repair, general repair, failure replacement and general repair. The researchers have contributed with multiple models which adopted the corrective maintenance actions. The general age replacement model was proposed by Block et al. (1988); Stadje and Zuckerman (1991), in which equipment failures are corrected with minimal repair and returned to their original working condition by identifying the probability repair and replacement.

Among the different corrective maintenance models, one of the appealing models was developed by Kijima (1989) to characterize the equipment and maintenance performances by calculating the new virtual age based on the idea of the primary age. Although corrective maintenance is easy to perform with less work and lower short-term costs, it increases the downtime and reduces the reliability of equipment.

The second concept that was introduced in the evolution of maintenance focused on prevention and originated sometime between the years 1950 and 1960. It was created to avoid a complete shutdown of equipment and disastrous failures. The actions include setting up regular inspections and maintenance at periodic time intervals, regardless of the current condition of the equipment. The critical parameter in PM is determining the optimal time interval for inspection. Savits (1988) and Block et al. (1990) developed one of the first PM models known as a block-replacement model. The model uses fixed time intervals to take action by removing each failure by replacement. Multiple authors (Tilquin and Cleroux, 1975; Boland, 1982; Boland and Proschan, 1982; Aven, 1983) published their work using block-replacement model. Later, Bazovsky (2004) initiated the use of optimization methods in PM models. Jardine and Tsang (2013) used an idea developed by Bazovsky (2004) while developing decision models to calculate the best time interval by extrapolating the historical reliability data and expected cost rate. Kelly (1989) did a survey of practitioners, which proved the unpopularity of fixed time intervals in PM. The strategies in PM reduces the number of failures. However, they do not eliminate immediate disastrous failures between the intervals and also increase maintenance activities, making PM labor intensive.

The evolution of PM was introduced in the late 1960s, when the United States civil aircraft industry developed reliability-centered maintenance (RCM) to reduce cost rate resulting from PM to achieve higher reliability while preserving the functionality of the equipment. It is based on the evolved form of Failure Mode Effect Analysis (FMEA) and involves the use of statistical parameters, particularly probability distributions.

The models in RCM use traditional reliability approaches where the parameters are analyzed based on the distribution of time-to-failure data obtained from similar equipment. The application of RCM using parametric models such as Weibull distribution Rausand et al.

(2004), non-homogeneous Poisson process Kothamasu et al. (2009) and Weibull distribution Van Noortwijk (2009) has helped improve the machine reliability. However, the disadvantage is that RCM provides the overall reliability estimate of the whole population of similar equipment in the organization, rather than the real-time reliability estimate of a particular equipment.

In the last two decades, condition-based maintenance (CBM) has been developed in the direction to reduce downtime by monitoring equipment health data without interrupting the normal working operation. CBM introduces maintenance tasks into the schedule only when there is an intervention detected in the measurements observed from the equipment. An able CBM can reduce unnecessary costs by eliminating the scheduled PM tasks. Nonetheless, the minimization of failures and costs require constant on-line monitoring of equipment health. Hess et al. (2008) identified some limitations in CBM traditional methods during a research conference held by National Institute of Standards and Technology, which are described below:

- failed to observe equipment constantly
- inaccurate results in prediction the equipment health
- inability to learn from the historical failure data and detect new failure patterns

In other words, CBM methods are limited by inefficiency in observing, reacting, and recommending actions to failures.

As the scope of maintenance gained more importance within the organizational performance parameters, researchers contributed to enhance CBM's approach which evolved into the concept of Prognostics and Health Management (PHM) (Hess et al., 2008). PHM can be defined as the ideology in maintenance which integrates the physics of failure mechanism and life-cycle management (Uckun et al., 2008). PHM is today's most widely accepted practice in high technology equipment based organizations, such as the aerospace industry and military. The United States has allocated special emphasis in this approach within NASA in their spacecraft (Osipov et al., 2007) and the military in two different programs. The programs are Joint Strike Fighter Program (Hess et al., 2004) and Future Combat Systems Program (Barton, 2007) for anomaly detection, efficient diagnostics, real-time performance monitoring and predicting failures.

PHM has been acting its part in the multiple areas helping industries from the equipment manufacturer to the end user. Some of the advantages of PHM compared to the other maintenance systems (Hess et al., 2008; Uckun et al., 2008; Asmai et al., 2010; Balaban and Alonso, 2012) are:

- improvement in equipment reliability (forecast failure-prone equipment)
- ability to recommend maintenance actions to increase life of an equipment
- reduction of downtime and operational costs by elimination of unnecessary maintenance actions

The main contribution of PHM is to provide the end users the knowledge of the future health of equipment. This job broadly consists of two different steps. The first step is monitoring and accessing the health condition; then anomaly detection techniques can be used here to detect various types of failure patterns. The second step aims at predicting the probability of failure, where machine learning methods are used (Si et al., 2011).

In summary, the evolution of maintenance has expanded from simple reactive approaches to data guided prediction, as the nature of business rely on effective strategies. The thesis focus on present day approaches while expanding the body of knowledge regarding PHM.

2.2 Prognostics and Health Management

PHM is a concept within equipment monitoring maintenance system which also includes fault analysis, equipment diagnostics, anomaly detection and online monitoring. Kothamasu et al. (2009) referred to prognostics as the complete form of CBM system. The authors have used PHM estimates to develop applications for maintenance assessment and scheduling. Pintelon and Parodi-Herz (2008); Hess et al. (2008) researched the use of PHM to predict failure for logistics systems. Table 2.1 lists the exciting research work using prognostics and health management.

Callan et al. (2006) divided the PHM system into five different steps: data acquisition, data manipulation, condition monitoring, health assessment and prognostics. Prognostics is the most important measure used to “estimate the time of failure or risk for one or more existing and future failure modes” (Katipamula and Brambley, 2005). The application of all the steps in the CBM will result in the improvement in production, reduction in downtime and failures, improved work performance of equipment, elimination of unnecessary downtime and a decrease in life-cycle cost.

The model considers the original data and produces the results in the form of probability of failure to schedule maintenance routines. Data is collected through continuous online observation of equipment and analyzed for pattern changes. The analysis can be performed with the help of different methods, including statistical and empirical models (Ma and Jiang, 2011). The current condition of the equipment is assessed and compared to the estimates of the degradation level; this helps determine if the equipment is operating abnormally. A statistical method utilizes the estimates distribution of normal working and degraded condition to determine the shift. If a shift is observed, it is important to determine the cause; equipment has different degradation levels based on the type of failure. Finally, different prognostics models can be used to determine the probability of failure of the equipment.

Prognostics algorithms can be classified into three types: physics-based, data-driven and hybrid prognostics as shown in Figure 2.1 (Si et al., 2011).

Table 2.1: Summary of traditional methods using prognostics and health management

Author	Title	Models	Advantages	Disadvantages
Goode, K. B., Moore, J., & Roylance, B. J. (2000)	Plant machinery working life prediction method utilizing reliability and condition-monitoring data	Weibull distribution	Predicts failure using both reliability and historical condition monitoring data	Assumes underlying distribution for input variables
Li, Y., Kurfess, T. R., & Liang, S. Y. (2000)	Stochastic prognostics for rolling element bearings	Crack growth modeling	Adapts to change in operational conditions	Failure is assumed to be directly correlated to the vibration signal
Marble, S., & Morton, B. P. (2006)	Predicting the remaining life of propulsion system bearings	Contact analysis	Failure prediction considers equipment geometry, size, load and speed	Physical parameters needs to be calculated and computationally expensive
Wang, W. (2007)	An adaptive predictor for dynamic system forecasting	Time series using neural network	Handles non-linear variables and dont require prior knowledge	Predicts failure only when the measurement exceeds the threshold and with a short horizon
Zhang, S., Ma, L., Sun, Y., & Mathew, J. (2007)	Asset health reliability estimation based on condition data	Recursive Bayesian technique	Predicts failure using condition monitoring data rather than failure event data	Performance of model depends on correct analysis of threshold
Sun, Y., Ma, L., Mathew, J., Wang, W., & Zhang, S. (2006)	Mechanical systems hazard estimation using condition monitoring	Proportional covariates model	Performs well even without failure event data	Assumes that failure pattern changes with covariates

2.2.1 Physics-based Prognostics

The physics-based algorithms for prognostics use mathematical techniques to model and understand the degradation of equipment (Pecht and Jaai, 2010). The equipment health estimates using this algorithm are based on the process information that causes abnormal activity and results in failure. They detect failure which occurs under the circumstances of mechanical, electrical, chemical, thermal and radiation disturbances (Pecht and Gu, 2009). The algorithm is selected based on the knowledge of loading conditions and equipment geometry (Pecht, 2008).

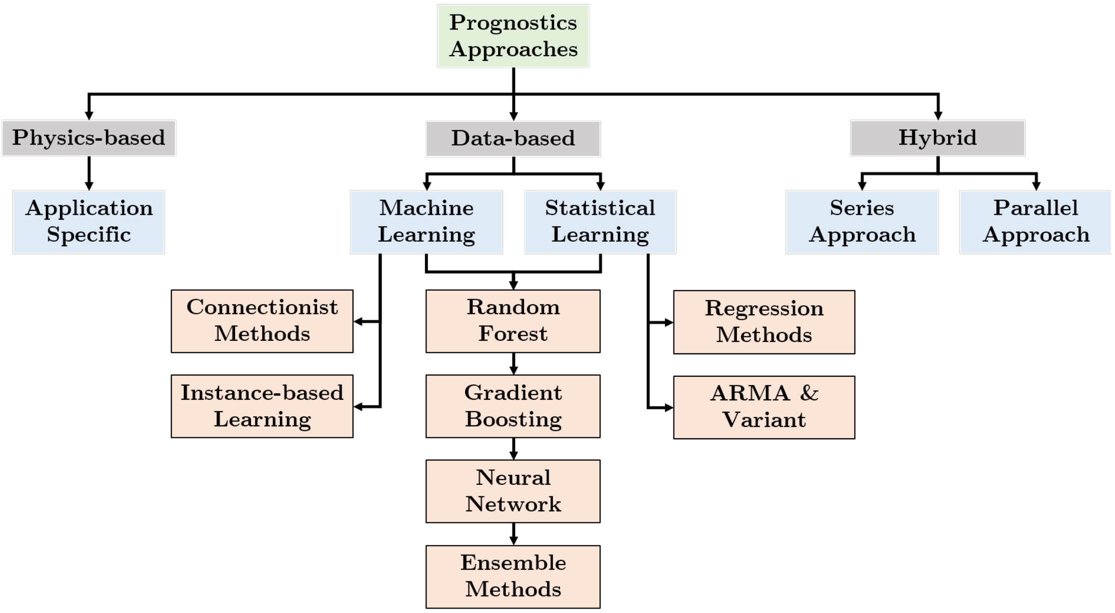


Figure 2.1: Classification of prognostics approaches

Physics-based prognostics are developed with an application in specific and applied at the lowest hierarchy. Heng et al. (2009) suggested that physics-based methods are inefficient to use in an industrial environment, because of the different types of failures observed in various types of equipment. Furthermore, method selection is hard when the geometry of the equipment is unavailable (Pecht and Jaai, 2010). Therefore it is ineffective to use this due to the various assumptions, errors, and uncertainty developed in a dynamic operating condition which leads to lower accuracy (Wang, 2010).

2.2.2 Data-driven Prognostics

Data-driven approaches develop models based on the condition monitoring data instead of models that depend on the comprehensive equipment physics. They assume that unless an abnormal activity occurs in the system, it keeps working in normal condition. This approach obtains relevant information from the raw data and behavior patterns of the equipment. Therefore, data-driven methods have a better application due to economic modeling, as they depend only on the historical data and don't require any human expertise (Javed et al., 2013; Pecht and Jaai, 2010). The data-driven prognostics methods are classified into machine learning and statistical approaches.

Machine learning methods are based on the concept of artificial intelligence. They learn from previous examples of failure patterns and modes in historical data. Based on the types of data available, different types of machine learning methods can be applied. Supervised methods are used when the data is labeled, while unsupervised methods are applied to unlabeled data. Atherton (1999) and Yam et al. (2001) developed prognostics model using recurrent neural networks (supervised method) to analyze the trends in the monitored data and forecast the equipment measured value for the future. Zhang et al. (2007) used a recursive Bayesian method on density function of measured values from the equipment to predict failure probability. The model mostly used historical degradation pattern rather than the failure event data.

Statistical approach predicts failure by fitting the monitored data to a probabilistic model, and fitted curve is extrapolated. Statistical methods are similar to machine learning methods: they are simple and use the condition monitored data to predict failure. However, the accuracy of the model can depend on the completeness and nature of data. Si et al. (2011) presented a survey of a literature review of statistical methods. The survey list includes multiple methods such as regression methods, Hidden Markov models, Kalman filtering methods, etc,. Orchard et al. (2005) employed a particle filtering method to forecast the non-linear projection of increasing degradation in the turbo engine blade. A priori estimate was calculated based on the previous state of the blade and used to generate the prediction horizon of the desired state.

Data-driven prognostics have the ability to extract relevant information from big noisy data to make prognostics decisions (Dragomir et al., 2009). When data is collected from equipment operating in a real industrial environment, it contains variability and noise. Therefore the preprocessing step is an essential step to extract relevant information to improve the accuracy of the model.

2.2.3 Hybrid Prognostics

Hybrid prognostic is a combination of physics-based and data-driven approaches. The significance of this method is to develop an optimized prognostics model using both historical monitoring and reliability data with minimal assumptions to handle uncertainty and predict failures of high accuracy. Sun et al. (2006) proposed proportional covariates model (PCM) which used system hazard as an explanatory variable and the monitoring data as response variables. This model was used to estimate the hazard function in the absence of historical failure data as long as the response variables were proportional to the hazard. Further research suggested that both event failure and condition monitoring data was used in measuring reliability estimation parameters. Hybrid prognostics is further divided into two types: series and parallel approaches. In a series approach, a physics-based model is paired with the data-driven method to update model parameters with new training data (Psichogios and Ungar, 1992). In a parallel approach, a combination of data-based and physics-based models work in concert to predict residuals in situations when other models cannot (Thompson and Kramer, 1994).

Heng et al. (2007) developed a new paradigm called intelligent product limit estimator which incorporated data composed of equipment health up to the time of repair or replacement (suspended or truncated data) to predict failure. The model built using the suspended data resulted in an excellent long-range prediction, as the equipment in operation are never allowed to run to failure and data are suspended. In the model, Kaplan-Meier develops estimates using the variation in equipment health, and these estimate probabilities are used as training targets in the feed-forward neural network. The research presents a model which utilizes the available information and provides accurate prediction in a probabilistic unit.

Despite the abundant research efforts, prognostics approaches are not perfect due to the assumptions inherent in every model (Sikorska et al., 2011). Furthermore, each approach has pros and cons, while limiting their application on the data available. A prognostics approach for a particular application is selected based on two important factors: performance and applicability (Jardine et al., 2006). Javed (2014) compared all the three prognostics approaches by assigning weights based on different criteria in performance and applicability. The data-driven approach had the maximum weight in the applicability factor as it can learn various types of failure patterns with its general methods. However, it requires improvement in the performance part. An et al. (2015) compared all the approaches with the help of case study, and data-driven method outperformed others with machine learning methods. The data-driven method performs better in the event of high levels of noise and large training data sets. Machine learning methods in data-driven approach is still an improving field, but it is heavily researched to improve some of the drawbacks. Because data-driven approach especially machine learning methods are the most efficient with good applicability, this method will be considered for further analysis in the thesis.

Some machine learning methods obtain high accuracy and some fails (Domingos, 2012), while the quality of the data has a big impact on the performance of the learning algorithm too. Inconsistent data will reduce the efficiency of a well tuned complex algorithm, while a good dataset can obtain high accuracy using a simple algorithm. Thus, developing features from raw data to extract the useful information is important (Domingos, 2012). It is also important to note that adding useless features to the data will result in overfitting the machine learner. Hence, in this study, the drawbacks are addressed by introducing a method to develop indicators of importance and selecting them to prevent curse of dimensionality.

2.3 Sensor Technology in Maintenance

Prognostics approaches have a long history of methods which evolved from using visual inspection to sensor signals to predict failures. Traditionally human interaction was required to diagnose equipment degradation. Fortunately, sensor technology has taken over the advanced maintenance approaches to help identify failures (Spencer et al., 2004). Utilization of sensor signals in parallel with the prognostic approach based data-driven methods will reduce unplanned maintenance costs, and improve availability and safety.

In spite of advancement in maintenance technologies, unplanned and hand-held based maintenance is still being used in some industrial equipments. At present, nearly 30% of the equipment is not using modern technology in maintenance practices (Hashemian et al., 2005). Emerson company reported the dataset containing pressure, level and flow transmitters measured using hand-held maintenance technology in multiple industries. Emerson found that 70% of the time maintenance was scheduled based on the measurement reading, while there was no breakdown in the transmitters (Hale, 2007). However, some nuclear plants utilized online sensor technology to get sensed reading and found that there were no problems 90% of the time in equipment (Hashemian et al., 1998). The above literature suggests that online sensor technology, rather than hand-held devices, reduces the failure rate and downtime.

Hashemian et al. (1995) describes that in online calibration monitoring, the equipment with sensor measurements drifted beyond the control-limits was identified and maintenance actions was performed during the plant downtime. This approach minimized the efforts of operators by 90%. Hashemian et al. (2005) developed the loop current step response method, which used active sensor measurements to schedule planned maintenance in cables, motors and thermocouples. As noted by Hashemian et al. (2007), sensor-based predictive maintenance methods were used to detect blockages in pressure sensing lines with the help of pressure sensor placed at the end of the sensing line.

Effective sensor technology and monitoring builds a good foundation for developing an efficient prognostic based maintenance. In fact, the data generated from sensors are a critical component for prognostics approach. However, the options of how the sensor data can be utilized is still being researched. The sensed data is collected from different sources which are not interconnected and an independent model is built for each source (Levis et al., 2004). With improvement in sensor technology, integrated online sensor monitoring systems were developed, but still lack the automated failure prediction model to utilize them efficiently (Madria et al., 2014). Therefore, this thesis discusses the ways to close the gap of automation using machine learning methods to build an automated failure prediction model.

2.4 Failure Prediction Models

The primary motivation for the development of prediction models is to understand the effect of the quality of historical data on the decision that help schedule planned maintenance to prevent failure in equipment. The failure event data and condition monitoring data can be efficiently utilized to identify failure patterns of different faults in equipment and guide maintenance decisions so as to reduce failure and downtime. In this section of the chapter, different machine learning techniques that can be used in the prediction of failure is discussed.

Recently there has been a significant increase in the use of predictive models in prognostics methodology. Multiple models have been developed for specific equipment or type of failures. The methods include random forest, gradient boosting, neural networks and ensemble learning methods. The focus is based on data-driven prognostics in the development of failure prediction models and moreover, to compare the performance to select the best technique for the study. The methods selected in this study were influenced by the application in a specific environment, the quality of historical data, predictive performance and computational requirements for applicability of the method.

Random Forest

Recently, random forest (RF) as a machine learning technique, has been utilized for failure prediction in multiple operations of engineering due to its robust ability to work efficiently with large number of indicators, small samples and also its simplicity in interpretation of tree-based models (Timofeev, 2004; Verikas et al., 2011). RF uses a tree-based classifier (Breiman et al., 1984a), integrated with bootstrap aggregation (Breiman, 1996). The algorithm exploits the use of trees in the method. Each failure pattern is trained in isolation in a tree, and the predictions of all trees are combined to get a sophisticated result. Using this method, Frisk et al. (2014) have predicted failures in lead-acid batteries with training data obtained from heavy duty trucks containing heterogeneous data of 300 variables. RF performed well even with imbalanced and missing data from trucks.

Santur et al. (2016) proposed the use of RF in a study to predict the failure that may occur in railway tracks. Video image data of railway tracks was used to extract indicators, and the data was used to predict the different types of faults like scouring, breaking and deficient fasteners on tracks. The three different methods: principal component analysis, singular value decomposition, and random forest were compared with RF achieving the highest accuracy of 85%. In the study health assessment of bearings, Satishkumar and Sugumaran (2016) used vibration signals of bearing to develop a failure prediction model using RF. An accuracy of 95.64% was obtained by initially performing a feature selection using decision trees.

Gradient Boosting Method

Friedman (2002a) developed gradient boosting machine (GBM), a machine learning classifier to improve the predictive performance in classifications. GBM is highly appreciated because of its robustness to interactions, missing values, imbalance and outliers (Hastie et al., 2009). Furthermore, it automatically selects variables and leaves out irrelevant variables.

Kelvin (2016) analyzed the occurrence of unexpected failures in 1100 automated teller machines and 280 cash acceptance machine. Multiple models were used in the modeling step, and GBM resulted in the best model with an area under the curve of 87% to predict the failure. Furthermore, there were key challenges of data format and volume which was efficiently handled by GBM. Cerqueira et al. (2016) faced multiple challenges with the air pressure system components from Scania trucks. There were missing values, outliers and imbalance in class distribution. IN this situation, GBM was selected as the best method to handle the challenges and prevent overfitting. Two different algorithms, random forest and extreme gradient boosting, were used to model the failure of air pressure components. The best model (extreme gradient boosting) was generated with parameter tuning using ten-fold cross-validation and obtained sensitivity cost of 3750 and lowest deviance.

In contrast to using raw sensor variables as training data and traditional statistical methods in Hu et al. (2016) and Liu et al. (2016) to predict failure, this thesis aims to develop relevant leading and lagging indicators from sensor variables that contain useful information to explain the failure patterns in equipment. The best subset of indicators are selected for modeling using the random forest and gradient boosting methods. The best failure prediction model is selected based on their applicability to utilize the indicators developed from the equipment and metrics in performance assessment. The following chapter will discuss the methodology followed in this research along with a detailed explanation of all the methods used.

Chapter 3

Methodology

This chapter describes the methodology developed to attain the objectives of the thesis. Initially, indicators are developed utilizing the historical dataset, and they are used in the machine learning algorithm. The details of the development of indicators and working of machine learning are discussed in the section of this chapter in detail. The methodology has been divided into the following phases:

1. Data preprocessing
2. Indicator development
3. Model development and selection
4. Evaluation metrics

The data preprocessing phase consists of a novel approach to clean and prepare the data for modeling. Indicator development describes the techniques to develop the leading and lagging indicators. The model development phase consists of the mathematical formulation and techniques used in formulating the machine learning algorithms for failure prediction. The evaluation metrics phase describes the different evaluation strategies for analyzing the quality of the prediction results.

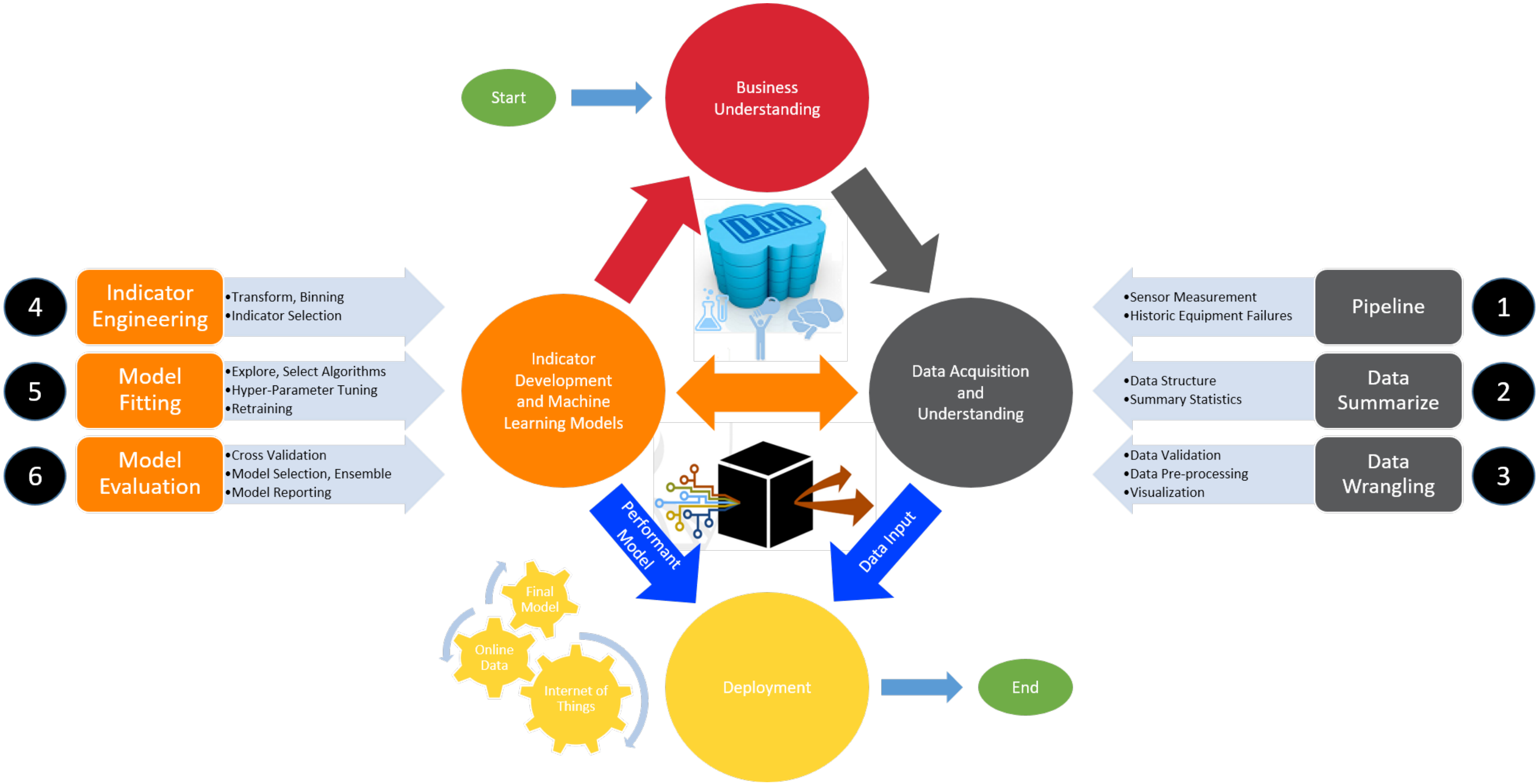


Figure 3.1: Methodology Flowchart

3.1 Data Preprocessing

Data preprocessing is the phase in the methodology where the data is prepared for further use in modeling. The challenge in this phase is the number of different activities it involves, including variable parsing, outliers checking and balancing the label variables. Due to an improvement in infrastructure, sensor measurement reliability increased linearly over the last decade when compared to antiquated, manual models of data processing. Therefore, preprocessing is a necessary task required for the model to predict failure accurately.

3.1.1 Data Organization

According to Wickham et al. (2014), the first step is to map the dataset to match its structure, which is performed by the following steps:

1. Each variable is allocated to its unique column
2. Each observation is allocated to its unique row
3. Each observational unit is assigned to its unique table

The above steps result in a standard structured data that is easier to read because the structured layout form paired values from different columns with the same row. However, it does not affect the further analysis which demands organization of the variables. The most efficient way of organizing the variables is by their importance in the analysis; the fixed time stamp variables should come first followed by the measured sensor measurements. Rows are then sorted based on the time stamp variables, cutting off ties with other fixed variables.

The data structuring step is followed by tidying the dataset read to the system. Initially, the dataset requires further preparation before jumping to the analysis part. Wickham et al. (2014) pointed out the most problems in messy datasets:

1. Column header has random names, not variable names.
2. Multiple information is stored in the same column

3. Variables are stored in both rows and columns

Commonly the messy dataset is arranged with values of the row and column rather than the variable names. This arrangement of data might sometimes prove useful for efficient storage and computation, but it complicates the analysis process failing to recognize the uniqueness of the variable. To improve the identification of variables, the values in the headers are converted to columns with a unique variable name. The names allocated to the columns based on the variable are easy to use and informative, maintaining the consistency throughout the dataset.

Often, after converting values to variable names, there are multiple variable names in the same column. The variable names in this format can be further broken down to form additional variables, which results in useful information. Preprocessing the data in this form extracts the hidden information resolving another problem in the messy dataset.

Variable values stored in both rows and columns is the most tricky part of the preprocessing. The information of a certain variable is spread all over the data across rows and columns making it difficult the analyze. The variable names are stored in the columns while the observation values are the headers. This issue is fixed by reconstructed the data to represent the variable names as the columns headers and each row represented by a sample.

3.1.2 Imbalanced Datasets

In the real world, datasets involving the health of equipment present imbalance in the dataset due to rare instances, i.e., failure, which makes it difficult to develop a model as there is little failure data compared to non-failure data to learn.

Developing a prediction algorithm with an imbalanced dataset results in a high accuracy of the model but the prediction classifier is biased towards majority class as the number of rows with minority class being small. For example, consider an imbalanced dataset with 20:80 ratio of minor to major label classes resulting in an accuracy of 95%. This model initially looks like the perfect model but the results could be deceiving since accuracy favors labels of major classes strongly while the minority classes are being misclassified.

There are a couple of methods to solve the imbalance problem in the dataset. The most efficient method found in the recent literature to re-balance the dataset is Synthetic Minority Oversampling Technique (SMOTE) Chawla et al. (2002). In this approach, the minority class is over-sampled by generating new examples to match the number of majority classes. SMOTE identifies the nearest neighbor of the minority class example using the Euclidean distance. A synthetic example is generated based on two examples and placed randomly somewhere between them.

Data preprocessing is an important phase in the methodology; good quality data improves the end results of the modeling. Like the usual saying garbage in and garbage out, a good quality data input to the model will lead to high-quality prediction output.

3.2 Indicators Development and Selection

Developing indicators is the process of using domain-specific experience and data insights to create features that help in the machine learning prediction. More than one indicator is used at a time in a prediction model; the more uncorrelated the indicators are from each other, the greater the information gain from the indicators. The set of indicators developed for the prediction model is referred to as feature space.

In the application of predicting failures using machine learning algorithms, the data acquired must be preprocessed before the development of feature space for effective results from the machine learning methods (Zhang et al., 2010). Every variable in the dataset is developed into a set of features that defines the feature space. This phase is not only limited to the development of indicators but also includes indicator selection which extracts the most important features affecting the trend of failure.

3.2.1 Indicator Development

Indicator development is based on different techniques that derive the required information from the raw sensor data in order to replace variables with new and better indicators. The development utilizes functions from various mathematical and modeling techniques to learn from the observed and measured sensor variables. Tan and Jiang (2013) has experimented with techniques like decomposing, filtering, translating and more to extract the hidden information in the sensor data. This thesis uses different techniques to develop indicators using the historical sensor measurements which signify strong statistical evidence between the trend in sensor variables and the occurrence of failures.

Binning Indicators

Binning is a process of transforming continuous variables into nominal or categorical indicators which help in creating density estimations of the measured values. When used correctly, the binning process can improve the simplicity of the model and decrease computation time (Kim and Han, 2000). There are multiple methods in binning but equal width interval binning was used because of its tendency to produce low discretization error. In equal width interval binning, the variables are sorted and divided into k equally sized intervals. For a variable x with the minimum and maximum values denoted by the x_{min} and x_{max}, the bin width is determined by:

$$\delta = \frac{x_{max} - x_{min}}{k} \tag{3.1}$$

The value k is determined by Sturges rule where $k = 1 + log_2(n)$, where n equals the length of the dataset (Yang and Webb, 2009). Then the method passes through the entire dataset once transforming each variable into binned indicators independently.

Time Domain Indicators

The trends of the measured variables can vary from time to time presenting non-stationarity in the data. According to Virili and Freisleben (2000), the time-dependent variables with trends can often cause complexity in modeling, leading to a decrease in the quality of predictions. Therefore, the transformed variables can display the clear change in patterns after removing the variability. In this section different time-dependent functions are used to extract the indicators in the time domain as represented in the equations 3.2 - 3.13.

$$\text{mean}: x_m = \frac{\sum_{n=1}^{N} x(n)}{N} \tag{3.2}$$

$$\text{standard deviation}: x_{std} = \sqrt{\frac{\sum_{n=1}^{N}(x(n) - x_m)^2}{N-1}} \tag{3.3}$$

$$\text{range}: x_r = max(x(n)) - min(x(n)) \tag{3.4}$$

$$\text{peak}: x_p = max|x(n)| \tag{3.5}$$

$$\text{root amplitude}: x_{ra} = \left(\frac{\sum_{n=1}^{N}\sqrt{|x(n)|}}{N}\right)^2 \tag{3.6}$$

$$\text{root mean square}: x_{rms} = \sqrt{\frac{\sum_{n=1}^{N} x(n)^2}{N}} \tag{3.7}$$

$$\text{skewness}: x_{ske} = \frac{\sum_{n=1}^{N}(x(n) - x_m)^3}{(N-1)x_{std}^3} \tag{3.8}$$

$$\text{kurtosis}: x_{kurt} = \frac{\sum_{n=1}^{N}(x(n) - x_m)^4}{(N-1)x_{std}^4} \tag{3.9}$$

$$\text{crest}: x_c = \frac{x_p}{x_{rms}} \tag{3.10}$$

$$\text{margin}: x_{ma} = \frac{x_p}{x_{ra}} \tag{3.11}$$

$$\text{shape}: x_{sha} = \frac{x_{rms}}{x_m} \tag{3.12}$$

$$\text{impulse factor}: x_{if} = \frac{x_p}{x_m} \tag{3.13}$$

where, n is the variable values from n = 1,2,3...,N and N is the length of the dataset.

Normalization Indicators

Normalization is the process of transforming the measured values to a common scale to deal with variables of different units and scales. Some of the machine learning methods are prone to outliers; normalization intends to compare the corresponding normalized values reducing the effects of exceptional values. The different functions used to normalize the variables are:

$$\text{unity-based normalization} : x_{un} = \frac{x(n) - min(x(n))}{max(x(n)) - min(x(n))} \tag{3.14}$$

$$\text{standard score} : x_{ss} = \frac{x(n) - x_m}{x_{std}} \tag{3.15}$$

$$\text{variation coefficient} : x_{vc} = \frac{x_{std}}{x_m} \tag{3.16}$$

Normalization is a tedious process as the future distribution of the variable is unknown, leading to the null maximum and minimum values of the variable. Therefore, normalization is to be performed after binning the variables which will eliminate outliers Gaber et al. (2005).

Frequency Domain Indicators

The frequency domain functions transform a given variable on each given frequency band. Finding the frequency at a particular point in time is irrelevant, but it is important to find how much of that particular frequency is in the variable. These frequency domain indicators are developed to find the distribution of frequency and filter the noise. Filtering in the time domain results in complexity and causes convolution. Therefore, the time-dependent variables are transformed into frequency domain, remove the noise with filtering and transform it to obtain the time-dependent indicators. Fourier transform functions are used in this case to develop the indicators.

Fourier transform treats the values in the variable as a point in the circular path and divides it into a group of cycles that hold the same information as the original variable (Bracewell and Bracewell, 1986). The properties of cycles are defined by strength, delay, and speed which is later used to recreate the original variable. Initially, the variables are passed through filters where each independent filter extract a cycle, i.e., the filters extracts all the values in the variables without leaving any observation. After the filtering, the original variable is obtained from the linear combination of the cycles. Fast Fourier Transform (FFT) algorithm is selected to perform the transformation and presented by the complexity of $O(n.log(n))$ operations. The FFT algorithm consists of two equations (Harris, 1978), where equation 3.17 represents the transformation from time domain to frequency domain and Equation 3.18 converts the frequency domain variables back to the original time-dependent variables.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i.2\pi kn/N} \tag{3.17}$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{-i.2\pi kn/N} \tag{3.18}$$

where,

X_k = amount of frequency k in the variable

N = number of samples

n = current sample, $n \in \{0, .., N-1\}$

k = current frequency

x_n = value of the variable at time n

The transformations result in the development of indicators like frequency peaks in the variable or the rate of change in the certain frequency.

Frequency Time Domain Indicators

The measured variables may contain hidden information in the frequency domain with continuously changing statistics with time; such information can be extracted using the time-frequency analysis. The methods analyze variables in time and frequency domain simultaneously to describe the behavior of variables over all time. The time-frequency distribution functions used to develop the indicators are described below.

The Short Time Fourier Transform (STFT) is related to the Fourier Transform introduced in the previous section. The STFT divides the variables into short segments of equal length based on the defined time window and analyzes each segment separately. The Fourier transform of these separate segments results in the sinusoidal frequency and phase content for that particular time window of the variable (Sejdić et al., 2009). The mathematical representation of STFT is (Allen and Rabiner, 1977):

$$X_n(e^{j\omega_k}) = \sum_{m=-\infty}^{\infty} w(n-m)x(m)e^{j\omega_k m} \tag{3.19}$$

where,

$x(m)$ = sample at time m

$w(m)$ = window size

$X_n(e^{j\omega}k)$ = STFT

ω_k = frequency value

The resolution of window size is fixed in the SFTF resulting in the trade-off between time and frequency resolution. A wide window size results in good frequency resolution but a poor time resolution, while a narrow window size does the opposite.

The wavelet transform is another indicator extraction method which overcomes SFTF weakness; they are based on small windows of time with limited duration. The fixed resolution in SFTF is replaced with a continuously changing resolution in both frequency and time domain in wavelet transformation. Also, it produces the shifted and scaled form of the original variable. The continuous wavelet transform which is the most common method in wavelet transformations is used. In continuous wavelet transform (CWT), the resolution is continuously changing with each time window to fit the variable's frequency and time. The equation of CWT is represented by:

$$Wt(a,b) = \int_{-a}^{a} x(t)\mathit{\Psi}^*_{(a,b)}(t)dt \tag{3.20}$$

where,

x(t) = time function of the variable

$\mathit{\Psi}^*_{(a,b)}(t)$ = continuous function in time and frequency domain

The above equation divides the variable with time and frequency domain information into wavelet decomposition coefficients. The coefficients or a combination of them are used as indicators.

Basic Expansion Indicators

The idea behind basic expansions is to create all combination of variables to develop new indicators. The indicators developed using basic expansions will help extract the non-linear behavior of variables. The different methods in basic expansions include the following:

- Logarithmic transformation
- Dividing variables to create ratios
- Linear and polynomial interaction
- linear and non-linear combination

Rolling Aggregate Indicators

The rolling aggregate is a method in which the lagged indicators are created by selecting a fixed window size and calculating the aggregate measures. The rolling aggregates can be computed on several different time domain indicators such as mean, standard deviation, peak, etc. Different lag window sizes are selected to create several indicators to explain the short-term and long-term history of the variables.

Summary

The measure of the deviation of an equipment from the normal working condition is dynamic and influenced by several internal and external factors. Hence, to accurately learn the degradation patterns and correctly predict the failures in equipment, a total of 346 indicators are developed as discussed in the section. Table 3.1 gives a summary of all the essential independent indicators reconstructed from the original equipment sensor data.

3.2.2 Indicator Selection

Following the indicator development phase, all indicators of the dataset are available with a high-dimensional feature space. Indicator selection is performed for multiple different

Table 3.1: Summary of Failure Indicators Developed

Indicator Description	Count	Indicator Category
Failure Count	4	Binning
Mean	14	Time Domain
Standard Deviation	14	Time Domain
Range	14	Time Domain
Peak	14	Time Domain
Root Amplitude	14	Time Domain
Root Mean Square	14	Time Domain
Skewness	14	Time Domain
Kurtosis	14	Time Domain
Crest	14	Time Domain
Margin	14	Time Domain
Shape	14	Time Domain
Impulse Factor	14	Time Domain
Unity-based Normalization	7	Normalization
Standard Score	7	Normalization
Variation Coefficient	7	Normalization
Fourier Transformation	7	Frequency Domain
Discrete Wavelet Transforms	5	Frequency Time Domain
Interactions	120	Basic Expansions
Date Features	7	Basic Expansions
Time of Last Repair	14	Rolling Average

reasons. First, selecting the most important indicators in the feature space will increase the ability of the prediction model to look at the most relevant data explaining the event patterns and hence resulting in higher accuracy. Second, selecting the important indicators reduces the feature space resulting in lower computation time and power. Third, smaller feature space leads to fewer indicators which simplifies the interpretation of results. Fourth, indicator selection avoids overfitting the data during modeling.

Indicator selection methods are divided into three types (Blum and Langley, 1997; Das, 2001):

(i) The *filter* method select indicators from the data independent of the classifier.

(ii) The *wrapper* method uses any statistical learning algorithm of interest to evaluate the useful indicators.

(iii) The *embedded* method select indicators that help improve the construction and accuracy of the classifier.

Filter method is computationally efficient but they select indicators which are generic, and does not take into account the chosen learning algorithm in the selection process. In turn ignoring the bias of the classifier and reducing the fragment of the prediction model. On the other hand, wrapper methods are reasonable as they select the indicators by training and testing based on the hypothesis of a predefined classifier and detect indicators dependencies. However, this is computationally expensive with a broad range of indicators.

The embedded indicator selection method is advantageous in this thesis case as there are many indicators with a big feature space and requires interaction with the classification algorithm. Embedded method performs computationally better than wrapper method and utilizes the hypothesis of the learning classifier, which is not offered by the filter method (Liu and Yu, 2005). The best technique for the embedded method is a random forest model (see section 3.3.2). A random forest consists of many classifiers in which each new forest is trained by selecting only those indicators that are the most important (Breiman, 2001). Multiple such forests are created with a different set of indicators to increase the probability that only the most important indicators are selected. After comparing, the forest with the lowest error rate and the most accurate contribution is selected as the best subset of indicators.

3.3 Model Selection

The initial selection of machine learning algorithm for the prediction of failure in equipment depends on the type of input data and output expected, as shown in Figure 3.2 . In this case, supervised machine learning is selected as the training data have failure outcome as the label variable. Furthermore, the output of the failure prediction is a class variable which categorizes the problem as a binary class classification method. Following the categorization of the problem, the algorithms that are applicable and practical to implement are identified.

There are numerous algorithms in classification to solve each task. But instead of selecting a specific algorithm initially, it is preferred to look at the data and start minimizing the

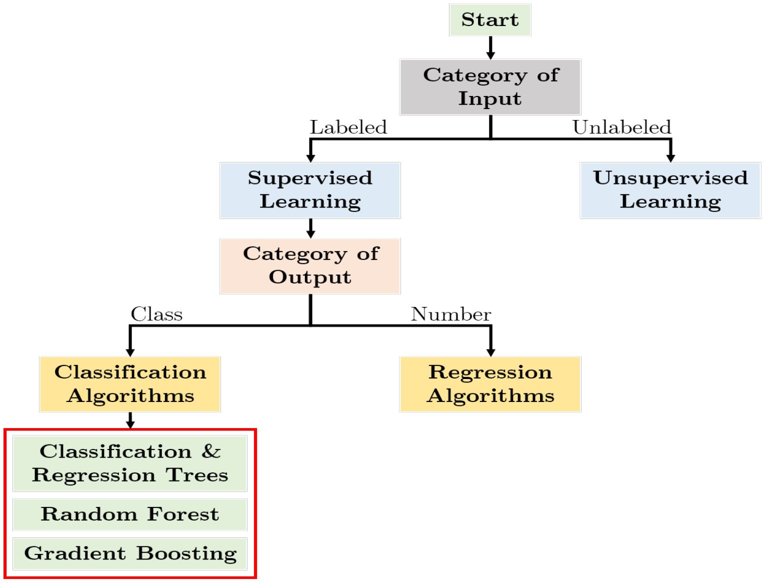


Figure 3.2: Guidelines for model selection

list of algorithms to something that makes sense with the data (Hastie et al., 2009). The exploration of algorithms is started with building models with naive tuning of parameters to just understand which model behaves better. Based on the result, the list of algorithms is shortened and the top four competing algorithms are chosen. Then the real modeling starts with parameters tuning and regularization to form a final model with the indicators selected to predict failure. These machine learning methods are explored in further detail in the following sections.

3.3.1 Classification and Regression Trees (CART)

Classification trees are used when the labels are provided in the data based on some derived rules. Classification and Regression Trees are constructed on a binary decision tree by obtaining child nodes from the parent node that contains the training sample (Breiman et al., 1984b). The following CART methodology is explained by Timofeev (2004) using t_p as the parent node and t_l, t_r as the left and right child nodes of the parent node t_p. The classification tree follows the splitting rule which splits the parent node into smaller parts with maximum homogeneity. The impurity function $i(t)$ is the measure of the maximum homogeneity of child nodes. The CART applies maximization problem at each split in the nodes:

$$\underset{x_j \leq x_j^R, j=1,\ldots,M}{argmax} \left[i(t_p) - P_l i(t_l) - P_r i(t_r) \right] \tag{3.21}$$

where X is a matrix with M number of variables x_j and K classes, x_j^R is the best splitting value of variable x_j and we assume that P_l, P_r are the probabilities of the left and right nodes.

There are many impurity functions in practice, but the most commonly used impurity function is the Gini index (Breiman et al., 1984b) and is represented by:

$$i(t) = \sum_{k \neq l} p(k|t) p(l|t) \tag{3.22}$$

where k, l 1,...,K are the index of class and $p(k|t)$ is the conditional probability of class k at node t.

Gini index can be obtained by applying the impurity function 3.22 to the maximization problem 3.21:

$$\underset{x_j \leq x_j^R, j=1,\ldots,M}{argmax} \left[-\sum_{k=1}^{K} p^2(k|t_p) + P_l \sum_{k=1}^{K} p^2(k|t_l) + P_r \sum_{k=1}^{K} p^2(k|t_r) \right] \tag{3.23}$$

With the help of equation 3.23 maximum tree is produced where the nodes were separated up to the last observation. The tree structure built may be highly complex and consist of multiple layers. Therefore, before applying the classification tree on testing data to validate the model, the trees must be optimized by choosing the right size tree by cutting off the insignificant subtrees.

There are two pruning algorithms in practice: optimization by a number of points in each node and cross-validation. In case of optimization, the splitting of the parent node is stopped when the number of observations is less that the predefined number, usually 10% of the learning sample size. Similarly, cross-validation finds the optimal point between the tree complexity and misclassification error which is obtained through cost-complexity function.

With the help of the above parameters, each of the observations in the sample will get to one of the nodes in the tree. Later, this value will be allocated to the dominant label value.

3.3.2 Random Forest

Random forest is a method in the ensemble learning (Ho, 1995) for classification and regression that produces multiple decision trees and outputs the classes. Breiman (2001) proposed random forests, which adds an extra layer by generating random components into the tree, which produces a distribution of trees. Moreover creating a distribution of predicted values for each sample label.

When different bootstrap samples are selected, the structure of the tree may look similar due to the fundamental relationship but they all look the same at the beginning and are correlated to each other. Therefore, Dietterich (2000) developed a way of random split selection, where at each split of the tree a random subset of top predictors are selected to build the next tree. Then each model in the ensemble casts a vote for the prediction probability of a new sample and the proportion of votes from the models is the predicted probability.

Algorithm 1: Random Forest Algorithm

```
Select the number of models to build, M
for $i$ = 1 to $M$ do
    Generate a bootstrap sample of the original data
    Train a tree model $T_i$ on this sample
    for each split do
        Randomly select $k$ ($< P$) of the original predictors
        Select the best predictor among the $k$ predictors and partition the data
    end
    Use typical tree model stopping criteria to determine when a tree is complete (but do not prune)
end
$h(\cdot) = \frac{1}{M} \sum_{i=1}^{M} T_i(\cdot)$
```

The tuning parameter that random forest uses to choose the number of randomly selected predictors is commonly denoted to as m_{try} which re-correlates the trees in the forest. Breiman

(2001) recommended setting m_{try} to the square root of a number of predictors, while m_{try} can also be selected optimally by using cross-validation for larger forests.

A general random forest algorithm (Kuhn and Johnson, 2013) to develop a tree-based model for classification can be implemented as shown in Algorithm 1. A random forest model is proven to perform accurately and computationally efficient compared to bagging and boosting. The linear combination of many different classifiers reduces the variance of the ensemble learner relative to the individual classifier, while random forest achieves this variance reduction by combing many complex classifiers under the ensemble. Since, each classifier is selected independently of all other classifiers, random forest exhibits improvement in error rates and is robust to noisy data.

3.3.3 Stochastic Gradient Boosting

Stochastic gradient boosting is one of the most efficient algorithms for modeling the failure of machines. Friedman (2001) utilized the boosting statistical framework to develop a simple and highly adaptable efficient algorithm called "gradient boosting machines." The algorithm works with a given loss function and a weak learner before minimizing the loss function by finding a suitable additive model. The boosting starts with the best estimate of the response sample and with the help of the initial estimates, gradient (i.e., residuals) is calculated, and the model is fit to the gradients to reduce the loss function. The fitting model is added to the previous model and continues during a specified number of iterations.

Since boosting works better with weak learners, any technique which uses tuning parameters can make a weak learner. The learner is an excellent match for boosting trees because of many reasons. Foremost, trees have the ability to demonstrate the ability of a weak learner by restricting their depth. These trees can be added together in a classification model to obtain better predictions. Lastly, trees structure can be produced rapidly. Therefore, the predictions from these trees can be aggregated which makes it perfectly suitable to be used in an additive modeling process. The algorithm for gradient boosting (Friedman, 2002b) is shown in Algorithm 2 as:

Algorithm 2: Gradient Boosting Algorithm

```
Initialized all predictions to the sample log-odds: $f_i^{(0)} = log\frac{\hat{p}}{1-\hat{p}}$
for interation j = 1,...,M do
    Compute the residual (i.e. gradient) $z_i = y_i - \hat{p}_i$
    Randomly sample the training data
    Train a tree model on the random subset using the residuals as the outcome
    Compute the terminal node estimates of the Pearson residuals: $r_i = \frac{1/n\sum_i^n(y_i-\hat{p}_i)}{1/n\sum_i^n\hat{p}_i(1-\hat{p}_i)}$
    Update the current model using $f_i = f_i + \lambda f_i^{(j)}$
end
```

Gradient boosting works with two tuning parameters: tree depth and number of iterations. Tree depth is also known as the interaction depth Kuhn and Johnson (2013), as each split of the parent node can be a higher level interaction term with the split in the previous tree predictors. From Algorithm 2 we can see that it has similar steps as the random forest where the trees are used as the base learners, and the final prediction is based on ensemble ranking of models. However, the ensemble created in gradient boosting differs compared to other methods. The models are dependent on the previously structured trees, in which the depth of the tree is minimized using the tuning parameter and contribute unevenly to the final model.

Gradient boosting machine can be liable to over-fitting because even the weakest learner moves towards optimally fitting the gradient (Friedman, 2001). This causes the weak learner to use greedy strategy to select an optimal solution at the current stage. But the learner fails to find an optimal global solution leading to over-fitting the training sample. A way to overcome the greediness is to engage shrinkage parameter in controlling the learning process. The shrinkage parameter can be regularized by adding a penalty to the sum of squared errors (SSE)if the estimates become large (Hoerl and Kennard, 1970) and is given by:

$$\mathrm{SSE}_{L_2} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{P}\beta_j^2 \tag{3.24}$$

The equation 3.24 comes in effect when there is a decrease in SSE; the parameter estimates shrink towards 0 while increasing the λ. The regularization parameter 3.24 can be added to the final step of the loop in Algorithm 2. In gradient boosting only a partial of the predicted results from current iteration is contributed towards the previous iteration's results. This ratio is known as the learning rate or denoted by the symbol λ and varies between 0 and 1.

The method also provides variable importance in classification. The importance is calculated by recognizing the improvement of each predictor at the split within each tree in the ensemble. The values are aggregated and averaged across the different ensemble models.

3.3.4 eXtreme Gradient Boosting

XGBoost (Chen and Guestrin, 2016), also known as eXtreme gradient boosting, is a famous scalable machine learning algorithm for tree boosting. The xgboost method is highly efficient due to its property of scalability and runs almost ten times faster compared to the existing popular methods on a single machine. It also handles billions of rows when running in parallelism. The scalability of xgboost is due to several algorithm optimizations and innovations (Chen and Guestrin, 2016) that include novel tree learning algorithm and a quantile weighted procedure which handles instance weights of different trees in the ensemble.

XGBoost modeling is based on least-square residual fitting with a squared-loss objective function which follows the sample principle as stochastic gradient boosting from previous section 3.3.3. However, the upgrade is located in the modeling framework, improving the boosting method that was originally developed by Friedman et al. (2000) to get better accuracy by controlling over-fitting through regularized model formalization.

The changes that Chen and Guestrin (2016) made in the regularized objective are explained as following with n samples and m variables including K additive functions.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \qquad f_k \in \mathcal{F}, \tag{3.25}$$

where $\mathcal{F} = \left\{ f(x) = w_{q(x)} \right\} (q : \mathbb{R}^m \longrightarrow T, w \in \mathbb{R}^T)$ is the space of functions containing all the classification trees. In Equation 3.25, q represents the structure of trees, T is the number of leaves, f_k corresponds to an independent tree structure q and leaf weights w. The decision rules are used to classify the trees into leaves and then calculate the final prediction by summation of leaf scores. To obtain the set of function, the following regularized objective is minimized.

$$\begin{aligned} \mathcal{L}(\phi) &= \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \\ where \quad \Omega(f) &= \gamma T + \frac{1}{2}\lambda \left\| w \right\|^2 \end{aligned} \tag{3.26}$$

In equation 3.26, l is the residual function of $\hat{y}_i$ and y_i. The second term Ω is the penalty term which works on the complexity of the model. Also, it helps smooth the weights to avoid over-fitting. The regularized objective will select the model with simple and best predictive functions. The Xgboost methodology developed by Chen and Guestrin (2016) is shown in Algorithm 3.

Algorithm 3: Xgboost Algorithm

```
Add a new tree in each iteration
Beginning of each iteration, calculate
  $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}), \quad h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$
Use the statistics to greedily grow a tree $f_t(x)$
  $Obj = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T$
Add $f_t(x)$ to the model $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$
      • Usually, instead we do $y^{(t)} = y^{(t-1)} + \in f_t(x_i)$
      • $\in$ is called step-size or shrinkage, usually set around 0.1
      • This means we don't do full optimization in each step and reserve
        chance for future rounds, it helps prevent over-fitting
```

Most of the existing machine learning tree algorithms require defined procedures, but Xgboost is a tool motivated by formal principle. Xgboost provides automation of optimization and regularization. Also, it overworks to obtain a scalable and accurate prediction.

3.4 Model Tuning

Three different algorithms are selected for the prediction of failures in equipment. Each algorithm has one or more parameters that can alter the working of the model and different ways of training the model on the dataset. These parameters ultimately affect the accuracy of the model. Therefore, model tuning is used to identify the optimal set of parameters to obtain the highest accuracy for the given dataset.

Each model is tuned and optimized to obtain the lowest test error. Time-series based cross-validation is used on the timestamped dataset considered in this thesis to find the optimal set of parameters as suggested by Bergmeir et al. (2015).

In cross-validation, the dataset is divided into two parts: training and testing dataset. The training set is used to adjust the parameters while the testing set is used to confirm the predictive performance of the model. During the model training process, the training set is further split into 10 subsets or folds of equal size where the first two folds are training set and third fold is validation set. The step is repeated by adding the succeeding subset of data to the training set while the data in fourth fold is used as validation set. A step-by-step procedure of the cross-validation process can be seen in Figure 3.3.

Each of the cross-validation steps produces a validation error, which acts a measure to describe the performance of the model. The average of these validation errors is considered to compare the machine learning models.

3.4.1 Tuning of Methods Used

Each machine learning algorithms use different parameters to improve the predictive accuracy of the classification. The parameters used for tuning the methods are described in Table 3.2. The methods are tuned using the caret package (Kuhn, 2008) based in R software. Using a 10-fold cross-validation with three repeats of the process as shown in Figure 3.4, the optimal value of tuning parameters is selected to obtain the highest accuracy from the chosen machine learning methods. These parameters are again plugged into the method and

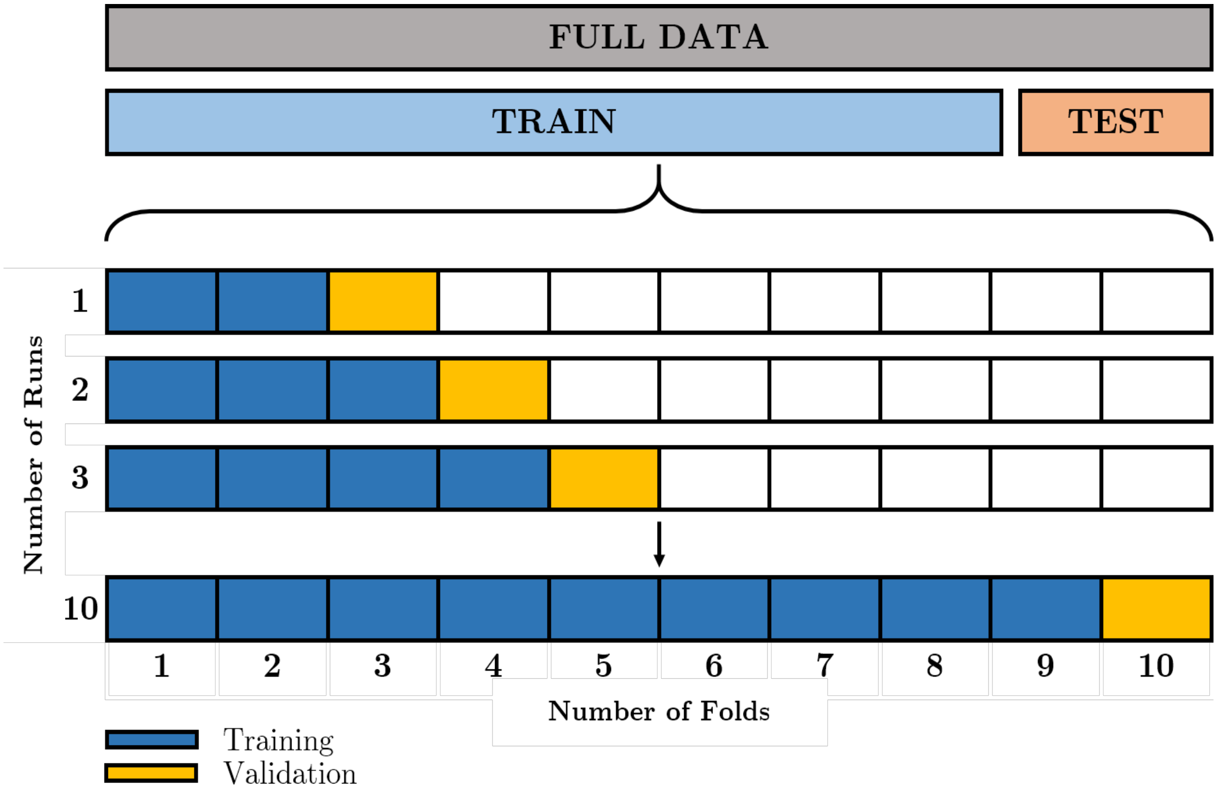


Figure 3.3: 10-fold cross validation process for time series data

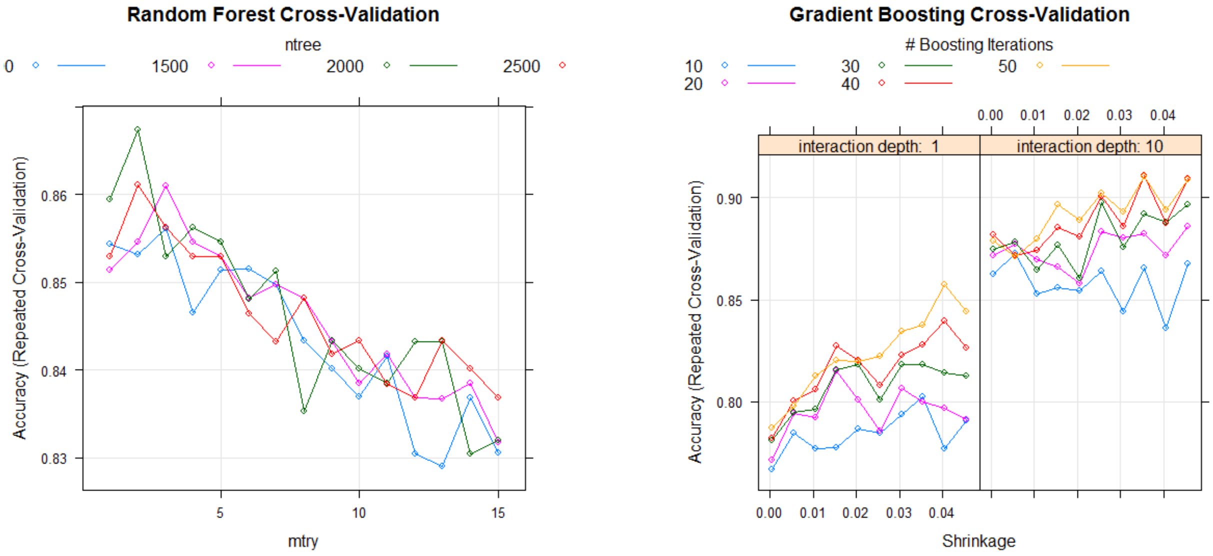
performed on the validation dataset to verify the improvement in accuracy. The methods with optimal parameters is then used on the test dataset to obtain failure time predictions.

3.5 Evaluation Metrics

The output of the machine learning failure prediction is obtained in the form of a binary decision whether the equipment status is failure-prone or not. The selected machine learning methods is first fit on training data to tune the model, then the trained model is used to prediction failures in equipments using the test data. The predictive accuracy of the model is evaluated using contingency table from which multiple metrics are derived. The decision rules setup in the algorithm is based on the tuning parameters used such as number of trees and number of variables at each split. While distribution estimates can be derived from the data set, finding optimal tuning parameters is specific to the method and is quite difficult.

Table 3.2: Tuning parameters for the methods used on training dataset

Method	Tuning Parameters
CART	cp - complexity parameter
Random Forest	n.tree - number of trees to grow mtry - number of variables at each split
Stochastic Gradient Boosting	n.tree - number of trees to fit interaction.depth - maximum depth of variable interaction shrinkage - parameter to control tree expansion n.minobsinnode - minimum number of observations in node
eXtreme Gradient Boosting	max_depth - maximum depth of a tree eta - shrinks the variable weights gamma - loss reduction parameter colsample_bytree - subsample ratio of columns min_child_weight - minimum sum of instance weight subsample - ratio of variables to train model



(a) Random forest uses two tuning parameters n.trees and mtry

(b) Gradient boosting uses four tuning parameters n.trees, interaction depth, shrinkage and n.minobsinnode

Figure 3.4: Tuning of random forest and gradient boosting methods

Indeed, the parameters in the failure prediction methods can be tuned to obtain better results than the other failure prediction method. Therefore, for this reason, multiple classification metrics are used to evaluate the performance of the failure prediction approaches.

3.5.1 Confusion Matrix

A confusion matrix, also known as an error matrix (Stehman, 1997) or contingency table, is a visual representation of the performance of a supervised machine method as shown in Table 3.3 . It presents the difference between incorrectly and correctly classified labels which will help for demonstrating the accuracy of the model. Consider a matrix of size $n \times n$ where n is the number of classes. Each column in the matrix represents the label counts from the testing data while each row represents the predicted label counts.

Table 3.3: Confusion Matrix

		True Class		
		Failure	**Non-Failure**	**Sum**
Prediction Class	**Failure**	True Positive (TP)	False Positive (FP)	Positives
	Non-Failure	False Negative (FN)	True Negative (TN)	Negatives
	Sum	Prediction	Non-Failures	Total

From confusion matrix, a different conclusion can be derived for the classification results of the prediction algorithm. The misinterpreted results are represented in two distinct ways. False negatives occur when the prediction algorithm predicts a failure but the machine, in reality, is working normally whereas false negative suggests that the machine is working normally, but in reality, it is failure prone. In failure prediction, the false positives, and false negatives are referred to as false and missing warnings. Similarly, the correctly classified results is presented in two different cases: true positive occurs with the prediction correctly classified as failure-prone, while the true negatives occur when the prediction are properly classified as not failure-prone.

From the confusion matrix, various metrics as shown in Table 3.4 can be derived to compare different prediction models. The metrics shown in the table is further explained in the next sections.

Table 3.4: Metrics derived from confusion matrix (Table 3.3)

Metric Name	Symbol	Formula
Accuracy	acc	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	p	$\frac{TP}{TP+FP}$

3.5.2 Accuracy

In classification methods, the performance is evaluated based on the accuracy metric, which is the ratio of a number of correct prediction to the number of all predictions. The accuracy performs as:

$$\text{Accuracy} = \frac{\text{true positives} + \text{true negatives}}{\text{true positives} + \text{false positives} + \text{false negatives} + \text{true negatives}} \tag{3.27}$$

Although accuracy is the most widely used metric, it is an inappropriate metric for failure measure. This is due to the fact that accuracy is not very descriptive when the failures are rare events. The model can be highly accurate but may still lack in the power of predicting the failures.

Consider an imbalanced training data set with 99% non-failures and 1% failures; the classifier will result in an excellent accuracy by correctly classifying the non-failures. In this case, the classifier may identify all non-failures and obtain high levels of accuracy without taking randomness into account. Rather, metrics such as precision and recall measures the correct number of failures predicted with imbalanced data, making them appropriate to be used in the failure prediction evaluation.

3.5.3 Precision and Recall

In recent days, the popularity of precision and recall are increasing with the classifier performance, which was originally introduced by van Rijsbergen (1979). Precision can be defined as the ratio of a number of correctly classified failures to the total number of events that are classified as failures. Recall is the ratio of correctly classified failures to the total number of true failures in the data set:

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} = \frac{\text{correct warning}}{\text{failure warnings}} \tag{3.28}$$

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} = \frac{\text{correct warning}}{\text{total failures}} \tag{3.29}$$

Based on the imbalanced data mentioned above, precision would be the failure warning generated as such, while recall would be a number of failure warnings correctly identified failures from all the labels in the dataset. Precision and recall are dependent on each other, each metric achieves better at the expense of other metric. A perfect failure prediction model will result in precision and recall of 1.0. Since the evaluation metrics has overcome the extreme imbalance in the data set, precision and recall are suitable where the failures are much more rare events compared to the non-failures.

3.5.4 F-score

Precision is inversely proportional to recall, i.e., improving precision reduces recall, i.e., reducing the false positives increases the false negatives. F-score incorporates the effect of both precision and recall. The F-score is a more stable measure of precision and recall in a single measurement which is shown in the equation below.

$$\text{F-score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{3.30}$$

Chapter 4

Results and Application

This chapter presents the case study and the results of the implementation of the failure prediction model. The case study was developed to test the predictive accuracy of the model using the asset performance data from Meridium (2017) and validated using benchmarked dataset from NASA's Prognostics Data Repository (NASA, 2008). The results are presented with a brief explanation of the dataset along with exploratory analysis. In the case study, indicators were developed and filtered to select the best subset which was introduced into a predictive model.

4.1 Asset Performance Data

The database of centrifugal pump records was obtained from Meridium, an asset performance management organization. The database consists of information from three tables: equipment records (EQ), work history data (WH) and sensor reading records (READINGS). The equipment records table consists of 906 centrifugal pumps with 10569 failure and non-failure recorded events, each with sensor measurements of about five to ten years. The performance of centrifugal pump is obtained from responses of seven sensed variables as shown in Figure 4.1. The equipment and sensor records consist of the date and time stamps; recorded corresponding to any event in the equipment, while the sensor measurements are recorded at an hourly interval for every equipment. The sensitive information which can identify the equipment are removed as requested by Meridium to maintain confidentiality. The variables extracted from the data tables are listed below in Table 4.1.

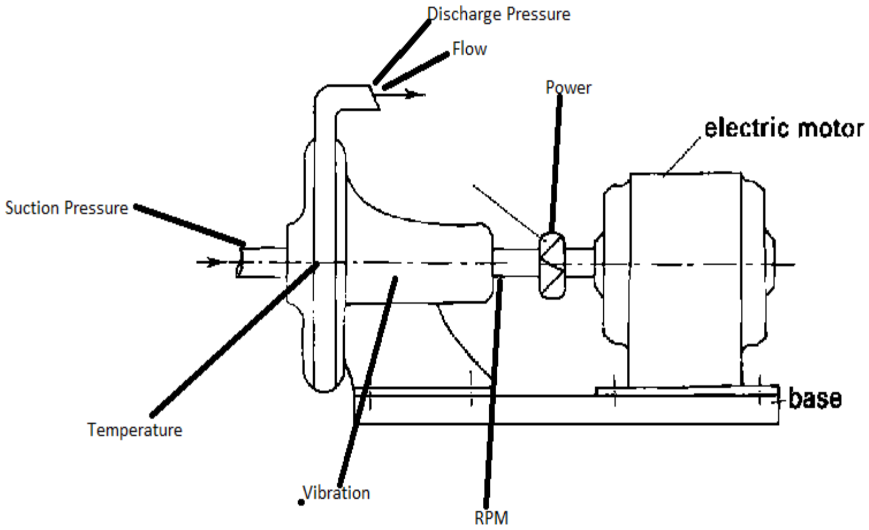


Figure 4.1: Sensors installed in centrifugal pump to monitor performance, adopted from Meridium (2017)

4.1.1 Preprocessing

Preprocessing is an important step to obtain consistent and outliers free data to input into the prediction model. Since the failure records were entered by the operators, the significant number of missing values, duplicate values, and inaccurate format of entry was found. This step is implemented to obtain good prediction results with high performance. Based on the specific assumptions considered in Chapter 1, the following steps are taken to make the data more accessible.

1. The variable *equipment type description* contains the equipment type and along with the manufacturer name encoded as a number. The manufacturer name may be useful in a general evaluation, but offer little importance to predict failure. Hence, the variable was parsed, and only the equipment type was considered.

Table 4.1: Variable description

Type of Information	Table Name	Description
Equipment Type Description	ER	Defines equipment category
Equipment Start Date	ER	Date-time when equipment started operating
Event Type	WH	Defines the event type like repair, preventive or predictive maintenance
Event Start Date	WH	Date-time when the event started
Maintenance Start Date	WH	Date-time when the maintenance started
Maintenance Completion Date	WH	Date-time when the maintenance ended
Equipment Available Date	WH	Date-time when the equipment is available after maintenance
Internal ID	WH	Unique identification number of equipments
Breakdown Indicator	WH	Status whether the equipment failed or not
Maintenance Order	WH	Type of maintenance performed: corrective, preventive or predictive
Maintenance Priority	WH	Type of maintenance assigned based on severity of event
Sensor Timestamp	READINGS	Date-time when the sensed variables are recorded; hourly interval
Suction Pressure	READINGS	Pounds per square inch
Discharge Pressure	READINGS	Pounds per square inch
Flow	READINGS	Gallons per minute
Vibration	READINGS	Millimeters per second squared
Temperature	READINGS	Degrees Fahrenheit
RPM	READINGS	Revolutions per minute
Power	READINGS	Horse power

2. After merging EQ and WH, 4 out of 11 variables had missing values. These equipment records were not considered in further analysis.

3. To ensure that the events are not recorded before the equipment start date, the following conditions were tested:

$$\text{Equipment Start Date} \leq \text{Event Start Date} \leq \text{Maintenance Start Date}$$

$$\text{Maintenance Start Date} \leq \text{Maintenance Completion Date} \leq \text{Equipment Available Date}$$

4. The variable names were encoded with special character and numbers which made it difficult to understand. The names were changed to match the description of data in the column.

5. The dates were specified in different formats in date-time columns. To make it consistent with the analysis, it was changed to MM:DD:YY HH:SS (month:day:year hours:seconds).

6. The imbalance due to a smaller number of failure occurrences leads to a higher number of non-failure records compared to failure records in the dataset. The failure records are increased by oversampling using SMOTE (Chawla et al., 2002) to balance the dataset.

After the preprocessing step, the tables EQ and WH were combined to obtain 11 variables with 10450 records. The number of event samples was reduced from 10569 to 10450 records after removing the rows with missing values and running the above-mentioned conditions. A general visualization of sensors can be seen in Figure 4.2.

4.1.2 Preliminary Modeling

Once the dataset is preprocessed, the historical sensor variables were used to create the preliminary failure prediction models using the selected machine learning methods (see Section 3.3). The sensor variables employed in the current evaluation will act as a benchmark to check if the model performs better after developing additional indicators with the historical data. For evaluation purpose, the model is created using only the training data, and then the accuracy of the failure predictions is checked using the test data. Failure of equipment

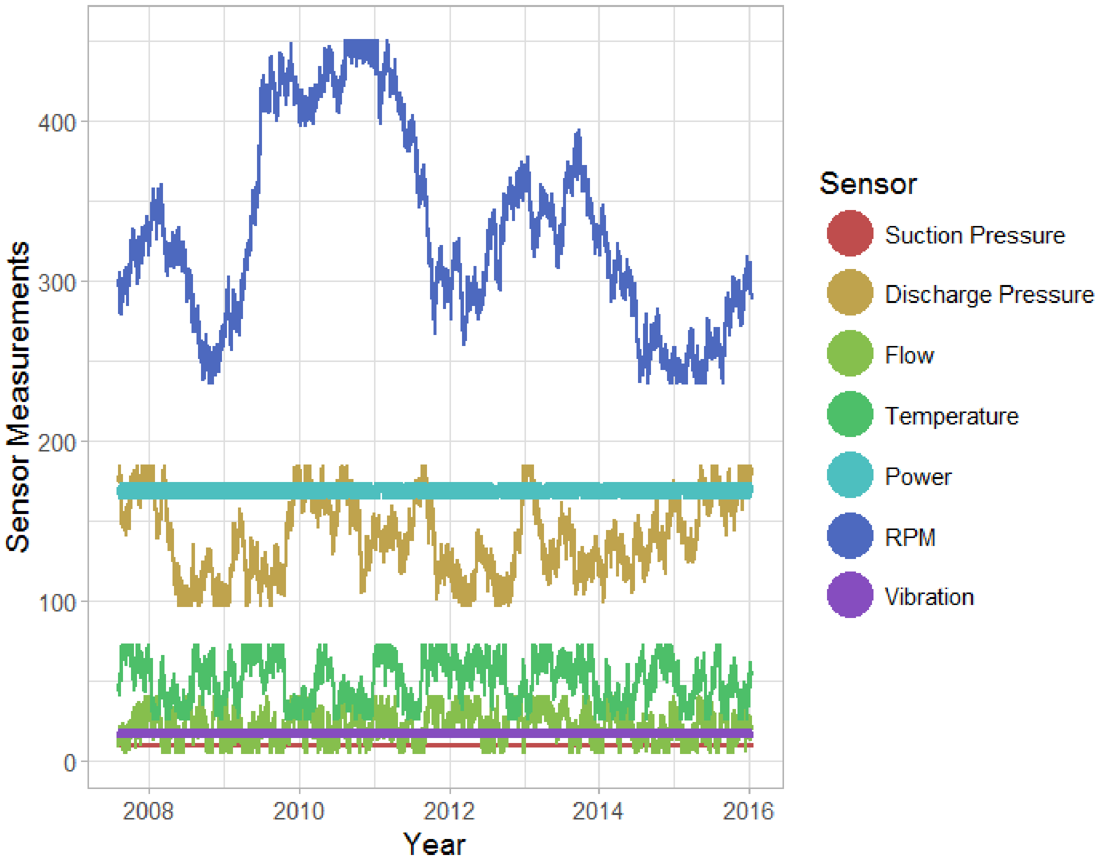
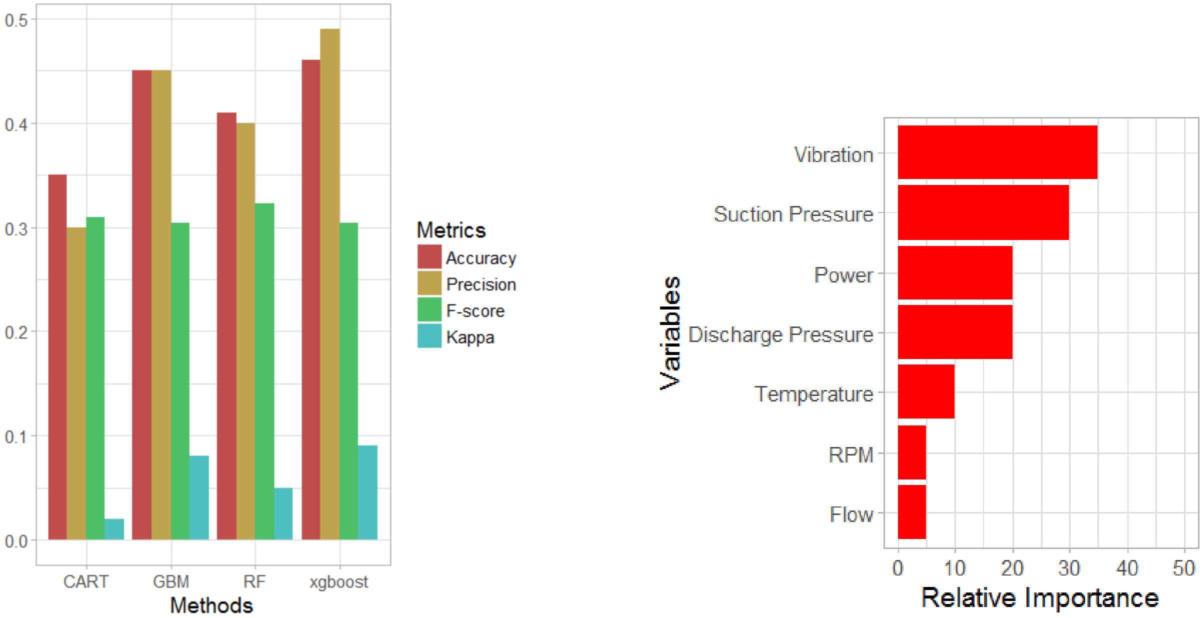


Figure 4.2: Centrifugal pump sensor measurements

were recorded in the testing data, so the failure predictions of the model was compared with the recorded event to check the performance.

The first 80% of the historical sensor measurements was used as the training data while the remaining 20% of the sensor measurements was used as test data. The preliminary model was created with default tuning parameters in the model. The objective of the preliminary model was to train the methods without spending much time and analyze the behavior of the dataset before developing the model for a production environment. The testing data was used to evaluate the performance of the model as there was no need to wait for the next failure to occur in the equipment.

Figure 4.3 shows the various evaluation metric values for the preliminary models (Classification and Regression Trees, random forest, stochastic gradient boosting and



(a) Evaluation of model using accuracy, precision, f-score and kappa

(b) Sensor variable importance plot using random forest

Figure 4.3: Performance of preliminary model

eXtreme gradient boosting). From the figure, the performance of the eXtreme gradient boosting is the best with 46% accuracy. However, a model with such low accuracy cannot be used in the production environment as it fails to predict the future failure in equipment. The variable importance plot identifies the relative importance of each variable contributing towards the accuracy of the model can be seen in Figure 4.3. The vibration sensor contributes the most for the prediction of failure. However, the contribution of the sensor variables is not much. The information comprised in the historical sensor variables is adding little value to help improve the accuracy of the model. Therefore, to obtain better information from the historical sensor variables, failure indicators should be developed using domain-specific knowledge.

4.1.3 Indicator Development and Selection

Additional indicators were developed from the historical sensor variables to improve the information contributed towards the prediction of failures without losing the original information. The measurements of seven sensor variables shown in Figure 4.2 were not directly useful in the prediction of failure as some of the sensors did not show any change in patterns during the failure of equipment. In other cases, the variables may be correlated proving it difficult to develop a good failure prediction model leading to a bad model with over-fitting or under-fitting the data.

A time-stamped data point in the dataset represents the working condition of the equipment at a point in time along with a label of either failure or non-failure. As seen in the methodology section 3.2, a recorded measurement from the sensor can be represented in multiple ways. Multiple functions were applied to the 7 historical sensor variables to create 346 failure indicators which were represented in vector form (shown in Figure 4.4).

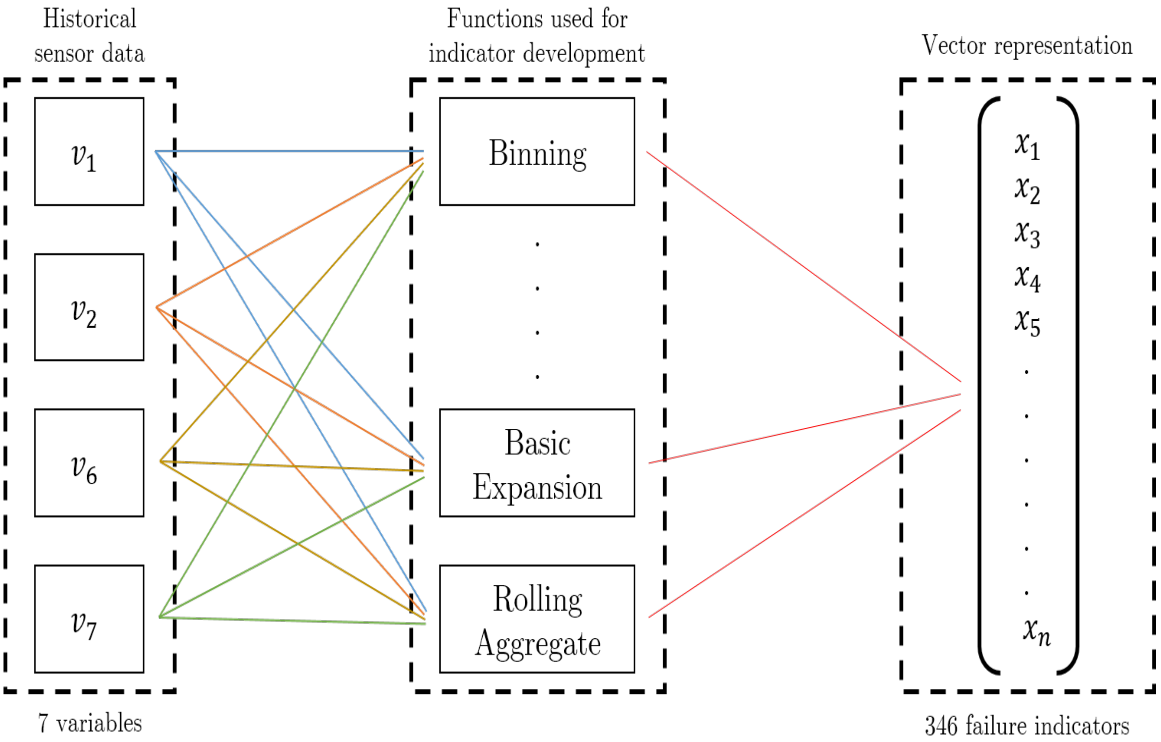


Figure 4.4: The process of creating failure indicators from the historical sensor data in the database

The dataset contains 346 unique failure indicators after the indicator development stage. Some indicators in the feature space had zero or near zero-variance which causes prediction model instability. Therefore, the best subset of failure indicators were selected to improve

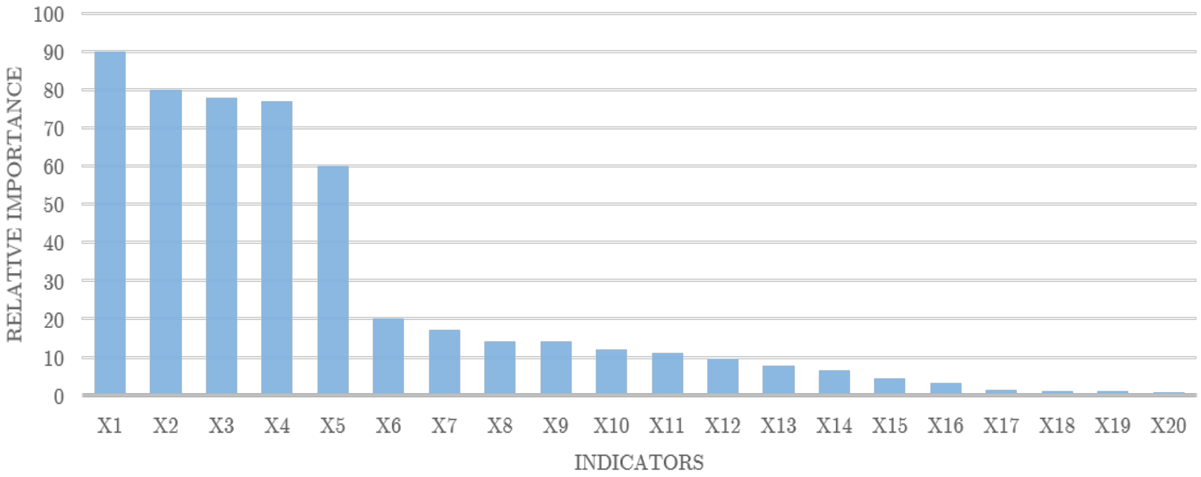


Figure 4.5: The process of creating failure indicators from the historical sensor data in the database

the prediction performance. Each time a selected indicator was added to the model, a performance gain was observed.

The best set of failure indicators to predict failure was identified with the help of variable importance feature of random forest method. From the 346 indicators developed, a total of 109 indicators was selected so as to accurately observe the change in patterns and predict the failure in equipment. Figure 4.5 shows the relative importance of the top 20 indicators selected based on the Gini impurity index using the random forest classification method. Table 4.2 describes the indicators referencing the Figure 4.5 in the decreasing order of importance chosen for failure prediction modeling. From the tabulated table, there were 14 time domain indicators, 4 basic expansions, and 2 frequency domain indicators.

The important indicators selected for predictive modeling will potentially help the maintenance practitioners look for any significant patterns leading to failures and schedule maintenance before the equipment stops working. The largest influence of failure consists of time domain indicators especially Kurtosis of sensor variables. The sudden change in the sensor measurements is captured by Kurtosis which differentiates the small changes which occur due to atmospheric conditions compared to the significant changes due to equipment degradation. Kurtosis had a higher value when there was a failure which helped the model setup conditions to classify between failure and non-failure.

The indicators derived from the timestamp also played an major role that can explain the seasonality of the age-related failure patterns (Wang et al., 2006). The identified basic expansion indicators as detailed in Table 4.2 are consistent with the finding from flight engine failure study by Keller et al. (2006) in which, the probability of failure increased on a particular day of the week due to long flight period. The basic expansion indicators explained one of the possible failure mechanisms associated with long working hours that created too much stress inside the pump walls (Wohlgemuth et al., 2006).

The failure indicators obtained from the selection process was used as input for the failure prediction model to predict the future failure probability of an equipment in the asset performance data. The section to follow will present the results of failure prediction models using different machine learning methods.

4.1.4 Performance Comparison of Failure Prediction Models

Four different machine learning algorithms: classification and regression trees (CART), random forest (RF), stochastic gradient boosting (GBM) and eXtreme gradient boosting (xgboost) methods are used in the development of the failure prediction model. The best method for the classification of failures was selected after comparison of the evaluation metrics. The goal is to find the best algorithm that can perform with the highest accuracy using the failure indicators developed. For this reason, the dataset is split into training and test data, where the first 80% of the recorded sensor measurements are treated as training data while the remaining 20% of the sensor measurements are used as test data. Table 4.3 lists the evaluation metrics accuracy, precision and F-measure for the methods utilized for modeling. Xgboost method was the best method compared to others with an accuracy of 90% followed by GBM method with 87% accuracy. When compared to random forest method, that is the most commonly used method for failure classification (Ngai et al., 2009), xgboost method performs with greater accuracy by correctly classifying the failures and non-failures.

Table 4.2: Description of top 20 indicators selected using random forest model

Indicator Description	Importance	Indicator Type	Reference
Kurtosis in vibration	90.22	Time domain	X1
Skewness in RPM	80.55	Time domain	X2
Kurtosis in RPM	78.87	Time domain	X3
Skewness in discharge pressure	77.08	Time domain	X4
Kurtosis in power	60.66	Time domain	X5
Interaction of discharge pressure, power and rpm	20.32	Basic expansion	X6
Skewness in power	17.05	Time domain	X7
Peak in rpm for 24 hour aggregation	14.56	Time domain	X8
Kurtosis in suction pressure for 3 hour aggregation	14.48	Time domain	X9
Fourier transformation of temperature	12.07	Frequency domain	X10
Peak in power for 24 hour aggregation	11.64	Time domain	X11
Day of the week	9.59	Basic expansion	X12
Skewness in temperature	7.73	Time domain	X13
Range in flow for 24 hour aggregation	6.63	Time domain	X14
Week of the year	4.87	Basic expansion	X15
Interaction of temperature and rpm	3.55	Basic expansion	X16
Standard deviation in temperature	1.75	Time domain	X17
Fourier transformation of flow	1.25	Frequency domain	X18
Kurtosis in vibration for 24 hour aggregation	1.06	Time domain	X19
Kurtosis in flow	0.94	Time domain	X20

Table 4.3: Performance of the failure prediction methods

Prediction Method	**Evaluation Metrics**			**Time(ms)**
	Accuracy	**Precision**	**F-measure**	
CART	60	80	43.5	1566.4
RF	71	90.7	60.7	2578.56
GBM	87	97.8	85	3600.26
Xgboost	90	99	90.1	2122.85

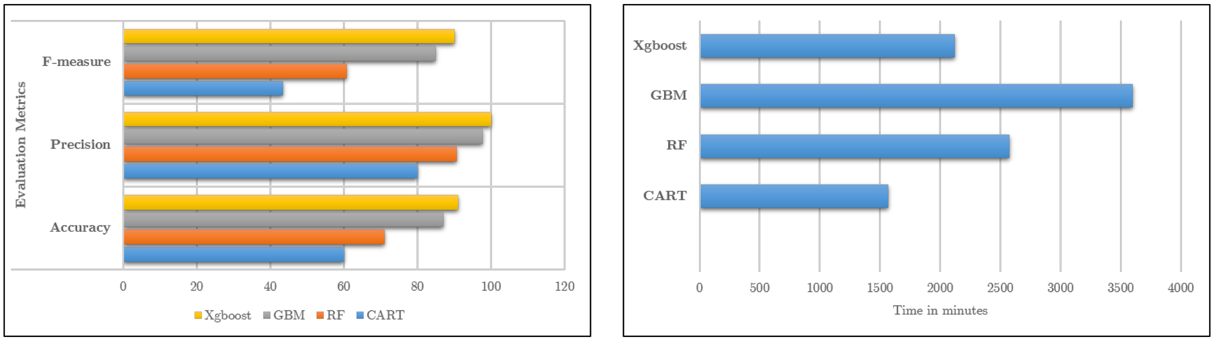


Figure 4.6: Comparison of performance metrics and computational time

Figure 4.6 is the visual representation of performance metrics across different failure prediction methods presented in Table 4.3. Another important metric to be considered while comparing the prediction methods is the build time and classification speed of the algorithm while training the model with failure indicators. This metric is important while working with real-time data of thousands of rows together. Figure 4.6 shows that CART method has the lowest build time followed by xgboost method with a 556 minutes difference. However, xgboost method is preferred for prediction in the production environment due to its higher performance and relatively low build time.

The next case study application utilizes a data set obtained from NASA data repository based on aircraft turbine engines. This application will further explain the efficacy of indicator development and selection methods for failure prediction.

4.2 NASA Benchmark Data

The NASA benchmark data set (NASA, 2008) consists of 100 engines each with multiple rows of multivariate data that includes time-stamped sensor data collected over every hour. The data described the working condition of aircraft turbine engines. The failure is detected at the component level in the engine with four different rotating components and exhibits output responses of pressure, vibration, voltage, and rotation as shown in Figure 4.7. The time series sensor dataset comprised of a total of 876100 entries. The sensor dataset consists of time-stamped sensor values at hourly intervals.

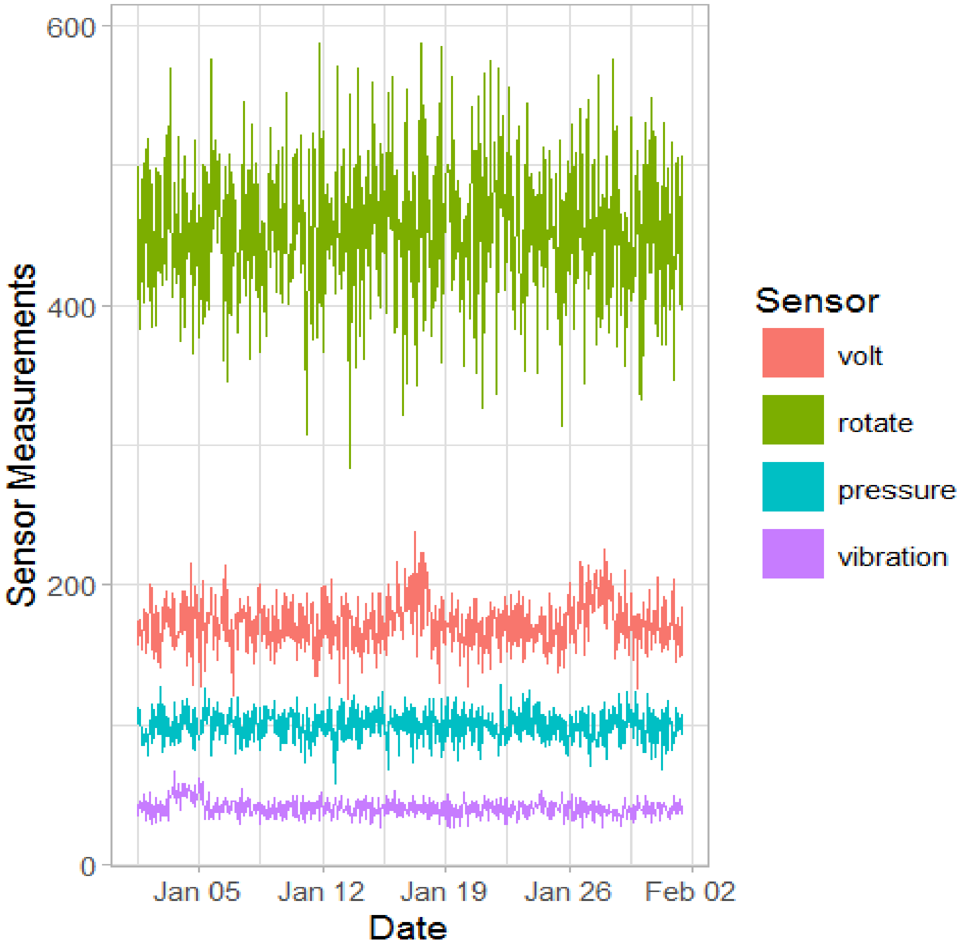


Figure 4.7: Engine sensor measurements

The historical dataset consists of maintenance and failure records. The maintenance dataset includes scheduled and unscheduled maintenance records which consist of both regular inspections as well as component failure. The records are entered into the maintenance dataset when a new component is introduced into the equipment during the scheduled maintenance or due to a failure. The entries created due to breakdown are entered into the failure dataset. The failure dataset consists of a total of 761 entries among four different components in the equipment which is shown in Figure 4.8. The historical records are merged with the sensor measurements using the timestamp column.

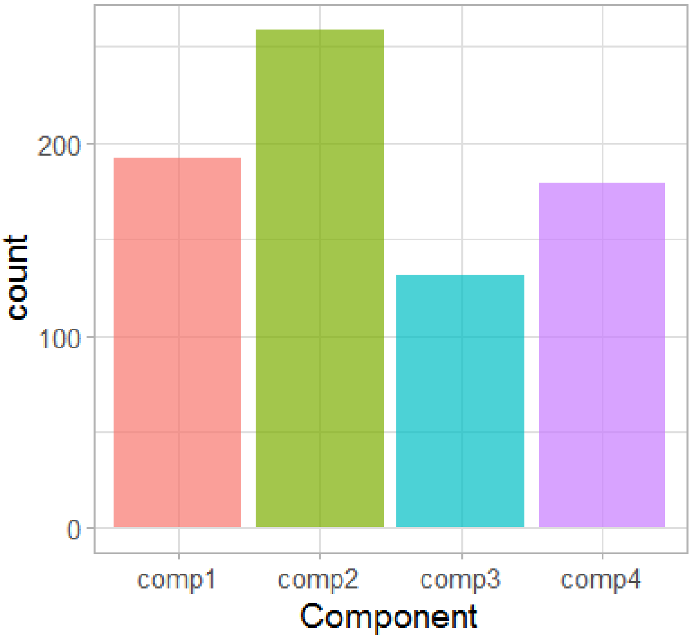


Figure 4.8: Failure in the components

The timestamped 4 historical sensor measurements were used to develop 94 leading and lagging failure indicators. However, because only some of indicators interpret the failure pattern, the 94 unique indicators passed through indicators selection method using random forest model. A total of 46 indicators were selected as the best subset that contributed to the failure prediction compared to the other indicators. These 46 indicators were used to develop a failure prediction model and predict the probability of failure of each component in the equipment.

4.2.1 Results

The process of finding a robust method for automatic prediction of failures was performed by training and cross-validation of four different machine learning algorithms. The classification performance of the algorithms was used for comparison to select the best model. The goal was to find best method that can predict failure using the incoming small real-time data. For this reason, the entire dataset was divided into training and testing set of 80:20 ratio. The training dataset was again split into increasing size of 10% to 90%. All the algorithms performed well with good accuracy. However, stochastic gradient boosting performed the best (see Table 4.4 and Figure 4.9).

Table 4.4: Performance of the failure prediction methods for engines

Prediction Method	Evaluation Metrics			Time(ms)
	Accuracy	Precision	F-measure	
CART	75	83.6	62.8	765.6
RF	83	85.5	82.6	720.56
GBM	89	90.9	85.7	980.2
Xgboost	85	91.7	89.2	920.7

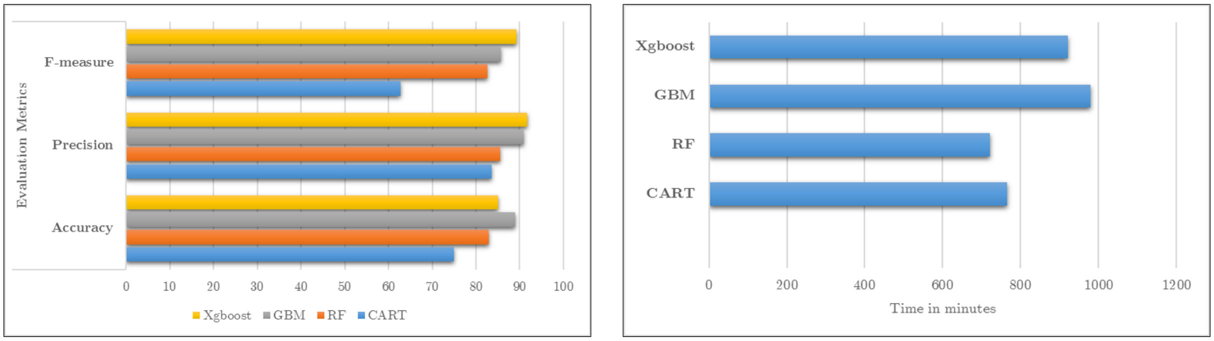


Figure 4.9: Right:Results of failure prediction models using evaluation metrics accuracy, precision and f-measure. Left: Comparison of computation time

In this work, automatic methods were presented to predict failures occurring in centrifugal pumps and aircraft turbine engines. The availability of failure occurrence in equipment or components in advance helps in scheduling planned maintenance activities. Furthermore, the reliability is enhanced which improves the throughput and increases the profit for an organization.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

A robust methodology that provides accurate estimates of failure in centrifugal pumps using extreme gradient boosting model is presented in this thesis. The developed model predicts the failure of pumps twenty-four hours in advance by accurately capturing the failure patterns and with the help of time-frequency indicators engineered from the historical sensor measurements. For every centrifugal pump, data from the sensor measurements is mined to obtain the failure patterns to analyze the intervention or degradation. The predictive accuracy of the extreme gradient boosting model is compared with random forest, stochastic gradient boosting, and classification and regression trees with the help of multiple evaluation metrics like accuracy, precision, and f-measure.

Extreme gradient boosting model is efficient as it proved to have a better fit for the data set and it takes less time to classify the failure events even for a higher order of magnitude of data as compared to stochastic gradient boosting model (second best model), which tend to over-fit the sensor measurements to non-failures. The best model was evaluated with benchmark dataset from NASA data repository and obtained a good accuracy model.

Relative the traditional data-driven methods which used historical sensor measurements and failure rates, there is an improvement in the prediction accuracy of the developed methodology because of the development of leading and lagging indicators.

Implementing the failure prediction model will help in scheduling planned maintenance by eliminating downtime in equipment. Providing the maintenance practitioners with the failure times will positively impact the performance of the system by increasing the reliability and availability. It also has the capability to improve maintenance costs and maintenance related logistics.

5.2 Contributions

The research described in this thesis leads to many contributions to the area of failure prediction and planned maintenance. These contributions result in the development of leading and lagging indicators from the historical data sources for use in the machine learning failure prediction methods. Additionally, a platform was built to identify and validate failure prediction models for different scenarios. The contributions are explained here.

1. Development of a set of leading and lagging indicators to characterize the failure patterns in the equipment for application of the machine learning failure prediction methods.

2. Development of an automated process to use suitable methods to identify an optimal or near-optimal subset of failure indicators, including preprocessing methods to remove less relevant data sources to improve the computation time.

3. Development of the R-Studio based program script to incorporate the application of four types of machine learning model development. Methods to aid in the development and selection of indicators and also to some extent to automate the process is included in the script.

5.3 Future Work

5.3.1 Model

The developed failure prediction model demonstrated its ability to classify the failures with good accuracy, however, there can be further improvements. By incorporating the individual failure messages with failed parts as potential predictor variables, a more detailed model can be developed resulting in the prediction of failure parts or mode. There also exists a potential to combine real-time environmental conditions with sensor measurements. This information can be used to develop more indicators which may lead to better estimation of failures. Furthermore, the model can be extended to other equipment by simply changing some of the indicators applicable to them.

5.3.2 Software

The developed model can be enhanced by integrating the existing scheduling tools used by maintenance practitioners to forecast the capacity. By doing this, the software can develop the bill of materials and manage the arrival of materials to the equipment. The same information can be displayed on the shop floor as a dashboard, providing the operators with updated information.

Bibliography

Ahmad, S., Schroeder, R. G., and Sinha, K. K. (2003). The role of infrastructure practices in the effectiveness of jit practices: implications for plant competitiveness. *Journal of Engineering and Technology management*, 20(3):161–191. 1, 5

Ahuja, I. P. S. and Khamba, J. S. (2008). Total productive maintenance: literature review and directions. *International Journal of Quality & Reliability Management*, 25(7):709–756. 4

Albino, V., Carella, G., and Geoffrey Okogbaa, O. (1992). Maintenance policies in just-in-time manufacturing lines. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 30(2):369–382. 4

Allen, J. B. and Rabiner, L. R. (1977). A unified approach to short-time fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11):1558–1564. 37

An, D., Kim, N. H., and Choi, J.-H. (2015). Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliability Engineering & System Safety*, 133:223–236. 23

Anderson, M. R., Antenucci, D., Bittorf, V., Burgess, M., Cafarella, M. J., Kumar, A., Niu, F., Park, Y., Ré, C., and Zhang, C. (2013). Brainwash: A data system for feature engineering. In *CIDR*. 6

Arlot, S., Celisse, A., et al. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79. 9

Asmai, S. A., Hussin, B., and Yusof, M. M. (2010). A framework of an intelligent maintenance prognosis tool. In *Computer Research and Development, 2010 Second International Conference on*, pages 241–245. IEEE. 17

Atherton, D. (1999). Prediction of machine deterioration using vibration based fault trends and recurrent neural networks. *Journal of vibration and acoustics*, 121:355–362. 21

Aven, T. (1983). Optimal replacement under a minimal repair strategya general failure model. *Advances in Applied Probability*, 15(01):198–211. 15

Aytug, H., Lawley, M. A., McKay, K., Mohan, S., and Uzsoy, R. (2005). Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161(1):86–110. 1

Balaban, E. and Alonso, J. J. (2012). An approach to prognostic decision making in the aerospace domain. Technical report, DTIC Document. 17

Barlow, R. E., Hunter, L. C., and Proschan, F. (1963). Optimum checking procedures. *Journal of the society for industrial and applied mathematics*, 11(4):1078–1095. 14

Barton, P. H. (2007). Prognostics for combat systems of the future. *IEEE Instrumentation & Measurement Magazine*, 10(4):10–14. 17

Bazovsky, I. (2004). *Reliability theory and practice*. Courier Corporation. 15

Bennett, R. B. (2009). Trade usage and disclaiming consequential damages: The implications for just-in-time purchasing. *American Business Law Journal*, 46(1):179–219. 2

Bergmeir, C., Hyndman, R. J., Koo, B., et al. (2015). A note on the validity of cross-validation for evaluating time series prediction. *Monash University, Department of Econometrics and Business Statistics, Tech. Rep.* 48

Bishop, C. M. (2006). Pattern recognition. *Machine Learning*, 128. 9

Block, H., Borges, W., and Savits, T. (1988). A general age replacement model with minimal repair. *Naval Research Logistics (NRL)*, 35(5):365–372. 14

Block, H. W., Langberg, N. A., and Savits, T. H. (1990). Maintenance comparisons: Block policies. *Journal of applied probability*, 27(03):649–657. 15

Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1):245–271. 39

Boland, P. J. (1982). Periodic replacement when minimal repair costs vary with time. *Naval Research Logistics (NRL)*, 29(4):541–546. 15

Boland, P. J. and Proschan, F. (1982). Periodic replacement with increasing minimal repair costs at failure. *Operations Research*, 30(6):1183–1189. 15

Bracewell, R. N. and Bracewell, R. N. (1986). *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York. 35

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140. 26

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32. 40, 43

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984a). *Classification and regression trees*. CRC press. 26

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984b). *Classification and regression trees*. CRC press. 41, 42

Callan, R., Larder, B., and Sandiford, J. (2006). An integrated approach to the development of an intelligent prognostic health management system. In *Aerospace Conference, 2006 IEEE*, pages 12–pp. IEEE. 18

Cerqueira, V., Pinto, F., Sá, C., and Soares, C. (2016). Combining boosted trees with metafeature engineering for predictive maintenance. In *International Symposium on Intelligent Data Analysis*, pages 393–397. Springer. 26

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357. 32, 57

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *arXiv preprint arXiv:1603.02754*. 46, 47

Choi, T. Y. (1997). The success and failures of implementing continuous improvement programs: Cases of seven automotive parts suppliers. *Becoming lean: Inside stories of US manufacturers*, pages 409–456. 3

Dal, B., Tugwell, P., and Greatbanks, R. (2000). Overall equipment effectiveness as a measure of operational improvement–a practical analysis. *International Journal of Operations & Production Management*, 20(12):1488–1502. 1

Darmoul, S., Pierreval, H., and Hajri-Gabouj, S. (2013). Handling disruptions in manufacturing systems: An immune perspective. *Engineering Applications of Artificial Intelligence*, 26(1):110–121. 1

Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. In *ICML*, volume 1, pages 74–81. Citeseer. 39

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer. 43

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87. 23

Dragomir, O. E., Gouriveau, R., Dragomir, F., Minca, E., and Zerhouni, N. (2009). Review of prognostic problem in condition-based maintenance. In *Control Conference (ECC), 2009 European*, pages 1587–1592. IEEE. 22

Emerson (2016). *2016 Cost of Data Center Outages Report*. goo.gl/C6bpZV. 4

Endrenyi, J., Aboresheid, S., Allan, R., Anders, G., Asgarpoor, S., Billinton, R., Chowdhury, N., Dialynas, E., Fipper, M., Fletcher, R., et al. (2001). The present status of maintenance strategies and the impact of maintenance on reliability. *IEEE Transactions on power systems*, 16(4):638–646. 5

Eti, M., Ogaji, S., and Probert, S. (2005). Maintenance schemes and their implementation for the afam thermal-power station. *Applied energy*, 82(3):255–265. 14

Fleischer, J., Weismann, U., and Niggeschmidt, S. (2006). Calculation and optimisation model for costs and effects of availability relevant service elements. *Proceedings of LCE*, pages 675–680. 1

Fluke (2007). *The basics of predictive/preventive maintenance*. http://goo.gl/HU9vWS. 14

Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407. 46

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232. 44, 45

Friedman, J. H. (2002a). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378. 26

Friedman, J. H. (2002b). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378. 44

Frisk, E., Krysander, M., and Larsson, E. (2014). Data-driven lead-acid battery prognostics using random survival forests. In *Proceedings of the annual conference of the prognostics and health management society. Fort Worth, Texas, USA*. 26

Gaber, M. M., Zaslavsky, A., and Krishnaswamy, S. (2005). Mining data streams: a review. *ACM Sigmod Record*, 34(2):18–26. 35

Gertman, D. I. and Blackman, H. S. (1994). *Human reliability and safety analysis data handbook*. John Wiley & Sons. 2

Gholami, M., Zandieh, M., and Alem-Tabriz, A. (2009). Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. *The International Journal of Advanced Manufacturing Technology*, 42(1):189–201. 3

Godinho Filho, M. and Uzsoy, R. (2011). The effect of shop floor continuous improvement programs on the lot size–cycle time relationship in a multi-product single-machine environment. *The International Journal of Advanced Manufacturing Technology*, 52(5):669–681. 3

Hale, G. (2007). *InTech survey: Predictive maintenance top technology challenge in 2007*. https://goo.gl/YmTwJ2. 24

Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83. 36

Hashemian, H. et al. (1995). On-line testing of calibration of process instrumentation channels in nuclear power plants. *US Nuclear Regulatory Commission, NUREG/CR-6343 (November 1995)*. 24

Hashemian, H. et al. (2007). Accurately measure the dynamic response of pressure instruments. *Power*, 151(11):72–72. 24

Hashemian, H., Riggsbee, E., Mitchell, D., Hashemian, M., Sexton, C., Beverly, D., and Morton, G. (1998). Advanced instrumentation and maintenance technologies for nuclear power plants. *US Nuclear Regulatory Commission, NUREG/CR-5501,(August 1998)*. 24

Hashemian, H. M. et al. (2005). *Sensor performance and reliability*. ISA-The Instrumentation, Systems, and Automation Society. 24

Hastie, T., Tibshirani, R., and Friedman, J. (2009). Unsupervised learning. In *The elements of statistical learning*, pages 485–585. Springer. 26, 41

Heng, A., Zhang, S., Tan, A. C., and Mathew, J. (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical systems and signal processing*, 23(3):724–739. 20

Heng, S. Y. A., Tan, A. C., Mathew, J., and Yang, B.-S. (2007). Machine prognosis with full utilization of truncated lifetime data. 22

Heo, G. Y. (2008). Condition monitoring using empirical models: technical review and prospects for nuclear applications. *Nuclear Engineering and Technology*, 40(1):49. 13

Hess, A., Calvello, G., and Dabney, T. (2004). Phm a key enabler for the jsf autonomic logistics support concept. In *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, volume 6, pages 3543–3550. IEEE. 17

Hess, A., Stecki, J. S., and Rudov-Clark, S. D. (2008). The maintenance aware design environment: development of an aerospace phm software tool. *Proc. PHM08*. 16, 17

Ho, T. K. (1995). Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE. 43

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67. 45

Hopp, W. J. and Spearman, M. L. (2011). *Factory physics*. Waveland Press. 2, 3

Horn, P. (2001). Autonomic computing: Ibm\'s perspective on the state of information technology. 9

Hu, Y., Chen, H., Li, G., Li, H., Xu, R., and Li, J. (2016). A statistical training data cleaning strategy for the pca-based chiller sensor fault detection, diagnosis and data reconstruction method. *Energy and Buildings*, 112:270–278. 27

Jabal Ameli, M. S., Arkat, J., and Barzinpour, F. (2008). Modelling the effects of machine breakdowns in the generalized cell formation problem. *The International Journal of Advanced Manufacturing Technology*, 39(7):838–850. 3

Jardine, A. K., Lin, D., and Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, 20(7):1483–1510. 23

Jardine, A. K. and Tsang, A. H. (2013). *Maintenance, replacement, and reliability: theory and applications*. CRC press. 15

Javed, K. (2014). *A robust & reliable Data-driven prognostics approach based on extreme learning machine and fuzzy clustering*. PhD thesis, Université de Franche-Comté. 23

Javed, K., Gouriveau, R., Zerhouni, N., and Nectoux, P. (2013). A feature extraction procedure based on trigonometric functions and cumulative descriptors to enhance prognostics modeling. In *Prognostics and Health Management (PHM), 2013 IEEE Conference on*, pages 1–7. IEEE. 21

Jeziorek, O. (1994). Lean production. In *Lean Production*, pages 3–27. Springer. 2

Katipamula, S. and Brambley, M. R. (2005). Review article: methods for fault detection, diagnostics, and prognostics for building systemsa review, part i. *HVAC&R Research*, 11(1):3–25. 18

Keller, K., Swearingen, K., Sheahan, J., Bailey, M., Dunsdon, J., Przytula, K. W., and Jordan, B. (2006). Aircraft electrical power systems prognostics and health management. In *Aerospace Conference, 2006 IEEE*, pages 12–pp. IEEE. 13, 62

Kelly, A. (1989). Maintenance and its management. Conference Communication. 15

Kelvin, L. W. B. (2016). Exploring accuracy and efficiency of preventive maintenance prediction with machine learning techniques. *Business Analytics: Progress on Applications in Asia Pacific*, page 286. 26

Kijima, M. (1989). Some results for repairable systems with general repair. *Journal of Applied probability*, 26(01):89–102. 15

Kim, K.-j. and Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert systems with Applications*, 19(2):125–132. 33

Kothamasu, R., Huang, S. H., and VerDuin, W. H. (2009). System health monitoring and prognostics–a review of current paradigms and practices. In *Handbook of maintenance management and engineering*, pages 337–362. Springer. 16, 17

Kuhn, M. (2008). Caret package. *Journal of Statistical Software*, 28(5). 48

Kuhn, M. and Johnson, K. (2013). *Applied predictive modeling*. Springer. 44, 45

Levis, P., Patel, N., Culler, D., and Shenker, S. (2004). Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proc. of the 1st USENIX/ACM Symp. on Networked Systems Design and Implementation*. 25

Liao, H., Zhao, W., and Guo, H. (2006). Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model. In *Reliability and Maintainability Symposium, 2006. RAMS'06. Annual*, pages 127–132. IEEE. 13

Liu, H. and Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*, 17(4):491–502. 40

Liu, L., Peng, Y., and Liu, D. (2016). Feser: A data-driven framework to enhance sensor reliability for the system condition monitoring. *Microelectronics Reliability*, 64:681–687. 27

Ljungberg, Õ. (1998). Measurement of overall equipment effectiveness as a basis for tpm activities. *International Journal of Operations & Production Management*, 18(5):495–507. 1

Ma, J. and Jiang, J. (2011). Applications of fault detection and diagnosis methods in nuclear power plants: A review. *Progress in nuclear energy*, 53(3):255–266. 18

Madria, S., Kumar, V., and Dalvi, R. (2014). Sensor cloud: A cloud of virtual sensors. *IEEE software*, 31(2):70–77. 25

Marsh (2014). *The 100 Largest Losses 1974-2013, Large Property Damage Losses in the Hydrocarbon Industry, 23rd Edition*. http://tinyurl.com/hfatcxo. 3, 4

Melnyk, S. A. (2007). Lean to a fault? *Council of Supply Chain Management Professionals Supply Chain Quarterly*, 2007(3):29–33. 4

Meridium, A. S. C. (2017). *Meridium, from GE Digital*. https://www.meridium.com/. xi, 6, 54, 55

Miller, R. B. (1953). A method for man-machine task analysis. Technical report, DTIC Document. 2

Mosley, S. A., Teyner, T., and Uzsoy, R. M. (1998). Maintenance scheduling and staffing policies in a wafer fabrication facility. *IEEE Transactions on Semiconductor Manufacturing*, 11(2):316–323. 4

NASA (2008). Degradation simulation data set. *NASA Ames Prognostics Data Repository*. 54, 65

Ngai, E. W., Xiu, L., and Chau, D. C. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert systems with applications*, 36(2):2592–2602. 62

Orchard, M., Wu, B., and Vachtsevanos, G. (2005). A particle filtering framework for failure prognosis. In *World tribology congress III*, pages 883–884. American Society of Mechanical Engineers. 21

Osipov, V., Luchinsky, D., Smelyanskiy, V., Kiris, C., Timucin, D., and Lee, S. (2007). In-flight failure decision and prognostics for the solid rocket booster. In *AIAA 43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Cincinnati, OH*. 16

Parida, A. and Kumar, U. (2006). Maintenance performance measurement (mpm): issues and challenges. *Journal of Quality in Maintenance Engineering*, 12(3):239–251. 14

Pecht, M. (2008). *Prognostics and health management of electronics*. Wiley Online Library. 20

Pecht, M. and Gu, J. (2009). Physics-of-failure-based prognostics for electronic products. *Transactions of the Institute of Measurement and Control*, 31(3-4):309–322. 20

Pecht, M. and Jaai, R. (2010). A prognostics and health management roadmap for information and electronics-rich systems. *Microelectronics Reliability*, 50(3):317–323. 20, 21

Pintelon, L. and Parodi-Herz, A. (2008). Maintenance: an evolutionary perspective. *Complex system maintenance handbook*, pages 21–48. 13, 17

Pintelon, L. M. and Gelders, L. (1992). Maintenance management decision making. *European journal of operational research*, 58(3):301–317. 4

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45. 9

Psichogios, D. C. and Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511. 22

Rausand, M., Arnljot, H., et al. (2004). *System reliability theory: models, statistical methods, and applications*, volume 396. John Wiley & Sons. 15

Santur, Y., Karaköse, M., and Akin, E. (2016). Random forest based diagnosis approach for rail fault inspection in railways. In *Electrical, Electronics and Biomedical Engineering (ELECO), 2016 National Conference on*, pages 745–750. IEEE. 26

Satishkumar, R. and Sugumaran, V. (2016). Vibration based health assessment of bearings using random forest classifier. *Indian Journal of Science and Technology*, 9(10). 26

Savits, T. H. (1988). A cost relationship between age and block replacement policies. *Journal of applied probability*, 25(04):789–796. 15

Savsar, M., Khan, M., and El-Tamimi, A. (1993). Modeling of maintenance polices in just-in-time production assembly systems. In *Proceeding of the Second Scientific Symposium on Maintenance, Planning and Operations, Riyadh*, pages 95–108. 4

Sawhney, R., Subburaman, K., Sonntag, C., Rao Venkateswara Rao, P., and Capizzi, C. (2010). A modified fmea approach to enhance reliability of lean systems. *International Journal of Quality & Reliability Management*, 27(7):832–855. 2

Sejdić, E., Djurović, I., and Jiang, J. (2009). Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Processing*, 19(1):153–183. 37

Si, X.-S., Wang, W., Hu, C.-H., and Zhou, D.-H. (2011). Remaining useful life estimation–a review on the statistical data driven approaches. *European journal of operational research*, 213(1):1–14. 17, 18, 21

Sikorska, J., Hodkiewicz, M., and Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, 25(5):1803–1836. 23

Spencer, B. F., Ruiz-Sandoval, M. E., and Kurata, N. (2004). Smart sensing technology: opportunities and challenges. *Structural Control and Health Monitoring*, 11(4):349–368. 24

Stadje, W. and Zuckerman, D. (1991). Optimal maintenance strategies for repairable systems with general degree of repair. *Journal of Applied Probability*, 28(02):384–396. 14

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62(1):77–89. 51

Sun, Y., Ma, L., Mathew, J., Wang, W., and Zhang, S. (2006). Mechanical systems hazard estimation using condition monitoring. *Mechanical systems and signal processing*, 20(5):1189–1201. 22

Tan, L. and Jiang, J. (2013). *Digital signal processing: fundamentals and applications*. Academic Press. 33

Thompson, M. L. and Kramer, M. A. (1994). Modeling chemical processes using prior knowledge and neural networks. *AIChE Journal*, 40(8):1328–1340. 22

Tilquin, C. and Cleroux, R. (1975). Periodic replacement with minimal repair at failure and adjustment costs. *Naval Research Logistics (NRL)*, 22(2):243–254. 15

Timofeev, R. (2004). *Classification and regression trees (CART) theory and applications*. PhD thesis, Humboldt University, Berlin. 26, 41

Tucker, R. M., Straub, T., Feng, S., et al. (2013). Unplanned downtime in the gulf of mexico: A significant production loss management opportunity for producers. *Journal of Petroleum Technology*, 65(07):72–75. 4

Uckun, S., Goebel, K., and Lucas, P. J. (2008). Standardizing research methods for prognostics. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–10. IEEE. 16, 17

Van Noortwijk, J. (2009). A survey of the application of gamma processes in maintenance. *Reliability Engineering & System Safety*, 94(1):2–21. 16

van Rijsbergen, C. (1979). Information retrieval butterworth and co. *London,*. 52

Veeger, C., Etman, L., Van Herk, J., and Rooda, J. (2010). Generating cycle time-throughput curves using effective process time based aggregate modeling. *IEEE Transactions on Semiconductor Manufacturing*, 23(4):517–526. 3

Verikas, A., Gelzinis, A., and Bacauskiene, M. (2011). Mining data with random forests: A survey and results of new tests. *Pattern Recognition*, 44(2):330–349. 26

Vineyard, M. and Meredith, J. (1992). Effect of maintenance policies on fms failures. *The International Journal Of Production Research*, 30(11):2647–2657. 4

Virili, F. and Freisleben, B. (2000). Nonstationarity and data preprocessing for neural network predictions of an economic time series. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 5, pages 129–134. IEEE. 33

Wang, B., Furst, E., Cohen, T., Keil, O. R., Ridgway, M., and Stiefel, R. (2006). Medical equipment management strategies. *Biomedical Instrumentation & Technology*, 40(3):233–237. 62

Wang, T. (2010). *Trajectory similarity based prediction for remaining useful life estimation.* PhD thesis, University of Cincinnati. 20

Wickham, H. et al. (2014). Tidy data. *Journal of Statistical Software*, 59(10):1–23. 30

Wohlgemuth, J. H., Cunningham, D. W., Monus, P., Miller, J., and Nguyen, A. (2006). Long term reliability of photovoltaic modules. In *Photovoltaic Energy Conversion, Conference Record of the 2006 IEEE 4th World Conference on*, volume 2, pages 2050–2053. IEEE. 62

Wu, K., McGinnis, L. F., and Zwart, B. (2008). Queueing models for single machine manufacturing systems with interruptions. In *Simulation Conference, 2008. WSC 2008. Winter*, pages 2083–2092. IEEE. 3

Yam, R., Tse, P., Li, L., and Tu, P. (2001). Intelligent predictive decision support system for condition-based maintenance. *The International Journal of Advanced Manufacturing Technology*, 17(5):383–391. 21

Yang, Y. and Webb, G. I. (2009). Discretization for naive-bayes learning: managing discretization bias and variance. *Machine learning*, 74(1):39–74. 33

Zhang, L., Xiong, G., Liu, H., Zou, H., and Guo, W. (2010). Bearing fault diagnosis using multi-scale entropy and adaptive neuro-fuzzy inference. *Expert Systems with Applications*, 37(8):6077–6085. 32

Zhang, S., Ma, L., Sun, Y., and Mathew, J. (2007). Asset health reliability estimation based on condition data. 21

Appendix

Appendix A

Asset Performance Data

A.1 Addressing the imbalance problem

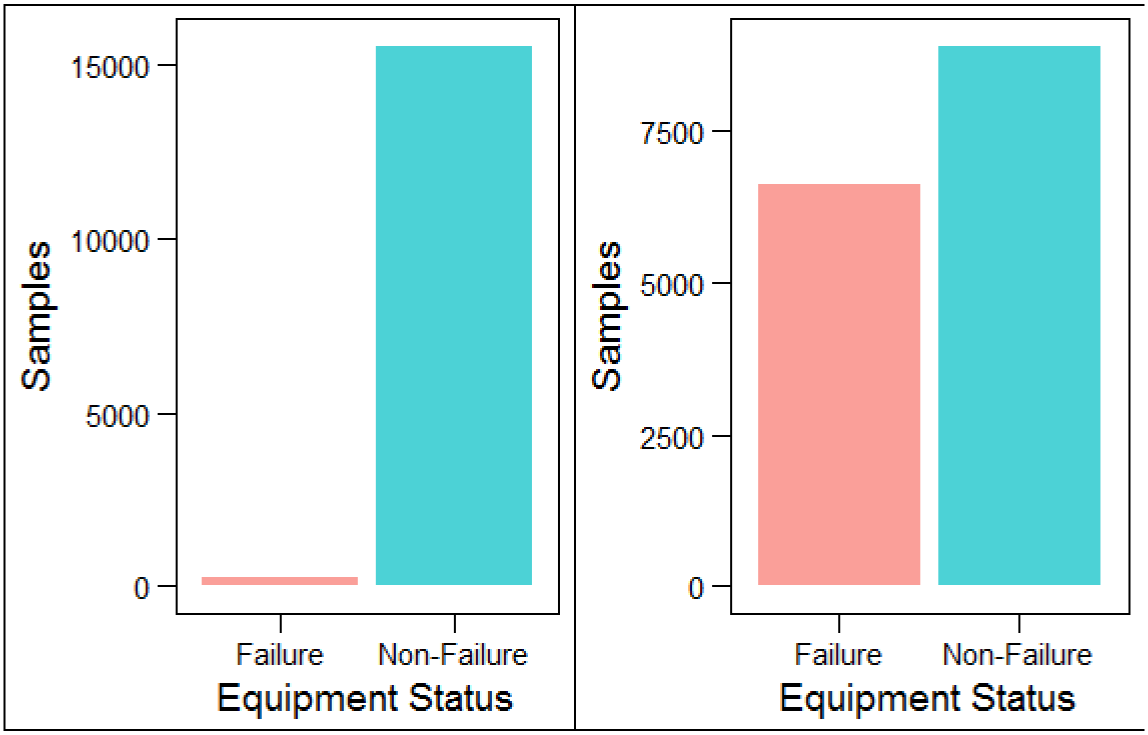


Figure A.1: Over-sampling failures to balance the data

A.2 Failure indicators

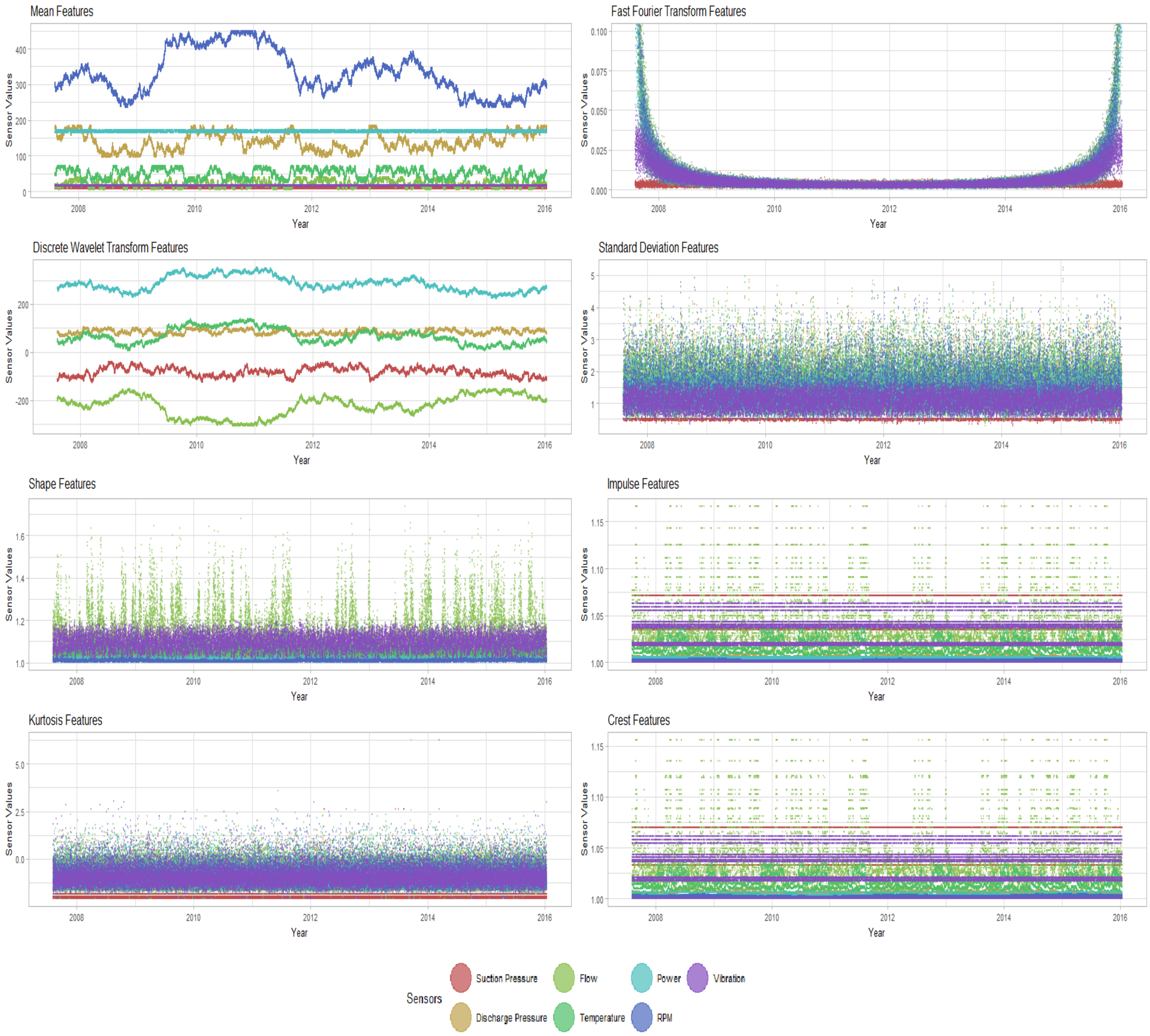


Figure A.2: Sample of indicators developed from sensor measurements

Vita

Dhanush Agara Mallesh is currently a graduate student pursuing two Master's, Master of Science in Industrial Engineering and Master of Science in Statistics at the University of Tennessee, Knoxville. Prior his graduate studies, Dhanush obtained a Bachelor of Engineering degree in Mechanical Engineering from AMC Engineering College, India in 2013. He attended two different schools Frank Anthony Public School and BGS International Residential School where he lettered in soccer, field hockey and track, graduating in May 2007.

Dhanush Agara Mallesh was born on May 1991 in Bangalore, India. He is the son of Mallesh Agara Mallegowda and Shantha Gorur Chickkegowda of India, and is the younger brother of Kavya Mallesh. At UTK, he was awarded the Extraordinary Professional Promise honor which is provided to graduate students who demonstrate professional promise in teaching, research or other contributions. His research interests include predictive modeling and natural language processing.