Summer 2012

Performance monitoring of wind turbines : a data-mining approach

Anoop Prakash Verma
University of Iowa

PERFORMANCE MONITORING OF WIND TURBINES: A DATA-MINING APPROACH

by

Anoop Prakash Verma

An Abstract

Of a thesis submitted in partial fulfillment
of the requirements for the Doctor of Philosophy
degree in Industrial Engineering
in the Graduate College of
The University of Iowa

July 2012

Thesis Supervisor: Professor Andrew Kusiak

ABSTRACT

The rapid growth of wind turbines in terms of turbine size, number of installations and rated capacity has a huge impact on its operations and maintenance costs. Monitoring the performance of wind turbines and early fault prediction is highly desirable.

To date, traditional maintenance strategies such as reactive maintenance, periodic maintenance etc. are more prevalent in wind industry. However, over the last couple of years, the research pertaining to wind turbine has been shifted towards the condition monitoring and maintenance.

Condition monitoring approaches have shown their potential in wind industry by providing continuous monitoring of the wind turbines, and identifying fault signatures in the event of faults. However, most of the studies reported in literature are based on the simulated dataset, or in constrained experiments. In reality, the external environment plays an important role in governing the turbine operations. Moreover, the cost associated with condition monitoring cannot be justified as it often requires installations of specific sensors, equipment.

Another stream of research focuses on utilizing historical turbine data for turbine performance assessment in real time. The cost associated with such approaches is almost negligible as most of the wind farms are equipped with SCADA systems which records turbine performance data in regular time-interval. Such approaches are called as performance monitoring.

In this dissertation, the performance monitoring of wind turbines is accomplished using the historical wind turbine data. The information from SCADA operational data, and fault logs is used to construct accurate models predicting the critical wind turbine faults. Depending upon the nature of turbine faults, monitoring wind turbines with different objectives is studied to accomplish different research goals.

Two research directions of wind turbines performance are pursued, (1) identification and prediction of critical turbine faults, and (2) monitoring the performance of overall wind farm. The goal of predicting critical faults is to facilitate planned maintenance, whereas, monitoring the performance of overall wind farm provides the status-quo of all wind turbines installed in a wind farm. Depending on the requirement, the performance of overall wind farm can be assessed on a daily, weekly, or monthly basis.

Solution methodologies presented in the dissertation are generic enough to be applicable to other industries such as wastewater treatment facilities, flood prediction, etc.

Abstract Approved: _____
Thesis Supervisor

_____
Title and Department

_____
Date

PERFORMANCE MONITORING OF WIND TURBINES: A DATA-MINING APPROACH

by

Anoop Prakash Verma

A thesis submitted in partial fulfillment
of the requirements for the Doctor of Philosophy
degree in Industrial Engineering
in the Graduate College of
The University of Iowa

July 2012

Thesis Supervisor: Professor Andrew Kusiak

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

_____

PH.D. THESIS

_____

This is to certify that the Ph.D. thesis of

Anoop Prakash Verma

has been approved by the Examining Committee
for the thesis requirement for the Doctor of Philosophy
degree in Industrial Engineering at the July 2012 graduation.

Thesis Committee: _____
Andrew Kusiak, Thesis Supervisor

_____
Pavlo Krokhmal

_____
Yong Chen

_____
Pablo Carrica

_____
David Bennett

To My Parents, Family and Friends

Your biggest opportunity probably lies under your own feet, in your current job, industry, education, experience or interests

-Brian Tracy

ACKNOWLEDGMENTS

I would like to express my gratitude to several people who contributed, in different ways, to the completion of this work. Inevitably, some names will be missed here.

Firstly, I would like to express my sincere gratitude and special thanks to my thesis and academic advisor Professor Andrew Kusiak, for his advice, support and patience. From the time I started my studies at University of Iowa, he has always been there to guide and encourage me. I was extensively exposed to the research of advanced data mining and optimization theories as well as real-world applications while working in the Intelligent Systems Laboratory.

I would like to thank Professor Pavlo Krokhmal, Professor Yong Chen, Professor Pablo Carrica, and Professor David Bennett for serving on my PhD dissertation committee and providing valuable suggestions and feedback on my research.

I would also like to thank my fellow students in the Intelligent Systems Laboratory for their suggestion and help. Also, special thanks to my undergraduate advisor Professor M.K.Tiwari who introduces me to the complex research world of industrial engineering and optimization.

Finally, I would like to express my heartfelt appreciation and love to my mom and brother for their support and understanding.

ABSTRACT

The rapid growth of wind turbines in terms of turbine size, number of installations and rated capacity has a huge impact on its operations and maintenance costs. Monitoring the performance of wind turbines and early fault prediction is highly desirable.

To date, traditional maintenance strategies such as reactive maintenance, periodic maintenance etc. are more prevalent in wind industry. However, over the last couple of years, the research pertaining to wind turbine has been shifted towards the condition monitoring and maintenance.

Condition monitoring approaches have shown their potential in wind industry by providing continuous monitoring of the wind turbines, and identifying fault signatures in the event of faults. However, most of the studies reported in literature are based on the simulated dataset, or in constrained experiments. In reality, the external environment plays an important role in governing the turbine operations. Moreover, the cost associated with condition monitoring cannot be justified as it often requires installations of specific sensors, equipment.

Another stream of research focuses on utilizing historical turbine data for turbine performance assessment in real time. The cost associated with such approaches is almost negligible as most of the wind farms are equipped with SCADA systems which records turbine performance data in regular time-interval. Such approaches are called as performance monitoring.

In this dissertation, the performance monitoring of wind turbines is accomplished using the historical wind turbine data. The information from SCADA operational data, and fault logs is used to construct accurate models predicting the critical wind turbine faults. Depending upon the nature of turbine faults, monitoring wind turbines with different objectives is studied to accomplish different research goals.

Two research directions of wind turbines performance are pursued, (1) identification and prediction of critical turbine faults, and (2) monitoring the performance of overall wind farm.

The goal of predicting critical faults is to facilitate planned maintenance, whereas, monitoring the performance of overall wind farm provides the status-quo of all wind turbines installed in a wind farm. Depending on the requirement, the performance of overall wind farm can be assessed on a daily, weekly, or monthly basis.

Solution methodologies presented in the dissertation are generic enough to be applicable to other industries such as wastewater treatment facilities, flood prediction, etc.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Wind is regarded as one of the most potential source of renewable energy with the competitive advantage in terms of its availability and environmental considerations. Due to the improvements in the technology and availability of space, an increasing number of wind-turbines are being installed every year across the globe. In the last few years, wind energy has gained attention among large firms, researchers, and scientists, and it is anticipated to grow in years to come (Dimitrovski and Tomsovic, 2006). The installed wind energy capacity has been steadily growing in the US and Europe (Amirat *et al.*, 2009).

Wind turbine industry has witnessed some shortcomings, such as drive train failures, spalled bearings, and fractured gears due to excessive loads. Such failures may lead to a catastrophic failure of the overall system and require expensive repairs (Rolf and Powers, 2006). The success of the energy industry can be estimated by its cost of energy. Cost of energy (COE) measures the cost of generating electricity from wind energy. Thus, COE can be used as a metric to evaluate the potential of wind turbines.

Studies reveal the strong association between cost of energy (COE) and operations and maintenance (O&M). In the wind energy research community, the COE is calculated using the following equation (Cohen *et al.*, 1989).

$$COE = \frac{ICC \times FC + LRC}{AEP} + O\&M \tag{1.1}$$

Where, COE is the cost of energy, ICC is the initial capital cost, FC is the fixed charge (%/year), LRC is the levelized replacement cost ($/year), AEP is the annual energy production, and O&M is operations and maintenance costs. Thus, in order to reduce the overall cost of energy, the energy production needs to be improved, whereas,

the operations and maintenance related costs needs to be minimized. As per the COE of low wind speed turbines (LWST), O&M costs can account for more than 10% of the total cost (Cohen *et al.*, 1889). The O&M related costs are expected to grow with the years of turbine operations. The advanced supervisory control and data acquisition (SCADA) systems is able to resolve several operations issues by automatically starting, stopping, and resetting the turbines in case of small fluctuations (Vachon, 2002).

However, maintenance cost is still a major concern. Extent of maintenance, type of maintenance, and component's age are the key factors discussed in operations and maintenance (O&M) related research and development (R&D) communities (Ribrant and Bertling, 2007). The maintenance cost comprises of the cost associated with scheduled maintenance, and cost associated with unscheduled maintenance. In order to better understand the maintenance related issues, fault analysis of wind turbines is essential. Gearboxes and blades are the most costly and fault prone components in the turbine, and therefore they have drawn the focus of researchers. Other fault prone components of the systems are electrical system and yaw systems. A graph showing the comparison of percentage failure is presented in Figure 1.1 (Ribrant, 2006). Studies reveal that generator; turbine blades and gearbox are the critical components which contribute more than 85% of the maintenance cost as well as downtime of the whole wind energy conversion systems (WECS) (Rademakers *et al.*, 2007).

Additionally, with an aim to harvest more energy, several modifications in the turbine design has been done. Now-a-days turbine blades are about 40 meters long with the tower height being increased from 60 meters to 100 meters, whereas, rotor diameter has now increased to more than 100 meters. Modification in the tower height made the maintenance and inspection task difficult, whereas, modification in the rotor size made turbine blades more sensitive to wind speed.

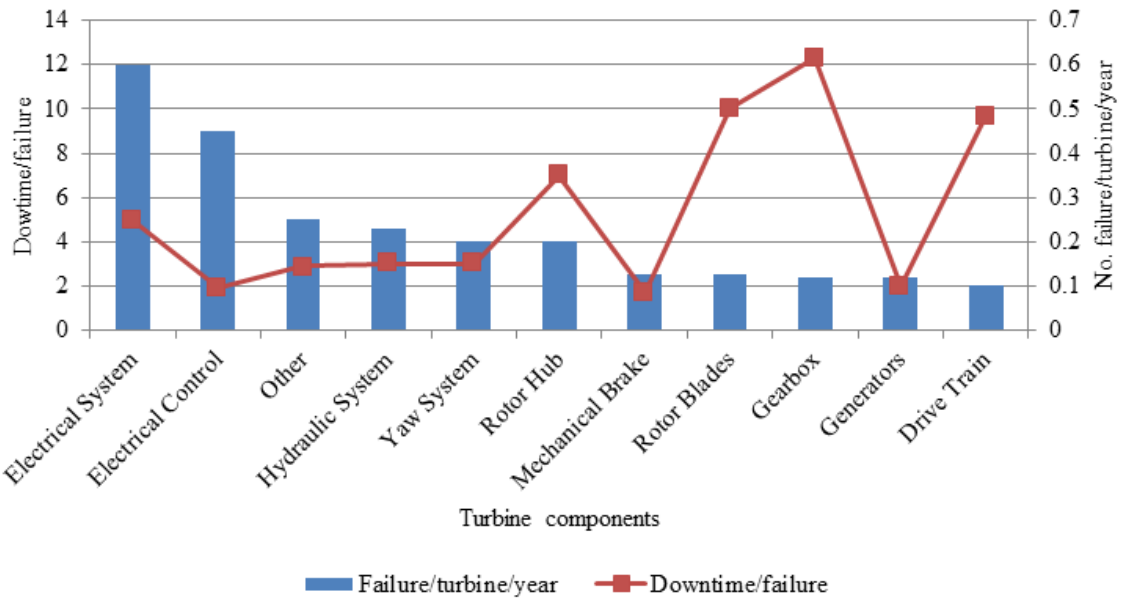


Figure 1.1. Percentage failures of turbine components.

In order to address the towering operations and maintenance requirements, the traditional maintenance strategies such as periodic maintenance and reactive maintenance etc. are being replaced with condition based monitoring and maintenance. Unlike traditional maintenance, condition based maintenance approaches continuously monitors the performance of wind turbine components with the help of sensors and equipment. Such approaches determine the equipment's health, and act only when maintenance is actually necessary. Vibration analysis (Caselitz and Giebhardt, 2005), optical strain measurements (Becker and Posta, 2006), and oil particle analysis (Caselitz *et al.*, 1997) are commonly used in condition monitoring. In the related research, Rademakers *et al.*, (2003) utilized a contamination monitoring approach to detect the presence of presence of ferro-magnetic debris, which is indicative of wear particles from rolling or rubbing contacts. Such approach can be applicable to whole wind energy conversion system (WECS). Wernicke *et al.*, (2004) performed spectral analysis to analyze the periodic oscillations in nacelle. They applied Fast Fourier Transformation (FFT) algorithm on the measured acceleration time signals. Schroeder *et al.*, (2006) developed strain monitoring approach to

continuously monitor the load in the wind turbine. They have used optical sensor in their study. Lading *et al.*, (2002) developed an ultrasonic inspection based approach to analyze the depth in the turbine structure. They termed it as optical coherence thermography.

The mentioned condition monitoring approaches provide reasonable solution and can be used to continuously monitor the turbine components. However, such studies are mostly limited to lab settings which do not truly reflect the turbine characteristics. In addition, condition monitoring approaches appear to be a costly option as they require additional sensors and equipment to be installed in wind turbine. Thus, efficient and cost effective monitoring of wind turbines still remains an issue.

Making use of already recorded wind turbine performance data can be an alternative way for turbine performance analysis. Over the past few years, wind turbine research community have utilized SCADA data in (1) Design of wind turbines (Laino *et al.*, 1993), (2) Control of wind turbines (Ko *et al.*, 2008), (3) Prediction of wind power (Ko *et al.*, Kusiak *et al.*, 2009a), and (4) Wind energy conversion (Kusiak *et al.*, 2009b, Kusiak *et al.*, 2009c) etc.

Such approaches make use of the data recorded by advanced supervisory control and data acquisition systems (SCADA), which most of the wind turbines are equipped with. Thus, monitoring the wind turbines by utilizing the operational SCADA data needs attention. In the present research, the historical wind turbine data is used to *tune* the wind turbines by providing time-ahead prediction of critical wind turbine faults.

Two research goals are set for this dissertation. The first goal is to develop non-parametric models capturing the turbine behavior in case of abnormalities. To accomplish the goals, both multi-class classification and multiple nonlinear regression based fault prediction models are constructed. Later, the overall wind farm is analyzed as a whole by continuously monitoring the progress of several wind turbines.

1.1Fault modes of wind turbines

Wind turbines are sophisticated aerodynamic machines with complex control system consisting of various assemblies, sub-assemblies, and parts etc. which are likely to fail. Depending upon external environment, and turbine condition itself, failure can occur anywhere in the wind turbine (Ashley *et al.*, 2007). However, the frequency and extent to which the particular failure affects the system performance varies. Gearbox failures are less frequent but are most costly to repair and replace, whereas, electrical system of the wind turbine often fails, but can be easily repaired. A description of failure associated with turbine components are shown in Figure 1.2

Every year various damages in the wind turbines are reported across the globe. Depending upon the extent of damage, fault associated with wind turbines are categorized into 3 categories (Pacot *et al*. (2003). Category A represents the most severe faults which can lead to the shutdown of the turbine. To deal with these fault types efficient monitoring is required so that it can be identified beforehand. Category B faults can partially affect the WECS, whereas, category C fault arises due to over speeding and therefore can be controlled easily (Figure 1.3).

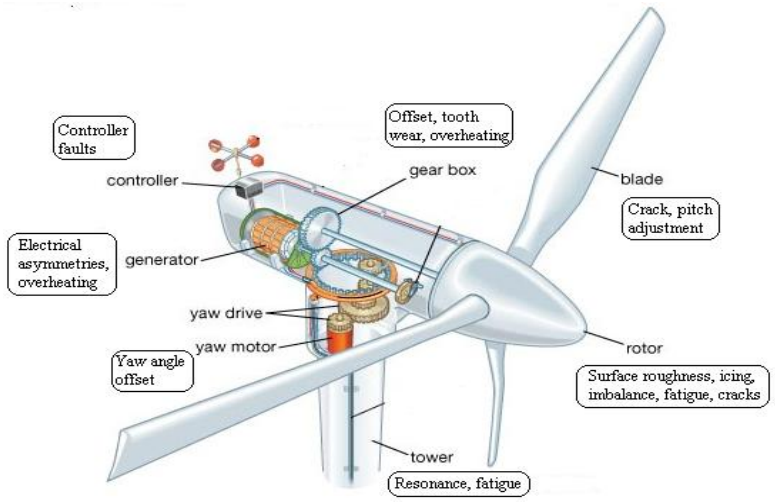


Figure 1.2. Wind turbine components and associated faults.

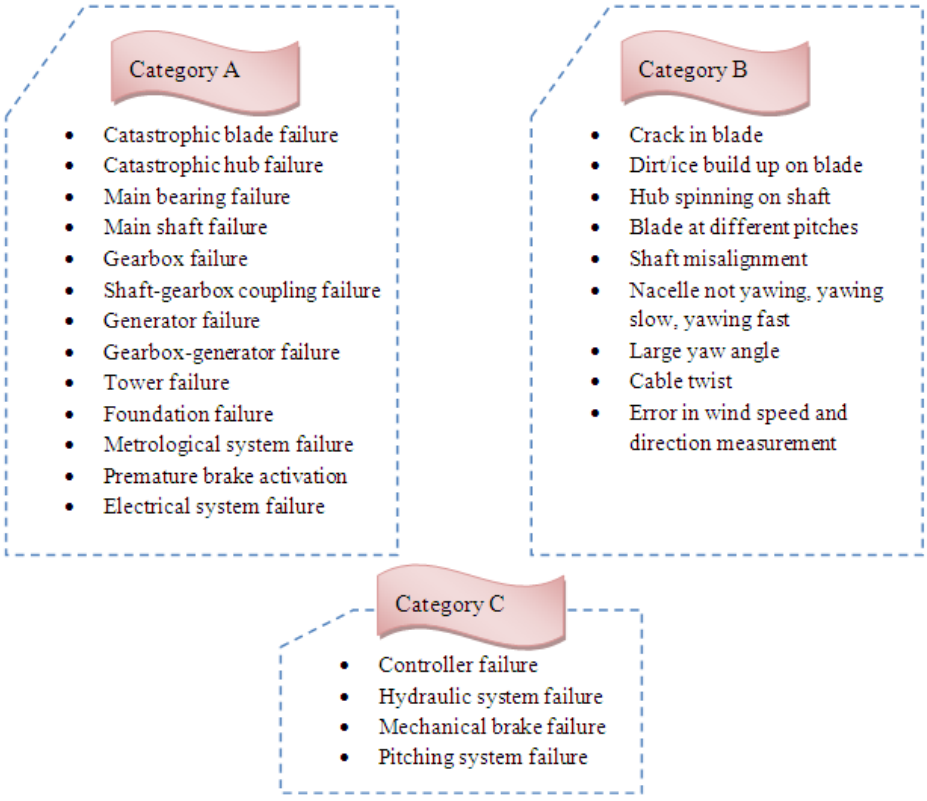


Figure 1.3. Failure modes of a wind turbine.

Alarm based monitoring approaches generate relevant alarm signals in the event of faults, or normal wind turbine operations. Typically, a wind turbine control system generates four kinds of alarm signals, i.e., (1) System update, (2) Information, (3) Warning, and (4) Error. The alarm signal namely *system update* indicates the wind turbine is operational, *information* alarm signals records the system change information with no-consequential effect on wind turbine performance. The alarm signals namely, *warning* indicates that certain components of wind turbine may fail in the future, whereas, *error* signal indicates the certain component is faulty.

Qiu *et al.*, (2011) performed Venn diagrams and decision tree based analysis with an aim to improve the system reliability. They claimed that using Venn diagram, relationship between wind turbine statuses can be clearly represented.

1.2. Wind Turbine Fault Detection

Wind turbines consist of several components and subassemblies which are likely to fail during its course of operations. Even with the advanced SCADA systems, certain faults are difficult to characterize, or often the alarm is triggered when the fault is already occurred. Thus, fault detection is critical in identifying faults in the system in a timely manner.

Signal processing units (SPU) evaluates the process parameters and classify them into normal operations and fault situation. In this approach fault indicators are derived from process measurements via limit and trend checking of the process signals. Fast Fourier Transformation, cepstrum analysis, envelope spectrum etc. are some common approaches that are utilized for signal analysis in frequency domain. Signal processing approaches are suitable for analyzing rotating components, i.e., turbine generators, gearboxes etc.

In system identification, the measured signal is compared against the set values, any significant deviation from the set values indicate fault. For complex processes, where analytical models cannot be aptly applied, artificial intelligence based approaches found its scope. AI based approaches can learn the complex behavior, and therefore, any significant deviation in the behavior will be a fault. Benefits of the AI based fault detection systems include: (1) Avoidance of premature breakdown, (2) Reduction of maintenance costs, (3) Remote diagnosis (Caselitz *et al.*, 2006). Thus, research utilizing the machine learning based approaches in fault detection is a viable option.

In the literature, turbine components namely turbine blades, generators, and gearbox are widely researched. Description about them is provided below

1.2.1. Turbine blades fault analysis

Blades play an important role in the wind power generation. Wind turbine blades are continuously adjusted to capture the maximum power.

In the literature, research has been focused on various control aspects of blade pitch. Analysis on both real and simulated data is found in the literature. Literature focusing on blade pitch is divided into two parts (1) Optimization and control of blade pitch for maximum power gain, and (2) Analysis of blade pitch faults. Muljadi and Butterfield, (2000) performed simulated study on the pitching mechanism of wind turbine blades in turbulent wind conditions. As per their analysis, electrical power can be controlled at any rpm; however, the limiting factor is the rating of power converter, and generator. They also found out that in higher wind speed region, the rotor speed must be controlled to avoid rotor speed being increased beyond the control limit. Kusiak *et al.*, (2011) optimized wind power and vibrations in drive train and tower by controlling the blade pitch angles. Ruba *et al.*, (2009) studied the fault tolerant mechanism of wind turbines in simulated environment. They combined two simulations environment namely FLUX 2D and Simulink to analyze the effect of different windings faults. As per their conclusion, by increasing the number of rotor poles, separating the phases/channels, setting new connections between the existing windings and using a complex control system can improve the fault tolerance in wind turbine drive train. Ganeriwala and Richerdson, (2011) analyzed blade pitch on wind turbine from structural health monitoring perspective. They introduced both edge and surface crack in 4 feet long turbine blades to study its impact on wind turbine performance using frequency response functions (FRFs). They considered model shape, frequency, and mode as their model parameters to validate their testing on cracked blade.

Normally servomotors are used to control the pitch mechanism. Overall safety of the mechanism can be assured by current and time measurement and difference in pitch angle differences. Due to non-stationary nature of the process, model based approaches are mostly preferred. A process based on trend analysis is used by Verbruggen, (2003), where the residual of the process and an estimator of output signals are used to determine the changing characteristics of the pitch mechanism (Figure 1.4 (a)). Another similar

trend analysis approach based upon model parameters is also used, whereby the related parameters of the model are continuously estimated and compared with the measured input/output values to determine any deviation in the system's characteristics (Figure 1.4 (b)).

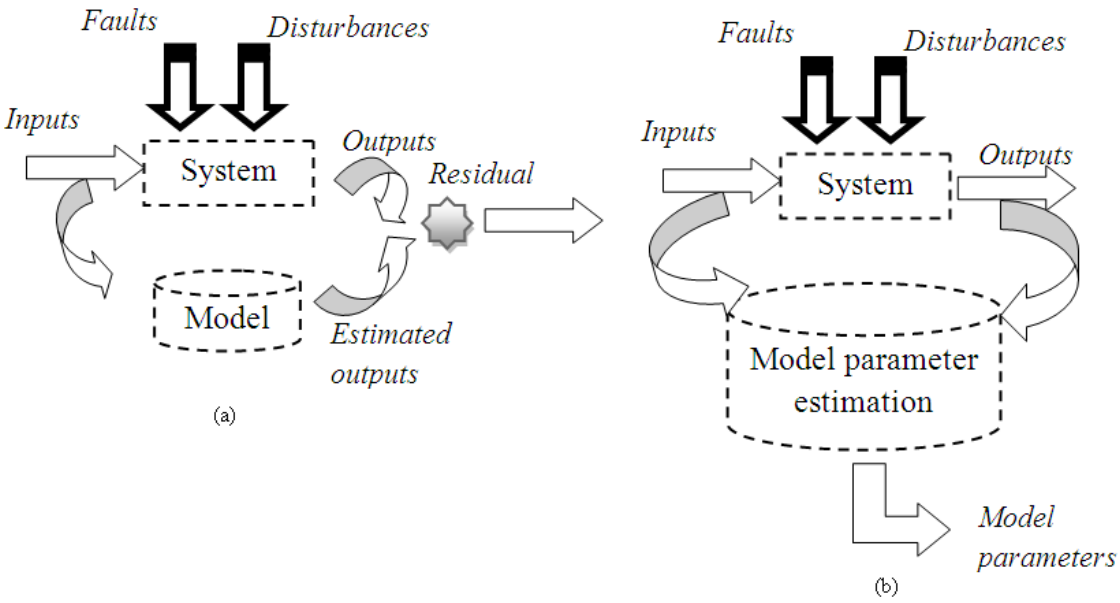


Figure 1.4. Trend analysis of pitch mechanism (a) residual based, (b) model parameters based.

1.2.2. Generator fault analysis

The generators of wind turbines consist of various sub-components which fails/degrades over time. Among them, generator brushes, bearings are severally affected. Klein and Lali, (1990) performed failure mode and effect analysis (FMEA) on wind turbine generators. They recommended need for safety device for various wind turbine rotating components including shaft over-speed, and gearbox vibration etc. They also suggested a need of disc brake on high speed shaft to minimize the effects of shaft failure. Numerous vibration based analysis is performed on generator bearings. Amirat *et al.*, (2010) performed fault detection in doubly feed induction generators (DFIG) bearings using first

intrinsic mode function. Using their proposed method, they improved the classical amplitude demodulation techniques for fault detection. Casadei *et al.*, (2006) developed a fault diagnostic system based upon rotor modulating signals.

1.2.3 Gearbox fault analysis

Gearboxes are one of the most sensitive and costly equipment of wind turbine with high downtime. Most of the early wind turbine design errors in gearboxes arise due to the underestimation of its operating loads. Main reason behind this is the lack of sufficient analysis on its design, operability and load prediction, which requires an effective collaboration between component manufacturers, lubricating engineers, and gear designers etc. The collaboration has resulted in internationally recognized design standards. Designers do a lot of modifications in their actual design, which can be good up to some extent but it cannot guarantee that the component will not fail. Literature survey reveals that on an average twenty percentage of the wind turbine downtime is due to gearbox failure and which needs more than ten days for its complete repair. . Other factors contributing to gearbox failures are torque overloads, wrong material type, damage during transportation and assembly, and misalignment of component etc.

Ribrant, (2006) did a literature survey about the reliability of the gearboxes and concluded that even though the percentage failure due to the gearboxes has reduced, the downtime of the system has increased. His survey was based upon the data collected from three different countries viz. Finland, Germany, and Sweden. Out of several possible causes of gearbox failure, misalignment between gear and generator, in-appropriate bearings are the most common. The process of gearbox failure is gradual which depends upon the equipment wear rate and therefore conditional monitoring has been used. Recently, Byon *et al.*, (2009) formulated a Markov decision process to determine the reliability of the gearbox. They developed an operation and maintenance

(O&M) decision model using probabilistic cost modeling approach to quantify risks and uncertainties.

Fault diagnosis in bearings is usually derived from envelope curve analysis using high frequency resonances to identify fault frequencies. Other methods are cepstrum analysis which is basically used to diagnose gearbox. Detailed analysis is reported in Caselitz *et al.*, (1997), where they demonstrated a network based CMS techniques for the fault diagnosis of turbine gearboxes and rotor. A simulation based test was performed by them to determine the effectiveness of their approach. Approaches based upon demodulation of current signal of an induction motor which drives gearbox is widely used to determine the fault associated with it. Applied both amplitude and frequency demodulation of induction motor current to determine the rotating shaft frequencies which was used as a base to identify faults in gearboxes. Being non-stationary in nature, the approach seems to be very interesting. Garcia *et al.*, (2006) used an intelligent search technique to identify and diagnose faults in gearbox. They validated their model on the real WECS and developed an optimized maintenance schedule.

1.3 Performance monitoring of wind farm

Supervisory Control and Data Acquisition (SCADA) system routinely collects wind turbine operations data which can be used for performance monitoring purposes. Even with the advanced SCADA systems, wind turbine faults are often recognized too late to perform a planned maintenance on the system. Data mining approaches are well-known to extract the hidden patterns in the data. With the data-mining algorithms, faults associated with wind turbines can be identified and predicted well ahead of their occurrence. In addition, the performance of wind turbines can be continuously monitored using the operational data such as power output, rotor speed, blade pitch angle etc.

Over the past few years, data-mining has been successfully applied in manufacturing, marketing, and medical informatics (Harding *et al.* 2006; Berry and

Linoff 2004; Shah *et al.* 2006). In the energy sector, data-mining based algorithms were used to forecast electricity market price (Jhao, 2007), optimization of combustions and heating, ventilation, and air conditioning (HVAC) systems (Kusiak *et al.*, 2010 a, b; Song and Kusiak (2010). In the wind energy itself, data-mining based approaches are used for (1) Optimization of wind power output

Therefore, in the research of predicting wind turbine faults, data-mining algorithms are employed to identify the association among wind turbine performance parameters. Depending on the nature of turbine fault and available data, both classification and regressions models are constructed. As the performance of data-mining algorithms solely depends on dataset at hand, advanced data preprocessing techniques are employed to develop robust fault prediction models.

At present, there is insufficient study on wind turbine performance monitoring. In this dissertation, performance monitoring of wind turbines is achieved by providing accurate and robust data-mining based fault prediction models. Figure 1.5 illustrates the structure of the thesis. In the present dissertation, three research topics are investigated.

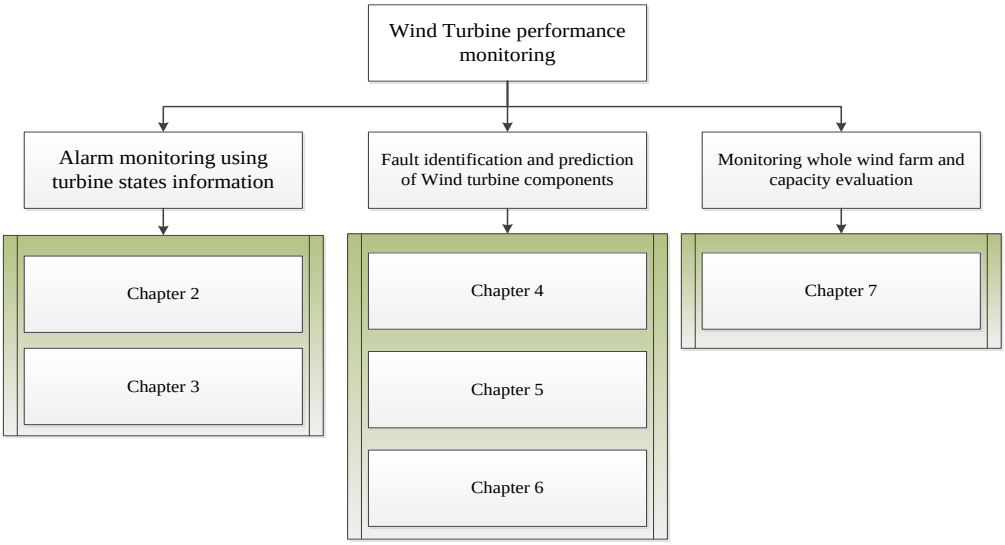


Figure 1.5. Dissertation structure.

The first research topic presented in Chapters 2 – 3 concentrates on identifying the hidden patterns in wind turbine faults. Chapter 2 utilizes the information from the wind turbine fault logs, whereas, in chapter 3, the routinely recorded parameters are analyzed. Chapters 4-6 constitute the second research topic in which the fault prone components of wind turbines are investigated. Faults associated with turbine components namely blades, generators, and gearboxes are analyzed and prediction models are developed. The third research topic presented in Chapter 7 concentrates on monitoring the performance of wind farm.

In Chapter 2, the status patterns in wind turbines are discussed. Wind turbine statuses are recorded whenever turbines changes its course of operation, i.e., operating normally, idling, maintenance downtime. Identifying pattern in the statuses is critical as sequence of statuses occurring over time may lead to component failure. Such statuses are event triggered and recorded as fault logs. Association rule mining algorithm is employed to generate the status patterns with high confidence and support. Later, data-mining based classification models are developed to separate the status pattern class from a normal class.

In Chapter 3, the fault prediction performance is improved using turbine states information. Unlike fault logs, the turbine states information are recorded within the operational data, and therefore reducing any risk of information loss while labeling the output data for prediction. The prediction process comprises of three phases, e.g., phase 1 where the states of wind turbines are predicted, phase II where the turbine fault modes are predicted, and phase III, where unseen/unlearned faults are identified.

In Chapter 4, faults related with wind turbine components i.e., blades and generators are analyzed. The fault associated with the turbine blades is blade angle implausibility, and blade angle asymmetry. Worn out of generator brushes is the second fault. Due to the nature of fault and data-limitation, the imbalance in the output class exists. In class imbalance problem, often the desired class (i.e. fault class) is

misrepresented, when compared with the other (i.e. normal class). Approaches based on cost sensitive classification, and instance selection is used. A heuristic is developed to identify the costs associated with misclassifying fault as normal, and vice-versa. Later, considering the NP-hard nature of instance selection problem, Genetic algorithm based optimization approach is employed. The results reveal the significant improvements in the prediction accuracy.

In Chapter 5, generator bearing temperature of wind turbines is analyzed with an aim to develop early alarm signals in the event of over temperature related issues. A method based on anomaly detection is employed. The standardized signal error is analyzed and any significant deviation is identified using the control chart based approaches. In Chapter 6 vibration analysis is performed to identify and predict faults in wind turbine gearbox system. Based on the data analysis, faults in ring gear of turbine gearbox is identified and predicted. Analysis in both time and frequency domain is performed to validate the severity of identified faults. Virtual models predicting vibrations and jerk in ring gear is developed using the information from a single and multiple sensors.

Chapter 7 proposes an effective and computationally efficient performance monitoring scheme to track the progress of wind farm. Three performance curves using wind turbine operational data namely power output, rotor speed, and blade pitch angle are developed. The reference curves for each of the performance curves are constructed by utilizing third and fourth order moment information, namely kurtosis and skewness. The reference curves are smoothed by detecting outliers in the data using Mahalanobis distance. Kurtosis and skewness of bivariate data provides a single value which can be easily tracked in 2-d scatter graph. Later, multivariate control chart approach is employed to track the progress of out of track wind turbines in time order. A case study monitoring the wind farm over a period of four years is presented.

Chapter 8 provides the overall summary of the dissertation with future research directions.

CHAPTER 2

IDENTIFICATION OF CRITICAL STATUS PATTERNS OF WIND TURBINES

2.1 Introduction

The developments in wind energy have materialized in the form of large-scale wind farms, wind energy cooperatives, wind turbines owned by individual investors, and multinational exploration of remote sites and offshore locations. Despite the increasing rated capacity of wind turbines, operations and maintenance (O&M) costs remain high due to failures of wind turbine components such as gearboxes and blades. Wind turbines undergo various states during its course of operations. Supervisory control and data acquisition (SCADA) system records the state changes in the form of status codes. A status has potential to become fault in the future if not properly addressed. Despite its potential related with wind turbine fault diagnosis, the relevant studies are limited. Chen *et al*., (2011) developed an expert system identifying the alarm pattern related with blade pitch status. However, the study developed by them was based on one day simulated data. In real world, wind turbines are surrounded by several such statues that need to be investigated. More specifically, the sequential occurrence of such statuses may leads to a fault.

This chapter investigates various statuses that a wind turbine can undergo. The status information is event triggered and is recorded by SCADA system along with numerous wind turbine parameters. Decision rules are developed using association rule mining approach to identify critical status patterns in wind turbines (Agrawal and Srikant, 1994).). Further, prediction models are constructed by merging the event triggered status information with time-stamped wind parameters.

2.2 Problem background and dataset description

2.2.1 Dataset description

The data used in this chapter is collected from the supervisory control and data acquisition (SCADA) system of a wind farm. A typical SCADA system records data on more than 100 parameters averaged over 10 min intervals (10 min data). Two different data types, i.e. operational data and status data are studied. The operational data include 10 min data of parameters such as power, wind speed, rotor speed, and generator speed recorded. The status data include the status codes, status description, wind speed, and power and are recorded at the time the system undergoes a status change. In total, 12 months of status data and 4 months of operational data is used.

2.2.2 Problem background

A wind turbine includes assemblies, systems, and components that may fail. A component failure usually develops in stages over a period of time (see Fig. 2.1). Changes in the values of turbine parameters are reported as status codes. For a typical large-scale turbine, over 400 different status codes can be generated. A status indicates a potentially emerging fault. The factors contributing to a fault can be internal (e.g., operational parameters, temperature, and vibrations) and external (e.g., extreme weather conditions). Depending on the severity of the problem, a status code may trigger three types of alarms: information, warning, and failure.

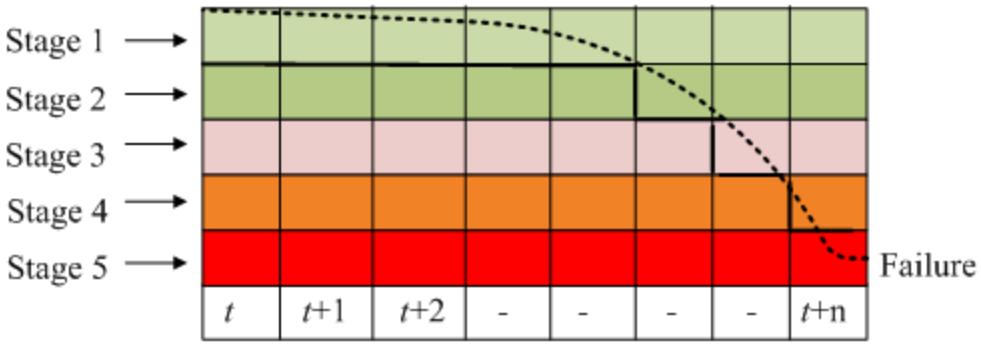


Figure 2.1. Component degradation curve.

The status code data is grouped in four categories, category 1 through 4, with category 1 being the most severe one (Kusiak and Li, 2011). Examples of status codes are provided in Figure 2.2. A category 4 status usually represents an inconsequential event during the

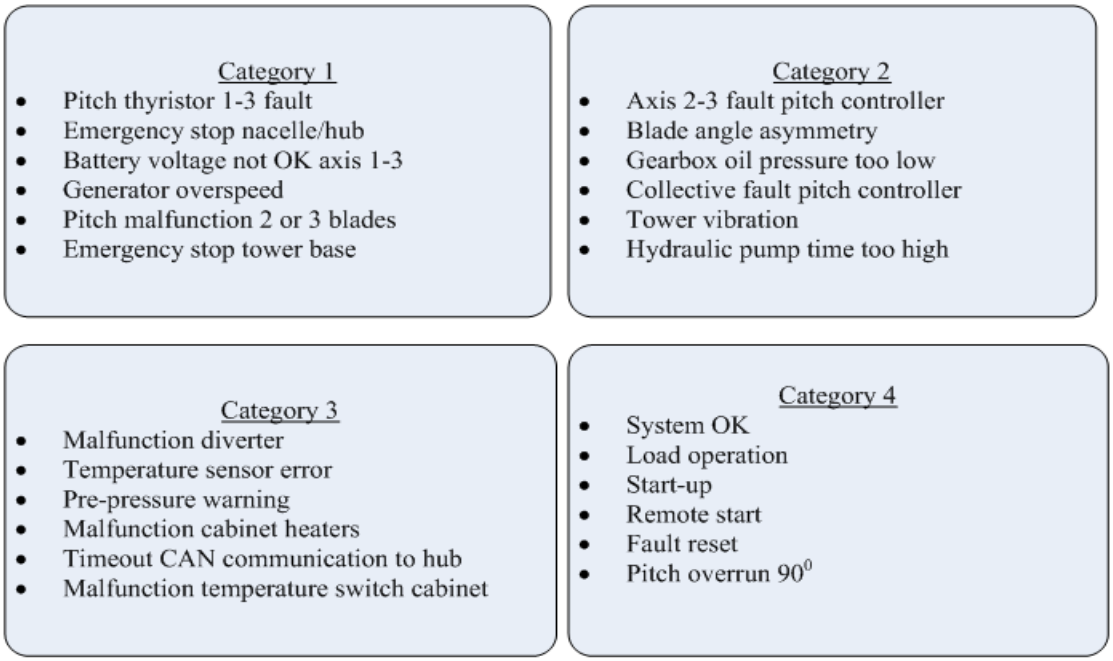


Figure 2.2. Example status descriptions of four categories.

normal operations of a wind turbine (Figure 2.2). In the analysis performed in this chapter, status codes corresponding to categories 1, 2 and 3 are considered.

The status data from 100 wind turbines has been analyzed. Figure 2.3(a) provides four discrete distribution functions, e.g., negative binomial, geometric, logarithmic, and Poisson of category 1 status data. The probability-probability plot in Figure 2.3(b) indicates the category 1 status closely follows the Poisson distribution. The distribution statistics of all four statuses are summarized in Table 2.1. The status frequency varies for each status category, with category 4 being the most frequent and category 1 status the least frequent. Accurate prediction of patterns involving category 1 statuses is most desirable.

Table 2.1. Distribution statistics of all category statuses.

Status Category	Distribution	No. Data Points
1	Poisson ($\lambda = 10.98$)	100
2	Poisson ($\lambda = 31.61$)	100
3	Poisson ($\lambda = 4.77$)	100
4	Poisson ($\lambda = 40.48$)	100

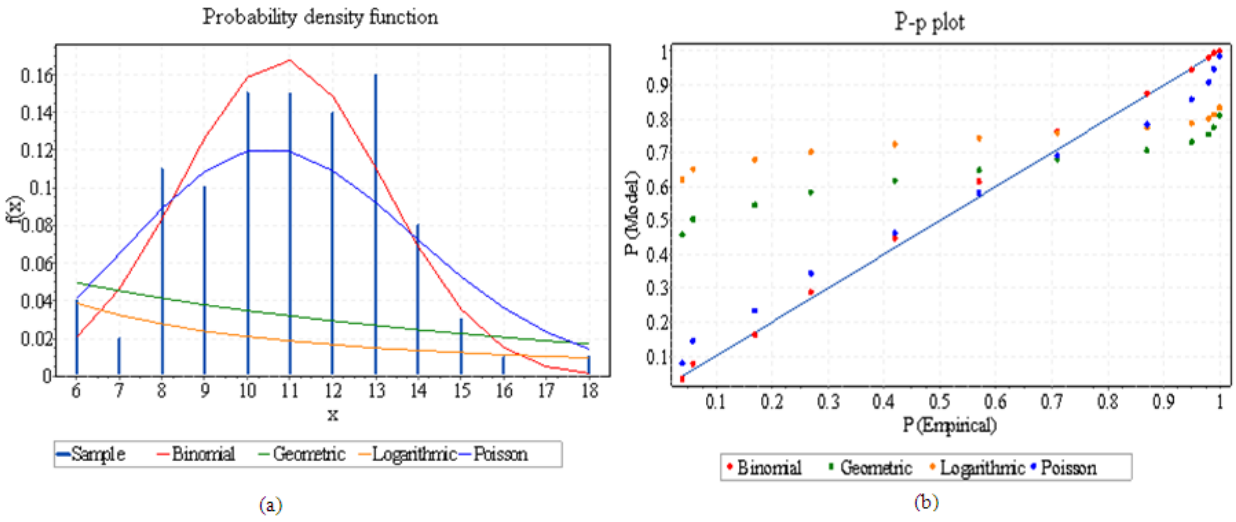


Figure 2.3. Category 1 status data for 100 turbines: (a) histogram and five probability density functions, (b) probability-probability (P-P) plot.

In general, individual statuses do not provide much information regarding the system/component health; rather the sequence of the statuses (called here a status pattern) occurring over time may lead to the component failure. Thus, early prediction of status patterns is needed for effective condition monitoring and maintenance.

2.3. Solution approach

This section presents a new approach to fault prediction (see Figure 2.4). The status data from 100 turbines is used to identify frequent status patterns, whereas the data obtained from the 10 representative turbines is merged with the SCADA operational data to generate a dataset for the prediction of status patterns.

Let S= {S_1, S_2… S_n} be the set of n statuses, TS= {t_1, t_2 ….t_m} be the set of time

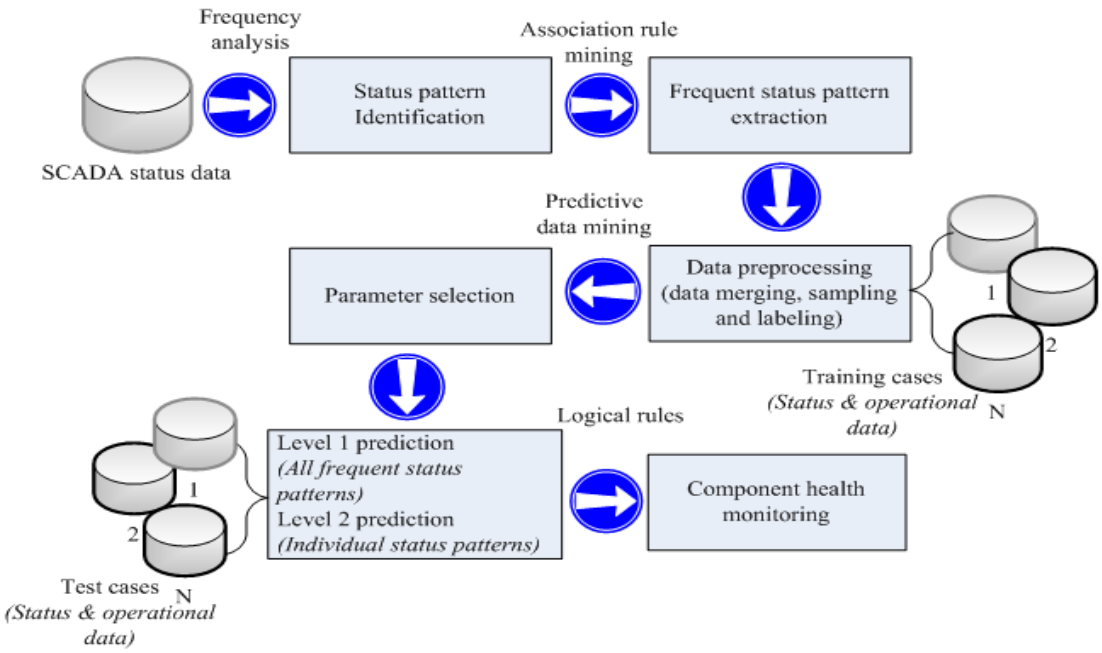


Figure 2.4. Framework for the prediction of status patterns.

when the status are recorded. Each TS contains a subset of status from S. A status pattern can be defined as $A \Rightarrow B$, where $A, B \subseteq S$, and $A \cap B = \varnothing$. The details of the framework of Figure 2.4 are presented next.

2.3.1. Frequency analysis

The aim of this step is to identify status patterns using data from 100 wind turbines. To accomplish this task only fault informative statuses (e.g., categories 1 through 3) are considered. The status data of 100 turbines are analyzed to determine patterns consisting of two of more statuses (see Figure 2.5). The limit on the time delay (Δ) is set to 60 second, e.g., the statuses recorded within 60 second are considered as a possible status

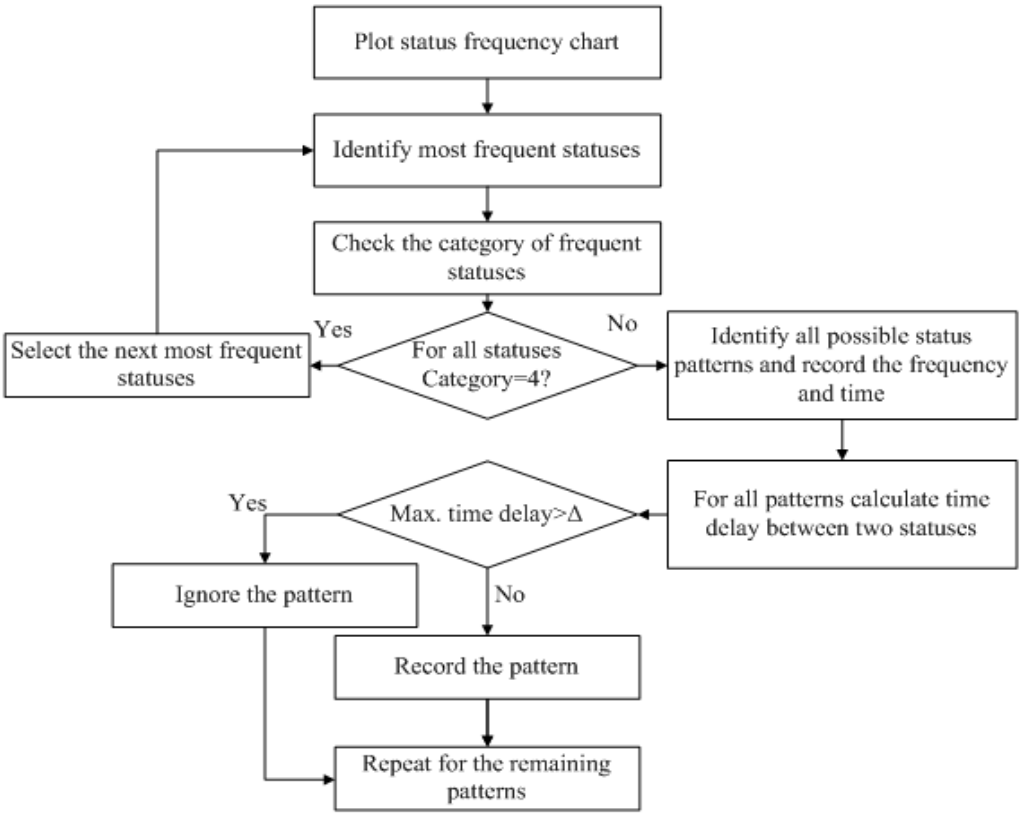


Figure 2.5. Flow chart for identification of status patterns.

pattern. In total 16 frequent status patterns have been identified. Figure 2.6 (a-e) provides the distribution of 16 status patterns across 100 wind turbines.

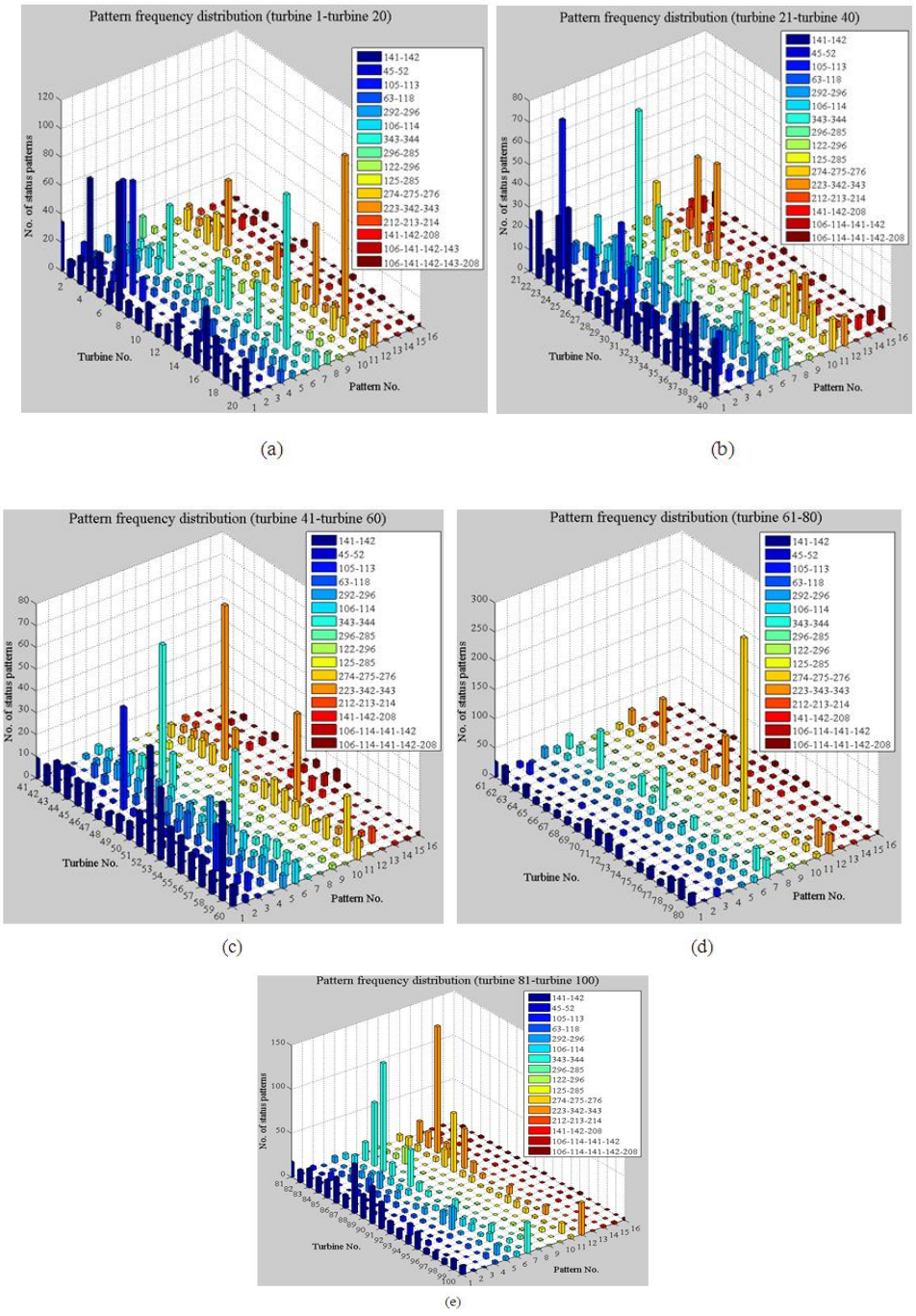


Figure 2.6. Frequency distribution of identified status patterns: (a) Turbine 1-20, (b) Turbine 21-40, (c) Turbine 41-60 (d) Turbine 61-80, (e) Turbine 81-100.

The identified status patterns along with their distribution statistics are summarized in Table 2.2. Among the identified status patterns, the rotor and turbine blade based status patterns are the most frequent. The histogram of all 100 turbines corresponding to the status patterns is shown in Figure 2.7. Status patterns closely follows a negative binomial distribution (n = 2, p = 0.02018). Ten most fault affected turbines e.g. 05, 17, 22,25,46,64,70,73,84, and 87 were considered for mining with association rule algorithms and fault prediction.

2.3.2. Association rule mining

Association Rule Mining (ARM) is used to determine interesting relations among parameters in a large dataset (Agrawal and Srikant, 1994). ARM has been used in web usage mining (Cho *et al.*, 2002), intrusion detection (Luo and Bridges, 2000), and bioinformatics (Besemann *et al.*, 2004). In this section, frequent patterns are determined with the apriori algorithm (Agrawal and Srikant, 1994; Agrawal *et al.*, 1993). The following metrics (see Equations 2.1-2.2) are used by the apriori algorithm to determine the goodness of a rule (e.g. status pattern).

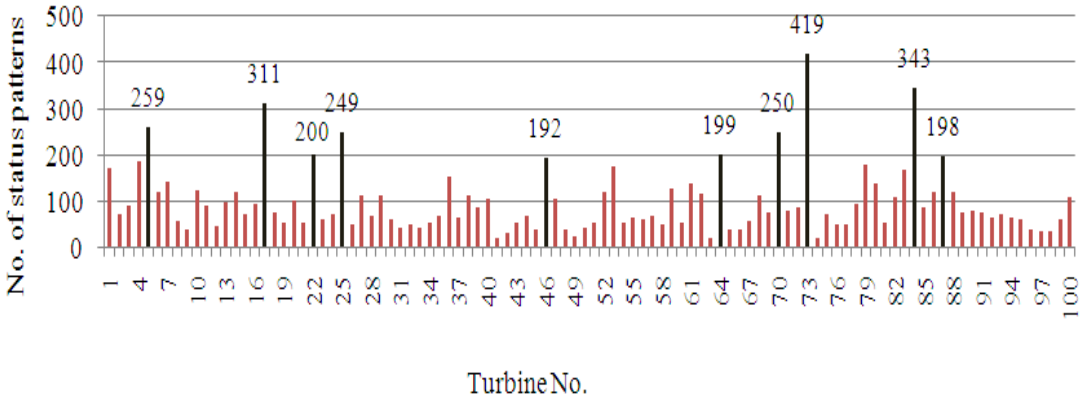


Figure 2.7. Frequency plot of all status patterns identified in 100 turbines.

Table 2.2. Identified status patterns.

No.	Status pattern (Category)	Description	Distribution
1	141(1), 142(2)	Rotor CCU collective faults, Line CCU collective faults	Neg. binomial (n = 2, p = 0.10072)
2	45(2), 52(2)	Hydraulic pump time too high, Gearbox oil pressure too low	Poisson (λ = 1.09)
3	105(2), 113(2)	Rotor CCU fault voltage, Line CCU fault voltage	Geometric (p = 0.139)
4	63(1), 118(1)	Safety chain, Emergency stop nacelle /hub	Poisson (λ = 3.47)
5	292(3), 296(3)	Malfunction cabinet heaters, Malfunction diverter	Geometric (p = 0.118)
6	106(2), 114(2)	Rotor CCU fault current, Line CCU fault current	Neg. binomial (n = 4, p = 0.442)
7	343(2), 344(1)	Blade angle not plausible axis 3, Pitch malfunction 2 or 3 blades	Geometric (p = 0.066)
8	296(3), 285(3)	Malfunction Diverter, Timeout CAN communication to hub	Poison(λ = 0.5)
9	122(2), 296(3)	Collective fault pitch controller, Malfunction of diverter	Poisson(λ = 1.23)
10	122(2), 285(3)	Collective fault pitch controller, Timeout CAN communication to hub	Poisson(λ = 0.28)
11	274(1), 275(1), 276(1)	Pitch thyristor 1 fault, Pitch thyristor 2 fault, Pitch thyristor 3 fault	Geometric (p = 0.0658)
12	223(2), 342(2), 343(2)	Blade angle not plausible axis 1, Blade angle not plausible axis 2, Blade angle not plausible axis 3	Geometric (p = 0.067)
13	212(1), 213(1), 214(1)	Battery voltage not OK axis 1, Battery voltage not OK axis 2, Battery voltage not OK axis 3	Poisson (λ = 0.9)
14	141(2), 142(2), 208(2)	Rotor CCU collective faults, Line CCU collective faults, No activity CAN-Bus CCU	Poisson (λ = 1.46)
15	106(2), 114(2), 141(2), 142(2)	Rotor CCU fault current, Line CCU fault current, Rotor CCU collective faults, Line CCU collective faults	Poisson (λ=1.23)
16	106(2), 114(2),141(2), 142(2), 208(2)	Rotor CCU fault current, Line CCU fault current, Rotor CCU collective faults, Line CCU collective faults, No activity CAN-Bus CCU	Poisson (λ = 1.23)

$$Sup_i = \frac{FP_i}{TFP} \tag{2.1}$$

$$Conf\,(A \Rightarrow B) = \frac{Sup(A \cup B)}{Sup(A)} \times 100 \tag{2.2}$$

In expression (2.1), Sup_i is the support of i^{th} status pattern, FP_i is the frequency of ith status pattern in a yearly status data, and TFP is the total status patterns found. Conf (A→B) in expression (2) represents the confidence of the status pattern *AB* consisting of status *A* and *B*. Confidence can also be interpreted as an estimate of probability $P(A/B)$.

Based on the identified status patterns, the minimum support was set equal to 10 (which signify a status pattern appearing less than 10 times in a year is discarded). The support value of 10 was selected as per the average frequency of the individual status patterns. Confidence threshold was set equal to 80% (signifies at least 80% of the time the statuses form a unique pattern). The association rule mining is performed on the data from 10 selected turbines and the common and frequent status patterns are selected for prediction at different time stamps.

2.3.3. Identification of frequent status patterns

The association rule mining algorithm is applied on the status data obtained from 10 most fault prone wind turbines. Using the mentioned threshold on support and confidence, more than 25 different rules (status patterns) were found for each turbine. The pitch thyristor, the pitch malfunction, and the blade pitch angle related status patterns were the most frequent patterns with the corresponding support greater than 100 and confidence equal to 100%. These status patterns are listed in Table 2.3, and are used for level 2 predictions discussed in Section 2.4.2.

Table 2.3. Frequent status pattern (support >100, and confidence =100%).

No.	Status Pattern (Category)	Description
1	343(2)=>344(1)	Blade angle not plausible axis 3=>Pitch malfunction 2 or 3 blades
2	274(1)=>275(1)=>276(1)	Pitch thyristor 1 fault=> Pitch thyristor 2 fault=> Pitch thyristor 3 fault
3	223(2)=>342(2)=>343(2)	Blade angle not plausible axis 1=> Blade angle not plausible axis 2=> Blade angle not plausible axis 3
*support>100 and confidence = 100%		

2.3.4. Data preprocessing

The data collected from a wind farm contains some noise due to sensor errors and malfunctions. Inconsistent data, e.g., abnormal wind speed, is deleted. After filtering the raw data, the final dataset for 10 turbines over a period of four months (e.g., from 1/1/2008 to 4/30/2008) was produced.

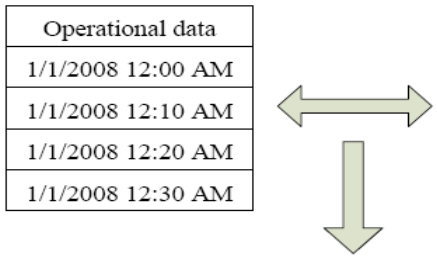
The approach described in the previous steps is used to determine the frequent status patterns where status data is used. However, the status data is merged with the operational data (recorded at 10-min intervals) to produce time-series dataset for prediction. The following decision variable is used to label the output of the prediction dataset (see Equation 2.3).

$$\Xi_i = \begin{cases} \text{status pattern}, & \text{if } t_o \le t_s \le t + i_o, i \in \{10, 20, 30, \ldots 60\} \\ \text{normal, otherwise} \end{cases} \quad (2.3)$$

In equation (2.3), i is the index of time stamps, Ξ is a decision variable labeling output as *status pattern* in the prediction dataset at time t_o, if the status pattern occurs (t_s) in between the t_o and next time stamp $t + i_o$, otherwise, the output label will be *normal*. The generation of the dataset is illustrated in Figure 2.8, where para *1*, para *2*,…, para *n*

represent the input parameters such as wind speed, power, generator speed, and so on which are obtained from operational data.

Status patterns are rare compared with the normal operations of a wind turbine. Merging the data across 10 wind turbine increases the number of status pattern instances, however, it also increases the number of normal cases, resulting in a highly imbalanced dataset. Sampling techniques, such as cost sensitive classification (Drummond and Holte, 2000), under-sampling (Liu *et al.*, 2008), and over-sampling (Liu and Ghosh, 2007), etc. are widely used in the literature to improve the prediction accuracy of imbalanced datasets. Cost sensitive classification assigns a penalty to the false prediction of a minority class in order to minimize misclassification errors (Drummond and Holte, 2000; Margineantu, 2000). In under-sampling, only a subset of data from the majority class is selected for the analysis. Over-sampling augments the size of the dataset by generating duplicate samples of the minority class.



Operational data
1/1/2008 12:00 AM
1/1/2008 12:10 AM
1/1/2008 12:20 AM
1/1/2008 12:30 AM

Status data
1/1/2008 12:08 AM
1/1/2008 12:35 AM
1/1/2008 01:46 AM
1/1/2008 02:20 AM

Final data set

Time Stamp	Para 1	Para 2	Para *n*	Output
1/1/2008 12:00 AM	----	----	----	Status pattern
1/1/2008 12:10 AM	---	---	---	Normal
1/1/2008 12:20 AM	---	---	---	Normal
1/1/2008 12:30 AM	----	----	----	Status pattern

Figure 2.8. Description of the dataset generation.

Due to the highly imbalanced dataset, a combination of over-sampling and under-sampling techniques, namely, (1) spread subsamples (to reduce the size of normal instances) and synthetic minority oversampling technique (SMOTE). A SMOTE based sampling technique resamples the data by applying a synthetic oversampling technique to the minority class dataset (e.g., class status patterns in the present case) (Chawla *et al.*, 2002).

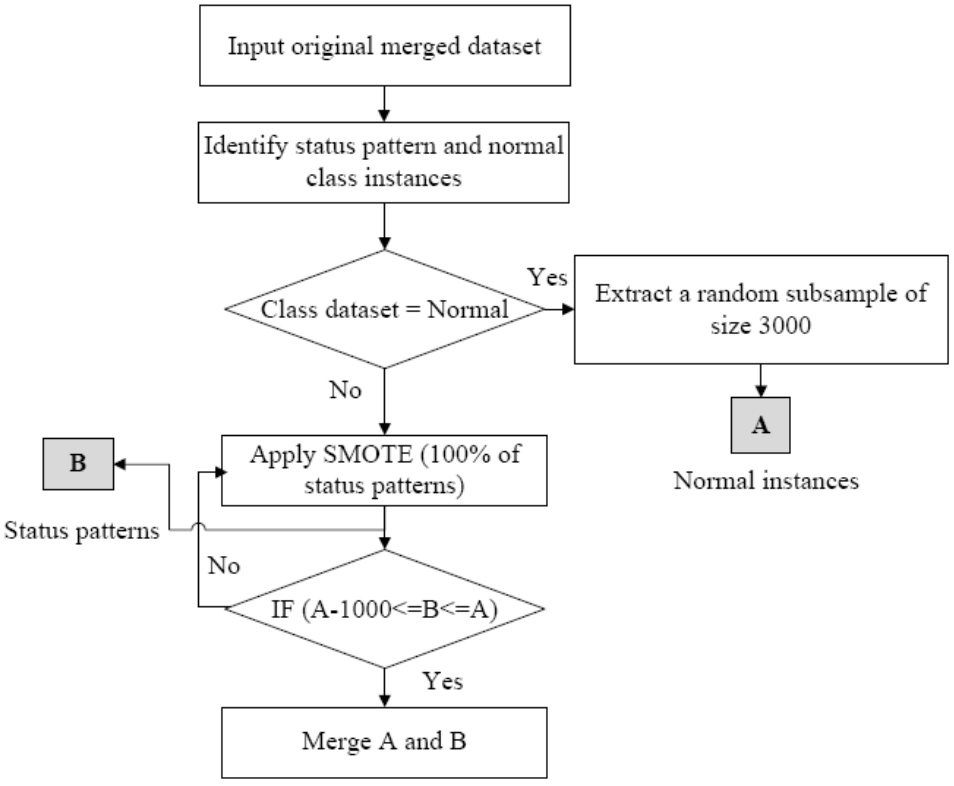


Figure 2.9. Data sampling steps.

In this chapter, initially a subsample of 3000 data points from majority class dataset (e.g., dataset with output label ‘normal’) is extracted and then SMOTE is repeatedly applied to the monitory class data (e.g., data with output label ‘status pattern’) until the number of instances satisfies the pre-defined limit. It was assumed that the ratio

of 60:40 (normal: status pattern) is sufficient for mining. Figure 2.9 illustrates the data sampling approach used in this research.

2.3.5. Parameter selection

For accurate predictions, the dimensionality of the dataset needs to be reduced, as not all the parameters are significant for the prediction. The removal of the unnecessary parameters improves prediction accuracy of the model and at reduced computational cost. Knowledge of the process is helpful in elimination of parameters that are not significant. The SCADA system usually records more than 100 parameters of a wind turbine. The recorded parameters can be grouped into three categories: (1) system related data, (2) control parameters, and (3) performance parameters. System related data, e.g., turbine number, index, time offset, is turbine specific and therefore it can be excluded from building a prediction model. Control parameters represent the desired values such as, set points of blade 1 to 3, torque set values, etc. Whereas, the performance parameters indicate the performance of a wind turbine necessary for prediction of status patterns. Usually the parameters directly collected by SCADA are used for status pattern predictions. In some cases they are transformed, e.g., absolute deviation might be used. They are referred to as derived parameters, e.g., blade 1-3 deviations (Figure 2.10).

Further reduction in the data dimensionality can be accomplished with data analysis. Algorithms such as the boosting tree algorithm (Smola and Scholkopf, 2004), the principal component analysis (PCA) (Jollife, 1986), and the wrapper approach integrated with the genetic or the best-first search algorithms (Tan *et al*., 2006; Espinosa *et al*., 2005) are widely used for parameter selection.

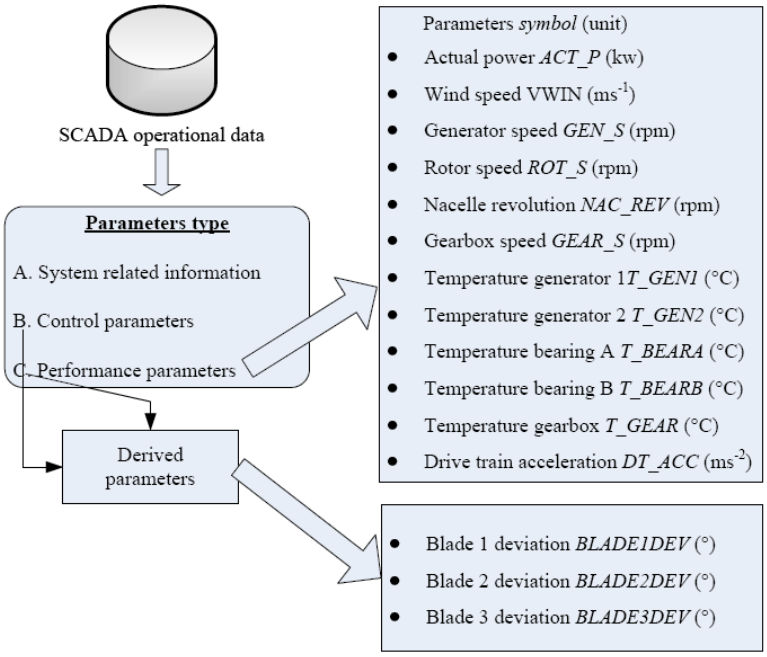


Figure 2.10. Wind turbine parameters selected for prediction of status patterns.

PCA is an unsupervised learning approach for dimensionality reduction that uses correlation coefficients of the parameters to combine and transform them into a reduced dimensional space (Miranda *et al*., 2008). It employs a ranker-based search algorithm to select the principal components. In this research, PCA was applied to select 15 parameters. The parameters, temperature generator 1, temperature generator 2, generator speed, and rotor speed are found be highly correlated. The six principal components shown in Table 2.4 explain 96.11% of the total variance; therefore, they are selected to build the prediction model. The dimensionality of the dataset is reduced from fifteen to six dimensions. The same principal components are used to reduce the dimensionality of all time stamped datasets.

Table 2.4. Eigen value of principal components determined by PCA.

Principal component	Eigen value	Variance (%)	Cumulative variance (%)
PC1	8.2865	55.24	55.24
PC2	2.03399	13.56	68.8
PC3	1.5629	10.42	79.22
PC4	0.99603	6.64	85.86
PC5	0.89348	5.96	91.82
PC6	0.64344	4.29	96.11

2.3.6. Metrics for prediction accuracy

The selection of prediction algorithms is evaluated on a test dataset. The metrics used in the analysis are based on the widely used confusion matrix illustrated in Figure 2.11.

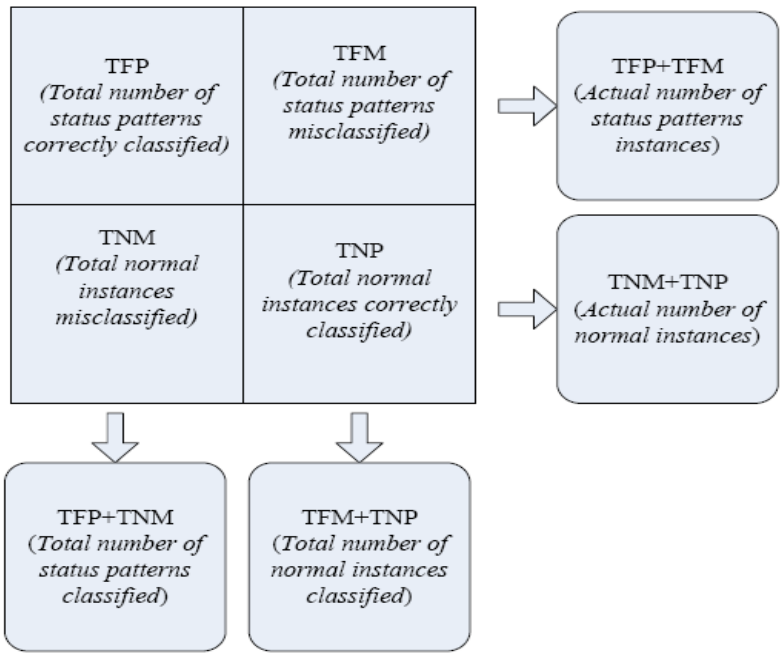


Figure 2.11. The confusion matrix for the performance evaluation of algorithms.

The metrics used to evaluate accuracy of the prediction model are listed in Equations (2.4)-(2.7).

$$Accuracy = \frac{TFP + TNP}{TFP + TFM + TNM + TNP} \quad (2.4)$$

$$Senstivity = \frac{TFP}{TFP + TFM} \quad (2.5)$$

$$Specificity = \frac{TNP}{TNM + TNP} \quad (2.6)$$

$$\Psi = \frac{(w1 \times Accuracy) + (w2 \times Sensitivity) + (w3 \times Specificity)}{w1 + w2 + w3} \quad (2.7)$$

The algorithm producing the best results based on the weighted prediction accuracy (ψ) defined in (2.7) is used for further analysis, where w_1, w_2 and w_3 are the weights associated with accuracy, sensitivity and specificity, respectively.

2.3.7. Algorithm selection

Five data-mining algorithms, bagging (Brieman, 1996), ripper (Cohen, 1996), rotation forest (Rodrigues *et al.*, 2006), random forest (Brieman, 2001), and k-nearest neighbor (k-NN, k =10) (Aha and Kibler, 1991), are used to construct the prediction model. Bagging is an ensemble meta-algorithm combining different classifiers. The ripper algorithm reduces classification errors by incremental pruning of an inductive rule algorithm to predict output classes. Rotation forest uses parameter selection to construct a classifier using a base learner and a projection filter. Random forest involves numerous decision trees and can be used for both classification and regression. The *k*-NN classifies objects based on the training examples in parameter space, whereby, objects are classified by the majority of votes from their neighbors.

The performance of the five data-mining algorithms for prediction of frequent status patterns was examined at t + 30 time stamps using the metrics (2.4)-(2.7). The best performing algorithm was used to build the prediction models at all six time stamps. Sampling of the large imbalanced dataset resulted in 5326 instances (see Table 2.5). Table 2.6 summarizes the prediction accuracy of the five data-mining algorithms.

Table 2.5. Dataset description for algorithm selection.

Dataset	Start Time Stamp	End Time Stamp	Description
t+30_*overall*	1/1/2008 12:00 AM	4/30/2008 11:30 PM	5326 observations
t+30_*training*	1/1/2008 12:00 AM	3/21/2008 09:30 PM	3512 observations
t+30_*test*	3/21/2008 09:60 PM	4/30/2008 11:30 PM	1814 observations

Table 2.6. Accuracy of data-mining algorithms predicting all frequent status patterns at time stamp t + 30.

Algorithm	Accuracy (%)	Sensitivity (%)	Specificity (%)
Bagging	93.88	94.3	93.6
Random forest	94.88	96.8	93.4
Ripper	88.58	88.4	88.8
k-NN	88.31	99.2	80.2
Rotation forest	94.32	95.5	93.5

Different weight values w_1, w_2, and w_3, in the rage (0-1), were assigned to accuracy, sensitivity and specificity, respectively, to identify the best performing algorithm. The random forest algorithm outperformed the remaining data-mining algorithms (see Figure 2.12) by providing the consistently best output. The weights assigned to the evaluation metric display the preference towards a particular output class. Performance of the three algorithms, bagging, rotation forest, and ripper, was found to be consistent for various weight values, however, less accurate; whereas the *k*-NN algorithm was found to be more sensitive to weight values. Based on the observation,

the random forest algorithm is selected to build the prediction model at various time stamps. Figure 2.13 compares the prediction output obtained by the random forest algorithm at time stamp t + 30 with the actual output. The dense clusters along the normal-normal (predicted-actual) and status pattern-status pattern (predicted-actual) axis depict that both normal and status pattern instances are well classified, whereas the sparse clusters along the normal-status pattern (predicted-actual) and the status pattern-normal (predicted-actual) axis indicate few misclassified cases.

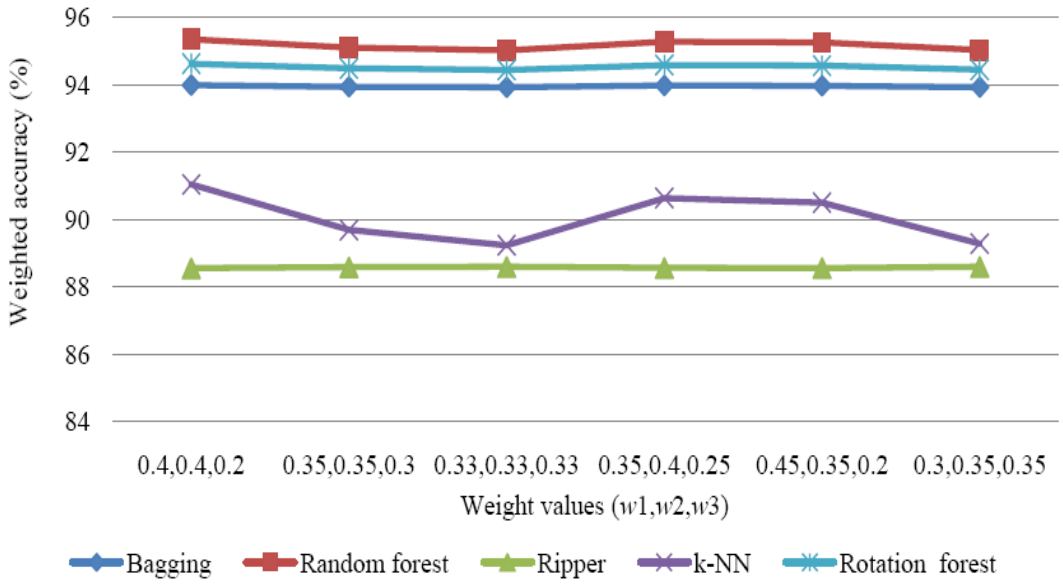


Figure 2.12. Prediction accuracy of five data-mining algorithms for different weight values.

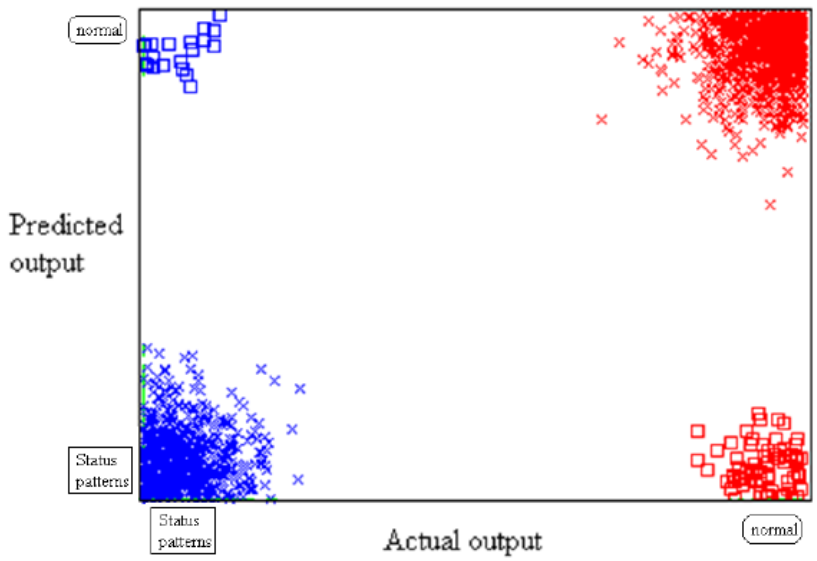


Figure 2.13. Comparison of the output classes (actual and predicted) by the random forest algorithm at time period t+30.

2.4. Computational results

In this section, various experiments are considered to analyze accuracy and robustness of the proposed approach.

2.4.1. Level 1 prediction

Level 1 prediction corresponds to the prediction of all three frequent status patterns obtained with association rule algorithm (see Table 2.3). The random forest algorithm is used to build the prediction model at six time stamps. A total of six predictive models were built, and the results obtained are shown in Table 2.7. The results indicate that the algorithm is consistent in accuracy and sensitivity at all-time stamps.

Table 2.7. Prediction accuracy of all status patterns with the random forest algorithm.

Time Stamp	Accuracy (%)	Sensitivity (%)	Specificity (%)
t + 10	80.4	87.1	73.4
t + 20	80.5	83.9	77.7
t + 30	76.6	79.1	74.5
t + 40	79.3	85.0	73.1
t + 50	78.6	80.5	77.1
t + 60	76.5	77.6	75.7

2.4.2. Level 2 prediction

Level 2 prediction corresponds to the prediction of the individual status patterns. Here, 18 prediction models were built with the random forest algorithm. Tables 2.8-2.10 summarize prediction accuracy of the models built for different horizons. The prediction accuracy for the random forest algorithm is in the range of 84.36%-96.08% with accuracy of predicting status patterns in the range 85.4%-96.4%. Figures 2.14 illustrates the comparison between actual output and predicted output of one such status pattern e.g. 274=>275=>276 predicted at time stamp t+10 and t+60. The relatively dense distributions of data points along the normal-normal (actual-predicted) and the status pattern-status pattern (actual-predicted) axis indicate that classification errors are much less.

Table 2.8. Prediction accuracy of the status pattern 274=>275=>276 with the random forest algorithm.

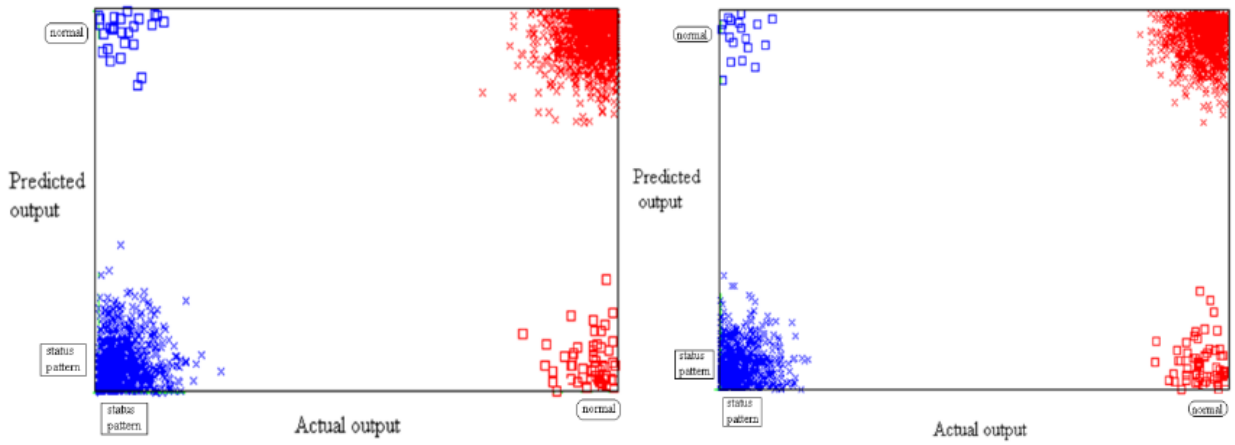
Time Stamp [min]	Accuracy (%)	Sensitivity (%)	Specificity (%)
t + 10	95.6	96.8	94.7
t + 20	95.3	97.1	94.0
t + 30	94.8	96.8	93.4
t + 40	94.1	96.3	92.4
t + 50	96.0	97.4	95.1
t + 60	93.7	95.0	92.9

Table 2.9. Prediction accuracy of the status pattern 343=>344 with the random forest algorithm.

Time Stamp [min]	Accuracy (%)	Sensitivity (%)	Specificity (%)
t + 10	86.1	88.2	84.2
t + 20	86.2	89.8	83.3
t + 30	85.2	90.1	81.0
t + 40	85.4	86.1	85.0
t + 50	85.5	88.6	82.8
t + 60	84.6	85.4	84.1

Table 2.10. Prediction accuracy of the status pattern 223=>342=>343 with the random forest algorithm.

Time Stamp [min]	Accuracy (%)	Sensitivity (%)	Specificity (%)
t + 10	86.7	88.4	85.3
t + 20	87.5	90.5	85.1
t + 30	86.5	90.6	83.1
t + 40	84.3	87.8	81.4
t + 50	86.9	89.2	85.1
t + 60	84.6	87.8	85.7



(a) (b)

Figure 2.14. The actual and the predicted status pattern 274=>275=>276: (a) t+10, (b) t+60.

2.4.3. Component performance monitoring

The monitoring scheme uses the predicted output to generate an alarm signal. At this time the alarm signal can be displayed up to 60 min ahead of the adverse event. A voting approach using the predicted output at all six time stamps is adopted to develop the performance monitoring system. Let η be the frequency of prediction output "status pattern" at six time stamps and ϵ_i be the possible output at time stamp i, ($i \in \{t + 10, t + 20, ..., t + 60 \text{ min}\}$. ϵ can be predicted as normal or status pattern depending on the input data. If $\eta >= 3$, and $\epsilon_i = \epsilon_{i-1} = \epsilon_{i-2}$= status pattern, the monitoring system will generate any alarm signal as *error*. However, if $2 <= \eta <= 3$ and $\epsilon_i = \epsilon_{i-1}$ = status pattern, the alarm signal will be *warning*, and for $1 <= \eta <= 2$ the alarm signal will be *information*. Otherwise, if $\eta = 0$, the alarm signal will be just a *system update*. Considering six time periods, there can be 64 (e.g., 2^6) possible prediction outcomes. It is important to mention that displaying alarm signals requires the prediction output at six consecutive time stamps. For displaying signals 60 min ahead, data mining algorithm will be used to get output at all 6 time stamps, whereas, for displaying alarm signals 50 min ahead or earlier, both past output information and data mining algorithm based prediction will be used.

Status patterns occurring at only one time stamp only appear to be as critical as their repeated sequence that might be an indication of a system or a component failure. In this section, analysis of the component monitoring approach is performed. One thousand (out of more than two thousands) cases are selected to form three different datasets. The distribution of the alarm signals for an example status pattern 274=>275=>276 is shown in (Figure 2.15).

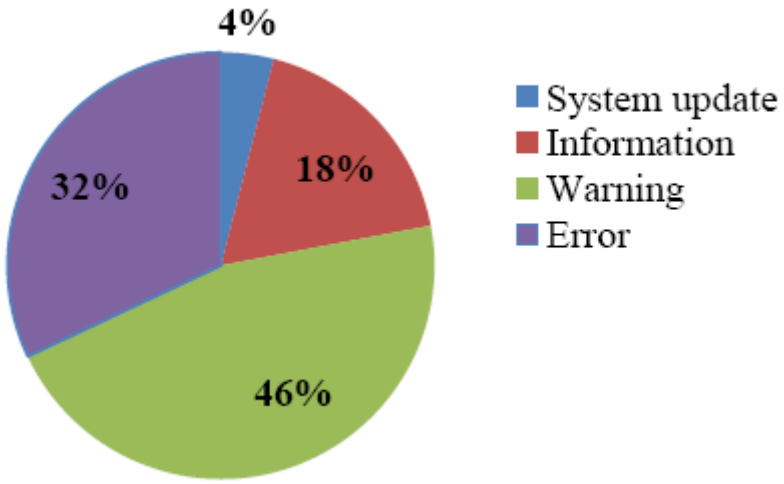


Figure 2.15. Distribution of alarm signals of the status pattern 274=>275=> 276.

Figure 2.15 indicate that alarm signals such as error and warning are more frequent than the system update or information signals in all three status patterns. Similar distribution was found for two other status patterns. Table 2.11 illustrates component monitoring scheme for status pattern 274=>275=>276 on turbine 73. Component monitoring scheme will use both past and future output values to generate alarm signal at time t.

2.5. Summary

A data-mining approach was applied to identify and predict status patterns of wind turbines. The identification of status patterns is important, as the system or component may not fail instantly, yet its health may gradually deteriorate. Early prediction of status patterns allows for predictive maintenance actions and possible avoidance of some faults. A prediction model was built using operational and status data collected at wind turbines.

Table 2.11. Alarm signals for status pattern 274=>275=>276 of turbine 73.

Date	Time	Time Stamp	Prediction Output	Alarm Signal	Time Interval considered
1/11/2008	12:00:00 AM	t - 30	Status pattern		
1/11/2008	12:10:00 AM	t - 20	Status pattern		
1/11/2008	12:20:00 AM	t - 10	Normal		
1/11/2008	12:30:00 AM	t*	Normal		
1/11/2008	12:40:00 AM	t + 10	Normal	Warning	{t - 40, t - 30,…, t+10}
1/11/2008	12:50:00 AM	t + 20	Normal	Warning	{t - 30, t - 20 ,..., t + 20}
1/11/2008	1:00:00 AM	t + 30	Status pattern	Warning	{t - 20, t - 10,..., t + 30}
1/11/2008	3:30:00 AM	t + 40	Status pattern	Warning	{t - 10, t,…, t + 40}
1/11/2008	4:00:00 AM	t + 50	Status pattern	Warning	{t, t + 10,..., t + 50}
1/11/2008	4:30:00 AM	t + 60	Status pattern	Error	{t + 10, t + 20,..., t + 60}
*Time for displaying alarm signals					

An apriori algorithm identified frequent status patterns. The principal component analysis was transformed the 15 dimension datasets into 6 dimension datasets. Of the five data-mining algorithms tested, the random forest algorithm was selected for building a prediction model. A component performance monitoring scheme was developed to generate alarm signals based upon the outputs predicted at different time intervals.

CHAPTER 3

ENHANCED PREDICTION PERFORMANCE OF WIND TURBINES USING STATES INFORMATION

3.1 Introduction

Chapter 2 discussed about the identification and prediction of status patterns using fault logs. While merging the fault log data with operation data, some information is lost.

In this chapter, fault described in time-stamped operational data is discussed. Apart from fault and normal classes, other commonly occurring states of wind turbines are maintenance downtime, and weather downtime.

While most of the literature related with wind turbine fault analysis is based on two-class classification models. Accurate prediction of all states of wind turbines is important in order to reduce the false alarms and miss. In this chapter, multiclass classification models are developed and accuracy for all four states of wind turbines is maximized by maximizing the overall geometric mean of individual output class. The four broad states of wind turbines are (1) Turbine OK, (2) Turbine off due to poor weather conditions, (3) Turbine off due to maintenance, and (4) Turbine off due to faults/abnormalities. Wind turbine SCADA systems record the event triggered fault logs in operational data as fault status. Obtaining fault specific information with the operational wind turbine data minimizes loss of information that is found while merging the event triggered information with the time-based operational data.

In this chapter, enhanced monitoring of wind turbines is done through the states information. The fault states of wind turbines are further analyzed in phase II prediction by breaking down them into the specific fault.

Later, to test the learning ability of data-mining algorithms, some unobserved faults are tested. This is done to ensure that the faults showing the same behavior can be identified irrespective of weather, data-mining algorithms were trained on it or not.

Figure 3.1 provides the distribution of turbine states (analyzed on a yearly wind turbine data). Wind turbines are in fault mode for almost 7.0% of the time, which is quiet significant. Thus, special attention is required to minimize the impact of the faults on wind turbine to improve its availability.

The results presented in the chapter are based on the analysis of data obtained from 17 wind turbines. The values of parameters recorded at 10-second intervals (10 s data) over a four-month period constitute the dataset for this research. Overall, three level prediction models are analyzed with an aim to predict any kind of wind turbine states, to predict specific and the frequent states, and to identify unseen states of the wind turbines.

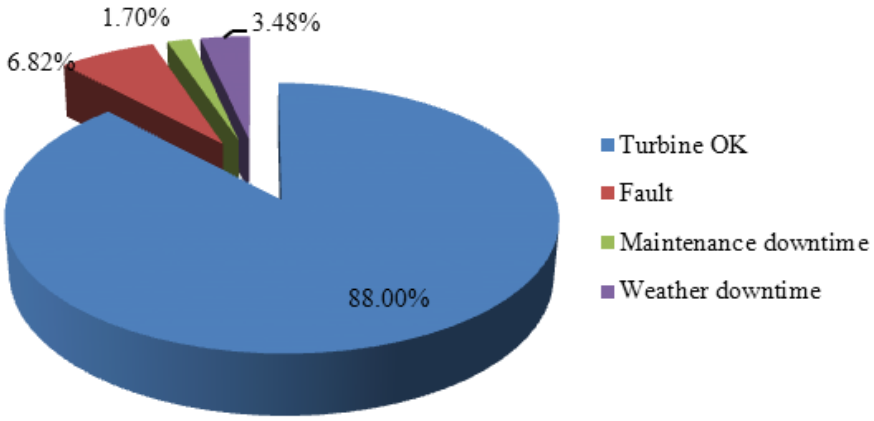


Figure 3.1. Distribution of wind turbine states.

3.2. Models for monitoring wind turbine states

The framework for building prediction models is provided in Figure 3.2. An abstraction of turbine states is used to categorize the output data into a number of states using expert knowledge. Model building involves using various data mining algorithms. The models are then tested. The generated dataset is used to construct models for phase-I and phase-II predictions. The main objective of phase-I is to predict a fault of any kind, whereas, predictions in phase-II target specific faults. In phase-III predictions, unseen faults from different wind turbines are identified. Descriptions of various wind turbine states are provided next.

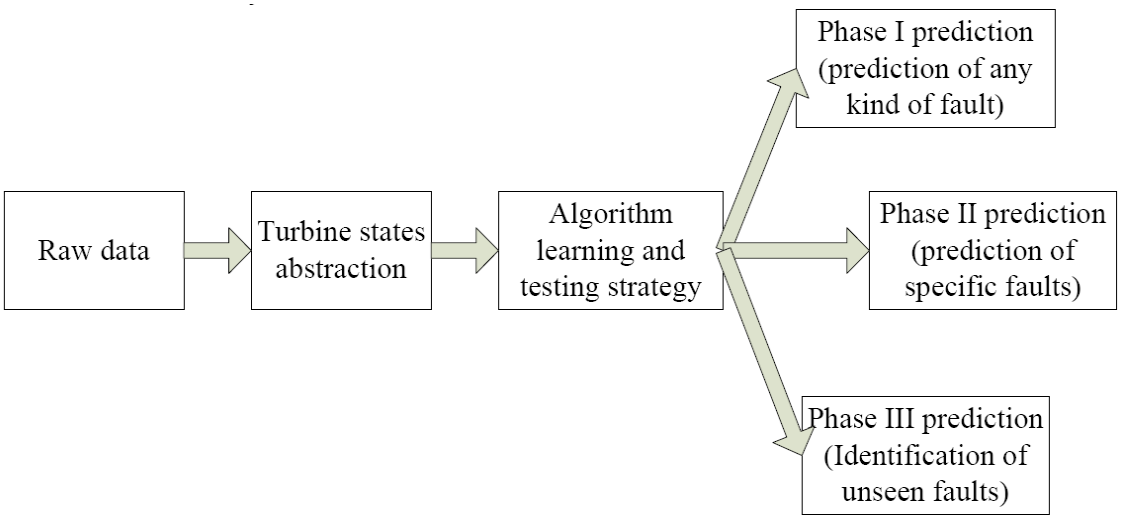


Figure 3.2. Framework of the proposed approach.

3.2.1 Turbine states description

The variability of wind speed impacts the performance of wind turbines and is recorded as fault states. Normal operations, weather-related downtime, maintenance downtime, fault mode, and emergency stop are some of the many states recorded by the SCADA system of a wind turbine. States changes may vary from insignificant (e.g., when a

turbine is changing its state from idle to normal operations) to a potential fault. Table 3.1 lists the 17 possible states of a wind turbine. State number 17 represents the fault mode of wind turbines and there can be more than 400 possible ways in which a wind turbine can be faulted. Gearbox oil over-temperature, blade angle asymmetry, pitch thyristor fault, and yaw runaway are some of the common fault modes of a wind turbine. In the research reported in this chapter, the main emphasis is to predict the fault mode of wind turbine ahead of actual occurrence.

Table 3.1. Turbines state information.

State Number	State Description	State Number	State Description
1	Turbine OK with no errors	10	Turbine stopped locally
2	Turbine running smoothly	11	Emergency stop
3	Turbine running up idling for cut in	12	Turbine stopped due to curtailment
4	Turbine in maintenance mode	13	Turbine stopped by customer
5	Turbine in repair mode	14	Turbine idling locally
6	Power failure/grid downtime	15	Turbine idling remotely
7	Weather downtime	16	Wind direction curtailment
8	Turbine stopped externally	17*	Turbine in fault mode
9	Turbine stopped locally		
* Primary focus			

3.2.2 Abstraction of turbine states

A typical turbine may undergo a number of different states including turbine normal operations, run-up idling, maintenance/repair mode, fault mode, weather downtime, etc. The prediction of a turbine's fault mode is of particular interest as it represents some potential fault in the system. Figure 3.3 shows the histogram of 17 wind turbines plotted over a period of four months (from 8/27/2010 to 12/4/2010). Based on the frequency of

fault mode, turbine 12 was considered in the analysis. In order to reduce the computational effort required by data-mining algorithms, the recorded states of wind turbines were further categorized using domain knowledge. Table 3.2 represents the initially recorded and categorized states of turbine 12. The initial 44 turbine states were categorized into four states: *Turbine OK, Fault, Weather downtime,* and *Maintenance downtime*. The *Turbine OK* category corresponds to normal functioning operations, including run-up idling, whereas, the *Fault* category corresponds to an actual or potential fault in the system. The *Weather downtime* category corresponds to turbine downtime due to poor weather conditions, whereas, any other downtime is considered as *Maintenance downtime*.

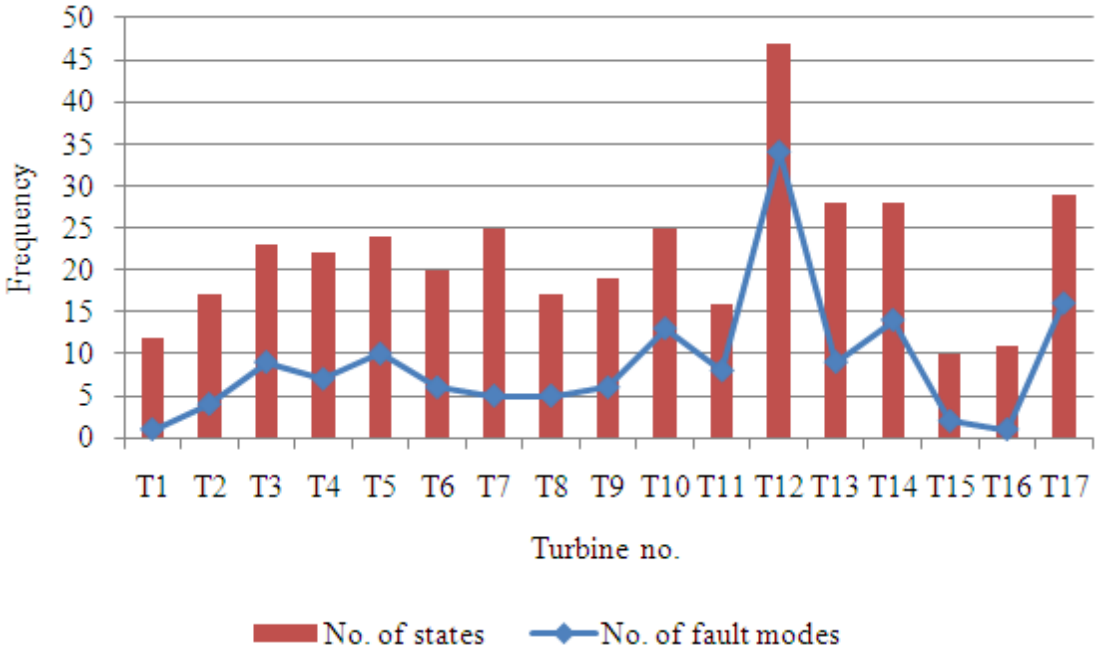


Figure 3.3. Comparison of wind turbines states.

Table 3.2. Turbine state categories.

Fault Mode States		States Other than Fault Mode	
State description	Output class	States description	Output class
Asymmetric generator current	Fault	Online	Turbine OK
Axle 1 fault pitch controller	Fault	Run-up idling	Turbine OK
Battery charging rotor blade drive	Turbine OK	Maintenance mode	Maintenance downtime
Cable twisting left	Turbine OK	Repair mode	Maintenance downtime
Cable twisting right	Turbine OK	Grid downtime	Maintenance downtime
Centrifugal switch	Fault	Turbine curtailment	Maintenance downtime
Gearbox oil over temperature	Fault	Stopped externally	Maintenance downtime
Gearbox oil temperature too low	Weather downtime	Stopped locally	Maintenance downtime
Hydraulic pump time too high	Fault	Stopped remotely	Maintenance downtime
Limit switch 90°-rotor blade defective	Maintenance downtime	Weather conditions	Weather downtime
Maintenance switch pitch	Maintenance downtime		
Maximum motor power	Fault		
Pitch overrun 0°	Fault		
Pitch thyristor fault	Fault		
Pulse sensor rotor monitor defect	Fault		
Reply generator high stage	Fault		
Temperature warning pitch motor	Turbine OK		
Wrong parameter check sum	Turbine OK		

3.3. Learning strategy

For both prediction phases, the dataset was divided into two parts, i.e., initial dataset and blind dataset. The data mining algorithms used two-thirds of the initial data for training,

and the remaining one-third was used for testing. The performance of the data mining algorithms on the test dataset was used for algorithm selection. The best performing algorithm was then used to construct prediction models on the unseen dataset. Details regarding the parameter selections are discussed next.

3.3.1. Selection of relevant wind turbine parameters

A Supervisory Control and Data Acquisition (SCADA) system records more than 100 wind turbine parameters that can be broadly categorized into: (1) wind turbine performance parameters, (2) wind turbine control parameters, (3) wind turbine non-controllable parameters. Parameters such as power, generator speed, and rotor speed are the performance parameters, whereas, blade pitch angle and generator torque are controllable parameters. Wind speed is the only non-controllable parameter. In the research reported in this chapter, a combination of turbine performance parameters, control parameters, and non-controllable parameters are used to predict the wind turbine states. To minimize the curse of dimensionality and to remove irrelevant parameters, a set of data-mining-based parameters selection algorithms are used. A month of data was used for parameter selection and algorithm learning. A stratified subset of the original data was used for parameter selection to make the process computationally efficient. Figure 3.4 displays the original and stratified data. Distribution of the output class is preserved in stratified data to avoid bias towards any specific class. Three different data mining algorithms, wrapper with genetic search (WGS) (Kusiak and Verma, 2010; Kusiak and Zhang, 2010), wrapper with best first search (WBFS) (Tavner *et al.*, 2006), and boosting tree algorithm (BTA) (Tavner and Xiang, 2005) were selected to determine relevant parameters for prediction of turbine states. Wrapper is a supervised learning approach using different search techniques to select the relevant parameters by performing 10-fold cross validation. Table 3.3 lists the 10 best parameters from each parameter selection algorithm. Parameters for *nacelle revolution, blade (1-3) pitch angle, current phase C,*

temperature hub, and generator/gearbox speed were finally selected to build the prediction models.

Table 3.3. Selected parameters using data mining algorithms.

No.	**WGS**	**WBFS**	**BTA**
	10 fold cross validation	10 fold cross validation	Parameter importance
1	Nacelle revolution (90)*	Blade 3 pitch angle (actual) (100)*	Blade2 pitch angle (actual) (100)*
2	Blade 3 pitch angle (actual)(90)*	Current phase C (80)*	Blade3 pitch angle (actual) (95)*
3	Current Phase B (70)	Temperature hub (80)*	Blade1 pitch angle (actual) (94)*
4	Nacelle Position (70)	Temp. control box axis 1 (60)	Generator/gearbox speed (86)*
5	Generator/gearbox speed (70)*	Voltage phase C (50)	Generator speed (85)
6	Temperature, bearing B (70)	Generator speed (50)	Rotor speed
7	Temperature top box (°C) (70)	Drive train acceleration (50)	Blade2 pitch angle (set)
8	Power (Actual) (60)	Temperature top box (50)	Blade3 pitch angle (set)
9	Tower deflection (60)	Nacelle revolution (40)*	Blade1pitch angle (set)
10	Wind deviation, 1 sec (60)	Temperature bearing A (40)	Drive train acceleration
*Selected parameters			

3.3.2. Evaluation metric

The evaluation of data mining algorithms is based on the prediction accuracy of each output class. Considering the imbalance in an output class, a weighted accuracy of each output class is used as criteria for selecting data mining algorithms for the prediction task. The evaluation of accuracy is presented in a confusion matrix (see Figure 3.5). Equation (3.1) defines the geometric mean (*gmean*) of the output class.

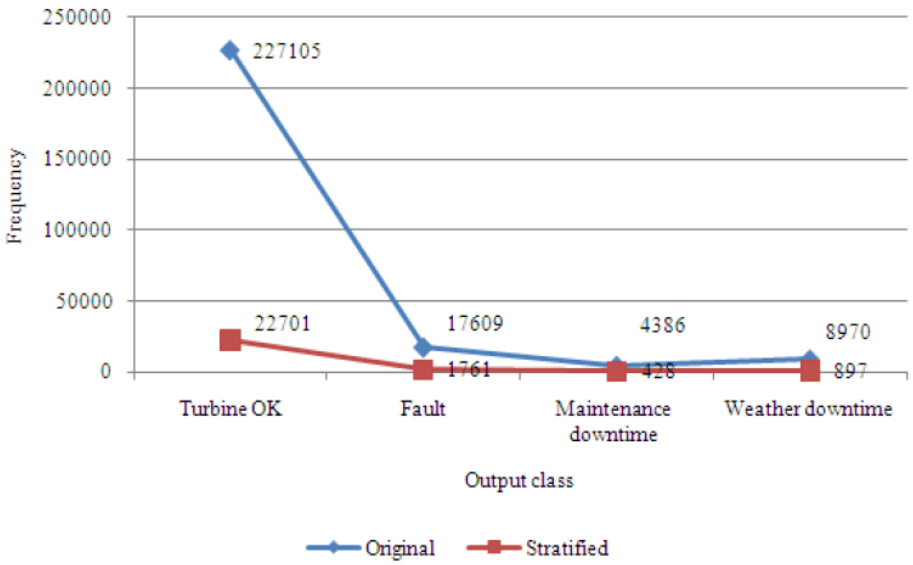


Figure 3.4. Output class distribution.

	Predicted			
Actual	TP11 *(Class1 correctly classified)*	FP21 *(Class2 classified as class1)*	...	FPn1 *(Class n classified as class1)*
	FN12 *(Class1 classified as class2)*	TP22 *(Class2 correctly classified)*	...	FPn2 *(Class n classified as class2)*
	...	...	...	...
	FN1n *(Class1 classified as class n)*	FN2n *(Class2 classified as class n)*	...	TPnn *(Class n correctly classified)*

Figure 3.5. Confusion matrix for multiclass classification.

$$gmean = \sqrt[n]{\prod_{i=1}^{n} acc_i} \tag{3.1}$$

In equation 3.1, acc_i is the accuracy of class *i*, *n* is the total number of output class.

3.3.3. Training algorithms

Five data mining algorithms: neural network (NN), support vector machine (SVM), random forest algorithm (RFA), boosting tree algorithm (BTA), and general chi-square automatic interaction detector (CHAID) algorithm were initially selected for building models at *t* time stamp. The prediction accuracy for each class (phase I predictions) is provided in Table 3.4. Accuracy in the range 95%-99% is obtained by five data-mining algorithms.

Table 3.4. Prediction of turbine states at time stamp *t*.

	Output Class				Overall Accuracy [%]
Algorithm	Turbine OK [%]	Fault [%]	Maintenance downtime [%]	Weather downtime [%]	
SVM	99.06	91.04	30.40	54.48	95.81
CHAID	99.00	89.66	35.48	67.09	96.08
NN	99.52	93.54	62.32	84.90	97.64
BTA	99.69	93.23	83.33	96.99	98.87
RFA	**99.88**	**99.67**	**78.41**	**97.99**	**99.45**

Essentially, all the algorithms performed well while predicting *Turbine OK* and *Fault* class, however, the output class *Weather downtime* and *Maintenance downtime* were predicted with relatively low accuracy. The geometric mean metric, *gmean*, indicates that when all classes are predicted with perfect accuracy its value is 1. The algorithm with the highest value of *gmean* was selected to build prediction models at different time stamps. From the graph in Figure 3.6 and phase-I prediction results (Table 3.4), both the boosting tree algorithm and the random forest algorithms outperformed the remaining three data

mining algorithms. However, RFA was selected to build the prediction models, as it possesses great generalization ability and it is almost insensitive to the size of the dataset. Figure 3.7 illustrates the tree complexity of the random forest algorithm as a function of the misclassification rate. The optimal number of trees was found to be 91. The same five algorithms were considered for constructing phase-II prediction models. The output class, *Fault*, from the phase-I prediction was replaced by actual fault type, resulting in overall 7 output classes. Figure 3.8 displays the distribution of data at time stamp *t*. In the figure, pitch overrun 0° is triggered when limit switch experience a non-positive angle at least one of the rotor blades. Pitch thyristor 2 fault is triggered when the thyristor is not ready even though the grid conductor is switched on. Pitch thyristor fault indicate defective axle cabinet. Axle 1 fault pitch controller reports axle disturbance. Pulse sensor rotor monitor defect is due to no pulses to over speed monitor when the generator over speeds. Table 3.5 displays the performance of different data mining algorithms on a *t* time stamped dataset.

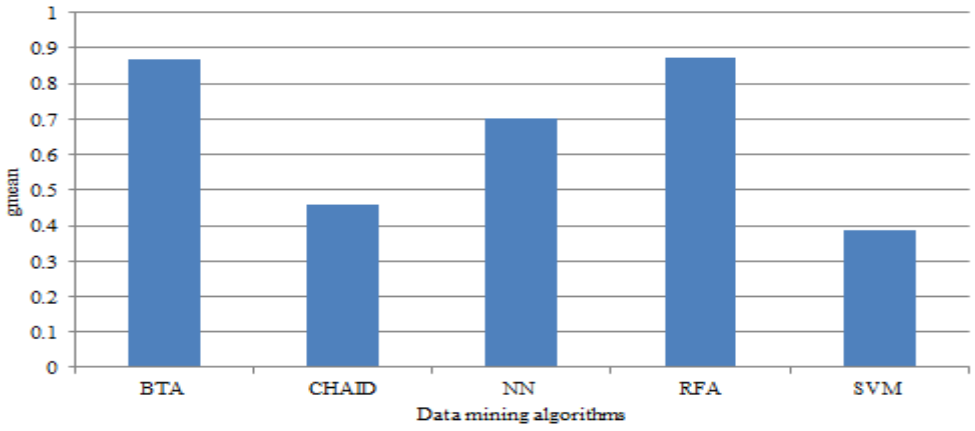


Figure 3.6. Performance of different data mining algorithms using *gmean* as criteria.

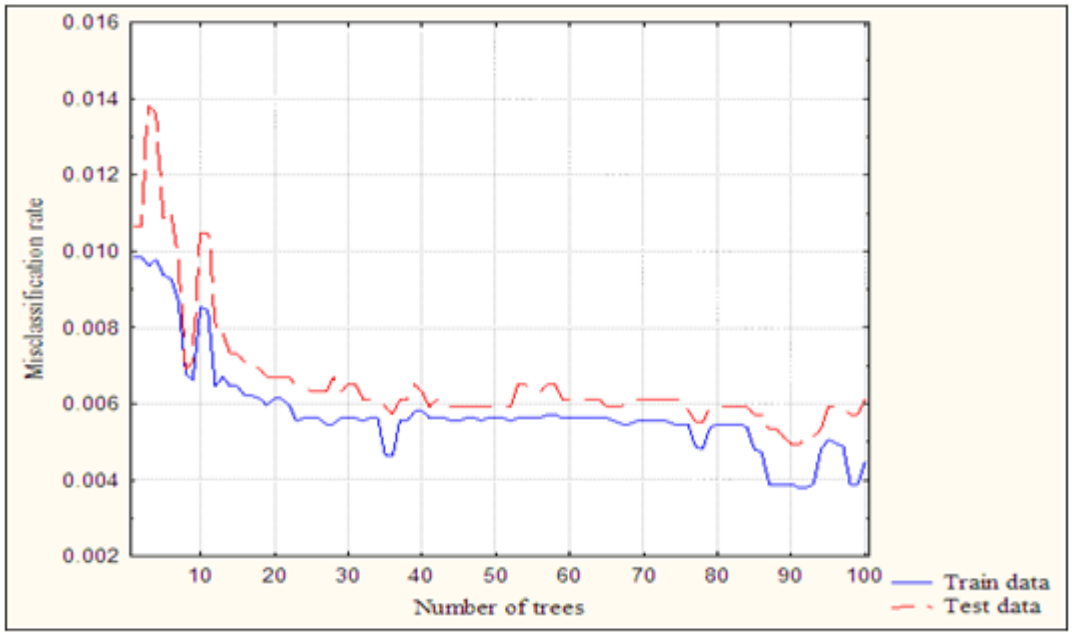


Figure 3.7. Misclassification rate of RFA as a function of tree size.

It can be seen from Table 3.5 that most of the algorithms failed to predict minority output classes (a class with few instances) and thereby resulted in a *gmean* equal to 0. Only NN and RFA yielded a *gmean* value greater than 0 (Figure. 3.9). As anticipated, RFA outperforms the other data mining algorithms, providing better accuracy for each output class.

Table 3.5. Prediction of turbine fault modes at time stamp *t*.

Algorithm	Output Class							Overall Accuracy [%]
	A [%]	B [%]	C [%]	D [%]	E [%]	F [%]	G [%]	
SVM	99.16	26.82	56.73	51.14	67.12	0.00	0.00	93.08
CHAID	98.98	0.00	97.17	100.0	08.30	30.55	0.00	93.42
NN	99.73	87.19	93.10	99.54	97.23	54.16	37.50	98.65
BTA	99.03	84.96	24.46	35.43	67.37	0.00	0.00	92.88
RFA	**99.64**	**82.70**	**99.29**	**100.0**	**98.93**	**87.83**	**61.90**	**98.83**
A: Turbine OK, B: Maintenance downtime, C: Weather downtime, D: Axle 1 fault pitch controller, E: Pitch overrun 0°, F: Pitch thyristor 2 fault, G: Pulse sensor rotor monitor defect								

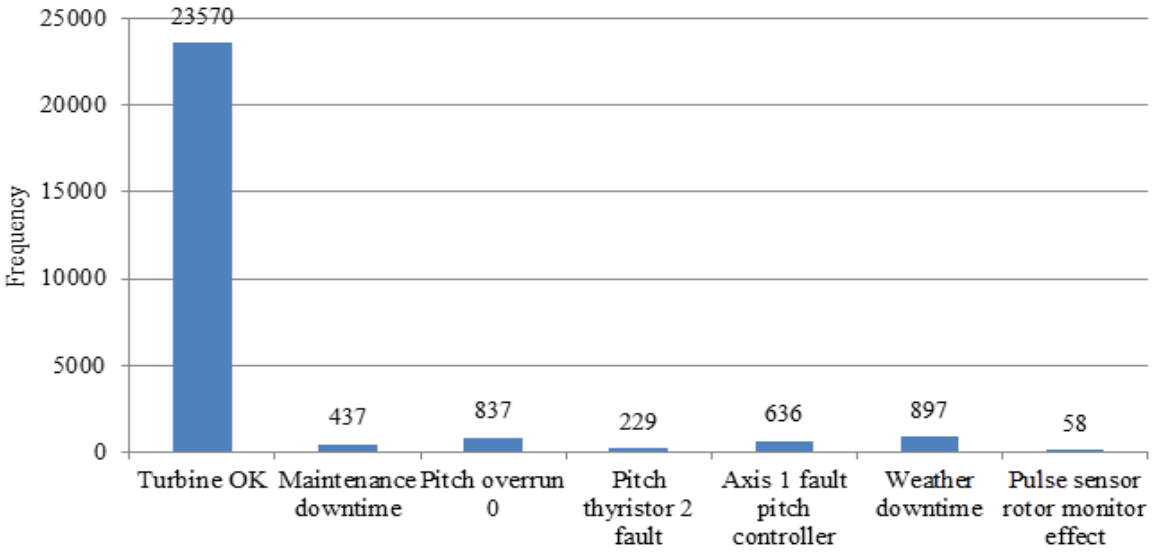


Figure 3.8. Distribution of output class at time stamp *t*.

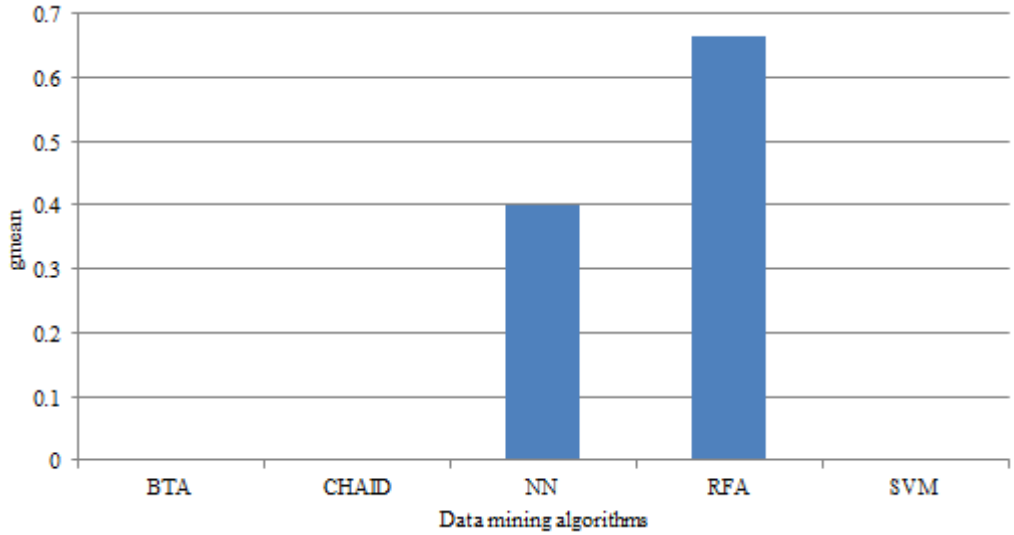


Figure 3.9. Performance of different data mining algorithms using *gmean* as criteria (phase-II prediction).

3.4. Experimentation results

3.4.1. Predicting turbine states

In this section, the random forest algorithm (RFA) was used to build 8 prediction models at various time stamps, with a maximum prediction length of 5 min. The maximum tree size for the random forest algorithm was set to 300. The accuracy was found to be in the range of 81 - 99% for all output classes (Table 3.6).

Table 3.6. Prediction accuracy of output class using RFA (Phase-I prediction).

Time stamp [s]	Output Class				Overall Accuracy [%]
	Turbine OK [%]	Fault [%]	Maintenance downtime [%]	Weather downtime [%]	
t	99.88	99.67	78.41	97.91	99.45
t + 10	99.56	99.00	77.22	95.04	98.39
t + 30	97.64	96.41	74.59	94.62	96.54
t + 60	95.70	95.64	71.67	92.55	94.43
t + 120	91.87	90.00	67.49	88.47	90.89
t + 180	88.58	87.34	64.94	84.43	86.82
t + 240	85.62	84.64	60.31	82.44	83.93
t + 300	83.05	82.76	59.67	80.39	81.76

3.4.2. Predicting turbine fault modes

In this phase, output class *Fault* was replaced with the actual fault types, these being pitch overrun 0°, pitch thyristor 2 fault, axle 1 fault pitch controller, and pulse sensor motor defect. Table 3.7 displays the prediction results obtained using the RFA at different time stamps. The accuracy of each output class was found to be in the range 68 - 100%, except for output class pulse sensor rotor monitor defect for which accuracy was low (e.g., 40.67 - 61.9%).

Table 3.7. Prediction accuracy of output class using RFA (phase-II prediction).

Time stamp[s]	Output Class							Overall accuracy [%]
	A [%]	B[%]	C [%]	D[%]	E [%]	F[%]	G[%]	
t	99.64	82.70	99.29	100.0	98.93	87.83	61.90	98.83
t + 10	99.34	81.08	97.84	98.67	97.77	85.43	58.94	96.44
t +30	97.15	79.26	95.23	97.13	95.52	83.29	55.03	94.09
t + 60	95.28	76.87	92.86	95.45	93.26	80.09	51.98	91.68
t + 120	90.10	71.58	88.41	91.90	90.23	77.67	48.83	87.82
t + 180	87.98	68.71	85.73	86.39	86.88	74.87	46.29	84.69
t + 240	84.45	65.32	83.66	82.55	83.34	71.45	43.91	81.53
t + 300	82.76	62.43	81.45	80.76	79.55	68.32	40.67	78.35
A: Turbine OK, B: Maintenance downtime, C: Weather downtime, D: Axle 1 fault pitch controller, E: Pitch overrun 0°, F: Pitch thyristor 2 fault, G: Pulse sensor rotor monitor defect								

Figure 3.10 displays the *gmean* value of both prediction phases. Phase-I prediction had overall better *gmean* values (0.435 - 0.817) than phase-II predictions (0.242 - 0.659), reason being the poor accuracy of one output class, the pulse sensor rotor monitor defect. In the next sub-section, phase-III prediction is illustrated and unobserved faults are also identified.

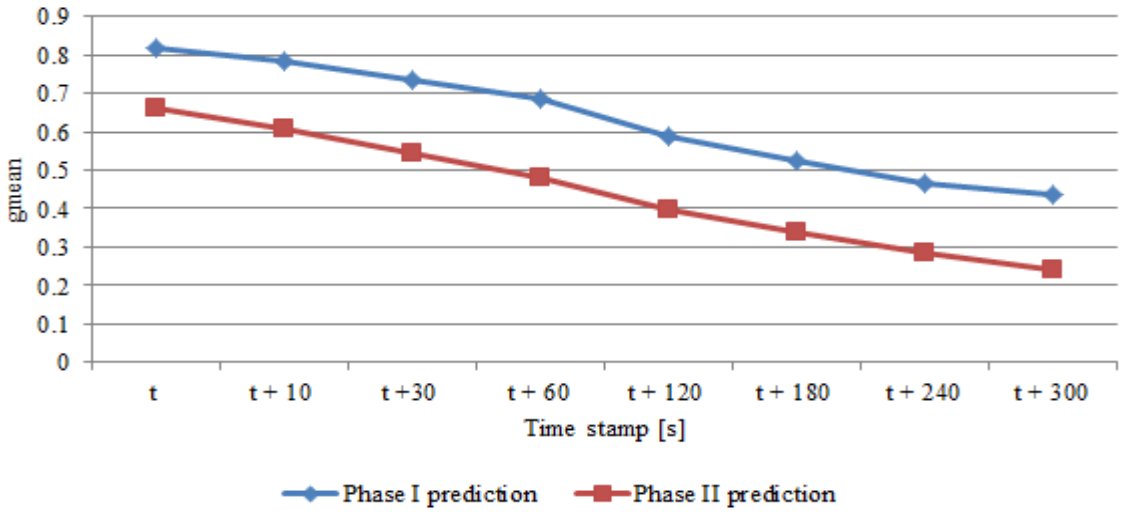


Figure 3.10. The values of *gmean* at various time stamps.

3.4.3. Identifying unseen faults

While the results on the testing dataset indicated the effectiveness of the random forest algorithm, in order to validate the robustness of the proposed model, data from other fault-prone turbines were analyzed with the additional objective of seeing how the model would respond to unseen states types. Due to the inherent variability in wind turbines, faults in a wind turbine vary from one to another. It is interesting to observe the models' responsiveness when some unseen faults are presented.

In this sub-section, data from 3 other fault-prone wind turbines, turbine 10, turbine 14 and turbine 17, are analyzed. Month-long data, from 8/28/2010 till 9/28/2010, was available for the analysis. The models built for phase-I prediction were deployed for this dataset. Faults such as yaw runaway, brush wear warning, blade angle implausibility, reply generator high stage were studied. Figures 3.11-3.13 display the actual distribution of output classes for turbines 10, 14 and 17, respectively. The number of faults varies from one turbine to another, however, the turbines were found to be operating normally with no errors most of the time. Tables 3.8 – 3.10 display the accuracy of output classes across turbines 10, 14 and 17, respectively. It is clear from the results that the algorithms are robust enough to identify unseen faults such as yaw runaway, blade angle not plausible axis 2, etc. The accuracy for correctly identifying unseen fault cases was found to be in the range of 60 - 100%, except for faults related to gearboxes (e.g., gearbox over-temperature, gearbox oil pressure too low) which were always identified as Turbine OK. The reasons for this include a lack of related input parameters (e.g., gearbox temperature, gearbox oil pressure, etc.) in the model. The results shown in Tables 3.8 – 3.10 confirm that the proposed model can be used to predict most wind turbine faults.

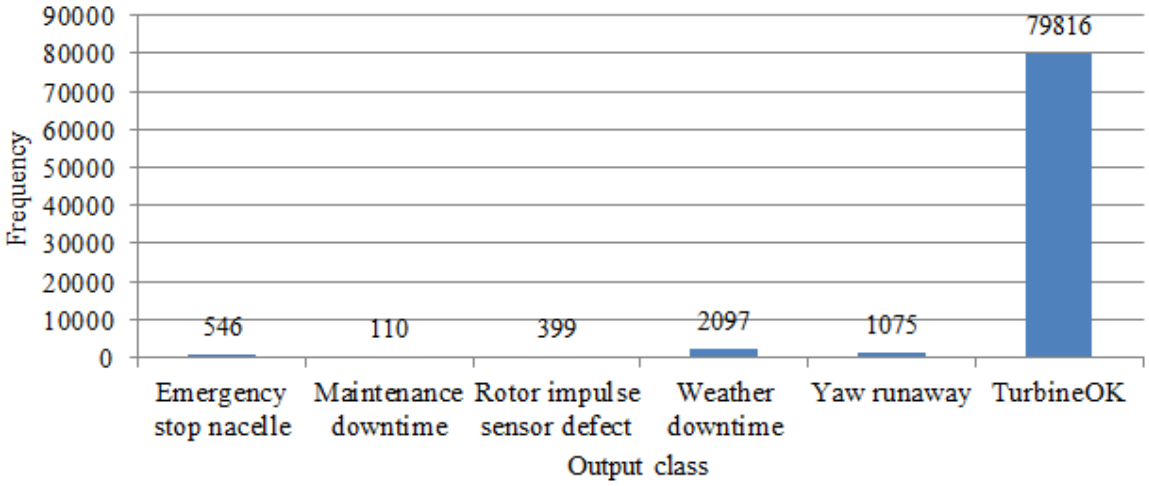


Figure 3.11. Distribution of output classes (turbine 10).

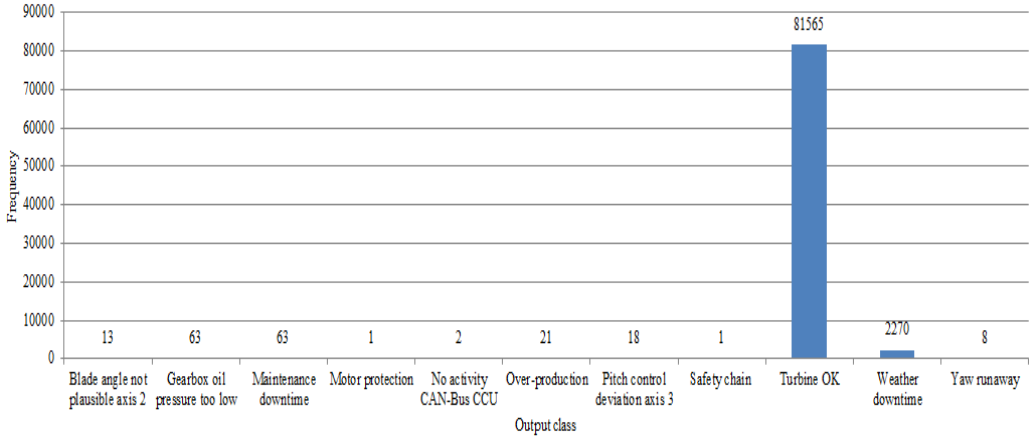


Figure 3.12. Distribution of output classes (turbine 14).

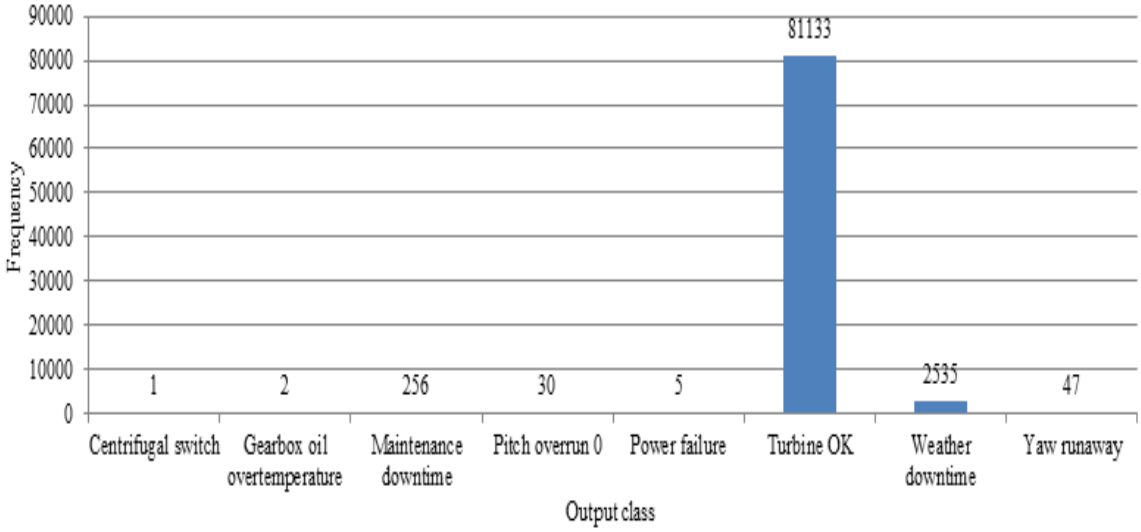


Figure 3.13. Distribution of output classes (turbine 17).

Table 3.8. Model analysis on turbine 10.

Actual output	Anticipated output	Correctly identified cases
Emergency stop nacelle	Fault	85.66%
Maintenance downtime	Maintenance downtime	100%
Rotor impulse sensor defect	Fault	60.90%
Weather downtime	Weather downtime	69.84%
Yaw runaway	Fault	99.62%
Turbine OK	Turbine OK	99.96%

Table 3.9. Model analysis on turbine 14.

Actual output	Anticipated output	Correctly identified cases
Blade angle not plausible axis 2	Fault	76.92%
Gearbox oil pressure too low	Fault	0.00%
Maintenance downtime	Maintenance downtime	100%
Motor protection	Fault	100%
No activity CAN-Bus CCU	Fault	50.0%
Overproduction	Fault	100%
Pitch control deviation axis 3	Fault	100%
Safety chain	Fault	100%
Turbine OK	Turbine OK	99.45%
Weather downtime	Weather downtime	60.70%
Yaw runaway	Fault	87.50%

Table 3.10. Model analysis on turbine 17.

Actual output	Anticipated output	Correctly identified cases
Centrifugal switch	Fault	100%
Gearbox oil over-temperature	Fault	0.00%
Maintenance downtime	Maintenance downtime	100%
Pitch overrun 0^0	Fault	100%
Power failure	Weather downtime	60.00%
Turbine OK	Turbine OK	99.27%
Weather downtime	Weather downtime	95.54%
Yaw runaway	Fault	100%

3.5 Summary

In this chapter, a methodology for predicting wind turbine states was presented. The proposed approach involved three key steps: turbine state abstraction, algorithm learning, and state prediction. In the first step, the initial wind turbine states were separated into classes using domain knowledge. To compensate the computational effort, data mining algorithms were trained using a stratified data set. The proposed model can be useful in isolating fault events from the normal turbine operations, also, at the same time the fault modes can be predicted in advance with good accuracy.

The model developed in this chapter is robust enough to identify various unseen faults.

CHAPTER 4

DEVELOPING PREDICTION MODELS OF FAULT PRONE COMPONENTS OF WIND TURBINES

4.1 Introduction

In chapters 2 and 3, frequent wind turbine statuses are analyzed and predicted. Quite often, certain components of wind turbines are most susceptible to damage than others. Thus, developing fault prediction models of the individual components/sub-components is more desirable.

In this chapter, component specific faults namely blade angle implausibility, and generator brush worn are investigated. In this chapter, first the mentioned faults are analyzed with an aim to identify the components/sub components affected by it. This is done by evaluating the snapshot files, which records the fault sequence. Analyzing snapshot files will be helpful in maintenance operations. In the literature, Chen *et al*., (2011) investigated blade pitch faults with an aim to reduce the occurrence the number of false alarms. Campbell and Adamson, (2003) developed an expert system identifying blade vibrations. They developed a rule base system based on two class classification model. Biegel *et al.*, (2011) developed a simulation model to optimize the blade pitch movement. In their model, they maximized power output while simultaneously minimizing fatigue loads. Muljadi and Butterfield, (2001) developed a simulated model for blade pitch control in turbulent wind conditions. In another research, Kusiak and Zhang, (2011) developed data driven approach to optimized the blade pitch movement

In nutshell, research pertaining to blade pitch system is new; and most of the research is done in a simulated environment by generating high frequency data. Advanced SCADA systems generate alarms in the event of such faults; however, early prediction of such faults is desirable to carry out planned and effective maintenance. The

output obtained from the present research will serve as input of maintenance planning and optimization tasks.

In general, such faults are frequent and can have downtime of 180 hours or more if persistent. Therefore, developing accurate prediction models is of interest.

Due to the nature of the dataset, classification models using several data-mining algorithms are developed in this chapter. The output dataset consists of two classes, namely fault and normal. Considering the imbalance in the output data, advance data balancing models are developed. The imbalance in the dataset is due to presence of relatively higher observations of turbine being functional, as compared with the fault cases.

4.2. Analyzing fault prone components of wind turbines

In this section, faults associated with two critical components of wind turbines namely turbine blades, and generators are discussed. The faults associated with the turbine blades are termed as blade angle implausibility, and blade angle asymmetry, whereas, generator brush worn and bearing over temperature faults are analyzed in wind turbine generators. Description of the mentioned faults is provided in coming sub-sections

4.2.1. Blade angle asymmetry

The blade angle asymmetry fault is attributed to the difference in blade pitch angles. Turbine components, namely motor, bearing, drive gear, electronic control and gearbox, are directly in contact with the blades, and therefore such faults adversely impact them. At a high wind speed, the blade pitch control system adjusts blade angle, whereas it usually remains constant at the low wind speed. When a blade angle lags behind the other blades, a turbine shutdown occurs to adjust the blade pitch angle (see Figure 4.1). Figure 4.1 provides blade angle asymmetry in one such turbine, where, pitch angle difference in blades is as high as 60°.

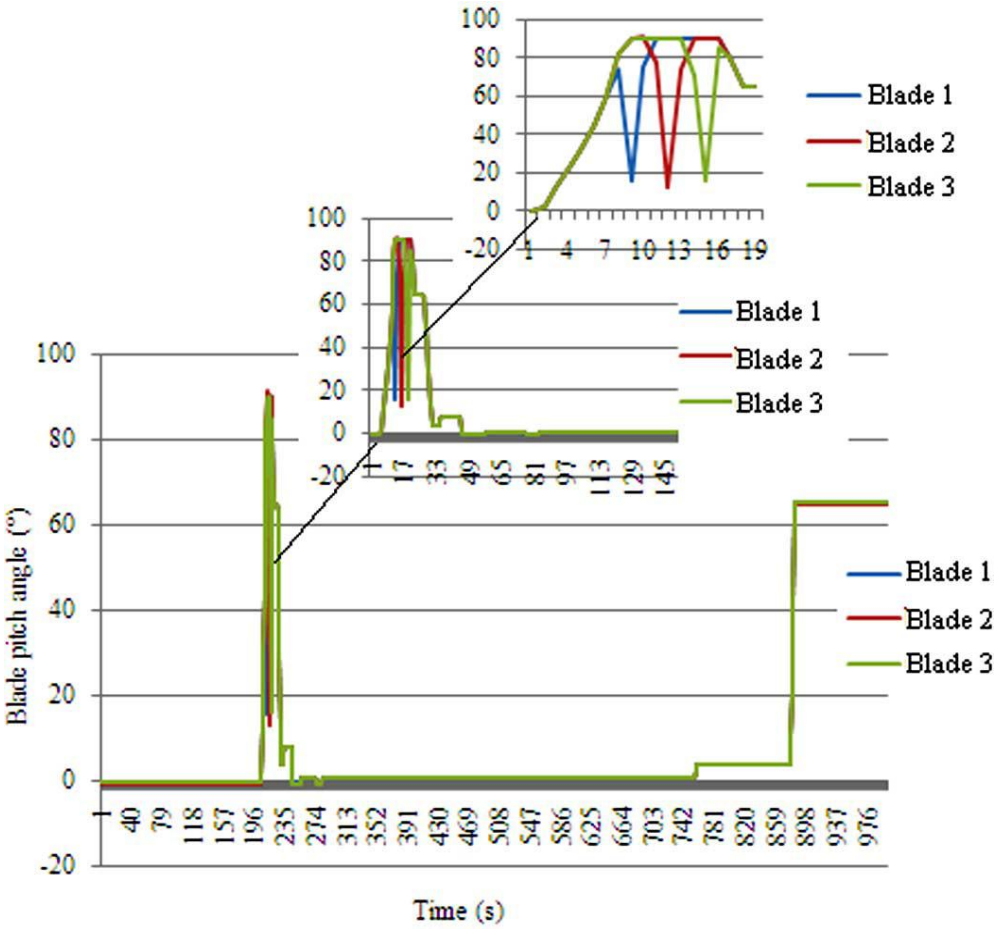


Figure 4.1. Blade angle asymmetry in a sample turbine.

System unavailability due to such a fault may reach 180 h, resulting in production losses.

4.2.2. Blade angle implausibility

The blade angle implausibility fault belongs to the group of category 2 faults, and is attributed to the difference in the actual and desired blade pitch angles. This fault status is recorded when the specified difference is exceeded for more than 1 s. Turbine components, namely the rotor, bearings, and the gearbox, are usually impacted. To overcome this fault, adjustment in the blade angles is performed by driving the blade

angles in the same direction until the deviation becomes zero. Fig. 4.2 (a-b) illustrates the blade angle implausibility fault along the axes 1 - 3.

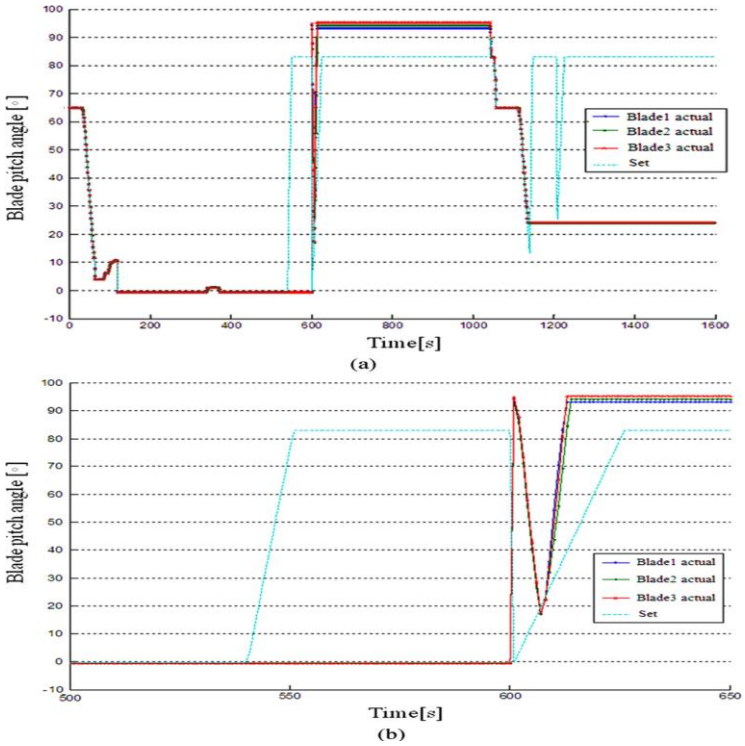


Figure 4.2. Blade angle implausibility in a sample turbine.

Even though the impact of the blade angle faults on the turbine system is obvious, it is difficult to identify the components/sub-assemblies directly affected by it. Therefore, special consideration is required to analyze the components associated with the fault.

In the coming section, wind turbine snapshots files are analyzed to identify the link between the actual fault and related components. Snapshot files are 10 s files recorded by the SCADA systems, whenever severe faults triggered in the turbine system.

4.2.3 Snapshot file analysis

Snapshot files are automatically generated operational data files whenever some critical fault occurs in the turbine. Snapshot files contain data at 1 second increments for the period of 7 seconds preceding and 3 seconds after the fault occurrence. Wind turbine status data was mapped with the operational data and the statuses recorded within the past 7 seconds and within the next 3 seconds of the mentioned faults are analyzed. Analyzing such files can provide more information about the fault sequence, and can be useful in identifying the actual faulty components. Figure 4.3-4.4 provides the frequency distribution of blade angle asymmetry and blade angle implausibility faults analyzed on 27 wind turbines (1.5 MW) over a period of 3 months.

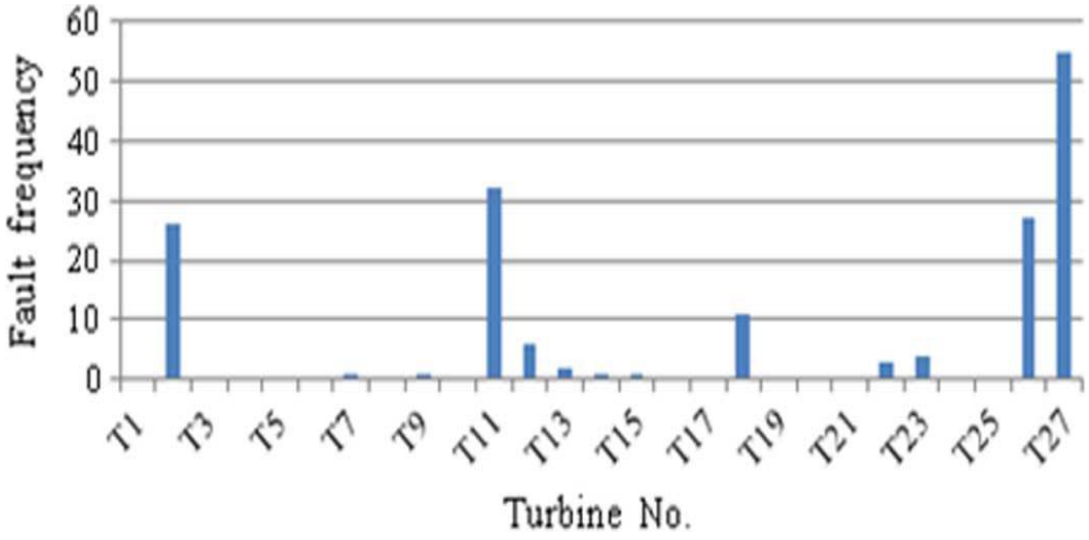


Figure 4.3. Blade angle asymmetry across 27 wind turbines.

Based on the distribution of faults across 27 wind turbines, turbine 5 (T5) and turbine 27 (T27) are considered for the analysis. Tables 4.1 and 4.2 presents the statuses triggered due to blade angle asymmetry and blade angle implausibility faults respectively.

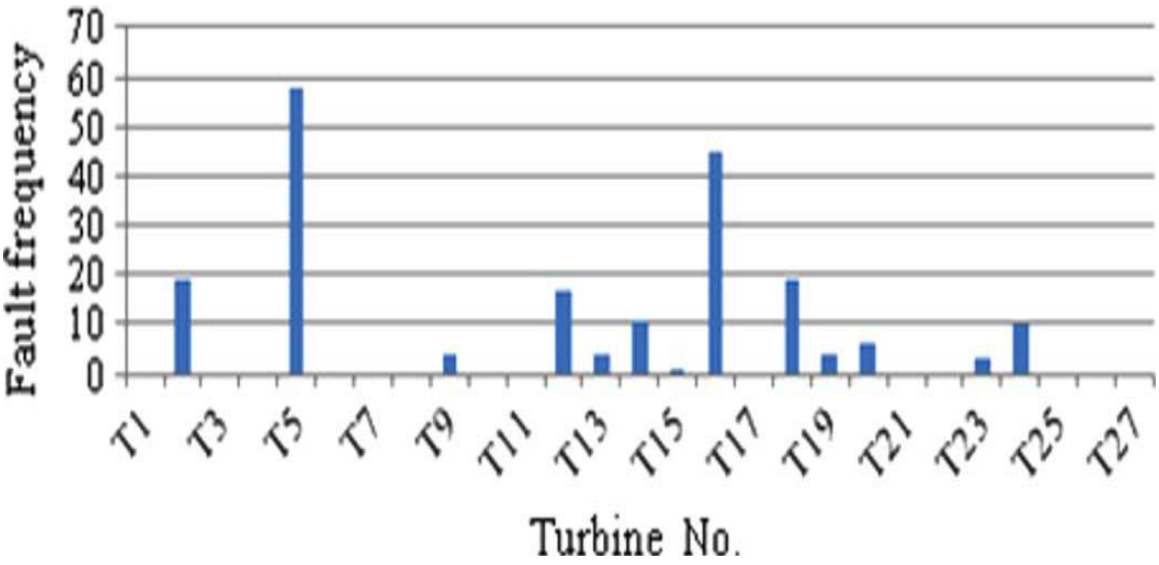


Figure 4.4. Blade angle implausibility across 27 wind turbines.

The fault sequence presented in Table 4.1 indicate the faulty pitch controller along axis 1, which is causing blade angle asymmetry. It also indicates that the pitch is overrun 90° to adjust the blade angles. The fault sequence obtained in Table 4.2 indicates the correlation among the pitch controller and blade angle implausibility faults. It also indicates that fault is reset to check the pitch motor batteries as faulty pitch motor, defective battery can be responsible for blade angle implausibility fault. Fault is reset by moving all three blades to feathered position 85. Wind turbine does not produce electricity till the turbine blades are in feathered position.

In next section, analysis on generator brush worn fault is performed.

4.2.4. Generator brush worn fault

The fault is attributed to the excessive wear of generator slip ring brushes, which is one of the most common faults occurring in wind turbines. This fault leads to unnecessary downtime because wind turbines are taken out of generation so that brushes can be replaced or cleaned.

Table 4.1. Statuses triggered by the blade angle asymmetry.

Status	**Category**	**Status**	**Category**
Programstart PLC	2	**Blade angle asymmetry***	**2**
Manual Stop	4	Pitch control deviation axis 1	2
Undervoltage	4	Repair	4
Emergency stop tower base / container	1	Manual operation pitch	4
Hydraulic pump time too high	2	Manual operation yaw	4
Safety chain	1	No activity CAN-Bus CCU	2
Restart time delay	4	Receiving external power command	4
Control call	4	Pitch thyristor 1 fault	1
Line fault voltage	4	Malfunction diverter	3
Line fault frequency	4	Idling command at WTG	4
Emergency stop nacelle / hub	1	Fault reset	4
Axis 1 fault pitch controller	2	Pitch control deviation axis 2	2
Collective fault pitch controller	2	Pitch control deviation axis 3	2
Pitch overrun 90°	4	Pitch malfunction 2 or 3 blades	1
Rotor CCU collective faults	2	Battery drive after grid fault	?
Line CCU collective faults	2	*Actual fault, ?unknown	

The turbine monitoring system, that is, the brush limit switch, reports excessive wear to maintenance personnel. In the absence of timely maintenance, however, the turbine needs to be shut down. Figure 4.5 illustrates the cause and effect diagram of "generator brush worn" faults. Design imperfections, turbine vibration, aerodynamic asymmetry, and generator over speed are considered the prime causes of "generator brush worn" faults.

Figure 4.5 clearly illustrates that the design is the main cause of such faults; however, other factors such as turbine vibration and aerodynamic imbalance also need to be considered.

Table 4.2. Statuses triggered by the blade angle implausibility.

Status	**Category**	**Status**	**Category**
No errors	4	Start-up	4
Manual stop	4	Load operation	4
Remote stop	4	Shut down	4
Remote start	4	Receiving external power command	4
System OK	4	**Blade angle not plausible axis 1***	**2**
Timeout pitch controller	2	Test pre-pressure active brake	4
Collective fault pitch controller	2	Malfunction cabinet heaters	3
Pitch overrun 90°	4	Malfunction diverter	3
Braking time rotor blade 1 too high	4	Idling command at WTG	4
Braking time rotor blade 2 too high	4	Fault reset	4
Braking time rotor blade 3 too high	4	Pitch control deviation axis 2	2
Battery charging rotor blade drive	4	Pitch control deviation axis 3	2
Limit switch 90°-rotor blade defective	2	**Blade angle not plausible axis 2***	**2**
Pitch control deviation axis 1	2	**Blade angle not plausible axis 3***	**2**
Tower vibration	2	Pitch malfunction 2 or 3 blades	1
Successful battery test is needed	1	Virtual battery test	4
Idling position	4	*Actual fault	

Research in component design has resulted in brushless generators where the drawbacks associated with the generator brushes have been eliminated. For existing wind farms, however, the cost of brushless solutions cannot be justified. While the cost to replace or maintain a worn out carbon brush cannot be eliminated, the cost corresponding to the lost operation can be minimized by predicting “generator brush worn” faults in advance.

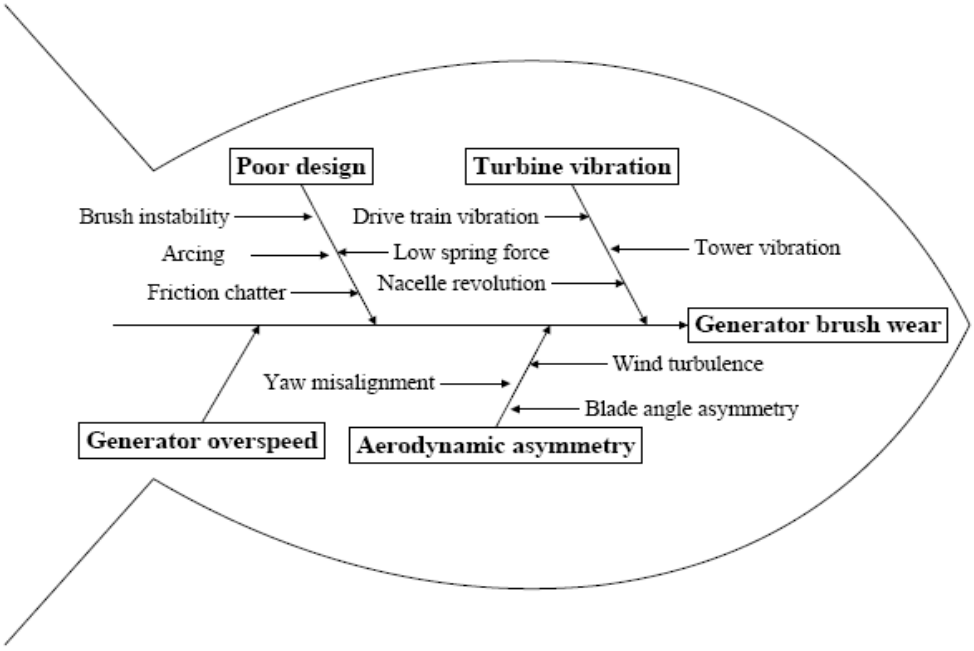


Figure 4.5. Cause and effect diagram of generator brush worn fault.

A typical carbon brush lasts for six months under normal operation; however, the presence of abnormalities in the system operation can reduce the brush life to a factor of 30% and thus demand early replacement.

4.2.4.1. Fault analysis based on the SCADA status data

The switch at the end of the brush reports the information to the SCADA system, where the information is stored in the form of status codes. Both status and operational parameters are used in this chapter. Status data is event triggered whereas, operational data is time triggered. Based on the data considered for this research, the delay between acknowledging the status code and maintenance action can be as long 168 hours; thereafter, the wind turbine will shut down. In the present research, data from 27 wind

turbines were analyzed. The distribution of faults across 27 wind turbines is shown in Figure 4.6. The faultiest turbine, Turbine 14, is considered for further analysis.

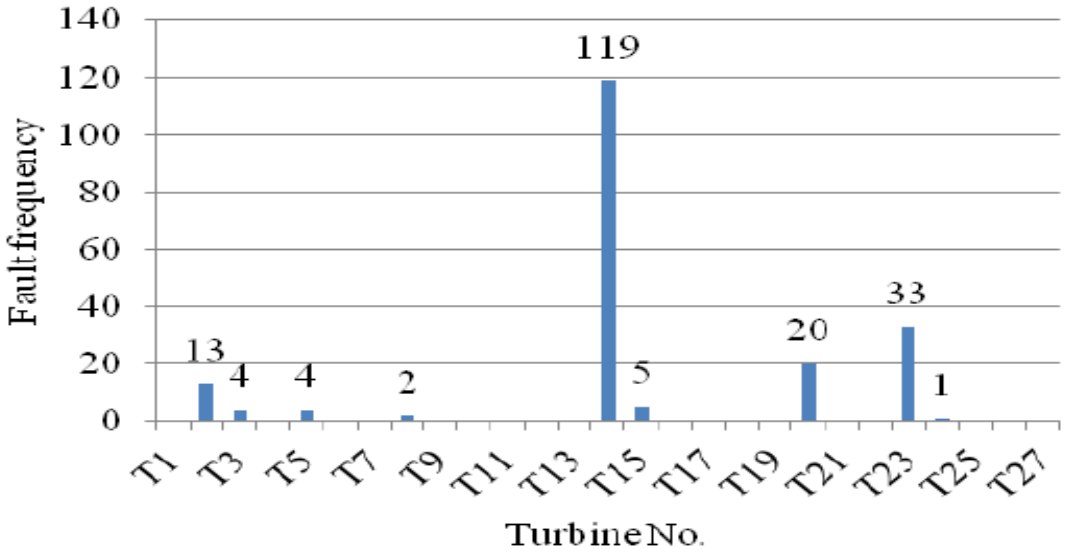
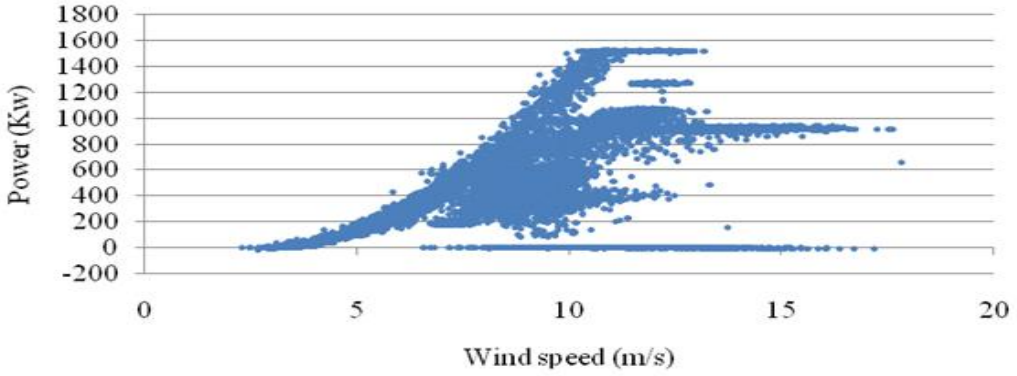


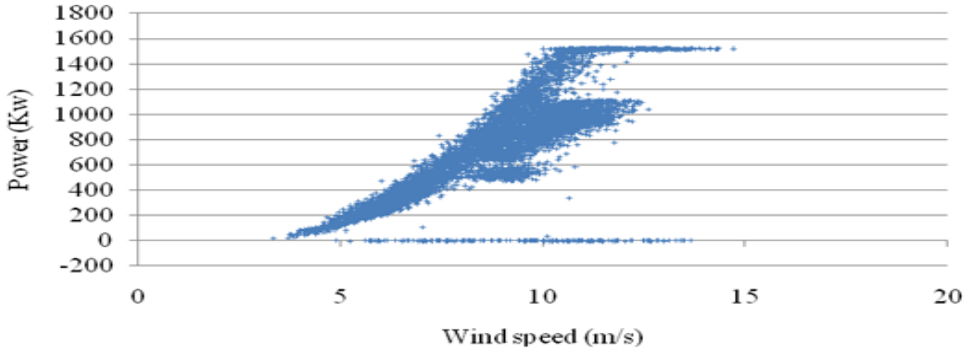
Figure 4.6. Generator brush worn fault across 27 wind turbines.

SCADA status data from Turbine 14 indicates that the "generator brush worn" fault occurred more than 100 times in the month during which data was collected. To identify the statuses associated with "generator brush fault," 1 second snapshot files were analyzed. On average, more than 30 different statuses were associated with this fault.

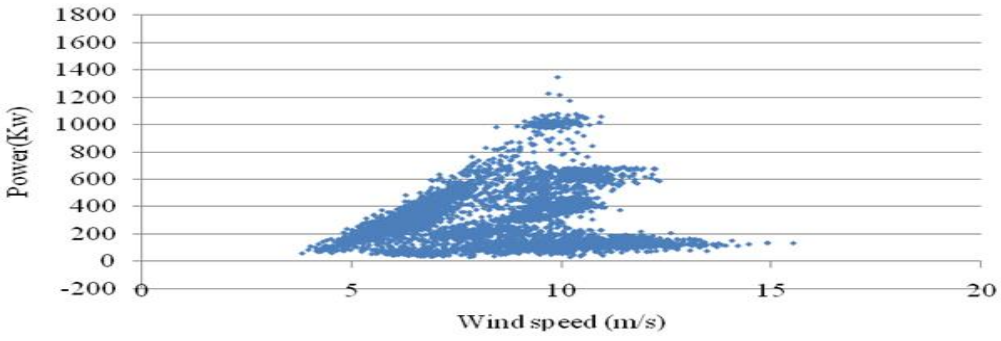
Figure 4.7 shows the power curve of a turbine (here Turbine 14) affected by the brush fault over three consecutive days, 4/14/2010-4/16/2010. The power curve had already started to deteriorate, but the turbine is still operational and producing power (Figure 4.7 (a)). The fault manifested itself for a time period resulting further deterioration in the power curve (Figure 4.7 (b-c)). It is also possible that other faults could be responsible for power curve deterioration; however, the impact of the "generator brush worn" fault prevailed in the two-day time period under consideration.



(a)



(b)



(c)

Figure 4.7. Powers curve a turbine during generator brush worn fault: (a) during fault emergence, (b) one day after the fault, (c) two days after the fault.

The analyses based on snapshot files are useful in identifying the turbine components or subcomponents affected during the fault. Early prediction of such faults is required to minimize the production loss. In next section, data-mining based prediction models are developed.

4.3. Data-mining based fault prediction models

4.3.1. Models predicting blade angle implausibility faults

The data collected at a wind farm is noisy due to sensor errors. Inconsistent data, e.g., abnormal wind speed, is deleted. After filtering the raw data, the final data set for turbine 16 over a period of three months (i.e., from 10/9/2009 to 1/16/2010) is produced.

The wind turbine status and data is merged for mining. The prediction task is performed at 13 different time stamps, with the maximum horizon of 10 min. In total, 13 different data sets are generated for modeling the blade angle implausibility. Table 4.3 describes the data set collected at turbine 16 and used to build prediction models for identifying the blade angle implausibility fault.

The parameters recorded by the SCADA system are broadly categorized as controllable parameters, non-controllable parameters, and performance parameters. Blade pitch angle and generator torque are examples of controllable parameters. Uncontrollable parameters are the stochastic parameters that cannot be controlled, e.g., wind speed, wind deviation. Performance parameters, such as power, generator speed, and gearbox speed, indicate the wind turbine performance. Parameters that are derived from the original parameters, e.g., blade angle deviation, can be informative and therefore are included in the parameter list. Since the main objective of this research is to develop a predictive model for wind turbine blades, only parameters related to wind turbine blades are selected for analysis. Based on the domain knowledge, the hundred dimensional data set has been reduced to 53 parameters. Further reduction in dimensionality is achieved with three different parameter selection approaches namely subset evaluator algorithm, relief, and information gain.

Table 4.3. Turbine 16 data set description.

No.	Data Type	Description
1	Data frequency	1 s
2	Fault	Blade angle implausible axis 1, blade angle implausible axis 2, blade angle implausible axis 3
3	Maximum fault instances	38
4	Maximum normal instances	10596
5	Prediction time stamps	5 s, 15 s, 30 s, 1min, 2 min, 3 min, 4 min,…,10 min
6	Maximum prediction length	10 min
7	Maximum input parameters	53

Subset evaluator algorithm evaluates the significance of a parameter subset by analyzing the prediction accuracy of individual parameters. Parameters that are highly correlated with the output and less correlated with each other are selected. Here, the five parameters are considered (see Table 4.4). Relief is an algorithm inspired by the instance-based learning. Given a training data *S*, a sample size *N*, and a threshold *Γ*, a relief algorithm identifies parameters that are statistically relevant to the target. The size of the nearest neighbor (*k*) is varied to identify the best subset of the parameters (Figure. 4.8). The maximum number of parameters to be selected is set to five. Nacelle revolution, rotor speed, generator speed, generator/gearbox speed, blade 1 deviation, blade 2 deviation, and blade 3 deviations have been determined to be the most relevant parameters. After removing redundant parameters, such as the generator speed and generator/gearbox speed, the five parameters are listed in Table 4.5.

Table 4.4. Parameters selected by the subset-evaluator-algorithm.

No.	Parameter Name
1	Tower deflection
2	Nacelle revolution
3	Blade 1 dev
4	Blade 2 dev
5	Blade 3 dev

Table 4.5. Average relevancy of best parameters.

No.	Parameter	Average Relevancy
1	Rotor speed*	0.15621
2	Generator/gearbox speed	0.15575
3	Generator speed	0.15511
4	Blade 2 dev*	0.13457
5	Blade 1 dev*	0.13296
6	Blade 3 dev*	0.13284
7	Nacelle revolution*	0.12793
8	Torque actual value	0.08895
9	Voltage phase B	0.08797
10	Voltage phase C	0.08477
11	Voltage phase A	0.07962
* Selected parameter		

The significance of a parameter is determined by the information gain with respect to the output class. The selected parameters and their information gain are provided in Table 4.6.

Table 4.6. Information gain of the selected parameters.

No.	Parameter name	Information gain
1	Blade 1 dev	0.06
2	Blade 3 dev	0.049
3	Blade 2 dev	0.046
4	Tower deflection	0.037
5	Nacelle revolution	0.036

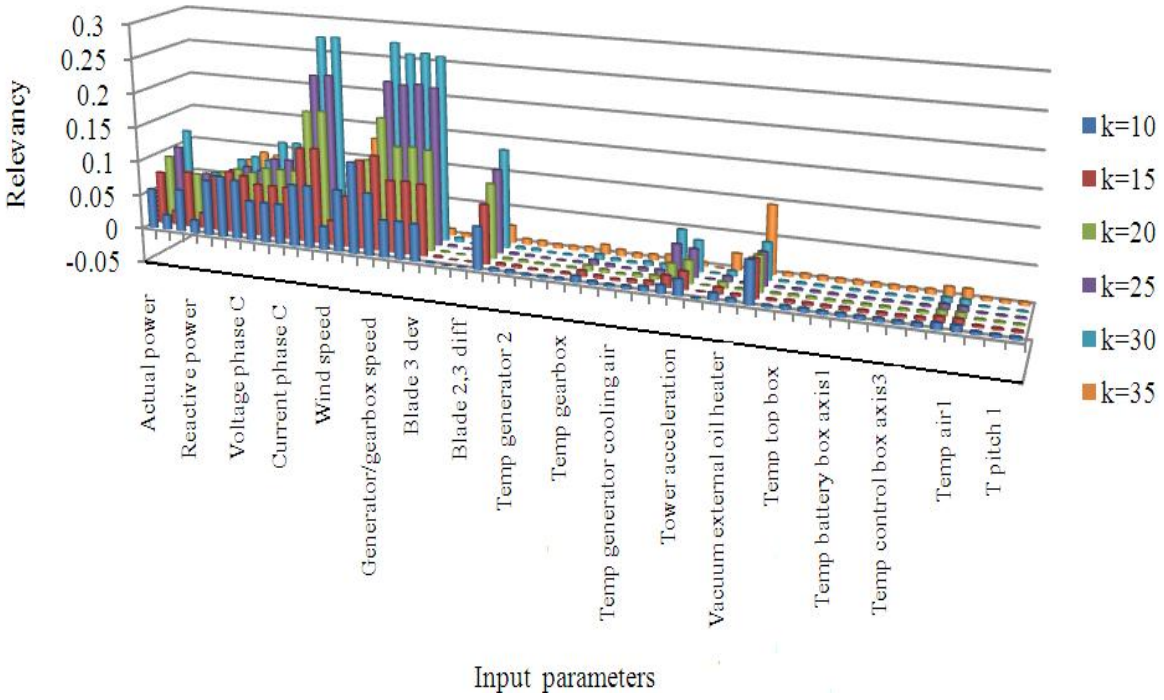


Figure 4.8. Parameter importance at various nearest neighbors using relief parameter evaluator.

Based on the different evaluation criteria mentioned in this section, five parameters, namely blade 1 dev (BD1), blade 2 dev (BD2), blade 3 dev (BD3), nacelle revolution (NR), rotor speed (RS) and tower deflection (TD), are selected to build the prediction model. The data set at time stamp t + 180 (see Table 4.7) is used to identify a suitable algorithm for constructing a prediction model for 13 time stamps. The description of data-mining algorithms considered in this chapter is provided next.

Table 4.7. Description of the dataset for model selection.

Dataset	Start Time Stamp	End Time Stamp
t +180 (Overall)	10/23/2009 0:00 AM	1/16/2010 23:57 PM
t +180 (Training)	10/23/2009 0:00AM	12/18/2009 23:57 PM
t +180 (Test)	12/18/2009 0:00 AM	1/16/2010 23:57 PM

Five data-mining algorithms, bagging, neural network (NN) (Brieman, 1996), PART (Bauer and Kohavi, 1999), *k*-nearest neighbor (*k*-NN, $k = 10$) (McCormick and Nandi, 1997) and genetic programming (GP) (Jack *et al*., 2003; Koja, 1992) are selected for building prediction models. Two meta-classifiers, the cost-sensitive classifier and the adaptive boosting (AdaboostM1) classifier, are used along with the base classifiers (Aha and Kibler, 1991). The cost sensitive classifier makes the base classifier cost sensitive, whereas Adaboost classifier improves the prediction accuracy. The algorithms are briefly discussed next.

The multi-layer perceptron was developed using training and validation data sets, and tested using a test set. The three data sets are mutually exclusive. Training terminates whether the prediction performance of the validation set diverges from that of the training set (i.e., the network becomes over trained on the training set).

PART is a tree-based classifier which generates a partial decision tree at each iteration, and the best node is used for rule construction. Unlike other classifiers, such as decision tree or RIPPER, PART does not perform global optimization. It adopts a divide-and-conquer strategy to build rules. Bagging is an ensemble meta-algorithm that combines various predictors to predict results. The *k*-NN classifies objects based on the training examples with objects being classified by the majority of vote from their neighbors.

Genetic programming (GP) is an algorithm inspired by biology. GPs represent individuals as executable programs (trees). The tree structure of GP consists of functional nodes and terminal nodes. Terminal nodes usually represent the system parameters' value, whereas functional nodes act as operators (logical or numerical) connecting individual nodes. GPs have been successfully used to solve complex problems and produced promising results (Koja, 1991; Jack *et al*., 2003).

The occurrence of blade angle implausibility faults are less, compared with when the turbines are functioning normally. This causes imbalance in the output class. For the current scenario, a class imbalance ratio is 0.0036 which is significantly higher. In order to get unbiased results, the output class needs to be balanced. Approaches namely under sampling of majority class, and oversampling of minority class are widely used in the literature to balance the output class. However, the approaches mentioned above suffer from poor generalization. In this chapter, cost sensitive learning scheme is employed for the data-mining algorithms mentioned earlier. Under normal learning, the cost of misclassifying normal class as fault and vice-versa are kept same. However, under cost sensitive learning, cost coefficients are usually kept higher for the minority class (usually the class of interest).

The difficulty associated with the cost sensitive learning is in the selection of the cost matrix. When the maximum specificity is the metric of interest, the false negatives (cases where fault instances are classified as normal instances) are penalized. However, at the same time, it is important not to compromise the solution for specificity, therefore maximization of the true negative instances (i.e., normal instances) is done by penalizing the false positives (cases where normal instances are classified as fault). Related research in cost sensitivity learning and classification involves: (1) Nature inspired algorithms (Li *et al*., 2005), (2) instance weighting method (Ting, 2002), (3) boosting tree method (Ting and Zheng, 1998), and (4) hybrid methods (Turney, 1995). However, the results reported by the above-mentioned approaches are promising; the computational time is the concern.

Moreover, the cost-matrix relies on data and hand. Finding the best possible cost coefficients is difficult and it requires expert knowledge. However, due to the lack of prior information on the misclassification cost, following heuristic algorithm is developed.

Step 1: **Identify** an appropriate predictor for false positives and false negatives.

Step 2: **Generate** N cost coefficients (C_{01} or C_{10}) for the selected predictor uniformly in the range (0.5-1.0) *(/* 1-generated cost coefficient value will be the cost coefficient for other predictor */)*.

Step 3: While i <= N,

Evaluate Misclassification cost (MC), accuracy, specificity and sensitivity for i^{th} cost coefficient using selected classifier.

$i = i+1$

End

Step 4: **Select** the best cost coefficient value and its two immediate neighbors.

Step 5: **Evaluate** MC, accuracy, specificity and sensitivity for all six possible combinations. Identify best cost coefficients.

Step 6: **Compare** best objectives of Step 4 and Step 5.

Step 7: If ($MC_{step4} < MC_{step5}$), stop

Else, best cost-coefficient = cost-coefficient$_{Step5}$ repeat, go to Step 4

Applying cost sensitive learning on the given dataset provides the cost coefficient for false negative cases (C_{01}) is 0.9, whereas 0.1 is found to be the best cost coefficient for false positive cases (C_{10}). The initial set of results produced by ANN based classifier is shown in Fig. 4.9. Table 4.8 provides the details of the best results. The results indicate that the cost of misclassifying a blade angle implausibility fault is nine times higher than

the cost of misclassifying a normal instance. The same cost coefficients and their two immediate neighbor cost coefficients are used to determine the best cost coefficients for other selected classifiers.

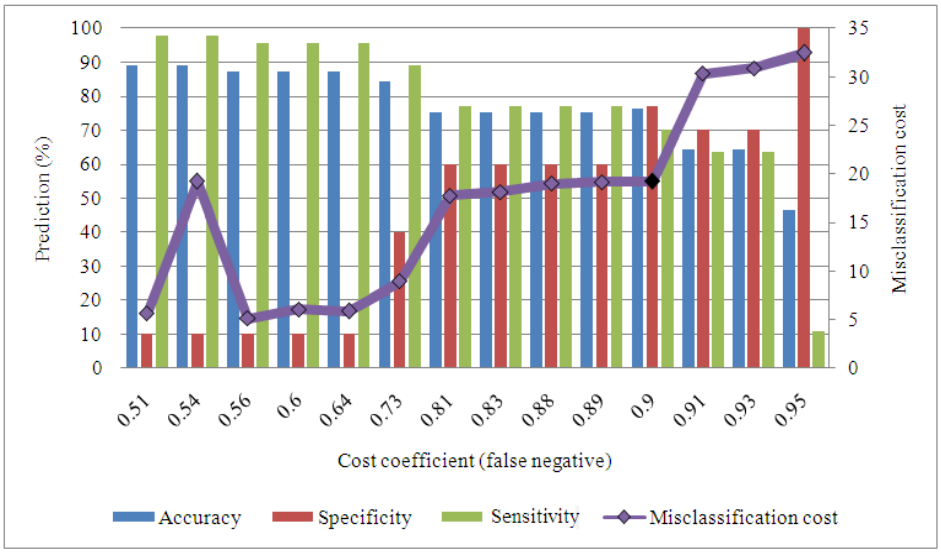


Fig. 4.9. Comparison of accuracy measures with misclassification cost.

Table 4.8. Results for cost sensitive classification.

No.	Parameter/Objective	Value
1	C_{01}, C_{10}	0.1,0.9
2	NFP*, NFN*	03,21
3	Accuracy	76.23%
4	Specificity	70.00%
5	Sensitivity	76.90%
6	Misclassification cost	19.2 units
*NFP = # of false positives; NFN = # of false negatives		

The performance of the data-mining algorithms was examined at t + 180 time stamps, and the best performing algorithm was used to build the prediction models at all 13 time stamps. Based on the results of the cost sensitive classification (Table 4.8), the cost coefficients for all the cases yielded best results in the C_{01} range of 0.1-0.0., and C_{02} range

of 0.9-0.99. Specificity signifies the accuracy in predicting normal class output, whereas, sensitivity signifies the accuracy in predicting fault cases. Table 4.9 reports the results obtained using five data-mining algorithms. The results shown in Table 4.9 indicate that the performance of most algorithms is similar, except PART, which yielded poor specificity. However, a genetic-programming (GP)-based classifier offers advantages in terms of solution interpretation and therefore is selected to construct the prediction model at all 13 stamps.

Table 4.9. Accuracy of data-mining algorithms for prediction at time stamp t + 180.

Algorithm	Accuracy (%)	Sensitivity (%)	Specificity (%)
Bagging	72.5	80.0	71.7
PART	75.5	60.0	77.2
ANN	76.2	76.9	70.0
k-NN (*k*=10)	73.5	80.0	72.8
Genetic programming	**74.7**	**80.5**	**75.3**

4.3.2. Experimental results predicting blade angle implausibility

In this section, results produced by the genetic programming (GP) algorithm are discussed. Various experiments are considered to ensure applicability and robustness of the approach. The ratio of coefficients of false negative and false positive cases was varied in the range 9-9.9 to facilitate cost sensitive classification. The control parameters of the genetic programming algorithm are varied to determine the best results. Table 4.10 lists the parameter values used by GP, applied to build model for prediction at all 13 time stamps. One third of the data is used for training, and the remaining two-thirds are used for validation. The classification is done by building two-class classifier from multiple one-class classifiers. Results are shown in Table 4.11.

Table 4.10. Description of GP parameters.

No.	GP Parameter	Descritpion/Value
1	Statistical normal pre-processor	Input data is normalized with 0 mean and 1.0 standard deviation
2	Population size	100
3	Validation test size	66%
4	Max. tree depth	6
5	No. of inputs for program	6
6	No. of available classes	2 (fault/normal)
7	No. of content for nodes (functional+terminal)	20 (14+6)
8	Function table	[+, -, /, *, If, >, <, Pow, &, |, Max, Min, Exp, Log]
9	One-class weight evaluator	Confidence of the class (0 or 1)
10	Tree population	Ramped half and half
11	Program selection	Fitness proportional
12	Size of elite program	1
13	Crossover operator	Proportional 0.9 (2 parents=>2 children)
14	Mutation operator	Proportional 0.07 (1 parent=>1 child)
15	New program tree operator	Proportional 0.03 (1parent=>1 children)

A graphical representation of the results obtained is shown in Figure 4.10. It is obvious from Figure 4.10 that as the time stamp increases, the prediction accuracy decreases. The GP classifier produced accuracy in the range of 69%-87% for 13 time stamps. A description of tree structure in LISP (Koja, 1992) format is shown in Tables 4.12-4.13. The results presented in Tables 4.12-4.13 show that most GP tree structures involve less parameters than the originally selected five parameters.

Table 4.11. Results obtained by the GP algorithm.

Time Stamp [s]	Evaluation Criteria		
	Accuracy (%)	Sensitivity (%)	Specificity (%)
t + 5	87.4	87.5	80
t + 15	85.1	86.3	78.2
t + 30	83.2	85.2	77.2
t + 60	81.3	83.2	76.6
t + 120	75.6	77.4	75.1
t + 180	74.7	75.3	80.5
t + 240	76.3	76.8	71.4
t + 300	75.6	76	70.3
t + 360	74.5	75.2	69.8
t + 420	73.6	74.7	68.6
t + 480	71.3	73.4	68
t + 540	69.8	72.5	67.2
t + 600	68.7	71.2	66.9

The GP produced trees provide an easy-to-understand relationship between the input parameters that classify an output as a fault/non-fault. The classification rules are easily understood and can be verified for other data sets. The multiple feature selection algorithms reduce the data dimensionality from 53 dimensions to 6, thereby facilitating faster convergence of the GP programs.

4.3.3. Models predicting generator brush worn faults

The data available for the analyzing generator brush worn fault was collected by SCADA systems at 27 wind turbines recorded at 10 minute intervals. In addition, the SCADA status data was used to label the fault conditions. Four months of the data collected from 3/1/2010 to 7/31/2010 was analyzed. Fault distribution is provided in Figure 4.6.

Table 4.12. GP based tree structure for both fault and normal class (t+5 to t+180 s).

Time Stamp [s]	Normal Class	Fault Class
t + 5	&(If(<(Nacelle Revolution, Pow(Co 0.603, Nacelle Revolution)), +(Co 0.3, Rotor speed)), If(Co 0.574, /(Co 0.668, Min(Rotor speed, Co 0.358))))	&(&(<(+(Blade2_dev, Co 0.138), +(Nacelle Revolution, Rotor speed)), >(Co 1, +(Nacelle Revolution, Blade2_dev))), If(<(Min(Rotor speed, Co 0.138), blade2_dev), Blade2_dev))
t + 15	&(If(<(Nacelle Revolution, Pow(Co 0.668, Tower deflection)), +(Co 0.3, Rotor speed)), If(Co 0.574,/(Co 0.844, Log(Nacelle revolution)))	&(<(Pow(Pow(Blade2_dev, blade2_dev), Pow(Blade2_dev, Rotor Speed)), Max(Blade1_dev, Co 0.434)), If(&(<(Co 0.939, blade2_dev), Co 0.909), -(Co 0.039, *(Blade3_dev, Rotor speed))))
t + 30	< (Nacelle revolution, /(Pow(Co 0.362, /(Blade3_dev, Co 0.575)), +(Co 0.549, +(Nacelle revolution, Co 0.549))))	<(+(Max(+(Blade1_dev, Co 0.905), *(Blade1_dev, Tower Deflection)), +(Max(Nacelle revolution, Tower deflection), /(Co 0.226, Tower deflection))), &(>(Exp(Nacelle revolution), /(Rotor speed, Tower deflection)), Tower deflection))
t + 60	Exp(/(Log(*(Co 0.893, Nacelle revolution)), blade3_dev))	/(-(*(Rotor speed, Nacelle revolution), Rotor speed), Min(Nacelle revolution, Exp(Pow(Nacelle revolution, Rotor Speed))))
t + 120	>(Log(Blade3_dev), -(-(Nacelle revolution, Log(Rotor speed)), Co 0.629))	If(>(Nacelle revolution, Rotor speed), Max(Max(Blade2_dev, blade2_dev), +(Nacelle revolution, Max(Blade3_dev, blade2_dev))))
t + 180	<(Blade1_dev, +(Min(Pow(Tower deflection, Nacelle revolution), Nacelle revolution), Rotor speed))	-(+(Co 0.244, Pow(Co 0.911, -(Tower deflection, Nacelle Revolution))), -(/(Min(Tower deflection, Tower deflection), Log(Nacelle revolution)), Max(Co 0.494, blade2_dev)))

To label the output data, the status data containing the fault information was merged with the operational data. The output (O_{i-1}) was labeled as *fault* or *normal* at time i-1 based upon the logic given in Equation 4.1.

Table 4.13. GP based tree structure for both fault and normal class (t+240 to t+ 300 s).

Time Stamp [s]	Normal Class	Fault Class
t + 240	-(Max(Pow(Co 0.256, Blade3_dev), Blade1_dev), Min(Blade1_dev, Exp(Pow(Co 0.518, Blade1_dev))))	/(-(-(*(Tower deflection, Blade2_dev), *(Co 0.998, Rotor Speed)), Nacelle revolution), /(-(*(Co 0.998, Rotor speed), Exp(Blade2_dev)), Exp(Exp(Blade2_dev))))
t + 300	|(<(Blade2_dev, Blade3_dev), >(Tower deflection, Nacelle revolution))	>(+(/(Min(Co 0.533, blade3_dev), Nacelle revolution), -(Rotor speed, -(Co 0.043, Rotor speed)), >(+(Nacelle revolution, -(Nacelle revolution, Co 0.533)), Max(Pow(Co 0.461, Rotor speed), Nacelle revolution))
t + 360	/(Log(+(/(Blade3_dev, Rotor speed), +(Rotor speed, Co 0.86))), Log(+(+(Rotor speed, Blade2_dev), Max(Nacelle revolution, Co 0.435))))	>(>(Log(/(Co 0.299, Rotor speed)), Log(+(Rotor speed, Blade2_dev))), +(Log(+(Co 0.299, Blade3_dev)), -(Co 0.805, Nacelle revolution)))
t + 420	>(Exp(-(Blade2_dev, Blade1_dev)), Max(/(Nacelle revolution, Rotor speed), Max(/(Co 0.049, Rotor speed), Pow(Rotor speed, Co 0.447))))	|(&(If(>(Rotor Speed, blade1_dev), Pow(Blade3_dev, Co 0.641)), If(Co - 1.004, <(Rotor speed, Nacelle revolution))), <(Blade3_dev, Min(Log(Nacelle revolution), Log(Rotor speed))))
t + 480	>(Min(Exp(-(Blade1_dev, Nacelle revolution)), Exp(-(Nacelle revolution, Blade1_dev))), Min(/(-(Rotor speed, Blade1_dev), Co 1.07), Exp(-(blade1_dev, Rotor speed)))	/(Max(Exp(*(Blade2_dev, Co 0.464)), Rotor speed), *(-(Rotor speed, Co 0.879), /(Nacelle revolution, Co 0.505)))
t+ 540	-(Min(Min(/(Blade3_dev, Blade1_dev), Pow(Tower deflection, Rotor speed)), Exp(+ (Rotor speed, Co 0.143))), *(*(Pow(Tower deflection, Rotor Speed), Pow(Tower deflection, Rotor speed)), -(/(Tower deflection, Nacelle revolution), /(Blade3_dev, blade1_dev)))	+(Co 0.617, *(/(+(Nacelle revolution, Co 0.617), +(Tower Deflection, Blade2_dev)), - (/(Blade3_dev, Nacelle revolution), Co 1)))
t+ 600	Pow(Pow(Rotor speed, +(Blade3_dev, Rotor speed)), Exp(+(Rotor speed, +(Rotor speed, Rotor speed))))	>(Max(/(Rotor speed, blade3_dev), If(/(Tower deflection, Rotor speed), /(Co 0.329, Nacelle revolution))), Nacelle revolution)

$$O_{i-1} = \left\{ \begin{array}{l} \text{fault, if } t + i - 1_o \le t_f \le t + i_o,\ i \in \{10, 20, 30, ...\} \\ \text{normal, otherwise} \end{array} \right\} \quad (4.1)$$

In Equation (4.1), t_f is the time at which the fault is recorded in the status data, whereas $t+i-1_o$ and $t+i_0$ are the time interval in operational data. The output is labeled as fault at time i-1, if the fault was recorded in between current time $t + i-1_o$ and the next time stamp $t + i_o$, otherwise the output is normal.

To avoid the curse of dimensionality and improve the prediction accuracy, irrelevant parameters were removed from the analysis. A combination of data-mining techniques and well-established statistic measures were used to select parameters. The SCADA system records more than 100 parameters to monitor the wind turbine performance, including fixed values. Using domain knowledge, the initial 100 dimensional data was reduced to 50 dimensions. Selecting a prediction algorithm for actual testing was based upon its performance on the test data. The metrics of Equations (4.2) through (4.5) were used to measure performance of the algorithms.

$$Accuracy = \frac{TFP + TNP}{TFP + TFM + TNM + TNP} \quad (4.2)$$

$$Senstivity = \frac{TFP}{TFP + TFM} \quad (4.3)$$

$$Specificity = \frac{TNP}{TNM + TNP} \quad (4.4)$$

The gmean metric is also used due to class imbalance of the dataset (Wang *et al*., 2009).

$$gmean = \sqrt{Sensitivity \times Specificit\,y} \quad (4.5)$$

In Equations (4.2)-(4.4), TFP is the total number of faults predicted, and TNP is the total normal cases predicted. TFM represents the total fault cased misclassified as normal, whereas TNP is the total normal cases misclassified as fault. The objective here is to maximize the gmean, while keeping the desired level of accuracy.

Two known parameter selection approaches—namely boosting tree, and wrapper algorithm with genetic search—were employed to identify the best subset of parameters for the prediction. Boosting tree uses a gradient boosting machine approach to rank the parameters. Wrapper methodology for selecting features uses the learning algorithm as a black box to rank/score subsets of features according to their predictive power. In the present research, a genetic algorithm based ranking approach was used. Table 4.14 describes the 10 parameters selected based upon the boosting tree, and wrapper methodology. The parameter selection approach has reduced 50 parameters to 10.

Table 4.14. Parameter selected based on different data mining algorithms.

Parameter	Boosting Tree Algorithm	Wrapper (Genetic Search)
	Rank/Score	Rank/Score
Nacelle revolution	1	1
Hydraulic pressure	0.98	8
Drive train acceleration	0.95	3
Generator/gearbox speed	0.9	10
Generator speed	0.89	5
Temperature, shaft bearing	0.89	4
Temperature, gearbox bearing	0.86	7
Rotor speed	0.83	2
Temperature, bearing A	0.82	9
Temperature bearing B	0.8	6

The class imbalance problem is prevalent in wind turbines where the fault cases are rare compared with normal cases (that is, when the turbine is fully operational). Considering the huge amount of data, for a typical fault (e.g., rotor imbalance, blade

angle asymmetry) the ratio of normal to fault instances can be as large as 1000:1. Class imbalance is considered one of the most critical problems in machine learning and data-mining based applications, and has gained attention from the researchers worldwide (Cleofas *et al*., 2009; Hripscak and Wilcox, 2002; Weiss, 2004).

Possible recommended solutions to balance the data include: (1) over-sampling of the minority class; (2) under-sampling of the majority class; and (3) cost sensitive classification. A detailed description of numerous data sampling approaches can be found in Bastia *et al*., (2004). Irrespective of their widespread applications, these solutions have certain limitations. For example, over sampling might create minority class data points very close to majority data points thereby making the classification task more challenging; under-sampling might cause over fitting; and the cost sensitive class classification requires cost information for misclassified cases. Another method of data sampling called data cleaning aims to remove redundant and noisy data points from the majority class. Common techniques include neighborhood preprocessing (Cano *et al*., 2003), Tomek links (Tomek, 1976), and so on. In this chapter, Tomek links based data sampling approach is used to remove the majority class data-points. In general, data from the majority class can be categorized into four types: noise, borderline, redundant, and safe. Removing noisy, borderline, and redundant data from the majority class can improve the prediction accuracy by making the output class more distinguishable. In this chapter, the Tomek links based data sampling method was used. Tomek links use Euclidean distance information of input data points to identify borderline and noisy data.

For example let us assume p_i and p_j to be two data points that belong to the minority class, and assume n_i to be the data point representing the majority class. The

distance between p_i and p_j is assumed to be $\delta(p_i,p_j)$, whereas the distance between n_i and p_i is assumed to be $\delta(p_i,n_i)$. The data point n_i will be Tomek link to data point p_i, if $\delta(p_i,n_i) < \delta(p_i,p_j)$. All identified majority class forming the Tomek links are removed iteratively. The process is repeated until the Tomek links are identified. In this research, the logic illustrated in Figure 4.10 was used for data sampling.

Figure 4.11 describes the application of the logic on one such dataset. For the particular dataset, the algorithm ran for 50 iterations and the data points from the majority class were removed. Comparing just the minimum distance interclass data points and minimum distance out of class data points made the process computationally efficient. Using the Tomek links, the imbalance in the dataset is reduced by 10%.

The application of Tomek links reduces the class imbalance only up to a certain limit as it depends on a distance function defined for the dataset. To further reduce the class imbalance, genetic algorithm is employed (Goldberg, 1986). The main aim is to enhance the capability of data-mining classifiers by selecting a near- optimal training set. Obtaining an optimal training set is not possible in polynomial time as the subset selection problem is *NP*-hard in nature. The operational wind turbine data recorded by SCADA system mostly have normal cases (e.g., case when wind turbine is functioning normally); whereas, the fault occurrence are very less, even if the concerned wind turbine is fault prone. This imbalance in the output class makes impairs the prediction ability of any data mining algorithms. Traditional approach to deal with the class imbalance in the dataset often lack in generalization, or cause over training of the dataset.

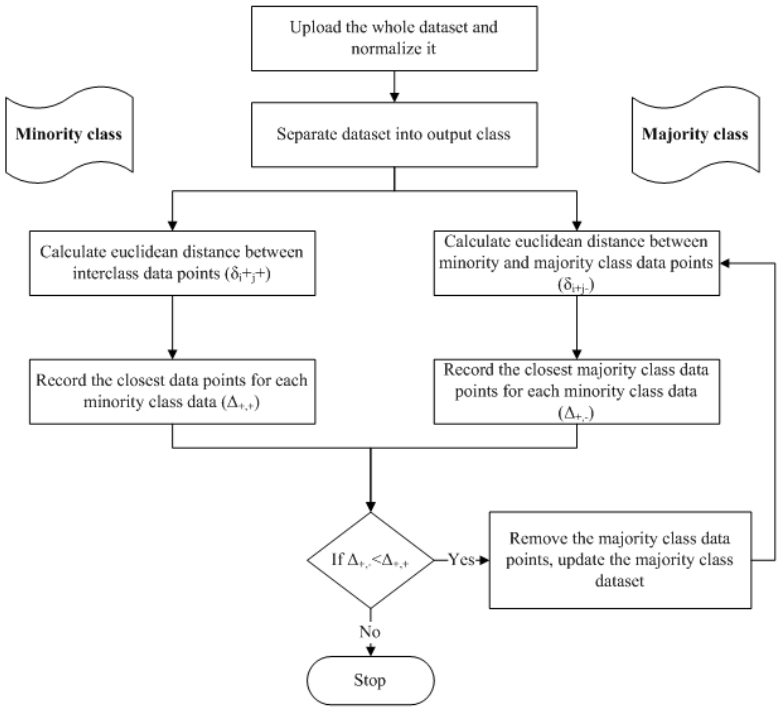


Figure 4.10. Identifying Tomek links.

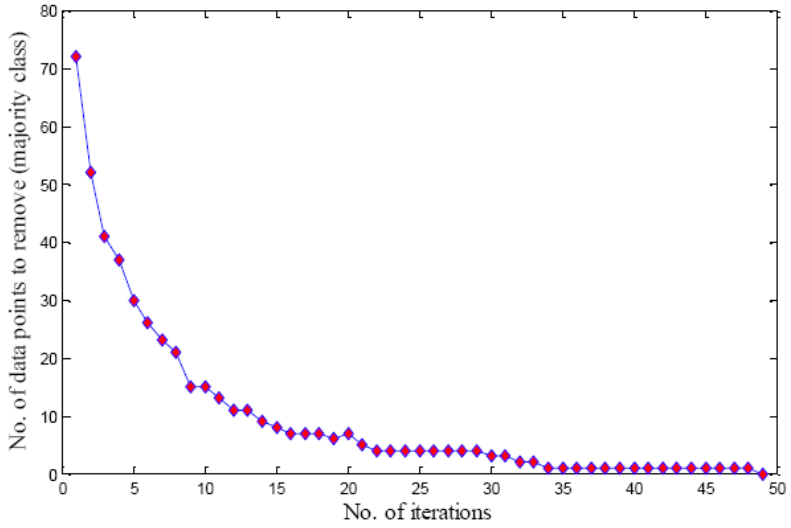


Figure 4.11. Iterative sampling of dataset *t* +21 using Tomek links.

Evolutionary algorithms such as genetic algorithm (GA), simulated annealing (SA), etc., have been successfully used in the data reduction task; also refer as feature

selection (Cano *et al*., 2003, Aha *et al*., 1991, Reeves and Bush, 2001). However, there are very few applications of evolutionary algorithms in balancing the output class. Description of the algorithm to obtain the near-optimal data sample is provided next.

An overall framework of the developed approach is shown in Figure 4.12. The keys steps of the approach are (1) initial sample selection, (2) application of evolutionary algorithm in identifying the best training set, (3) Learning data-mining classifier on training set, and (4) Testing the learned algorithm over blind test set. Genetic algorithm and data-mining based classifier control the process. Genetic algorithm provides the feasible training subset which is evaluated by the data-mining classifier. Depending upon the class imbalance ratio, the population size of the algorithm was determined, which was kept constant throughout the evolution. Mathematically, it can be written as:

$$Pop\_sz = ceil\left(size(Majority\_class)/Size(Minority\_class)\right) \quad (4.6)$$

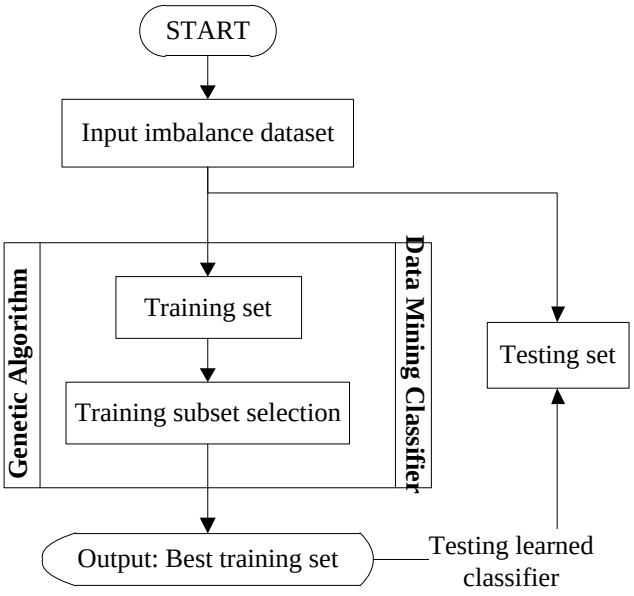


Figure 4.12. Overall framework of the algorithm.

The sampled data obtained after Tomek link application is divided into two parts e.g. training set and testing set. Two third (66.6%) of the data is used as training set, whereas, remaining one third (33.3%) is used for blind testing. The training data is divided into balanced subsets (equal to the population size) in which majority class instances are picked randomly from the original training data without replacement and all minority class instances are used in all training subsets (Figure 4.13). The search space associated with the algorithm constitutes all the subsets of training set. A binary representation is used where the value 1 indicates the instance at the particular location is included in the training subset, 0, otherwise. The binary representation is applied in selecting data from majority class. In case of minority class, all the instances are kept in all training subsets. A sample representation is shown in Figure 4.14.

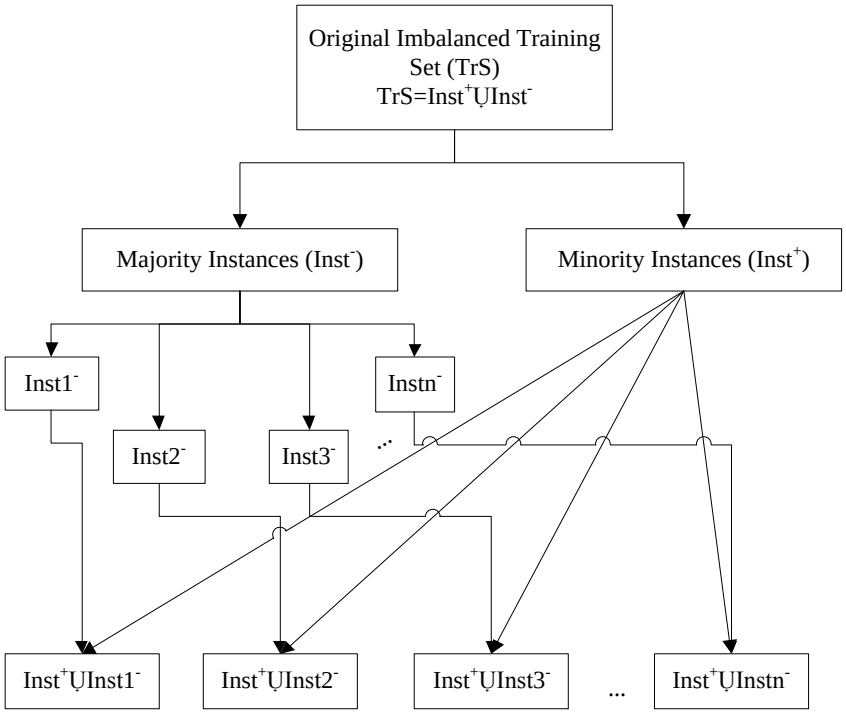


Figure 4.13. Approach to create balanced training subsets.

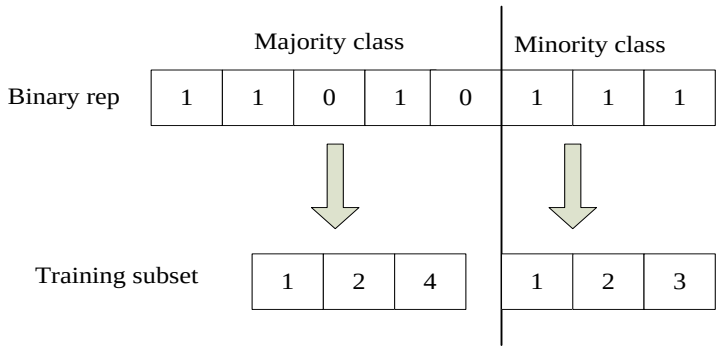


Figure 4.14. Chromosome representation of a solution for two class dataset.

The fitness function to evaluate the performance of the selected training set is based upon the confusion matrix. Considering the class imbalance in the dataset, accuracy obtained from each class is included in while evaluating the fitness function. A well-known metric e.g., geometric mean (see Equation 4.5) is used as our final fitness function which includes the accuracy obtained from both normal and fault class. The value of gmean lies in between 0 and 1. Higher the value of gmean better is the selected training subset.

A single point cross-over is used on selected parents, which is followed by a flip mutation. A constraint related with maximum number of training subset from majority class is imposed in order to maintain a balance in the output class. In mutation, randomly selected bit with value 0 is replaced with 1 and a bit with value 1 is replaced with 0. It is important to mention that only the bits belonging to majority class dataset is used in both recombination operators. Overall, the number of children equal to the size of initial population is generated and their fitness is evaluated against the parents' population to select the best candidates for next generation. Following stopping criteria are used (1) No improvement in the solution for 10 consecutive generations, (2) Maximum number of generation (e.g., GEN_MAX 100).

Four well-known data-mining algorithms, namely, Multilayered perceptron (MLP), boosting tree, *k*-NN (*k*=10), and support vector machine (SVM) were used to build a prediction model at the t + 21 time stamp. The best performing algorithm was used to perform predictions at 24 time stamps. A description of the dataset for algorithm selection is shown in Table 4.15.

Table 4.15. Description of the dataset for model construction.

Data Set	Start Time Stamp	End Time Stamp
t + 21 (overall)	3/1/2010 12:00 AM	3/31/2010 11:20 PM
t + 21 (train)	3/1/2010 12:00 AM	3/20/2010 09:30 PM
t + 21 (test)	3/21/2010 09:60 PM	3/31/2010 11:20 PM

Table 4.16 describes the results obtained using various data-mining algorithms. Compared with other data-mining algorithms, boosting tree algorithms produced the best accuracy for fault class and overall best gmean. It is therefore considered best to build prediction models at all 24 time stamps. Poor sensitivity obtained using the remaining data mining algorithms depicts their inability to classify the minority (fault) class instances, whereas the boosting tree algorithm appears to be insensitive to class imbalance by producing good sensitive results.

Results on the test dataset indicated that out of 37 fault instances, the boosting tree algorithm was correctly able to predict 31 instances, whereas in the case of normal instances, 101 were correctly predicted out of 127.

Table 4.16. Performance of data-mining algorithms at t + 21 time stamp.

	Accuracy (%)	Specificity (%)	Sensitivity (%)	Gmean
k-NN(k=10)	80.8	82.8	61.5	0.713
MLP	78.5	91.3	50.0	0.675
SVM	78.7	78.7	0.00	0.000
Boosting tree	**80.4**	**79.5**	**83.7**	**0.815**

4.3.4. Experimental results predicting generator brush worn faults

Two-thirds of the dataset was used to train the boosting tree, whereas the remaining one-third was used for testing. Table 4.17 describes the results of the analysis on time-series datasets. The results of three cases, (1) the original dataset; (2) the sampled using Tomek links; and (3) the sampled dataset using Tomek links and genetic algorithm, are displayed in Table 4.17.

Prediction accuracy on the sampled dataset using both Tomek links and genetic algorithm was found in the range 82.1%-97.1% for all time stamps, whereas accuracy in predicting fault cases was found in the range 71.4 %-95.8.0%. The significant improvement in accuracy indicates the effectiveness of data sampling methods.

Table 4.17. Results obtained from boosting tree algorithm.

Time Stamp [10 min]	Original			Tomek links			Tomek links + genetic algorithm		
	Acc (%)	Sen (%)	Spe (%)	Acc (%)	Sen (%)	Spe (%)	Acc (%)	Sen (%)	Spe (%)
t + 3	75.8	77.7	64.9	77.9	77.2	82.5	97.1	100	94.7
t + 6	74.3	74.0	77.5	76.1	76.7	73.4	95.9	96.0	95.8
t + 9	75.0	75.5	70.2	69.1	69.0	69.7	94.3	95.2	93.7
t + 12	76.3	76.7	69.7	76.7	77.0	74.7	94.2	100	88.8
t + 15	74.2	75.7	60.0	80.4	79.5	83.7	91.3	95	88.4
t + 18	73.9	74.6	66.6	74.5	76.4	65.3	83.4	98.5	82.8
t + 21	74.7	74.6	78.1	79.5	79.7	77.2	87.7	100	80.0
t + 24	72.5	73.9	63.3	70.8	74.7	58.0	87.5	91.6	86.2
t + 27	81.1	83.3	55.0	73.2	74.1	65.3	86.3	92.3	84.0
t + 30	75.7	81.6	53.3	75.4	80.2	55.0	88.2	86.6	89.4
t + 33	83.1	95.7	28.1	72.2	69.8	76.6	86.2	89.4	84.6
t + 36	83.8	90.7	40.0	82.2	89.4	70.8	87.4	84.4	89.0
t + 39	76.3	82.3	41.6	76.0	78.7	64.0	88.2	90.6	81.8
t + 42	83.4	92.8	29.1	77.7	86.7	50.0	84.8	87.5	83.9
t + 45	88.7	97.7	37.5	76.8	76.0	79.1	83.1	86.6	83.3
t + 48	76.0	84.1	22.2	77.2	80.0	70.5	84.1	83.3	84.4
t + 51	68.7	79.2	45.0	84.1	81.4	90.0	84.0	83.3	84.2
t + 54	74.7	79.2	54.5	74.1	80.9	56.2	84.2	81.2	86.3
t + 57	67.9	69.3	60.8	74.0	76.9	66.6	82.2	83.3	81.8
t + 60	75.0	78.3	46.6	77.2	79.0	70.5	80.0	92.8	71.4
t + 63	70.9	74.1	58.3	76.3	72.9	91.6	77.6	77.7	77.6
t + 66	60.6	61.6	53.8	64.0	64.5	63.1	82.1	59.2	92.9
t + 69	63.0	64.5	53.3	55.0	54.1	57.1	77.0	69.5	78.4
t + 72	67.9	72.6	50.0	76.1	82.1	66.6	72.3	60.8	83.3

*Acc=overall accuracy, *Sen=sensitivity (accuracy in predicting fault cases),
*Spe=specificity (accuracy in predicting normal cases)

Figure 4.15 compares the performance of all three cases in terms of the gmean measure. The gmean obtained using Tomek links and genetic algorithm based data sampling was always found to be better than the other two cases. In addition, it is important to mention that the misclassification costs for both fault and normal class output were kept the same; that is, equal to 1. The reasonable amount of accuracy for all 24 time-stamps indicates that the boosting tree algorithm is able to learn efficiently even in the case of class imbalance.

The results presented in this section offer early prediction of emerging faults. This allows operators to schedule maintenance and minimize operations and maintenance cost. In addition the potential of collateral and severe faults is reduced.

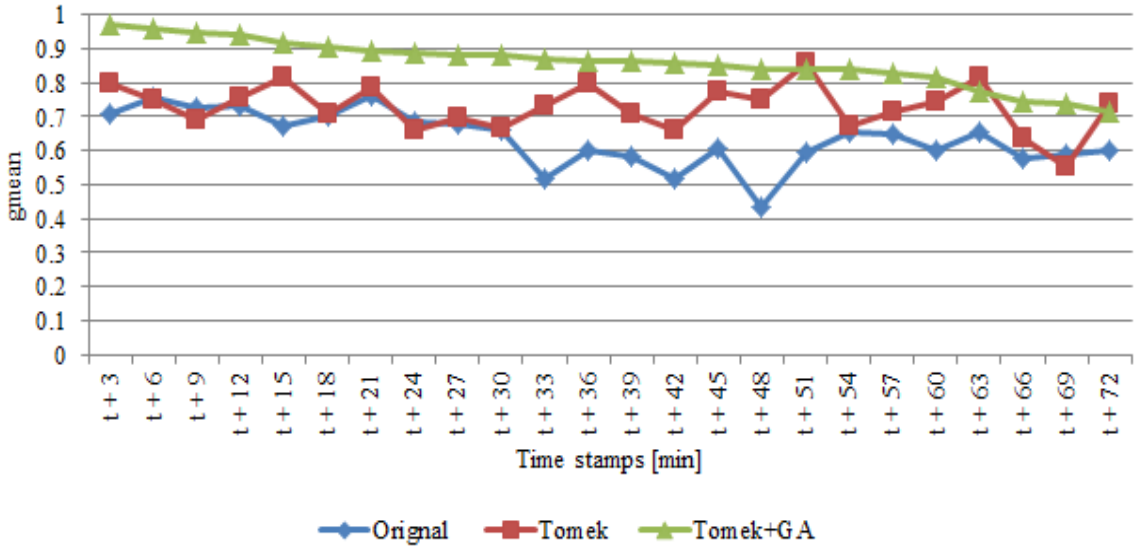


Figure 4.15. Relative improvement in gmean score over 24 time stamps.

4.4. Summary

In this chapter, fault associated with wind turbine components namely turbine blades (blade angle implausibility), and generators (generator brush worn out) were investigated. Data balancing approaches based on Tomek links, genetic algorithms were applied to

make the output class balance. The prediction accuracy is greatly improved by applying advanced data pre-processing methods. The solution obtained by genetic programming and boosting tree algorithm can be easily interpreted. Analyzing fault related snapshot files contains important information about fault sequence and can be used in identification of other components/sub-components affected by the faults under consideration.

CHAPTER 5

ANOMALY DETECTION BASED APPROACH TO ANALYZE BEARING OVERTEMPERATURE EVENTS

5.1 Introduction

Bearings are the essential part of turbine generators and gearbox. The dynamic and unpredictable stress cause the bearings to wear prematurely, leading to increased turbine maintenance costs, and could lead to sudden, expensive turbine breakdowns. Over-temperature is another issue impacting the performance of turbine bearings.

The temperature monitoring systems of wind turbines issue alarms in the event of faults. However, such alarms are usually signaled once the damage to the component has already occurred. There is a need to find solutions for predicting faults ahead of time, in order to avoid extensive damage of turbine components. Data-mining algorithms build fault prediction models using data collected by the supervisory control and data acquisition (SCADA) systems. Such data—e.g., power output, gearbox bearing temperature, and generator speed—is usually acquired for over one hundred turbine parameters. Amplitude demodulation (Amirat *et al*., 2010), Fast Fourier transformation (Lin *et al*., 2002), wavelet decomposition (Cusido *et al*., 2008) etc. are some common method used in the literature to analyze bearing faults. Such approaches require vibration data which often not been recorded by the typical SCADA systems. Another stream of research pertaining with bearing failures is anomaly detection in which the normal bearing behavior is modeled and tested against abnormal behavior (Zaher *et al*., 2009, Schlechtington and Santos, 2011). Such anomaly detection approaches ensures fault prediction ahead of the time.

In the present chapter, an approach based on anomaly detection is applied to identify and predict the bearing over-temperature. Various abnormal trends in the generator bearing temperature are analyzed.

5.2 Data description and analysis

The data used in the research reported in this chapter has been collected from the supervisory-control and data-acquisition (SCADA) systems of a large wind farm. Here, high-frequency data (i.e., 10 s) from twenty-four wind turbines over a period of four months (Aug 2009-Dec 2009) is used to analyze bearing faults. Table 5.1 displays the temperature range for the turbine bearings of the generator, gearbox, and shaft. Turbines 3 and 15 are affected by over-temperature generator bearing B and thus are considered for construction of data-mining prediction models. In the next section, training and testing strategies of the data-mining algorithms are discussed.

5.2.1 Training set selection

Modeling the normal behavior of a generator bearing requires input data with varying temperatures. Therefore, the data from different wind turbines is used for model development. The box plot in Figure 5.1 displays the generator-bearing temperature data from 24 wind turbines that is highly variable. To increase the number of training examples, data from multiple wind turbines has been used. Box plot is used to select training data from turbines behaving differently. The data from turbines 2, 5-8, 13-14, 16, 19, and 21-23 constitutes the training dataset.

Figure 5.2 depicts of the range of the generator bearing B temperature used to model normal turbine behavior (i.e., behavior not affected by faults). According to current wind-turbine operations practices, the generator bearing B temperature exceeding 90°C requires turbine shutdown. Turbines 3 and 15 have shown some abnormal behavior (see Table 5.1) and therefore their data has been used for testing abnormal behavior. The description of training and testing dataset is provided in Table 5.2.

Table 5.1. Turbine bearing temperature ranges.

Turbine index	Temp. generator bearing A(°C)		Temp. generator bearing B(°C)		Temp. gearbox bearing(°C)		Temp. shaft bearing(°C)	
	min	max	min	max	min	max	min	max
1	14	66	20	87	24	70	-4	36
2	13	74	16	77	22	71	-7	35
3*	-37	70	**-37**	**101**	-37	69	-36	36
4	-39	46	-39	64	-39	69	-38	36
5	-273	62	-273	73	-273	70	-273	34
6	-38	57	-38	73	-37	68.8	-37	34
7	-37	57	-37	69	-36	66	-36	33
8	10	43	13	62	32	67	3	37
9	-36	42	-39	49	-36	64	-35	36
10	5	42	8	48	23	68	4	36
11	6	40	15	46	26	64	0	36
12	-273	77	-273	69	-273	64	-273	32
13	-39	48	-40	52	-39	68	-38	35
14	12	52	21	76	26	69	-2	32
15*	-273	54	**-273**	**107**	-273	68.2	-273	34
16	8	48	12	64	25	67	-1	36
17	-37	78	-37	82	-37	70	-36	36
18	8	46	11	59	26	70	2	37
19	-37	49	-37	80.3	-36	77	-35	37
20	-39	44	-38	53	-38	65	-37	36
21	8.6	80	10	86	15	69	-8	34
22	-37	74	-37	77	-37	65	-36	34
23	-38.5	59	-38	75	-38	69	-37	36
24	-37	47	-37	47	-37	70	-36	36
* Of interest								

Thus, the training dataset consists of the data from 11 wind turbines. In order to reduce the data-dimension, data from 11 turbines are averaged to construct the training dataset.

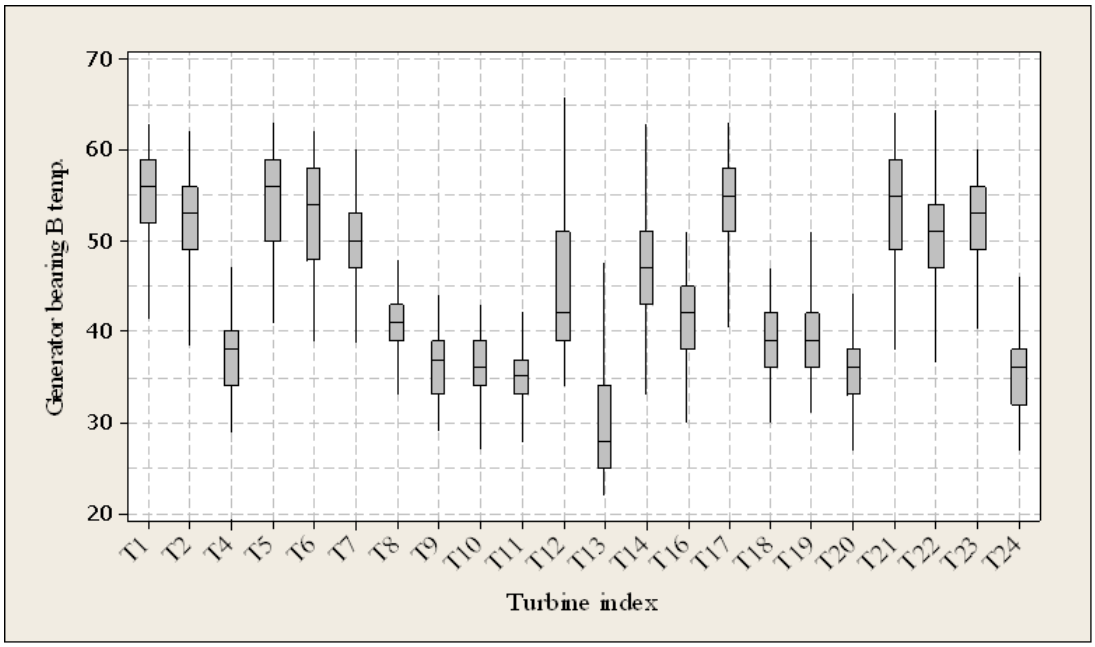


Figure 5.1. Box plot of the generator bearing B temperature of wind turbines.

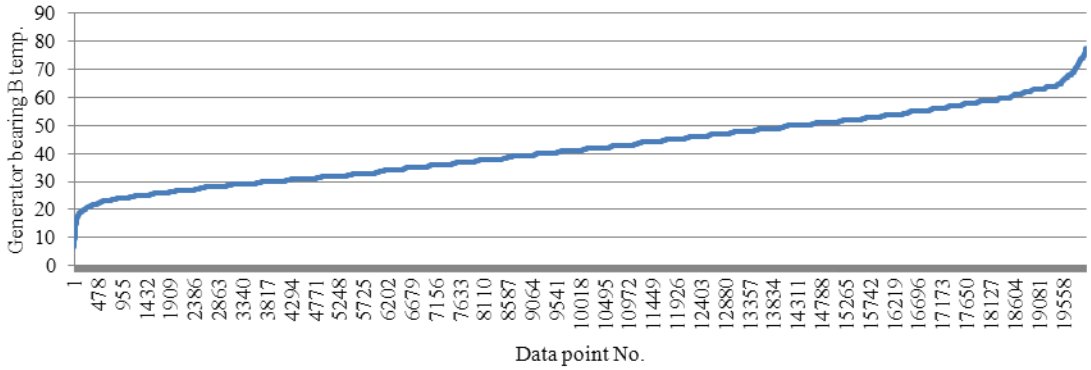


Figure 5.2. The generator bearing B normal temperature range.

Table 5.2. Dataset for the anomaly detection.

Dataset	Start time stamp	End time stamp	Turbines considered
Training and model validation	8/1/2009 12:00:00 AM	12/8/2009 11:59:50 PM	5-8, 12-14, 16, 19, 21-23
Testing-normal behavior	8/1/2009 12:00:00 AM	10/1/2009 11:59:50 PM	1,12
Testing-abnormal behavior	10/2/2009 12:00:00 AM	12/8/2009 11:59:50 PM	3

5.3 Modeling the normal bearing behavior

5.3.1 Parameter selection for model construction

To capture the normal behavior of the generator bearing, parameters impacting the bearing temperature are selected initially using domain knowledge (reduction from 100 to 50 parameters) and the final selection with data-mining algorithms. Three different data-mining algorithms—wrapper with genetic search (WGS) (Kohavi and John, 1997; Goldberg, 1989), wrapper with best first search (WBFS) (Sbihi, 2007), and boosting tree algorithm (BTA) (Kudo and Matsumoto, 2004)—have been applied to select the most relevant parameters for prediction of the generator bearing B temperature. The wrapper approach uses supervised learning to select relevant parameters by performing 10-fold cross validation. Table 5.3 lists the 10 most relevant parameters. In total, 18 different input parameters are used for the development of regression models.

5.3.2 Model construction

The model for predicting the generator bearing B temperature is presented in (5.1).

$$y(t) = f\begin{pmatrix} VA(t), VC(t), CA(t), CC(t), NR(t), G2T(t), GBT(t), \\ AT(t), NT(t), GCAT(t), VEH(t), HP(t), MBT(t), \\ TBT(t), HBT(t), TCB1(t), TH(t) \end{pmatrix} \tag{5.1}$$

Where y (t) is the target generator bearing B temperature expressed as a function of 18 input parameters. The model (5.1) is built with four neural network (NN) algorithms.

Table 5.3. Relevant parameters selected by data-mining algorithms.

No.	**WGS**	**WBFS**	**BTA**
	10-fold cross validation	*10-fold cross validation*	*Parameter importance*
1	**Voltage phase A,C (100)**	**Nacelle revolution (100)**	**Temperature, gearbox 1.00**
2	**Current phase A,C(100)**	**Current phase C (80)**	**Temperature, generator cooling air 0.98**
3	**Nacelle revolution (100)**	**Temperature hub (80)**	**Temperature, generator 2 0.95**
4	**Torque (100)**	**Temp. control box axis 1 (60)**	**Temperature Main Box (°C) 0.94**
5	**Temperature generator 1 (100)**	**Voltage phase C (50)**	**Temperature Control Box Axis 1 (°C) 0.70**
6	**Temperature gearbox (100)**	**Temperature top box (50)**	Temperature, generator 1 0.64
7	**Temperature ambient (100)**	Drive train acceleration (50)	Temperature Control Box Axis 3 (°C) 0.62
8	**Temperature Nacelle (100)**	Generator speed (50)	State fault 0.62
9	**Temperature generator cooling air (100)**	Blade 3 pitch angle (actual) (40)	Blade 3, actual value 0.59
10	**Temperature main box (100)**	Temperature bearing A (40)	Blade 1, actual value 0.58

To select the best performing algorithm metrics such as the absolute error (AE) mean absolute error (MAE), relative error (RE), mean relative error (MRE), and R^2 goodness of fit have been used (see (5.2) – (5.4)).

$$MAE = \frac{\sum_{i=1}^{N} \left| \hat{y}_i(t) - y_i(t) \right|}{N} \tag{5.2}$$

$$MRE = \frac{\sum_{i=1}^{N} \left| \frac{\hat{y}_i(t) - y_i(t)}{y_i(t)} \right| \times 100}{N} \tag{5.3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(\hat{y}(t) - y(t) \right)^2}{\sum_{i=1}^{N} \left(y(t) - \bar{y} \right)^2} \tag{5.4}$$

Table 5.4 presents prediction results produced by five neural networks (NNs) based on the training data (Rumelhart and McClelland (1986)). The training dataset is divided into three parts namely training, testing, and validation in the ratio 80:10:10. The algorithm shown in Table 5.4 were produced the best results among 100 randomly selected NNs. In order to obtain the best five NNs, the process is repeated five times. The NNs are generated by varying the no. of neurons in hidden layer, and hidden, output activation functions. The no. of neurons were kept in the range 5-25, whereas, activation functions namely tanh, exponential, identity, and logistics are used as activation functions. The networks structures were optimized using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method (Broydon, 1970; Fletcher, 1970; Goldfarb, 1970; and Shanno, 1970). The neural network algorithm NN2 outperforms the other data-mining algorithms. It provides consistently best results for training, testing, and validation dataset. The structural details of the neural network algorithm are provided in Table 5.5.

Table 5.4. Performance of NN on training set.

Net. name	MAE			MRE (%)			R^2		
	A^*	B^*	C^*	A^*	B^*	C^*	A^*	B^*	C^*
NN 1	0.802	0.804	0.919	1.740	1.715	2.032	0.993	0.992	0.984
NN 2	**0.765**	**0.775**	**0.860**	**1.650**	**1.642**	**1.880**	**0.994**	**0.993**	**0.987**
NN 3	1.050	1.036	1.115	2.360	2.250	2.521	0.987	0.986	0.984
NN 4	0.899	0.921	0.992	1.940	1.963	2.168	0.991	0.990	0.984
NN 5	0.810	0.844	0.896	1.750	1.796	1.964	0.993	0.991	0.986
*A: Training data, B: Test data, C: Validation data									

Table 5.5. Description of neural networks.

Net. Name	Network structure	Training algorithm	Hidden activation	Output activation
NN1	MLP 18-16-1	BFGS 380	Tanh	Identity
NN2	**MLP 18-17-1**	**BFGS 622**	**Logistic**	**Identity**
NN3	MLP 18-5-1	BFGS 214	Logistic	Exponential
NN4	MLP 18-15-1	BFGS 370	Logistic	Logistic
NN5	MLP 18-16-1	BFGS 377	Logistic	Exponential

Figure 5.3 displays the run chart of actual and predicted bearing temperature of validation dataset. The results indicate that algorithm was able to correctly predict the normal bearing temperature range. Compared with the anomaly detection models developed in the literature (Schlechtingen, and Santos, 2011), the prediction error was found to be the least in the present model. This is due to optimal NN structure over fixed number of runs.

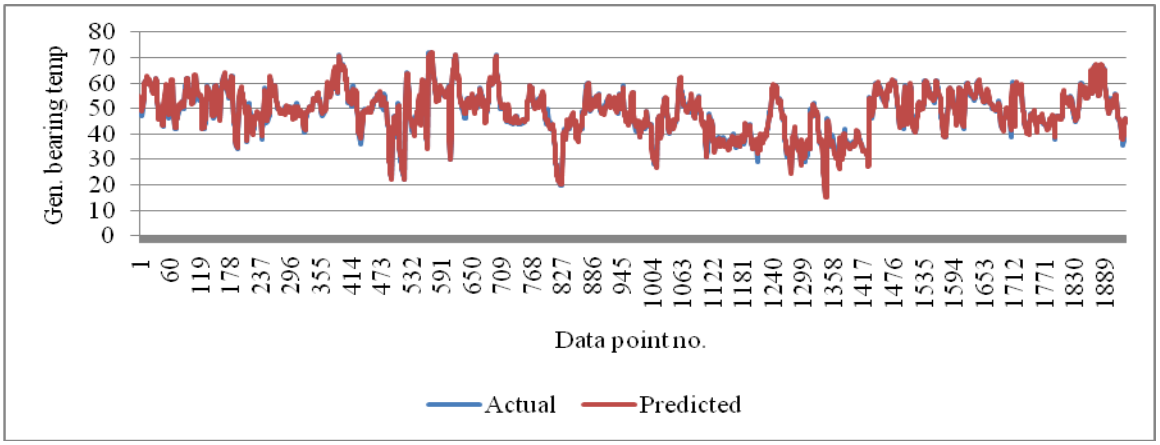


Figure 5.3. Run chart of actual and predicted bearing temperature (NN2).

5.3.2.1 Residual analysis

In this section, the best performing algorithm, i.e., NN2 is further analyzed with respect to error residuals. The aim here to make sure that the error residuals lies within the certain limit so that the false alarm during testing the abnormal behavior can be

minimized. In order to better observe the signal errors, the residuals are standardized (Equation 5.5). With 95% confidence interval, standardized residuals should lie within ±2.00 error band.

$$Stand\_res_i = \frac{r_i}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - \bar{r})^2}} \quad (5.5)$$

Where, r_i is the error residual for observation *i*, N is the total number of observations. Figure 5.4 displays the standardized error residuals of the run chart shown in Figure 5.3. With the specified control limits on residuals, almost 6.00% of the data points were found out of limits.

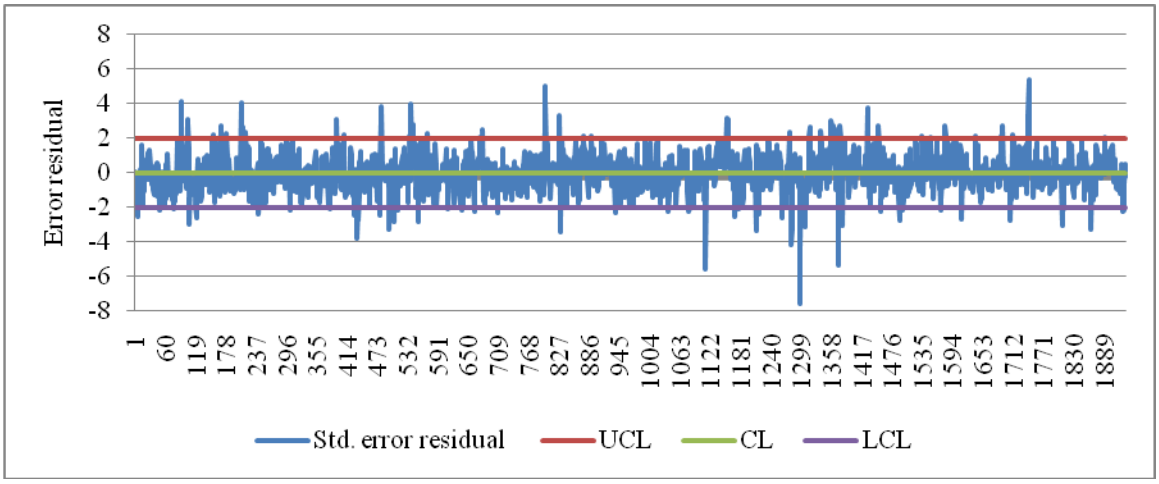
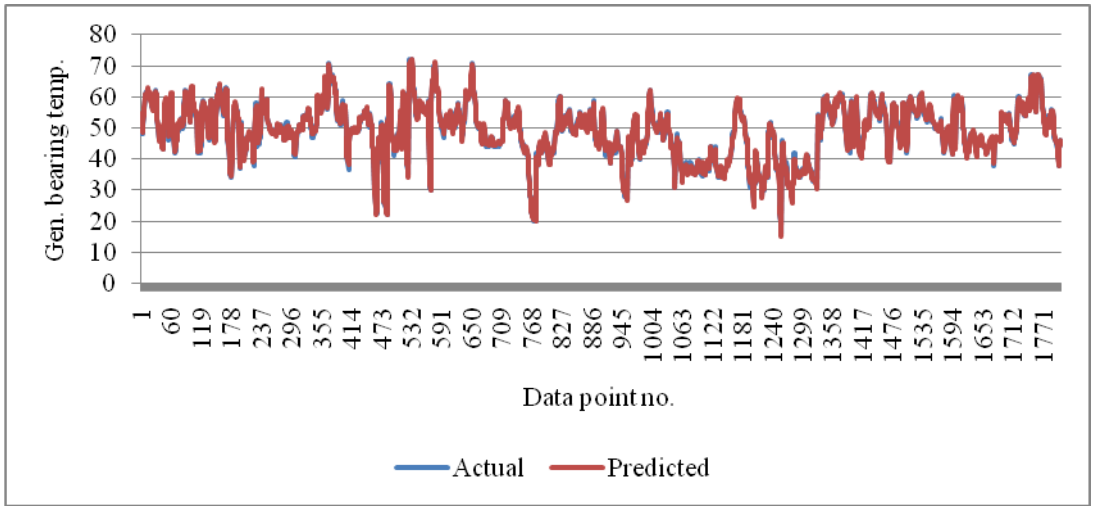


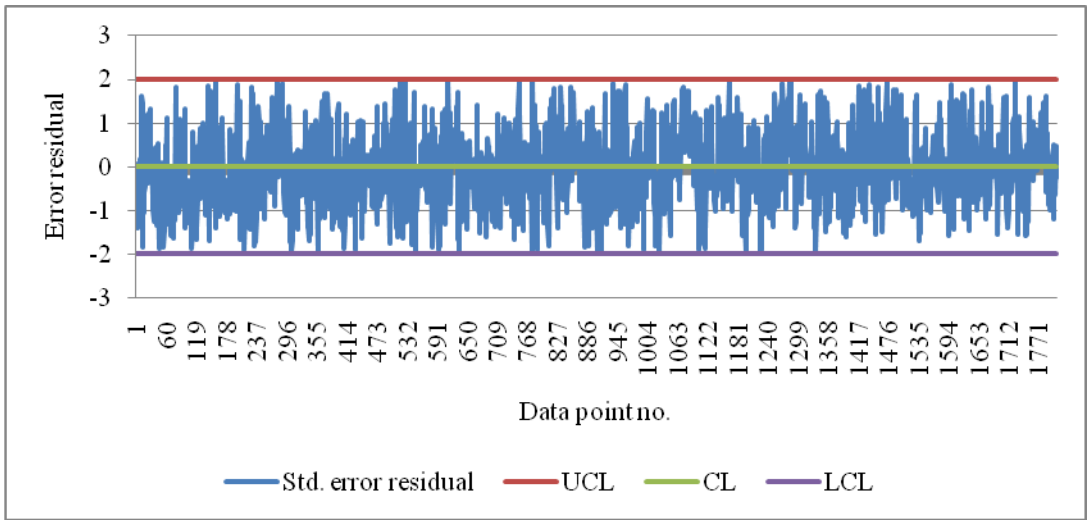
Figure 5.4. Standardized error residual of Figure 5.3.

The NN2 algorithm is trained again after removing the data-points causing out of control error residuals from the whole training dataset. The training process is repeated till all the error signals are not found within the control limits. After three iterations, the error residual were found within the bounds. Almost 7% of the data points were removed in the process. Figure 5.5(a) displays the run chart of validation data points, and Figure 5.5 (b) displays the error residuals of the data-points after three iterations. The NN2 algorithm

used less learning iterations (i.e. BFGS 320) to get all the residuals within control. The improved result obtained by NN2 is provided in Table 5.6.



(a)



(b)

Figure 5.5. Results obtained by NN2 after algorithm retraining (a) run chart comparison, (b) error residuals.

Table 5.6. Performance of NN2 after retraining.

Net. name	MAE			MRE (%)			R^2		
	Training error	Test error	Validation error	Training error	Test error	Validation error	Training perf.	Test perf.	Validation perf.
NN2	0.659	0.663	0.693	1.39	1.379	1.47	0.9916	0.9905	0.9903

In the next section, analysis on the turbine normal behavior is performed. Two turbines namely T1 and T12 are analyzed.

5.3.3 Testing normal bearing behavior

The best performing algorithm (i.e., NN2) is selected to test the normal behavior of wind turbines. Two turbines, Turbine 1 and Turbine 12, have been selected for testing. Figures 5.6 and 5.7 display the scatter plot of actual and predicted results.

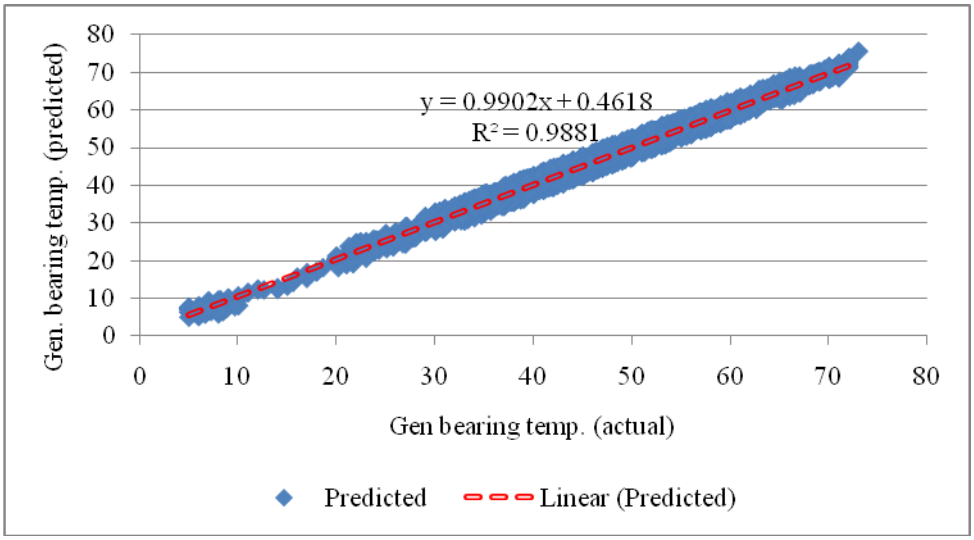


Figure 5.6. Comparison of actual and predicted values by NN2 (test turbine 1).

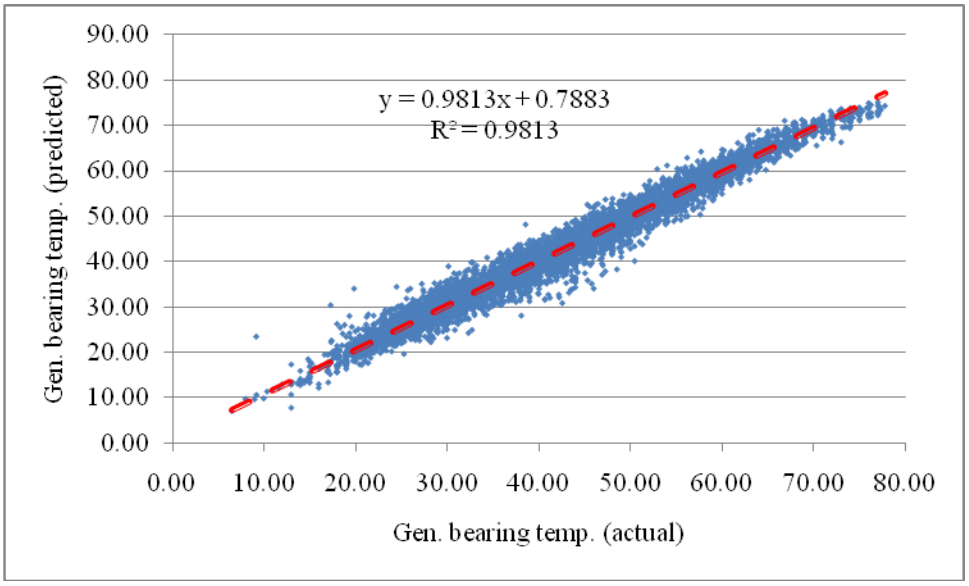


Figure 5.7. Comparison of actual and predicted values by NN2 (test turbine 12).

The results shown in Figures 5.6-5.7 indicate a high correlation between actual and observed values. NN2 applied on T1 provides MAE 0.89°C, and MRE 1.98%. Whereas, in case of T12 MAE and MRE (%) were 1.22°C, 3.19% respectively. The MAE and MRE values are within the acceptable range for both turbines. Thus, the model accurately captures the normal behavior of the wind turbines.

5.4. Analysis of experiments

5.4.1. Analyzing bearing abnormal behavior

In this section, results of abnormal turbine behavior are discussed. The normal behavior model constructed by best performing algorithm NN2 i.e., MLP 18-17-1, hidden activation: logistic, output activation: identity is exploited to detect abnormal behavior in Turbine 3. One week data i.e. from 3/12/2011-3/19/2011 is used where the generator bearings experience over temperature issues. Figure 5.8 display the histogram of testing turbines which indicates over-temperature instances.

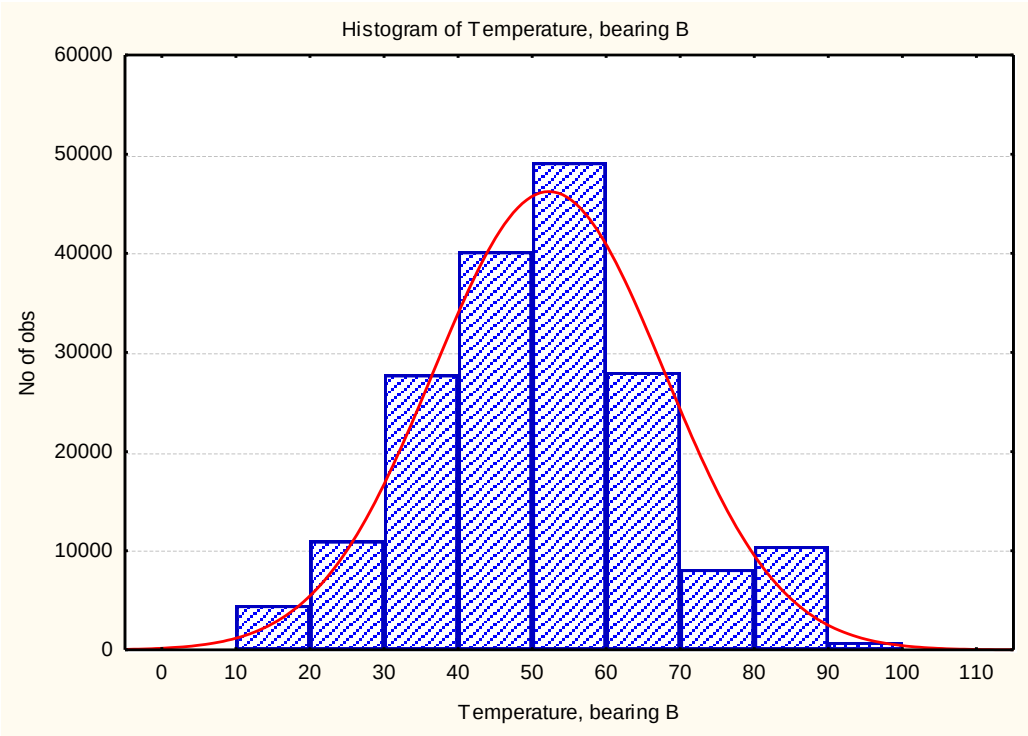


Figure 5.8 Histogram of testing turbine (i.e. test turbine 3)

Figure 5.9 illustrates the trend of the actual and predicted generator bearing B temperature of Turbine 3. The temperature at the peak points is more than 90°C, which is above the specified upper limits for the generator bearings. Such temperature may cause damage to the bearing and can adversely affect operations of the wind turbine; thus, the ability to predict this condition ahead of the time is desirable.

The predicted results (Figure 5.9) closely follow the measured values of the temperature. However, in the event of abnormal temperature the error is signaled, i.e., residual temperature exceeds the allowable limit. The standardized error residuals are provided in Figure 5.10. The error residuals at over temperature points clearly exceed the specified limits, and therefore alarm can be signaled.

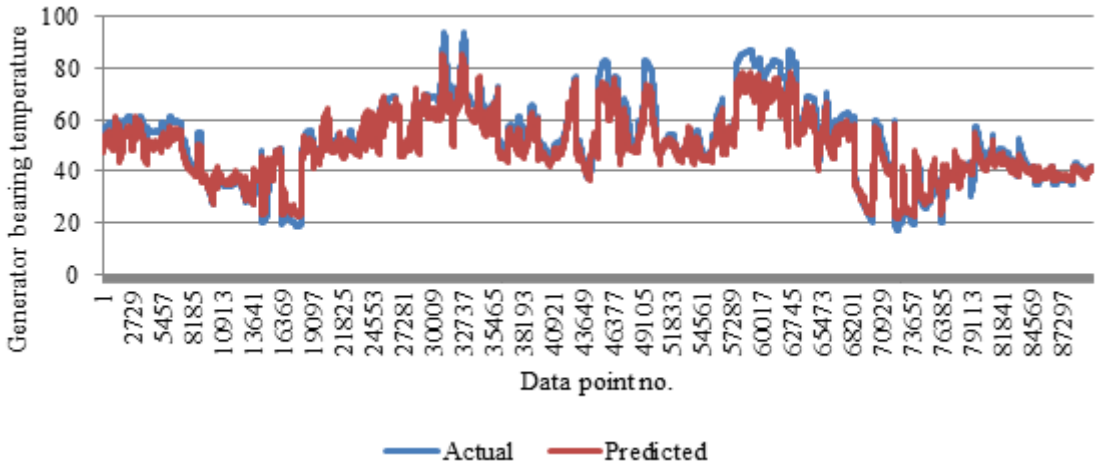


Figure 5.9. Run chart of actual and predicted values (test turbine 3).

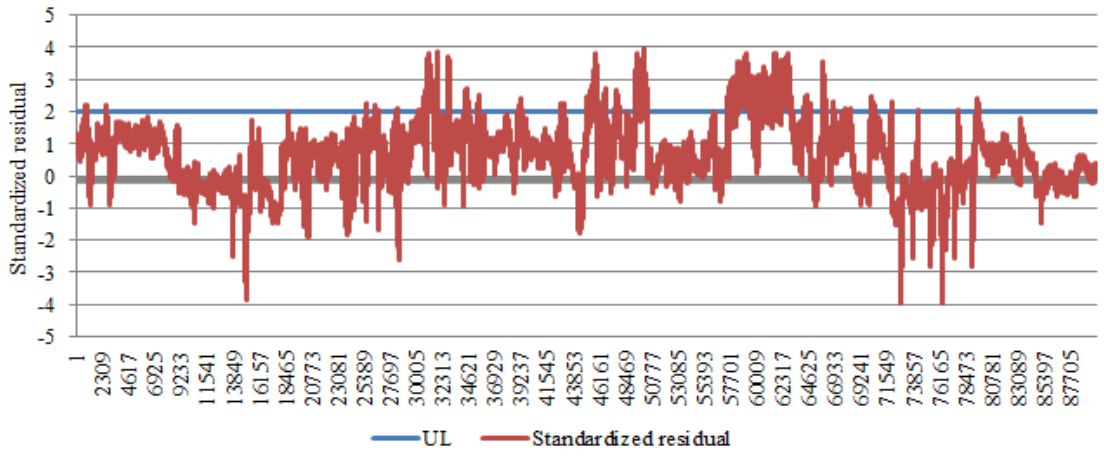


Figure 5.10. Standardized error residuals of run chart (test turbine 3).

5.4.2 Moving average filtering and improved residual analysis

The error residuals displayed Figure 5.10 indicate the abnormal events. However, in order to remove the effect of noise, a low pass averaging filter is applied. For bearing temperature monitoring, the 10-sec data is too frequent as temperature change may not be so abrupt due to thermal inertia. Thus, in this chapter, an approach based on moving average window is applied (Kusiak *et al.*, 2010). Figure 5.11 displays the moving average window, where N is the total data points, and i is the window width. A window of size

600 data points is arbitrarily considered. Figure 5.12 displays the smooth residuals, where four over temperature events are clearly identified.

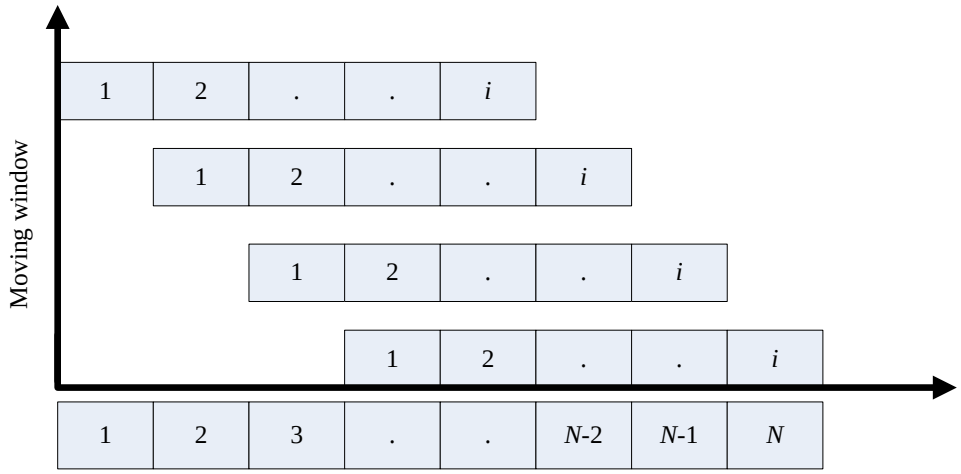


Figure 5.11 Error residual moving average window.

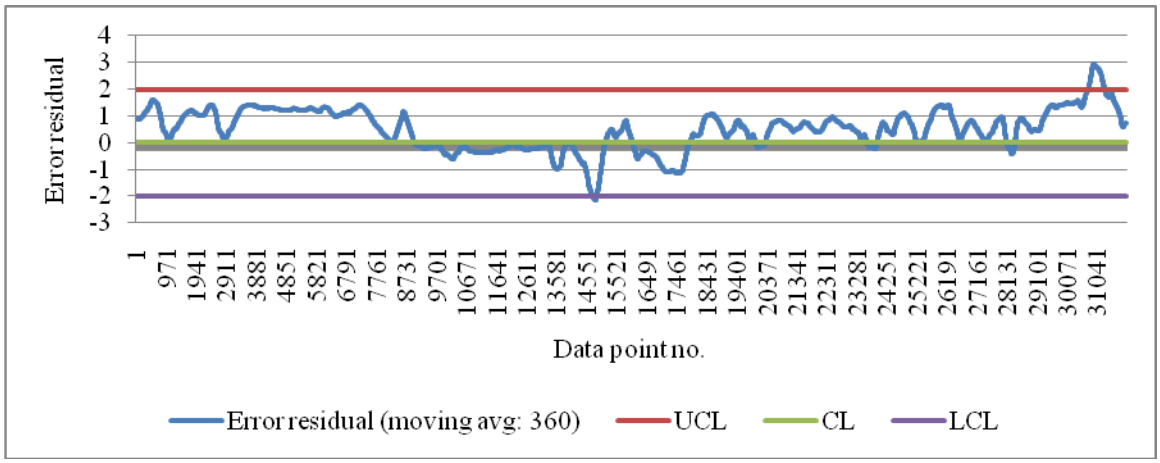


Figure 5.12 Denoised error residual with moving average window of size 360.

5.4.3. Computing prediction length

The analysis shown in previous section clearly demonstrate the role of anomaly detection approach in identifying the over temperature issues. However, in order to avoid bearings undergo over temperature, predicting such cases in advance is of interest. Figure 5.12

indicate six events, where, the standardized error exceeds the specified error limit (i.e., 2°C). It is important to mention that, the analysis presented in this chapter is targeted to identify and predict over temperature cases only.

Figure 5.13 provides the zoomed-in display of the first over temperature pattern found in Figure 5.12. Based on the set error limits (i.e. error residual equals 2°C), the alarm will be signaled at least 210 time steps before the actual event. In the Figure 5.12, each data-point is separated at 10 second interval, thereby signaling the alarm almost 35 minutes ahead of the actual case.

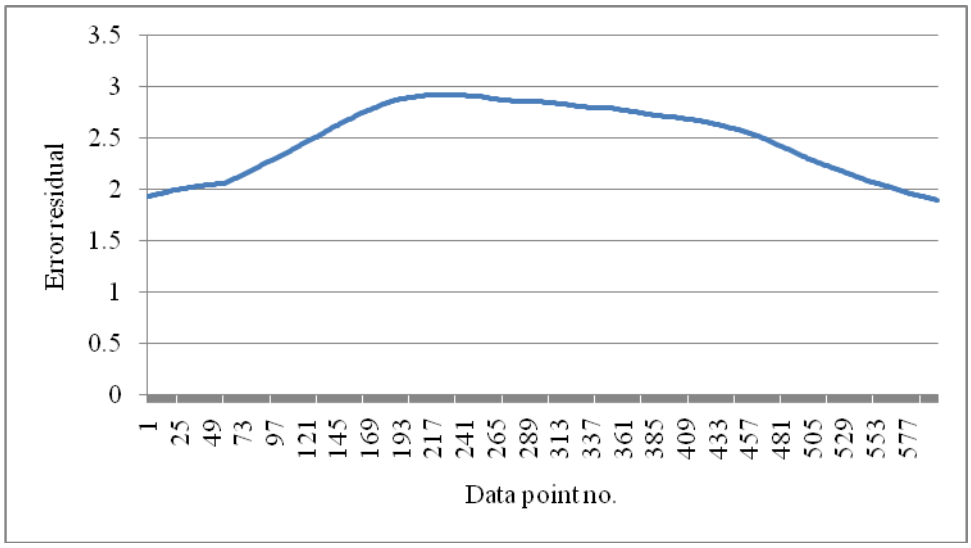


Figure 5.13 Error residuals trend during first over temperature event.

All five over temperature cases were analyzed in the similar fashion, resulting average prediction length close to 1.5 hrs.

5.5. Summary

In this chapter, a simple but effective approach to analyze the bearing over temperature is presented. The models developed by neural network algorithms are able to predict the

normal temperature trend with error less than 2%. The approach developed in this chapter can predict the over temperature up to 1.5 hours in advance, thereby providing decent time for maintenance. The models developed in this chapter are generic enough to detect under temperature bearing issues if required. Analysis on gearbox and shaft bearings can also be done in the similar fashion.

CHAPTER 6

VIBRATION ANALYSIS OF WIND TURBINE GEARBOX

6.1 Introduction

Gearboxes of wind turbines are fault-prone and costly to repair. Since its inception, the wind energy industry has experienced high gearbox failure rates (McNiff, 1990). Proper monitoring of gearboxes is necessary to avoid high repair and maintenance costs, which over time could exceed the turbine procurement cost. Vibration is one of the condition indicators of wind turbine components, and vibration data are useful in minimizing the damage of turbine components.

Vibration analysis of wind turbine data can be performed in frequency and time domains. The frequency domain may reveal intrinsic characteristics of the components that could be difficult to observe in the time domain. Fast Fourier analysis (FFT), wavelet analysis, and cepstrum analysis are commonly used in the frequency domain (Becker, 2006; Verbruggen, 2003).

This chapter utilizes high frequency data from a study of wind turbine gearboxes, performed by the National Renewable Energy Laboratory (NREL). The faults of the components are identified in the time domain and then validated by analysis in the frequency domain. The vibration of the faulty component is predicted with data mining algorithms.

6.2 Data for analysis

The data used in this research was provided by the National Renewable Energy Laboratory (NREL) through a consortium called as Gearbox Reliability Collaborative (GRC). The test turbine used in this study is stall-controlled, three-bladed with a rated power of 750 kW. The data was sampled at high frequency (i.e., 40 kHz), and was recorded for a period of 10 min under controlled testing conditions. In the research

reported in this chapter, the test condition with the main shaft speed of 22.09 rpm, the high speed shaft (HSS) speed of 1800 rpm, at 50% of the rated power is investigated. The data was collected over a period of 10 min.

The gearbox under study includes three stages – the low speed stage (LSST), the intermediate speed stage (ISST), and the high speed stage (HSST). The LSST interfaces the rotor, and the HSST is connected to the generator. A schematic description of the gearbox is shown in Figure 6.1. Twelve accelerometers were mounted on the gearbox, generator, and the main bearing to collect the vibration data (Figure 6.2). Table 6.1 provides the description of the sensors and their locations.

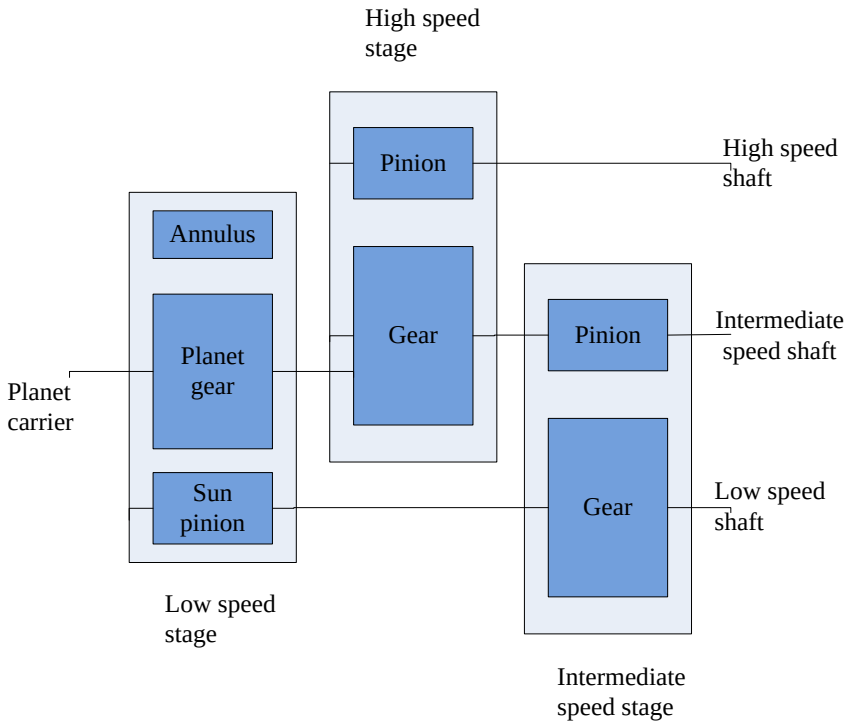


Figure 6.1. Schematic of gearbox used in current study (Courtesy of NREL).

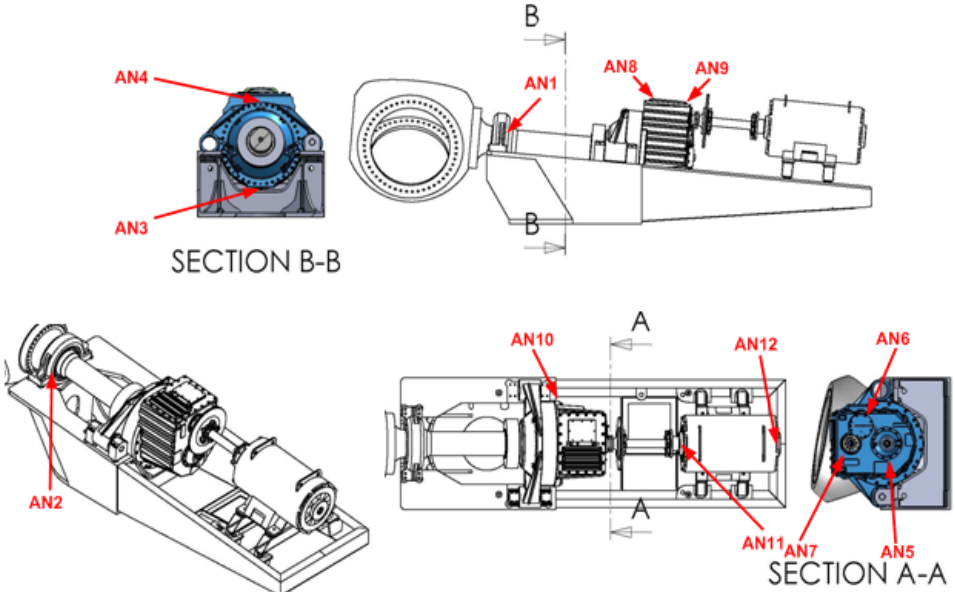


Figure 6.2. Sensor locations across the gearbox unit (Courtesy of NREL).

Table 6.1. Vibration sensors used in the study.

Sensor No.	Location	Unit
AN1	Main bearing radial	m/s^2
AN2	Main bearing axial	m/s^2
AN3	Ring gear radial 6 o'clock	m/s^2
AN4	Ring gear radial 12 o'clock	m/s^2
AN5	Low speed shaft radial	m/s^2
AN6	Intermediate speed shaft radial	m/s^2
AN7	High speed shaft radial	m/s^2
AN8	High speed shaft upwind bearing radial	m/s^2
AN9	High speed shaft downwind bearing radial	m/s^2
AN10	Carrier downwind radial	m/s^2
AN11	Generator upwind radial	m/s^2
AN12	Generator downwind axial	m/s^2
AN13	Encoder on high speed shaft	rpm
AN14	Strain gauges on low speed shaft	kNm

6.3 Damage identification

In this section, analysis of the vibration data is performed to identify faulty components. First, the jerk was calculated to determine the impact of vibration across various sensor

locations. '*Jerk*' is defined as the rate of change of acceleration and indicates the excitement of vibration. Mathematically, jerk is approximated as shown in Equation (6.1) (Zhang *et al*., 2012).

$$\vec{j} = \frac{d\vec{\alpha}}{dt} \approx \frac{\vec{\alpha}_t - \vec{\alpha}_{t-T}}{t-(t-T)} = \frac{\vec{\alpha}_t - \vec{\alpha}_{t-T}}{T} \tag{6.1}$$

In equation (6.1), $\vec{j}$ is the jerk $\vec{\alpha}_t$ is the acceleration at time index *t*, and T is the sampling interval (1/40000 sec in present case). Figure 6.3 illustrates the progression of the average jerk across 12 vibration sensors using 10 min of data. The initial 40 kHz data was averaged over 15 second intervals. The plot indicates that jerk monitored by the sensors AN3 and AN4 is significant relative to the other sensors. The sensors AN3 and AN4 are located in low speed shaft (LSS) and measure the vibration in the annulus/ring gear.

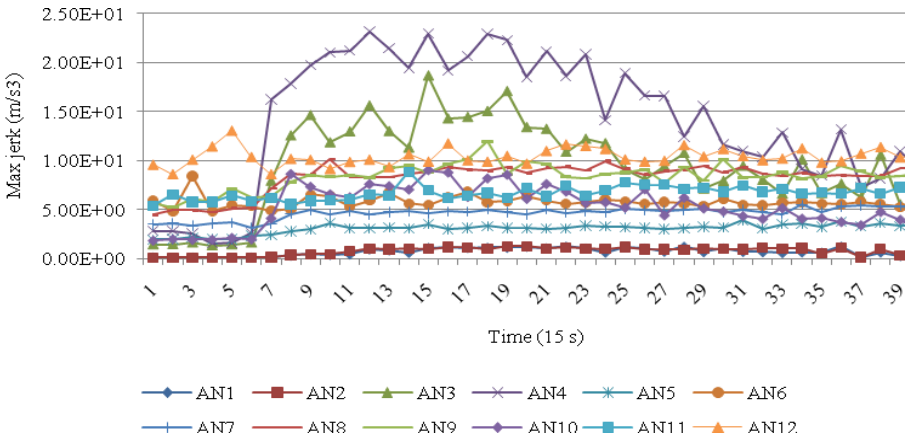


Figure 6.3. Maximum jerk across 12 vibration sensors.

The analysis demonstrated in Figure 6.3 does not provide any specific information; therefore, statistical analysis of the vibration waveforms is performed.

Specifically, metrics such as the root mean square (RMS), crest factor (CF), and kurtosis (Ku) are computed. The details of this analysis are provided next.

6.3.1 Time-domain statistical analysis

A run chart of vibrations values over time is a common way of analysis in the time domain; however, due to the large volume of the data, the fine details regarding the faults are often not visible. Thus, to overcome this issue, various statistical quantities are derived by averaging the high frequency data. Statistical quantities such as RMS, crest factor, clearance factor, impulse factor, shape factor, and kurtosis provide approximate information about faults and can be easily calculated (Patil *et al*., 2008). Crest factor and Kurtosis are less dependent on the vibration level, however, are sensitive to spikiness in the vibration signals, whereas, the impulse and shape factors are functions of redressed signal average. Details about these factors are provided next.

Root mean square is the simplest metric for measuring defects in the time domain. RMS value can be used to detect unbalanced rotating elements. It is a statistical measure of the magnitude with varying quantity as expressed in (6.2)

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i)^2} \tag{6.2}$$

Crest factor (CF) is a measure of changes detected in the signal pattern due to impulsive vibration sources, such as tooth breakage. It can be useful in detecting a few high peaks in the signal; the higher the magnitude of the peak and the fewer the number of peaks, the lower the RMS value will and the higher the peak value, denoting a higher crest factor. A crest factor with values between 2 and 6 represents normal operations, whereas a value higher than 6 represent defects in the component. The crest factor is calculated easily by dividing the peak level of the signal average by the standard deviation (RMS) of the signal average as expressed in (6.3).

$$CF = \frac{Peak\ level}{RMS} \tag{6.3}$$

Kurtosis is defined as the fourth statistical moment of an array of values about the mean. It indicates the existence of major peaks. Kurtosis values of less than 3 indicate that the component is in normal health condition, whereas any value greater than 3 represents abnormality. The more the peaks in the signal, the larger the kurtosis, as expressed in (6.4).

$$Ku = \frac{N.\sum_{i=1}^{N}(x_i - \bar{x})^4}{\left(\sum_{i=1}^{N}(x_i - \bar{x})^2\right)^2} \tag{6.4}$$

All of these metrics are applied to the data provided by sensors AN3 and AN4 (see Figure 6.2 and Table 6.1). Figure 6.4(a-c) illustrates values of the RMS, CF, and kurtosis value for sensor AN3 and AN4. In this test case, the RMS values of AN3 and AN4 are increasing over time, whereas the CF increases and then decreases. The higher crest factor is also congruent with the high kurtosis of sensor AN4. The crest factor and kurtosis begin to decrease in the later portion of the recorded data as the vibration pattern becomes more random due to increased damage level. To provide further insights in the time-domain, RMS, kurtosis, crest factor, and peak value are combined (Sassi *et al.*, 2006) as shown in (6.5) and (6.6).

$$COM_1 = \log\left[Ku + \left(\frac{RMS}{RMS_0}\right)\right] \tag{6.5}$$

$$COM_2 = \log\left[(Ku)^{CF} + \left(\frac{RMS}{RMS_0}\right)^{Peaklevel}\right] \tag{6.6}$$

In equations (6.5)-(6.6), RMS_0 is a constant representing the value corresponding to the healthy ring gear. In this research, the RMS value recorded during the startup of the

experiment (1st min) is denoted as RMS_0. Sassi *et al.*, (2006) tested the combined metrics to analyze the defects in bearings and found the metrics to be more insightful than conventional metrics. The combined metrics, COM1, and COM2, follow the same pattern and found higher value of AN4. Fig. 4 (d)-(e) illustrate the values of both metrics. A higher value at 5th min indicates that the damage in ring gear has begun and is becoming more severe as the time progresses.

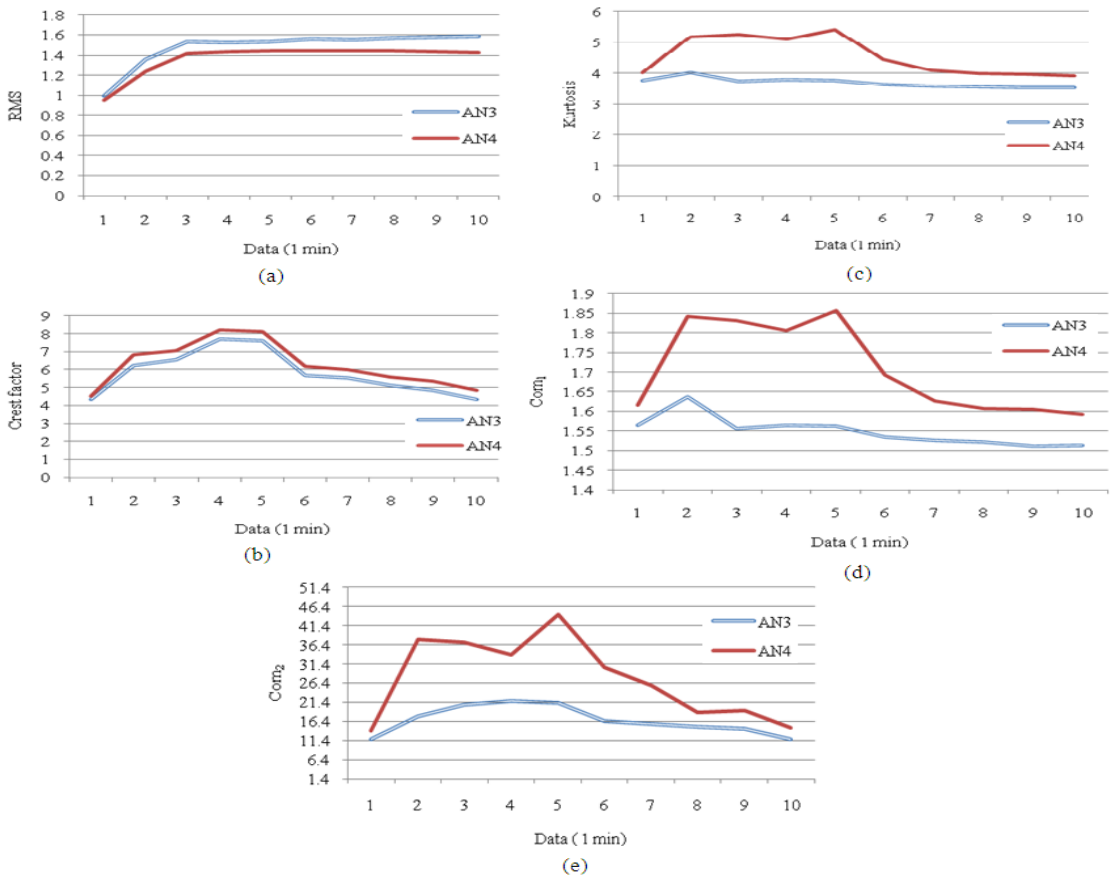


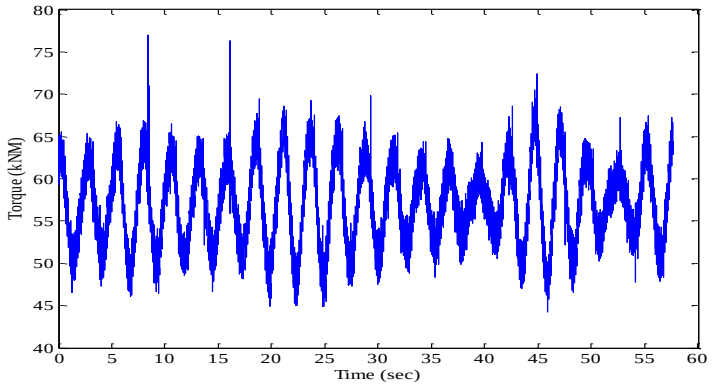
Figure 6. 4. Analysis of sensor AN3 and AN4 data: (a) root mean square, (b) crest factor, (c) kurtosis, (d) combined1 (COM1), (e) combined2 (COM2).

Since almost all metrics follow a similar trend for both sensors, sensor AN4 is analyzed in greater depth due to its high amplitude. The sensor (AN14) measures the torque at low speed shaft which could provide a good indication about the vibration impact in ring gear. Studies in the past have shown strong correlation among the torque and vibration (Kusiak and Zhang, 2009; Zhu *et al*., 2006).

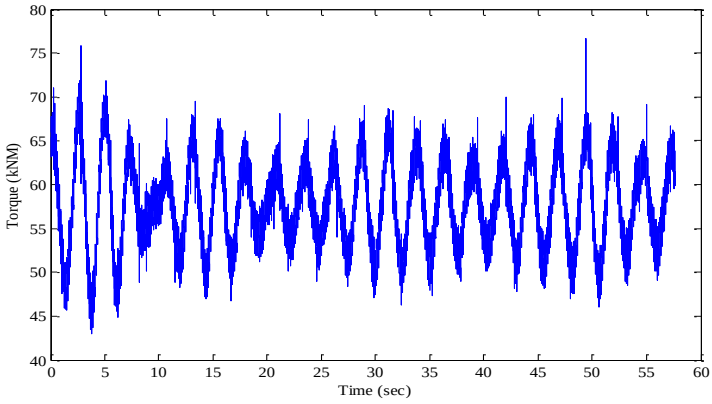
Figure 6.5 displays the torque at 1^{st}, 2^{nd}, 5^{th}, and 10^{th} min interval. The torque signal indicates one peak per revolution with frequency 0.37 Hz (i.e. time 2.72 second). The high spikes in the torque signal indicates fault in the annulus which interfaces with the shaft. The analysis on the torque signal indicates the emergence of fault in annulus gear. In next section, analysis in the frequency domain is done to determine the severity of the fault.

6.3.2. Frequency-domain analysis

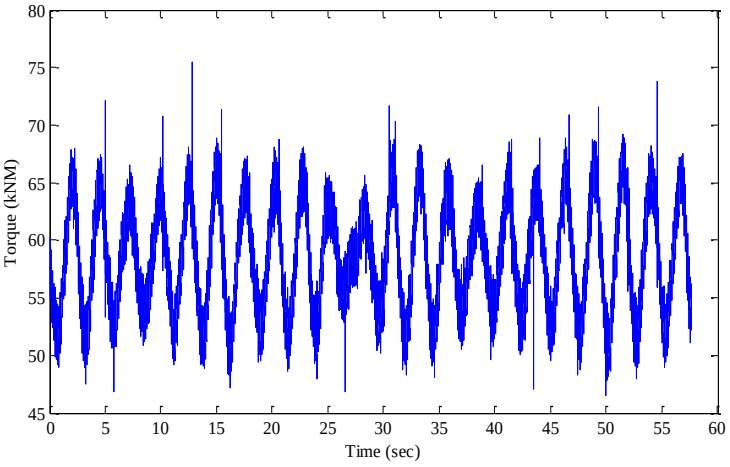
The data obtained from sensor AN4 is analyzed in the frequency domain. Specifically, Fast Fourier transformation (FFT) has been used to generate a frequency spectrum of the vibration. The frequency spectrum displays the portion of a signal's power (expressed in g) falling within a given frequency bin. The baseline spectra of healthy gearbox measured at high speed shaft is provided in Figure 6.6.



(a)



(b)



(c)

Figure 6.5. Analysis on torque signal: (a) 1st min, (b) 2nd min, and (c) 4th min.

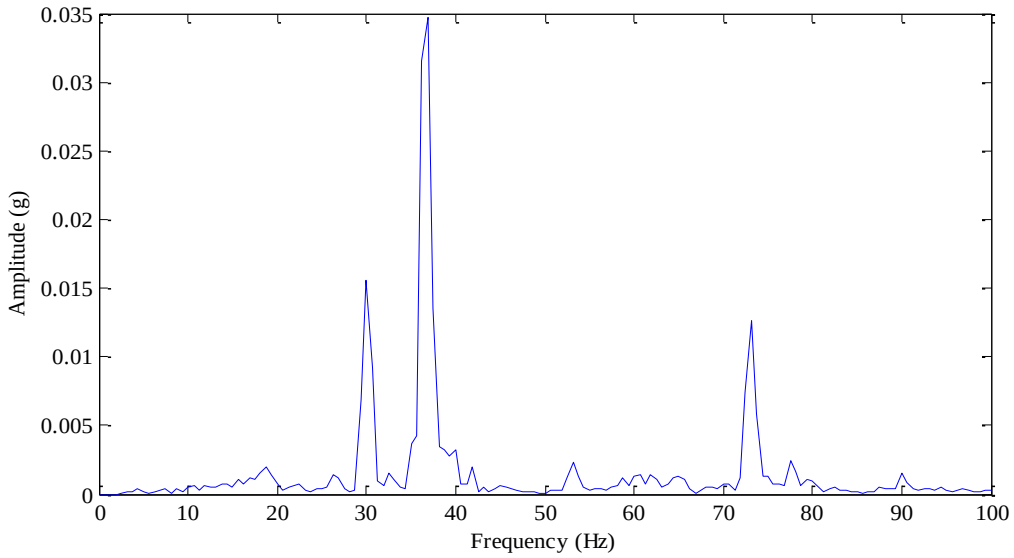
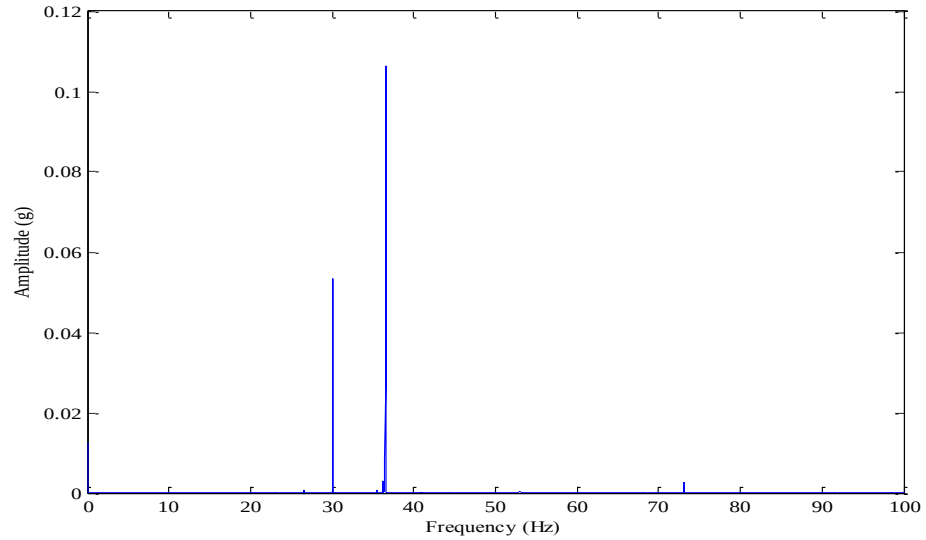


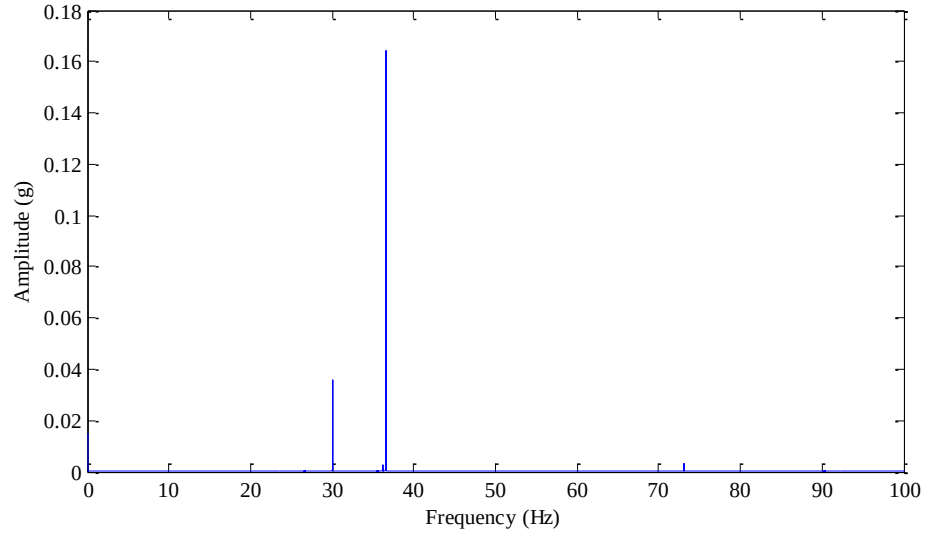
Figure 6.6. Baseline spectrum of healthy gearbox.

The Figure 6.6 indicates very low energy (<0.035 g) content at 36.45 Hz (gear mesh planet and ring). Figure 6.7 illustrates the frequency spectrum of high speed stage (HSS). With HSS at 1800 rpm, the gear mesh frequency at planet gear and annulus is 36.45 Hz. Since the HSS contains the spectra up to 20000 Hz, only low frequency component (i.e., 0-100 Hz) is displayed to better illustrate the signal power. Since, the initial data was available in the subset of 1 min, frequency spectrum at the 1st and last minute is provided in Figure 6.7 (a-b). Power spectrum is close to the healthy gearbox spectrum during the startup of the test run, however, in the last minute, the power spectrum jumps to 0.16 g showing mid-severe damage.

Figure 6.8 displays the trend of power spectrum for the overall test run, which is continuously increasing. This indicates the fault progression.



(a)



(b)

Figure 6.7. Power spectrum of vibrations across the ring gear: (a) 1st min, (b) 10th min.

The approaches discussed earlier clearly indicate faulty ring gear. Thus, model predicting the vibration across ring gear is important. In this chapter, sensor located near the ring gear (i.e. AN4) is considered as output, whereas, the vibration sensed by other sensors is considered as input.

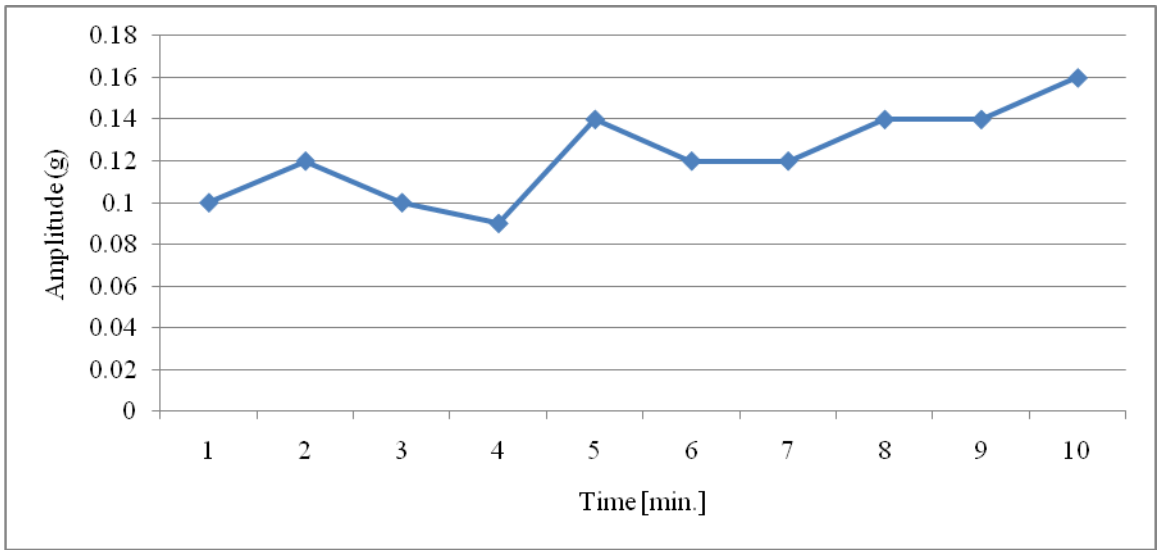


Figure 6.8. Trend of vibration amplitude across ring gear over the test run (10 min).

In the next section, models predicting jerk and acceleration in the ring gear (AN4) are discussed.

6.4. Models for predicting faults in ring gear

Based on the analysis in Section 6.3 and the available data, four prediction models are developed. Model 1 is called the one-parameter prediction model, in which the acceleration values are the target output, whereas the input parameter includes the historical acceleration values. In model 2, data from sensors other than AN4 are used as input to predict the acceleration of AN4. Models 3-4 are the same as models 1-2, except the jerk is the target output.

In addition, the statistical metrics (i.e., the mean, max, standard deviations, crest factor, kurtosis, RMS, clearance factor, impulse factor, and shape factor) are also included in the input parameter list to predict the target output. To reduce the dataset dimensionality, the initial high frequency data (i.e., 40 kHz) is converted into 10 Hz data (0.1 sec). The description of the dataset used in the study is provided in Table 6.2. For each model, first 80% data-points are used to build the model, which are then tested on

next 10%. Algorithm proving the best result on test dataset is used to perform prediction on the last 10% dataset.

Table 6.2. Description of the data scenario.

Scenario	Description	Target output
1	Single sensor model	Acceleration in ring gear (AN4)
2	Multiple sensor model	Acceleration in ring gear (AN4)
3	Single sensor model	Jerk in ring gear (AN4)
4	Multiple sensor model	Jerk in ring gear (AN4)

6.4.1. Parameter selection

In this section, different parameter selection algorithms are employed to identify the relevant parameters for predicting vibrations in ring gear. Three algorithms (i.e., the boosting tree (Kudo and Matsumoto, 2004), relief attributes (k=10) (Hinton and Sejnowski, 1999), and subset evaluator (Sikonja and Kononenko, 1997) are selected to perform the analysis.

The boosting tree algorithm generates parameter ranking based on the sum of the squared errors, which is computed at each split of input parameters (Kudo and Matsumoto, 2004). The average sum of square error is calculated for all splits. The parameter with the best split is assigned a value of 1, and so on. In the boosting tree algorithm, the relative influence of the parameters is calculated by using equation (6.7).

$$\tilde{J}_j^2(T) = \sum_{t=1}^{L-1} \tilde{I}_t^2 1(v_t = j) \tag{6.7}$$

In equation (6.7), $\tilde{J}_j^2(T)$ is the relative importance of parameter j, i is the index of the tree, v_t is the splitting feature associated with node t, and L is the number of terminal nodes in the tree. $\tilde{I}_t^2$ is the improvement of the squared error. The other two approaches, namely the relief and the subset evaluator, are filter-based approaches that belong to unsupervised learning (Hinton *et al*., 1999). Relief is an unsupervised learning algorithm

inspired by instance-based learning. Given training data, sample size, and a threshold, the relief algorithm assign a relevance weight to each parameter (Sikonja and Kononeko, 1997).

In the research reported in this chapter, the number of nearest neighbors is set at k = 10. Subset evaluator is another unsupervised learning approach that uses greedy step-wise learning to rank the input parameters (Hall, 1999). Parameters that are highly correlated with the output and less correlated with each other are selected. The same algorithms are used for analyzing all four models.

Tables 6.3-6.6 provide lists of the 10 best parameters obtained using the three parameter selection algorithms on four different scenarios. Algorithms, namely the relief and the subset evaluator, were employed using 10-fold cross-validation. Values of the average merit and standard deviations are reported in Tables 6.3-6.6. The results obtained from the parameter selection indicate that the actual vibration values are more important than the transformed statistical quantities discussed in the previous section.

The notation AANi is used for a neural network predicting acceleration, whereas JANi represents the NN predicting jerk, and index i represents the sensor location. In the case of single parameter models (i.e., scenario 1 and 3), the (t-k) represents the historical value of the target output, and k is the time index.

Table 6.3. Selected parameters for scenario 1.

Boosting tree		Relief (k=10)		Subset evaluator	
Parameter	Imp	Parameter	Average merit	Parameter	Average merit
AAN4(t-1)	1.00	AAN4 (t-1)	0.015	AAN4(t-1)	0.427
AAN4 (t-3)	0.96	AAN4(t-3)	0.005	AAN4(t-10)	0.509
AAN4 (t-2)	0.84	AAN4(t-2)	0.004	AAN4(t-3)	0.509
AAN4 (t-10)	0.81	MAX_AAN4(t-2)	0.004	AAN4(t-6)	0.488
AAN4 (t-9)	0.81	AAN4(t-5)	0.004	AAN4(t-4)	0.486
AAN4 (t-5)	0.78	MAX_AAN4(t-1)	0.003	AAN4(t-9)	0.485
AAN4 (t-6)	0.72	AAN4(t-6)	0.003	AAN4(t-5)	0.451
AAN4 (t-4)	0.66	MAX_AAN4(t-6)	0.003	AAN4(t-8)	0.423
MAX_AAN4(t-2)	0.38	AAN4(t-4)	0.003	AAN4(t-2)	0.391
Std_AAN4(t-1)	0.33	AAN4(t-10)	0.003	AAN4(t-7)	0.361

Table 6.4. Selected parameters for scenario 2.

Boosting tree		Relief (k=10)		Subset evaluator	
Parameter	Imp	Parameter	Average merit	Parameter	Average merit
AAN8	1.00	AAN8	0.022±0.001	AAN8	0.62±0.003
AAN3	0.95	AAN3	0.017±0.001	AAN3	0.71±0.003
AAN7	0.88	AAN7	0.009±0.000	AAN11	0.7±0.002
AAN10	0.68	AAN10	0.008±0.000	AAN1	0.68±0.002
AAN11	0.46	AAN9	0.005±0.000	AAN2	0.65±0.002
AAN9	0.38	Std_AAN3	0.004±0.000	AAN12	0.61±0.002
Std_AAN6	0.32	AAN5	0.003±0.000	AAN9	0.59±0.003
AAN1	0.27	AAN1	0.003±0.000	AAN7	0.56±0.005
AAN6	0.25	Std_AAN6	0.003±0.000	AAN6	0.55±0.006
AAN2	0.24	Max_AAN10	0.003±0.000	AAN5	0.52±0.004

Table 6.5. Selected parameters for scenario 3.

Boosting tree		Relief (k=10)		Subset evaluator	
Parameter	Imp	Parameter	Average merit	Parameter	Average merit
JAN4(t-1)	1	Max_JAN4(t-1)	0.028±0	JAN4(t-1)	0.846±0.002
Std_JAN4(t-1)	0.98	Max_JAN4(t-4)	0.026±0.001	JAN4(t-4)	0.888±0.002
Std_JAN4(t-4)	0.95	Max_JAN4(t-5)	0.023±0.001	Std_JAN4(t-1)	0.879±0.002
Max_JAN4(t-4)	0.95	Std_JAN4(t-1)	0.02±0.001	JAN4(t-10)	0.883±0.002
Max_JAN4(t-1)	0.92	JAN4(t-1)	0.018±0.000	RMS_JAN4(t-8)	0.879±0.003
JAN4(t-4)	0.9	Max_JAN4(t-3)	0.015±0.000	RMS_JAN4(t-1)	0.88±0.002
Std_JAN4(t-5)	0.85	Std_JAN4(t-4)	0.015±0.000	JAN4(t-5)	0.88±0.002
JAN4(t-7)	0.84	JAN4(t-4)	0.013±0.000	Std_JAN4(t-4)	0.88±0.002
Max_JAN4(t-5)	0.84	Std_JAN4(t-5)	0.013±0.000	RMS_JAN4(t-2)	0.876±0.004
Std_JAN4(t-7)	0.8	Max_JAN4(t-2)	0.012±0.000	JAN4(t-6)	0.878±0.003

Table 6.6. Selected parameters for scenario 4.

Boosting tree		Relief (k=10)		Subset evaluator	
Parameter	Importance	Parameter	Average merit	Parameter	Average merit
Std_JAN10	1	Std_JAN3	0.032±0.001	Mean_JAN3	0.945±0.001
Mean_JAN3	0.98	Max_JAN3	0.028±0.001	Std_JAN10	0.955±0.001
Std_JAN3	0.96	Mean_JAN3	0.021±0.001	Mean_JAN1	0.955±0.001
Mean_JAN10	0.91	Max_JAN10	0.019±0.000	Mean_JAN2	0.955±0.001
Max_JAN10	0.91	Std_JAN10	0.012±0.000	Std_JAN1	0.953±0.001
Max_JAN3	0.83	Mean_JAN10	0.007±0.000	Mean_JAN13	0.952±0.001
Std_JAN2	0.77	Max_JAN8	0.007±0.000	Std_JAN2	0.948±0.002
Std_JAN1	0.76	Mean_JAN8	0.006±0.000	Mean_JAN10	0.945±0.001
Max_JAN1	0.69	Max_JAN9	0.006±0.000	Std_JAN3	0.953±0.001
Std_JAN6	0.64	Std_JAN8	0.005±0.000	Std_JAN5	0.955±0.001
Mean_JAN1	0.63	Max_JAN5	0.005±0.000	Std_JAN14	0.954±0.002

Using the parameter sets generated by the parameter selection algorithms, the four scenarios are mathematically described in equations (6.8-6.11). The left-hand side of each equation is the target output, whereas the right-hand side represents the input parameters. The input parameters include the memory parameters of the target output and the parameter recorded at sensor locations other than AN4. The initial dimensionality of the data intended for scenarios 1–4 was 60. Using the parameter selection approaches, the dimensionality of scenarios 1, 2, 3 and 4 was reduced to 14, 16, 18 and 18, respectively.

$$AAN4(t) = f\begin{pmatrix} AAN4(t-1), AAN4(t-2), AAN4(t-3), AAN4(t-4), AAN4(t-5), AAN4(t-6), \\ AAN4(t-7), AAN4(t-8), AAN4(t-9), AAN4(t-10), MAX\_AAN4(t-1), \\ MAX\_AAN4(t-2), MAX\_AAN4(t-6), Std\_AAN4(t-1). \end{pmatrix} \quad (6.8)$$

$$AAN4(t) = f\begin{pmatrix} AAN1(t), AAN2(t), AAN3(t), AAN5(t), AAN6(t), AAN7(t), \\ AAN8(t), AAN9(t)AAN10(t), AAN11(t), AAN12(t), \\ MAX\_AAN10(t), Std\_AAN3(t), Std\_AAN6(t). \end{pmatrix} \quad (6.9)$$

$$JAN4(t) = f\begin{pmatrix} RMS\_JAN4(t-1), RMS\_JAN4(t-2), Std\_JAN4(t-1), Std\_JAN4(t-4), Std\_JAN4(t-5) \\ Std\_JAN4(t-7), Max\_JAN4(t-1), Max\_JAN4(t-2), Max\_JAN4(t-3), Max\_JAN4(t-4) \\ Max\_JAN4(t-5), JAN4(t-1), JAN4(t-4), JAN4(t-5), JAN4(t-6), JAN4(t-7), JAN4(t-10). \end{pmatrix} \quad (6.10)$$

$$JAN4(t) = f\begin{pmatrix} Max\_JAN1(t), Max\_JAN3(t), Max\_JAN5(t), Max\_JAN8(t), Max\_JAN9(t), Max\_JAN10(t) \\ JAN3(t), JAN8(t), JAN10(t), JAN13(t), Max\_JAN4(t-4), Std\_JAN1(t), Std\_JAN2(t) \\ Std\_JAN3(t), Std\_JAN5(t), Std\_JAN6(t), Std\_JAN8(t), Std\_JAN10(t), Std\_JAN14(t). \end{pmatrix} \quad (6.11)$$

The next section describes the data mining algorithms used to extract the four models.

6.4.2. Algorithm and scenario selection

The vibration data is highly non-linear; therefore, the traditional model-based approaches such as regression and Box-Jenkins are unsuitable for this study. In the literature, neural networks (NNs) are widely used to approximate the non-linear relationship between the

input data and output (Boydon, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970). NNs are self-adaptive as they learn from examples (training data instances) to capture the complex functional relationship among the data. Thus, NNs are used for the present analysis, as the functional relationship among various sensors is unknown. In the present study, a multi-layered perceptron (MLP) with different parameter settings is optimized to obtain the best network for the four models (see equations (6.8–6.11). In the research reported in this chapter, 50 different NNs are used for training, whereas the number of hidden units varied between 5 and 25. Activation functions, namely Tanh, exponential, identity, and logistic, are analyzed for both hidden and output node. Metrics such as the mean absolute error (MAE), and mean relative error (MRE) (see equations (6.12-6.13) to select the best NN).

$$MAE = \frac{\sum_{i=1}^{N} AE_i}{N} \tag{6.12}$$

Where,

$$AE = |\hat{y}(t) - y(t)|$$

$$MRE = \frac{\sum_{i=1}^{N} RE_i}{N} \tag{6.13}$$

Where,

$$RE = \left| \frac{\hat{y}(t) - y(t)}{y(t)} \right| \times 100$$

6.4.2.1. Selection of NN training algorithm

In this section, NN algorithms such as gradient descent (GD) (Moller, 1993), conjugate gradient (CG) (Zweiri *et al*., 2002), and radial basis function (RBF) (Yee and Haykin, 2001) are analyzed to select the best algorithm to train the neural networks. The acceleration data (scenario 1) is used to evaluate the performance of these NN training algorithms. Table 6.7 presents correlation coefficients between models derived by different NN training algorithms. The test results shown in Table 6.7 indicate that the NN-BFGS has a higher correlation with the actual output than the other NN algorithms.

When compared with algorithms such as GD, CG, and RBF neural networks, the BFGS neural network was found to perform better by providing 17.39%, 12.63%, and 17.7% improvement in MAE, respectively. Whereas, in terms of MRE, an improvement of 16.6%, 12.22%, and 17.00% is found. This justifies the use of BFGS as the NN algorithm for model construction.

Table 6.7. Correlation coefficients of the models constructed with various NN training algorithms.

	Average jerk	Actual	BFGS-NN	GD-NN	CG-NN	RBF-NN
Actual	0.420178	1	**0.753**	0.595	0.644	0.566
BFGS-NN	0.420079	**0.753**	1	0.788	0.852	0.744
GD-NN	0.418766	0.595	0.788	1	0.909	0.883
CG-NN	0.420428	0.644	0.852	0.909	1	0.844
RBF-NN	0.420178	0.566	0.744	0.883	0.844	1

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is used to train the networks as it has been found superior to other training algorithms, such as back propagation (BP) and radial basis function (RBF) (Boydon, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970). BFGS uses an approximate search scheme to improve the computational speed and to achieve global convergence. It involves the following four basic steps:

1. Setting the search direction;
2. Determining the step length along the search direction;
3. Updating the hessian matrix; and
4. Checking the convergence rate by some specified criteria.

Tables 6.8-6.11 contain the test results produced by the five best neural networks. The 100 neural networks are randomly selected by varying the number of neurons, hidden activation function, and output activation function. All 100 NNs were trained using

BFGS training method. The process is repeated 5 times and the best NNs for each run are selected for performance comparison on test dataset.

The description of the parameter settings of the NNs that generated the results in Tables 6.8-6.11 are displayed in Tables 6.12-6.15. The NN settings displayed in Tables 6.12-6.15 indicate that the MLP 14-5-1 with Tanh function in the hidden and output layers is most suitable for scenario 1. MLP 16-18-1 with Tanh as a hidden activation function and Logistic function as output activation is best for scenario 2. For the scenarios based on the jerk data, MLP 18-22-1 with Tanh and exponential as hidden and output functions, respectively, is best for scenario 3; whereas MLP 18-22-1 with exponential and identity functions as hidden and output activations is appropriate for scenario 4.

Table 6.8. Performance of data mining algorithms on test set (scenario 1).

No.	Average acceleration in ring gear (observed)	Average acceleration in ring gear (predicted)	Correlation coefficient	MAE	MRE (%)
NN1	0.01046	0.010402	0.6992	0.2731	52.75
NN2	0.01046	0.010403	0.7022	0.2712	53.58
NN3	0.01046	0.010402	0.7404	0.2551	48.81
NN4	**0.01046**	**0.010402**	**0.7530**	**0.2712**	**29.23**
NN5	0.01046	0.010419	0.775	0.2418	45.28

Table 6.9. Performance of data mining algorithms on test set (scenario 2).

No.	Average acceleration in ring gear (observed)	Average acceleration in ring gear (predicted)	Correlation coefficient	MAE	MRE (%)
NN1	**0.010437**	**0.010402**	**0.8328**	**0.2089**	**37.5**
NN2	0.010437	0.010403	0.8229	0.216	41
NN3	0.010437	0.010401	0.7754	0.2398	45
NN4	0.010437	0.010407	0.8081	0.2223	40.5
NN5	0.010437	0.010405	0.8186	0.2161	39

Table 6.10. Performance of data mining algorithms on test set (scenario 3).

No.	Average jerk in ring gear (observed)	Average jerk in ring gear (predicted)	Correlation coefficient	MAE	MRE (%)
NN1	0.394768	0.394765	0.9532	0.0186	4.46
NN2	0.394768	0.394766	0.9547	0.0187	4.48
NN3	0.394768	0.394862	0.9145	0.0253	6.09
NN4	0.394768	0.394862	0.9146	0.0253	6.09
NN5	**0.394768**	**0.394705**	**0.9725**	**0.0151**	**3.71**

Table 6.11. Performance of data mining algorithms on test set (scenario 4).

No.	Average jerk in ring gear (observed)	Average jerk in ring gear (predicted)	Correlation coefficient	MAE	MRE (%)
NN1	**0.394216**	**0.394214**	**0.9819**	**0.0126**	**3.12**
NN2	0.394216	0.394175	0.9628	0.0179	4.51
NN3	0.394216	0.39419	0.9823	0.0126	3.14
NN4	0.394216	0.394175	0.9628	0.0179	4.51
NN5	0.394216	0.394216	0.9623	0.0178	4.48

Table 6.12. Parameter settings of neural networks (scenario 1).

No.	Network code	Hidden activation	Output activation
NN1	MLP 14-5-1	Exponential	Tanh
NN2	MLP 14-5-1	Logistic	Tanh
NN3	MLP 14-19-1	Exponential	Exponential
NN4	MLP 14-5-1	Tanh	Tanh
NN5	MLP 14-24-1	Logistic	Identity

Table 6.13. Parameter settings of neural networks (scenario 2).

No.	Network code	Hidden activation	Output activation
NN1	MLP 16-18-1	Tanh	Logistic
NN2	MLP 16-15-1	Logistic	Tanh
NN3	MLP 16-5-1	Exponential	Exponential
NN4	MLP 16-14-1	Exponential	Identity
NN5	MLP 16-18-1	Exponential	Exponential

Table 6.14. Parameter settings of neural networks (scenario 3).

No.	Network code	Hidden activation	Output activation
NN1	MLP18-11-1	Exponential	Identity
NN2	MLP 18-8-1	Logistic	Exponential
NN3	MLP 18-21-1	Identity	Logistic
NN4	MLP 18-16-1	Identity	Logistic
NN5	MLP 18-22-1	Tanh	Exponential

Table 6.15. Parameter settings of neural networks (scenario 4).

No.	Network code	Hidden activation	Output activation
NN1	MLP 18-22-1	Exponential	Identity
NN2	MLP 18-17-1	Identity	Tanh
NN3	MLP 18-12-1	Tanh	Exponential
NN4	MLP 18-12-1	Identity	Tanh
NN5	MLP 18-20-1	Identity	Identity

Figures 6.9(a-d) presents the run chart of the observed and predicted values for models 1-4 obtained by the best NNs (the first 100 data points). It can also be observed that the NN in jerk models yields better results than the models based on the acceleration data. This indicates that jerk is more suitable for constructing prediction models at various time stamps. Among scenarios considering jerk as an output, scenario 4 yields a better result than scenario 3, indicating the importance of other sensors in predicting vibrations in ring gear.

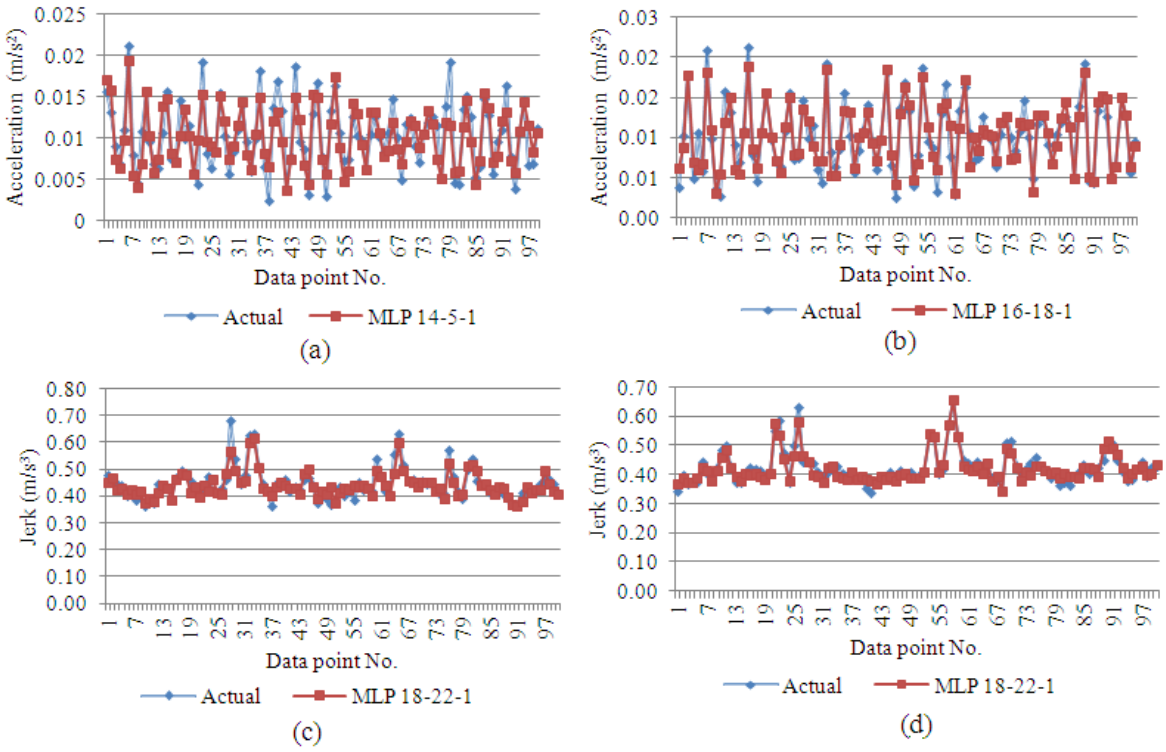


Figure 6.9. Run chart of the test results obtained using NN based models: (a) scenario 1, (b) scenario 2, (c) scenario 3, and (d) scenario 4.

In next section, prediction results of scenario 4 are presented.

6.5. Results of experiments

In this section, results produced by the neural network (NN) algorithm are discussed. The best performing NN model (MLP-18-22-1) is used to perform predictions. The model uses data from sensor locations AN1-AN3 and AN5-AN12 to predict the jerk at ring gear (measured by sensor AN4). Since, the data used in this study was recorded for short durations (e.g., 10 min); jerk is predicted for 15 time stamps with the data sampled at 0.1 s intervals. The results obtained are shown in Table 6.16. The mean absolute error (MAE) ranges from 0.021 to 0.345, whereas the mean relative error (MRE) ranges from 5.21% to 8.32%.

Table 6.16. Prediction results at 15 time stamps.

Time stamp [0.1 s]	Average jerk (actual)	Average jerk (predicted)	MAE	MRE (%)
t+0	0.4202	0.4286	0.021	5.213
t+1	0.4202	0.4288	0.026	6.307
t+2	0.4202	0.4288	0.028	6.676
t+3	0.4202	0.4288	0.029	6.991
t+4	0.4202	0.4288	0.030	7.220
t+5	0.4202	0.4287	0.030	7.320
t+6	0.4202	0.4286	0.031	7.461
t+7	0.4202	0.4286	0.031	7.480
t+8	0.4202	0.4286	0.032	7.635
t+9	0.4202	0.4288	0.032	7.715
t+10	0.4202	0.4288	0.033	7.931
t+11	0.4202	0.4287	0.033	7.995
t+12	0.4202	0.4286	0.034	8.208
t+13	0.4202	0.4286	0.034	8.208
t+14	0.4202	0.4286	0.034	8.283
t+15	0.4202	0.4286	0.035	8.326

Figure 6.10 displays the values of MAE and MRE for 15 time stamps. The values of MAE and MRE increase over time. The results shown in Figure 6.10 indicate that the proposed approach accurately predicts higher values of the jerk that contribute the most to the component failure.

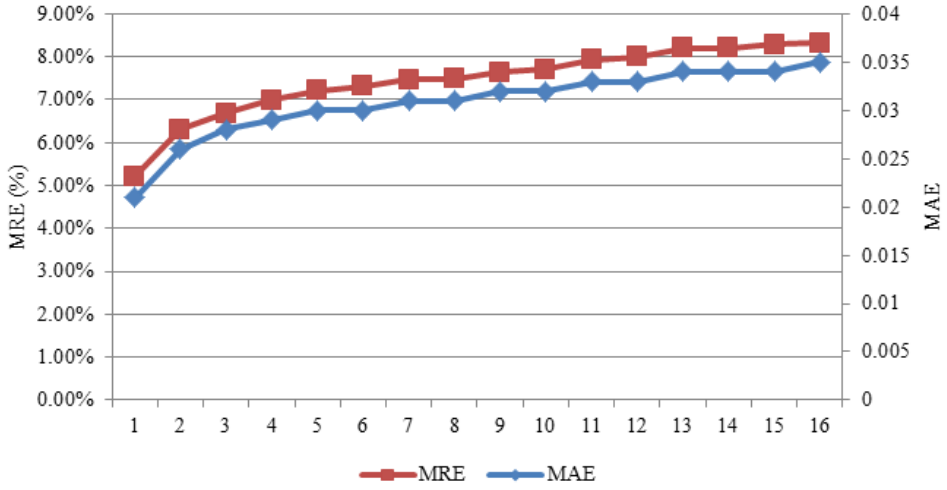


Figure 6.10. The values of MAE and MRE for different time stamps.

6.6. Summary

A methodology for identifying and predicting faults in the ring gear of a wind turbine gearbox was developed. Analysis in the time and frequency domain was performed to identify fault patterns in the ring gear. Jerk and acceleration data were used to generate models with data mining algorithms. Four different scenarios based on data from a single sensor (AN4) and multiple sensors (AN1-AN3 and AN5-AN12) were analyzed. Single sensor models assumed that only one sensor was installed, whereas multiple sensor models predicted vibration at selected drive train location.

The data mining models developed with jerk data (scenario 3 and 4) provided better accuracy than the models generated based on the acceleration data (scenario 1 and 2). A model developed on multiple sensor data (scenario 4) was used for jerk prediction. A neural network using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) learning method outperformed algorithms such as conjugate gradient (CG), gradient descent (GD), and radial basis function (RBF). The Broyden-Fletcher-Goldfarb-Shanno neural network (BFGS-NN) models accurately predicted jerk in a ring gear at different time intervals. The mean absolute error (MAE) was in the range of 0.021-0.345 m/s^3 and mean relative error (MRE) in the range of 5.21%-8.32%. The analysis presented in this chapter shows the importance of data-mining algorithm in vibration analysis.

CHAPTER 7
MONITORING WIND FARM USING TURBINE PERFORMANCE CURVES

7.1. Introduction

SCADA records hundreds of wind turbine parameters at high frequency (10 min or 10 sec). Analyzing high frequency data can be computationally expensive in order to monitor the progress of overall wind farm.

An efficient way is to measure the impact of internal factors is through turbine operations. The operational characteristics of turbines depend on parameters such as rotor power, torque, and pitch angle. Continuous monitoring of these parameters can be useful in assessing wind turbine performance. In the literature, power curve has been extensively used to monitor the progress of wind turbines. Kusiak *et al*., (2009) developed an advanced control chart approach to monitor the progress of wind turbines. In their research, they compared parametric power curve model with non-parametric data driven models. In another research, Yan *et al*., (2010) identified several stages of wind turbine power curves for monitoring purposes. They used inverse data transformation for change detection in turbine performance. Caselitz and Giebhardt (2005) developed a heuristic to identify the alarm limits in power curve for rotor condition monitoring. Harman, and Raftery, (2003) suggested the use of performance indicator namely availability, windiness, long-term wind speed and power performance in order to evaluate the overall wind farm performance.

In this chapter, the distinctive shape of wind turbine performance curves is exploited to perform wind farm monitoring. Three performance curves namely power curve, rotor curve, and blade pitch curve are investigated. To perform such assessment, operational wind turbine data is needed. Supervisory control and data acquisition (SCADA) systems record wind turbine parameters at different time intervals. SCADA data may be effectively used to tune a wind farm and provide early warning of possible failures. In the research reported in this chapter, the historical wind turbine data is used to

extract reference curves. The high frequency data is transformed into a single value using bivariate kurtosis and skewness metric.

7.2 Data for turbine performance curves

The data used in this chapter has been collected from a wind farm of over 100 wind turbines. Three wind turbine performance curves: power curve (power vs. wind speed), rotor curve (rotor speed vs. wind speed), and blade pitch curve (blade pitch angle vs. wind speed) are constructed for wind turbine performance. The dataset analyzed in this chapter is divided into three parts. First, a four year historical data (August 2005 – August 2008) from a 22 turbines is available to extract the reference curves. The data is averaged over 10 min intervals (10-min data). The reference curves are validated with the following year (August 2009) data. To perform continuous performance monitoring, one month long data from 22 wind turbines collected in August 2011 is used. Table 7.1 presents the data used in the research.

Table 7.1. Task description and related data period.

Task	Data Period
Extracting reference curves	August 2005-08
Validating reference curves	August 2009
Monitoring wind turbines	August 2011

Description of the three turbine performance curves is provided in the next section.

7.2.1. Description of turbine performance curves

A power curve indicates power generated by a wind turbine at various wind speeds. Malfunctions of a wind turbine will impacts its power generation capability. A typical wind power curve resembles to a sigmoid function, however due to various malfunctions, e.g., sensors and components, the power curve acquires its own shape.

Rotor curve represents a mapping between rotor speed and wind speed. Failures of turbine components impact its shape. A typical rotor curve is a monotonically increasing function of the wind speed.

Blade pitch curve shows the relationship between the turbine pitch angle and wind speed. Turbine's control system adjusts the blade angle for maximum power capture. A malfunction of the control system and high wind speed causes a turbine to stall, i.e., blade pitch angle becomes 90^{O}. During normal operations of a wind turbine, the pitch angle is set to, e.g., 0^{O}, 66^{O}, and 83^{O}. In general, during startup of a wind turbine, the blade pitch is set to a high value. A negative value of the pitch angle reflects the presence of a strong wind.

In the cut-in to cut-out region of the wind speed, the blade pitch settings are adjusted by the control system for the maximum power output. At the rated wind speed, the blade pitch angle is continuously adjusted to maintain the power required.
In next section, the proposed solution methodology is discussed.

7.3. Solution methodology

The proposed solution methodology includes four phases (see Figure 7.1). The historical wind farm data from several wind turbines is scanned initially to select wind turbine data (phase 1). Due to stochastic nature of the wind and inherent variability in the individual turbines, the noisy raw data is processed using multivariate outlier detection approach (phase 2). The resulting reference curves are used as a benchmark to evaluate performance of individual turbines (phase 3). Skewness and kurtosis of performance curves are calculated for each wind turbine and compared against the corresponding reference curves. In phase 4 a quality control chart is used for continuous monitoring of wind turbines.

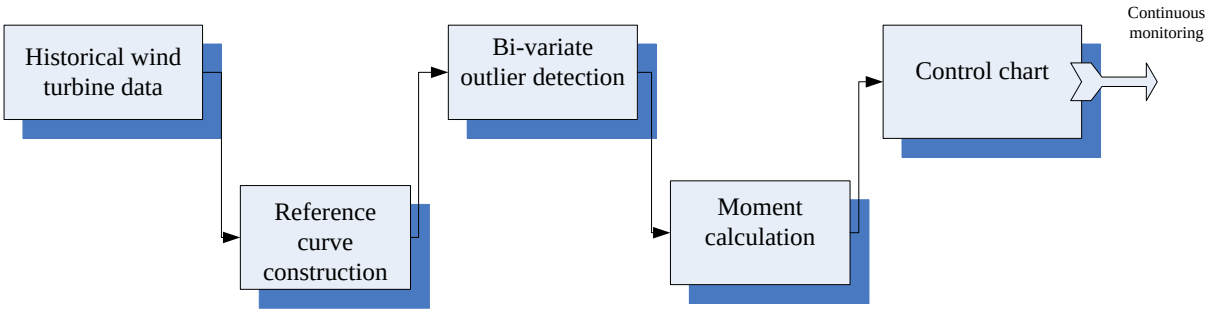


Figure 7.1. Proposed solution methodology.

All solution phases are discussed in the subsequent sections.

7.3.1. Reference curve construction

The performance of wind turbines depends on the wind speed. The data used in this research is obtained from a large wind farm, located in Blairsburg, Iowa. The monthly average distribution of wind speed at this location is investigated. Figure 7.2 provides the monthly distribution of wind speed. The average wind speed varies across different months. Constructing reference curves using the yearly performance data may not be ideal. Based on the distribution of the wind speed, the reference curves for individual months are constructed.

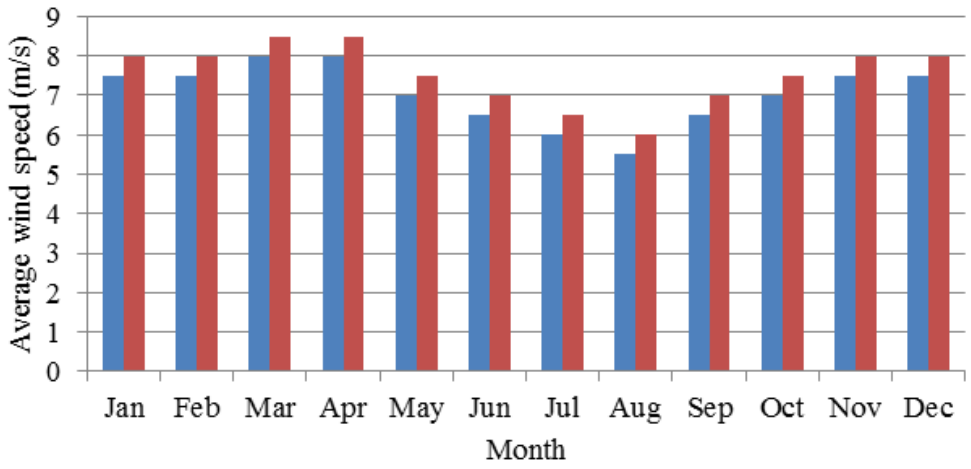


Figure 7.2. Average monthly wind speed distribution near the wind farm location (source: Iowa Energy Center).

Based on the completeness of data across different years, performance curves for the month of August are extracted from the turbine data collected on 22 turbines over a four year period, i.e., August 2005-August 2008. Based on the analysis, data from a single turbine is used. Figures 7.3(a)-(c) provide the references power curve, rotor curve, and blade pitch curve of the four year data.

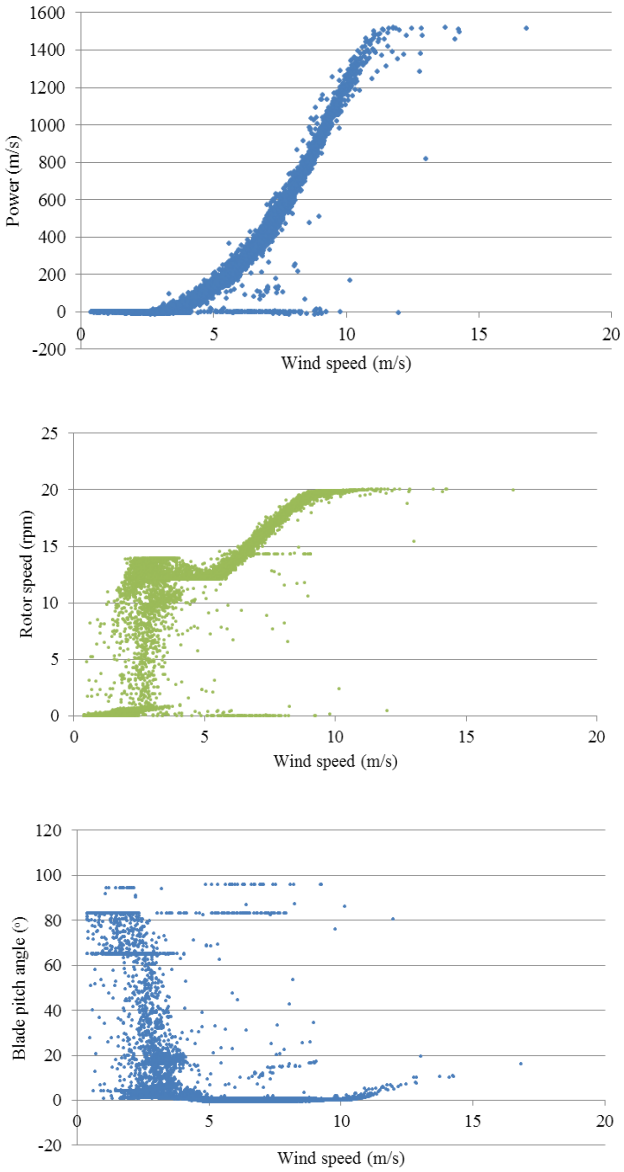


Figure 7.3. Performance curves for the month of August (August 2005-August 2008).

In next section, a bivariate outlier selection approach is discussed.

7.3.2. Bivariate outlier detection

The reference curves constructed from the historical data contain outliers which need to be removed for clear depiction of normal turbine behavior. These outliers are largely due to the sensor errors and fluctuations in the turbine performance. In this chapter, a multivariate outlier detection approach based on Mahalanobis distance is used. The Mahalanobis metric expresses the distance of an instance to the centroid in the multidimensional space (Mahalanobis, 1936) and it is calculated based on the correlation-covariance matrix. Therefore, Mahalanobis distance indicates whether an instance is an outlier with respect to the independent variable values (see Equation (7.1)).

$$D_{st}^2 = (x_s - x_t)Cov^{-1}(x_s - x_t)^{-1}, s \neq t \tag{7.1}$$

In equation (1), D_{st}^2 is the Mahalanobis distance between instance *s*, and *t*, and Cov^{-1} is the inverse of covariance matrix.

Due to distinct shape of performance curves, calculating Mahalanobis distance for an overall curve can be misleading as the centroid (usually for the wind speed between 4.5 m/s and 7 m/s) will consider the extreme data points (points close to cut-in wind speed, or/and near rated wind speed) as outliers, which in fact they are not. Thus, in order to improve the outlier detection, the performance curve data is grouped into smaller clusters. The *k*-means clustering algorithm determines the number of clusters for each curve minimizing the cost function in Equation (7.2) (Inaba *et al*., 1994).

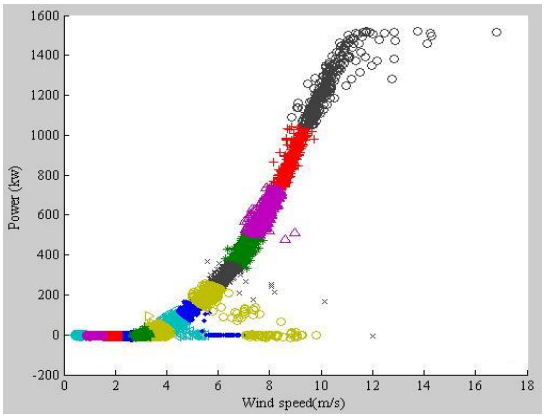
$$\Delta(k, x, c) = \frac{\sum_{i=1}^{k}\left(\sum_{x \in C_i} \|x - C_i\|^2\right)}{\sum_{i=1}^{k} Obs_i} \tag{7.2}$$

where Δ is the clustering cost, k is the number of clusters, Obs_i is the number of data points in cluster i, x represents the data points, and C_i represents cluster i. The proposed procedure for identifying outliers in the bivariate performance curves is presented next.

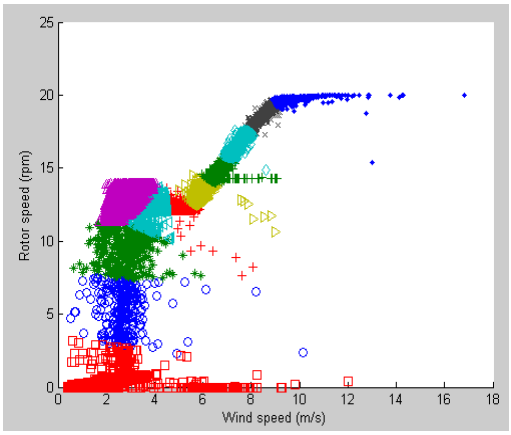
Procedure: Extracting smooth performance curves
Parameters: *Perf_C = {PC, RC, BPC}, optimal number of clusters (kopt), maximum number of data subsets (Max_fold), Mahalanobis distance threshold (δ)*
Begin
For each x € Perf_C
Set initial number of clusters $k = 2$
    Divide the dataset into Max_fold
    For f = 1:Max_fold
        Randomly select 90% subset for training and 10% for testing.
        Initialize k centroids
        Repeat, until the centroid does not change
        Evaluate the training error using the cost function in Eq. (2)
    End for
    Output: *k*opt (Optimal number of clusters)
    For j = 1:*k*opt
        Evaluate the Mahalanobis distance (Dpq) Eq. (1) for each data pair (p, q) from the cluster mean
        Sort the data pairs (p, q) based on the distance (Dpq)
        Retain data pairs (p, q) with Dpq <= δ
    End for
    Output: Mahalanobis distance between all data points for *k*opt clusters
Repeat
Do until x = |Perf_C|
End
Output: Smooth performance curves

The above procedure computes the Mahalanobis distances between the data points and the cluster centers (centroids). The *k*-means clustering algorithm applied to the monthly reference curves provides 14, 11, and 9 clusters for the power curve, the rotor curve, and the blade pitch curve, respectively. Figures 7.4(a)-(c) illustrate the clustered reference curves with training errors 0.039, 0.062, and 0.075 for the power curve, the rotor curve, and the blade pitch curve, respectively. Figures 7.5-7.7 depict the Mahalanobis distance of performance curves for individual clusters. The outlier data points can be easily identified in Figures 7.5-7.7. The aim here is to extract smooth reference curves, therefore a conservative approach based on the Mahalanobis distance

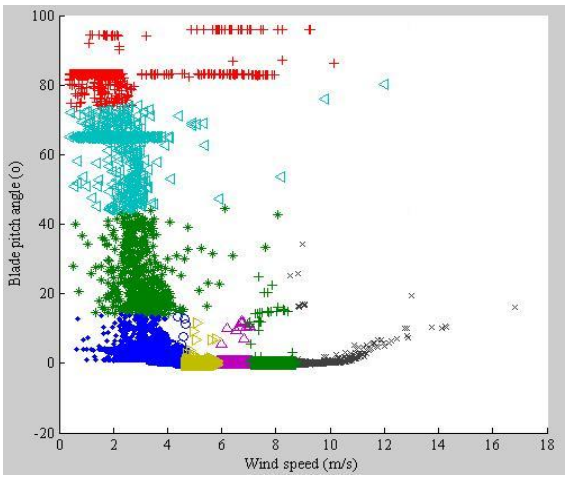
metric is used to remove the outlier data points. The threshold distance is chosen in a way that the data points corresponding to the high density clouds in the clusters are selected. Table 7.2 presents the threshold distance for each cluster. Using the threshold distance indicated in Table 7.2, 10%-15% of the data points were considered as outliers and thus are discarded. The refined performance curves are illustrated in Figures 7.8(a)-(c).



(a)

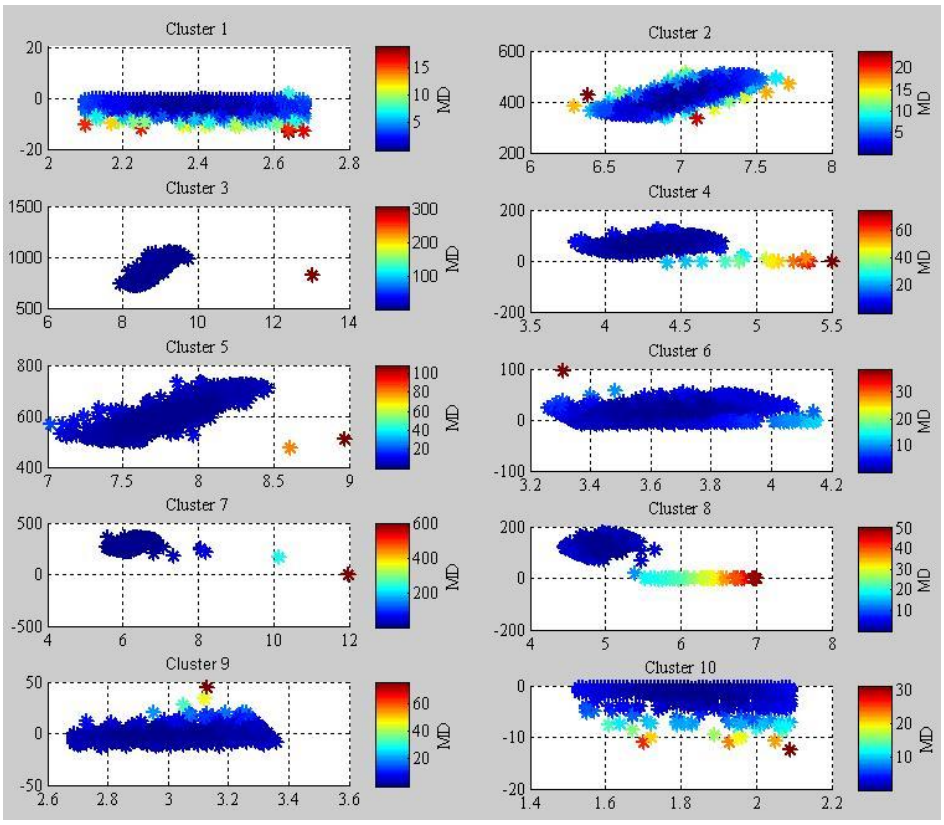


(b)

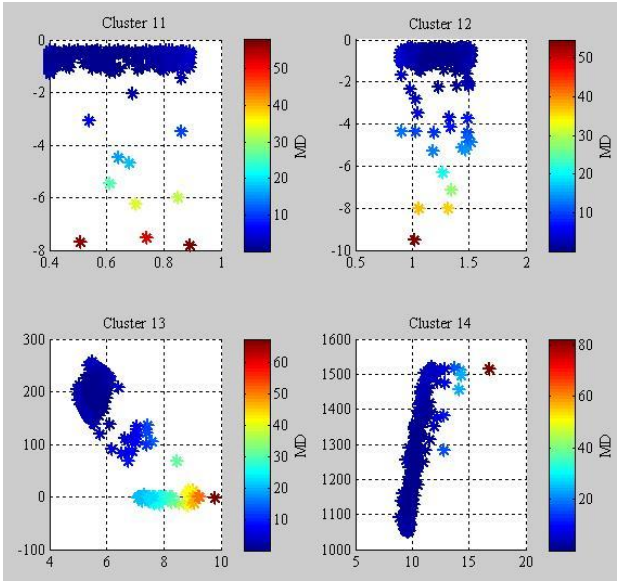


(c)

Figure 7.4. Performance curves with clusters: (a) power curve, (b) rotor curve, and (c) blade pitch curve.

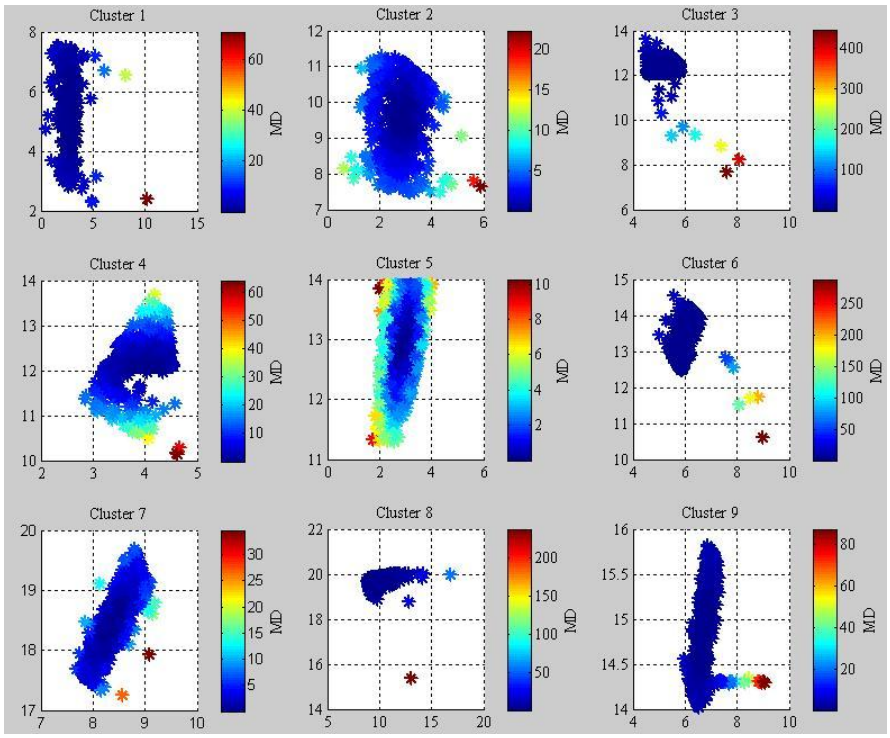


(a)

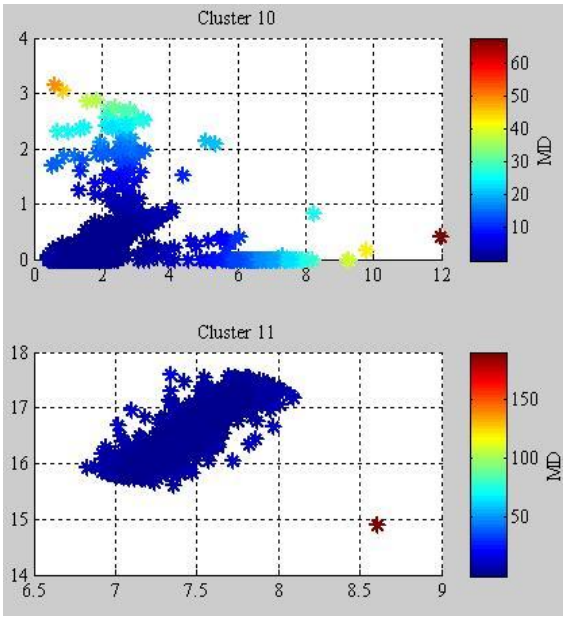


(b)

Figure 7.5. Mahalanobis distance (MD) of power curve based clusters: (a) Cluster 1-10, (b) Cluster 11-14.



(a)



(b)

Figure 7.6. Mahalanobis distance (MD) of rotor curve based clusters: (a) Cluster 1-9; (b) Cluster 10-11.

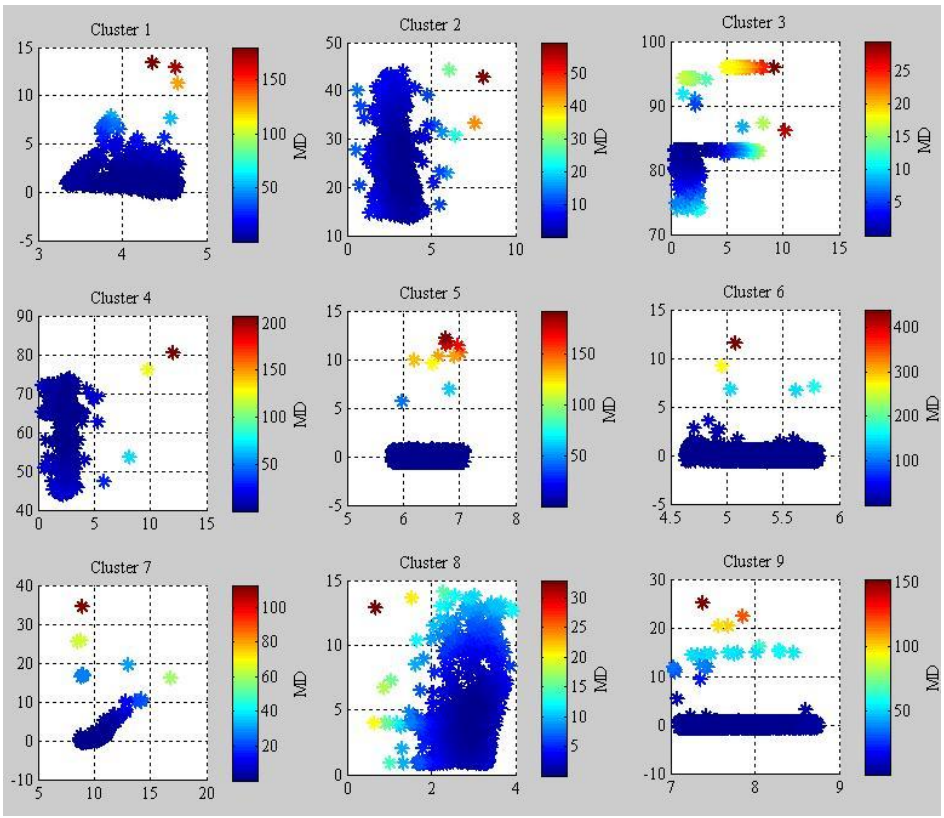
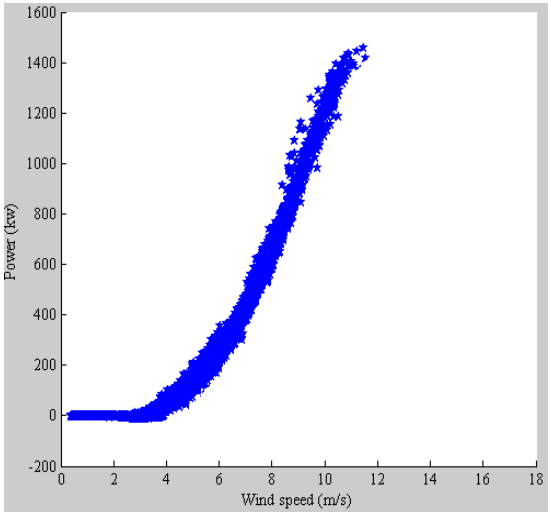


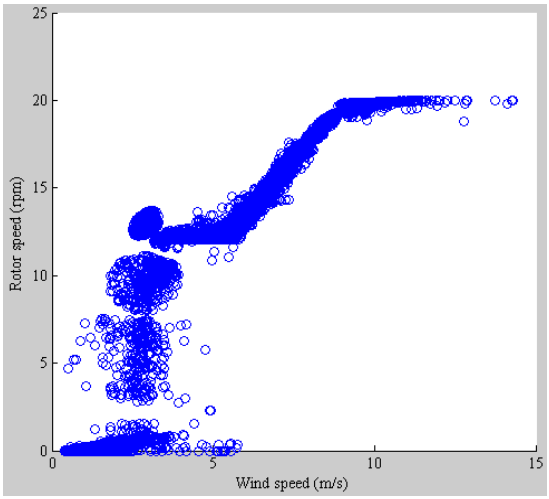
Figure 7.7. Mahalanobis distance (MD) of blade pitch curve based clusters.

Table 7.2. Mahalanobis distance threshold for performance curve clusters.

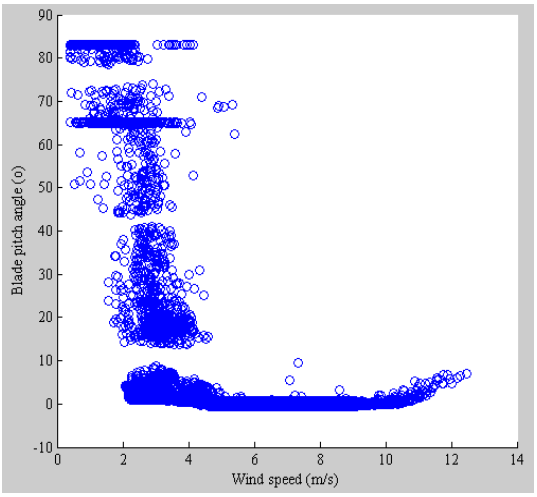
Cluster No.	Power Curve	Rotor Curve	Blade Pitch Curve
1	2.5	10	25
2	2.5	2.5	5.0
3	50	50	5.0
4	10	5	25
5	10	1	25
6	5.0	25	50
7	100	2.5	10
8	5.0	25	2.5
9	10	10	25
10	5.0	10	NA
11	5.0	25	NA
12	5.0	NA	NA
13	10	NA	NA
14	10	NA	NA



(a)



(b)



(c)

Figure 7.8. Refined performance curves: (a) power curve, (b) rotor curve, and (c) blade pitch curve.

In next section, the moment of performance curves is discussed. Scalar performance matrices, namely, skewness and kurtosis of bivariate data are evaluated.

7.3.3. Moment calculation

The third and fourth order moments namely kurtosis and skewness are often used to describe the shape of the data distribution. The multivariate kurtosis and skewness can also be used a data compression technique providing a single value describing the shape of the distribution of high frequency multivariate data. More literature pertaining to skewness and kurtosis is related with depicting normality in multivariate analysis. Few practical applications of kurtosis and skewness includes (1) identifying initial component in independent component analysis (ICA) (Kollo, 2008), (2) identification of dynamics in N-dimensional market. The refined reference curves obtained in the previous section are used as a benchmark for continuous performance monitoring of the wind farm. Multivariate skewness is a univariate measure of skewness for multivariate data, where a value close to zero indicates elliptical symmetry. The multivariate skewness is defined in Equation (7.3) (Mardia, 1970; Mardia, 1974).

$$Skew_{multi} = \frac{1}{n^2}\sum_i \sum_j \left((x_i - \bar{x})^T \cdot Cov^{-1} \cdot (x_j - \bar{x})\right)^3 \tag{7.3}$$

Where, $\bar{x}$ is the matrix mean, Cov-^{1} is the estimated population covariance matrix.

Similarly, multivariate kurtosis is a univariate measure of kurtosis for multivariate data. For *p* columns matrix, a value of kurtosis coefficient close to *p (p+2)* indicates approximate multinormality. Multivariate kurtosis is mathematically described in equation (7.4) (Mardia, 1974; Mardia, 1980).

$$Kurt_{multi} = \frac{1}{n^2}\sum_i \left((x_i - \bar{x})^T \cdot Cov^{-1} \cdot (x_i - \bar{x})\right)^2 \tag{7.4}$$

In general, underperforming wind turbines will deviate from the reference curves, resulting in different values of kurtosis and skewness that can be tracked in 2-*D* graph. Depending on the requirements, performance monitoring can take place on a daily, weekly, or monthly basis. Table 7.3 compares the kurtosis and skewness of yearly and monthly reference curves with respect to the test data (August 2009). The kurtosis and skewness of reference curves constructed based on the yearly data (January 2008-December 2008) is obtained in the similar way. The values presented in Table 7.3 indicate that the skewness and kurtosis of monthly reference curves are much closer

when compared with the yearly reference curves. Thus, monthly reference curves are used for monitoring the wind farm.

Table 7.3. Multivariate kurtosis and skewness of reference curves.

Data Type	Criteria	PC	RC	BPC
Yearly	Skewness	2.25	2.11	4.17
	Kurtosis	9.34	8.41	9.12
Monthly	Skewness	8.24	1.75	2.38
	Kurtosis	15.13	7.89	8.35
Test	Skewness	10.25	1.98	3.18
	Kurtosis	17.33	8.87	8.67

7.3.4. Monitoring wind farm

In this section, 22 wind turbines over a period of a month (August 2011) are analyzed using kurtosis and skewness for three performance curves. The analysis is based on the 10-min average data. Turbines located farthest from the reference points (see Table 7.3) are considered to be abnormal. Euclidean distance is used to evaluate the distance of individual wind turbines from the reference points. Figures 7.9-7.11 provides the 2-*D* scatter plot of the performance curves, where, each point (diamond) represents individual wind turbines. In addition, kurtosis and skewness of the reference curves are also included in the Figures 7.9-7.11. Depending on the distribution of data points across the performance curves, the kurtosis and skewness distribution varies. Due to the distinct shape of the power curves, the kurtosis and skewness values are higher and spread out, than those of the rotor and blade pitch curves.

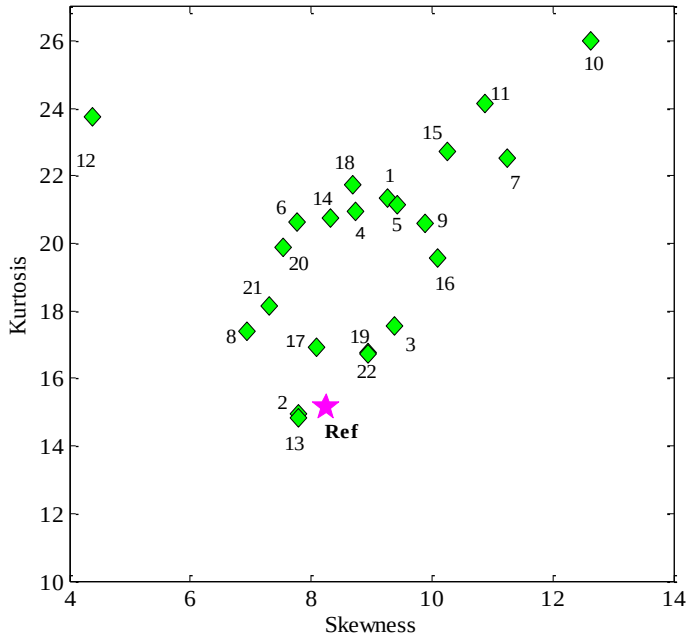


Figure 7.9. Status of a wind farm reflected by the power curve.

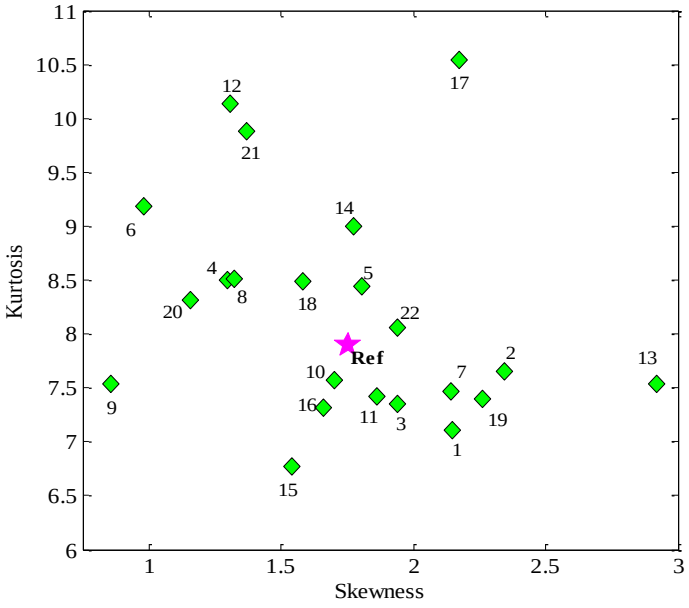


Figure 7.10. Status of a wind farm reflected by the rotor curve.

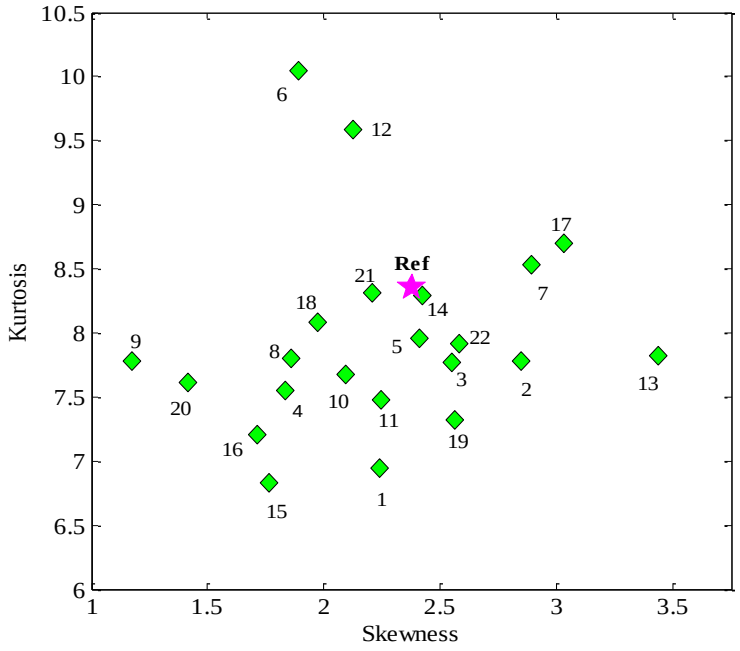


Figure 7.11. Status of a wind farm reflected by the blade pitch curve.

In 2-*D* skewness-kurtosis graph, wind turbines performance can be assessed by: (1) relative location of individual turbines with respect to the reference curves, and (2) location of individual turbines with respect to the turbine clusters. In general, turbine showing the same behavior will form a distinct cluster. Any abnormal turbine behavior can be easily visualized in a 2-*D* scatter graph. The possible reasons for distinct location of individual turbines in skewness-kurtosis plot could be: (1) underperformance due to system abnormalities, (2) underperformance due to different wind speeds, (3) over performance due to errors in wind speed measurement.

Using the guidelines mentioned earlier in this section, power curve based skewness-kurtosis graph identifies turbine 10, and turbine 12 as abnormal, whereas, turbine 13, turbine 17, and turbine 9 behave differently in the rotor and blade pitch curves. Figure 7.12 illustrates the power curve of turbine 10. The abnormal behavior of turbine 10 is clearly visible as the fault logs confirm the faults associated with generator windings. The fault log data recorded by the SCADA systems confirm the faults associated with generator windings. More information about turbine fault logs is provided in (Hyers, 2006; Kusiak and Verma, 2011).

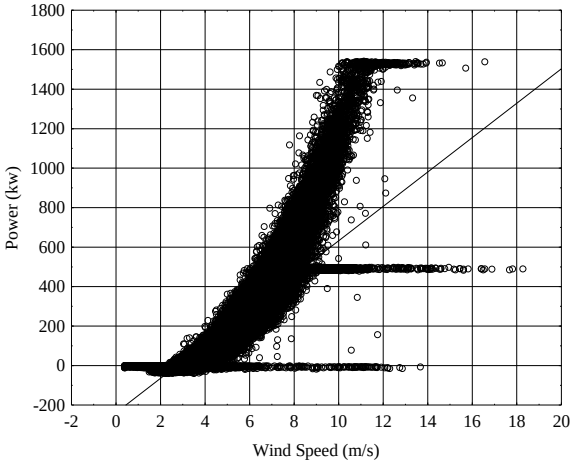


Figure 7.12. Power curve of turbine showing abnormal behavior (turbine 10).

Performance of wind turbines can be assessed with the 2-*D* kurtosis-skewness graph. However, the time is not depicted in the scatter plot. Therefore, to keep track of the time, control charts are utilized.

7.4. Continuous monitoring of wind turbines

In this section, continuous monitoring of wind turbines is performed using a quality control chart. The overall monitoring of the wind farm can be done on a weekly or monthly basis. However, for performance monitoring of wind farm in time, quality control charts are required. Two output metrics namely the skewness and the kurtosis are used. Monitoring the matrices independently can be misleading. Therefore, bivariate process monitoring using Hotelling's T^2 control chart is employed. In the literature, Hotelling's T^2 chart has been widely used to simultaneously monitor two or more output variables (Johnson and Vichern, 2002). Equations (7.5)-(7.7) define the T^2 statistic.

$$T^2 = \left(x - \bar{X}\right) S^{-1} \left(x - \bar{X}\right)' \tag{7.5}$$

In equation (7.5), x is the individual observation, $\bar{x}$ is the variables mean, and S is the covariance matrix inverse. Since, the subgroup size is 1, the covariance matrix is evaluated by pooling all observations (equation (7.6)) (Williams *et al.*, 2006; Sullivan and Woodall, 1995).

$$S = \frac{1}{m-1}\sum_{i=1}^{m}\left(x_i - \bar{X}\right)\left(x_i - \bar{X}\right)' \tag{7.6}$$

The lower control limit (LCL) is always 0, whereas, the upper control limit (UCL) is calculated from Equation (7.7).

$$UCL = \frac{p(m+1)(m-1)}{m^2 - mp} F_{\alpha,p,m-p} \tag{7.7}$$

In equation (7.7), p is the number of output variables, $F_{\alpha,p,m-p}$ is obtained from F distribution. The value of α is set to 0.001. The kurtosis and the skewness can be monitored simultaneously; however, the Hotelling T^2 test requires the data to be normal.

Therefore, the initial data is normalized using the Box-Cox approach. A value λ varies between -5.0 to 5.0. Figure 7.13 provides the comparison of initial and transformed skewness data of turbine 1 obtained using box-cox approach. For λ equal -0.552, the transformed data resembles to a normal distribution. This process is repeated for all turbines.

The transformed bivariate data of turbines is divided into two training and testing of control limits. Using the information presented in the power curve based kurtosis-skewness data (see Figure 7.9), turbine 7, turbine 10, turbine 10, turbine 11, turbine 12, and turbine 15 are used for testing, whereas, the control limits is obtained using the data from remaining turbines. Figure 7.14 provides the UCL of 10.507, resulting in 12 data points out of control. The out-of control data points are removed and the training process is iterated until all data-points meet the control limits. After three training process iterations, all data-points were found in control, with the resulting UCL of 10.505 (Figure 7.15).

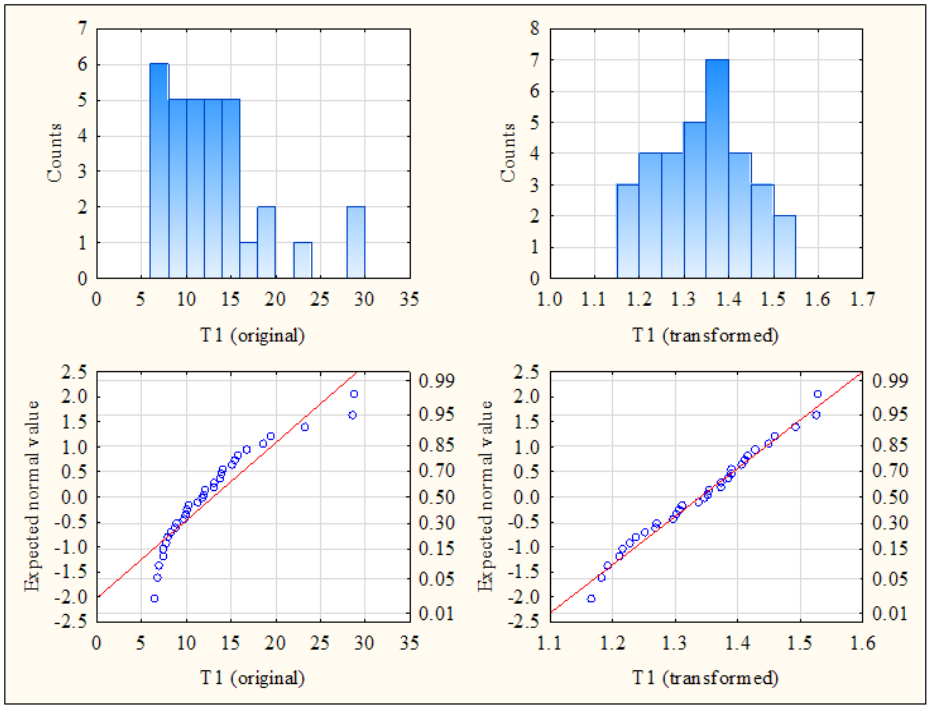


Figure 7.13. The Box-Cox transformation of turbine 1 data (λ = -0.552).

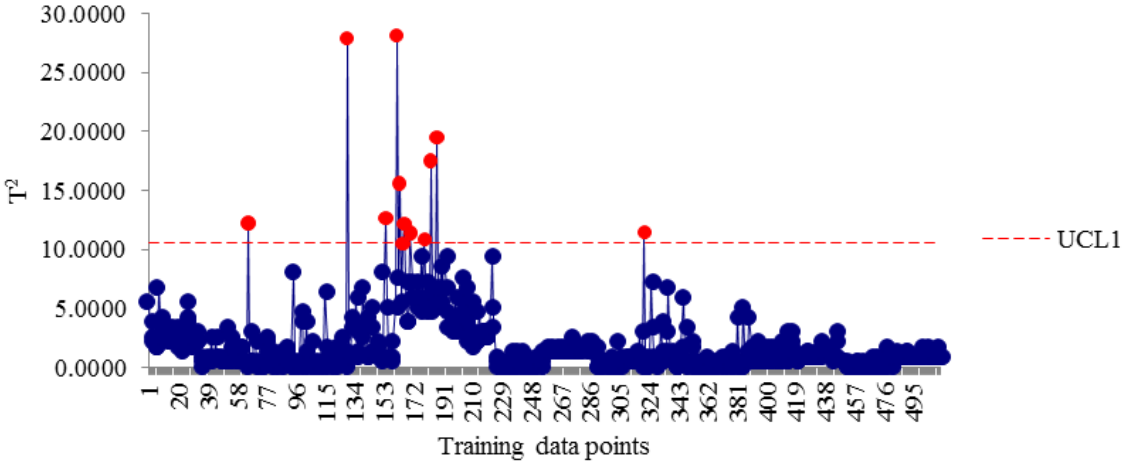


Figure 7.14. Control limits for training data points (iteration 1).

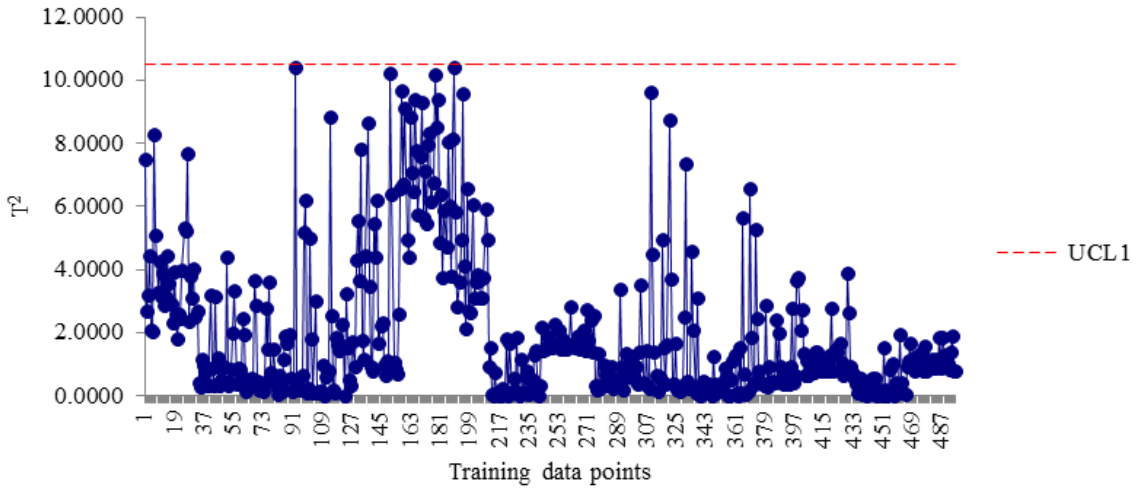


Figure 7.15. Control limits for training data points (iteration 3).

Figures 7.16(a)-(e) illustrate the test data corresponding to turbines turbine 7, turbine 10, turbine 11, turbine 12, and turbine 15. Using the obtained UCL values, turbine 7, turbine 10, turbine 11 point at least one abnormal day, whereas, turbine 12, and turbine 15 were found to operating normally. In general, faults namely cable twisting left, faulty pitch controller, blade angle implausibility fault were present in turbine 12. However, the main reason for abnormal days is power curtailment. The power curtailment causes generator to rotate half or their actual rpm as depicted in Figure 7.12. Due to a limited number of observations, no significant patterns in T^2 values have been observed.

7.4.1. Power curve monitoring: A case study

The power curve of 10 wind turbines over a period of four years (Jan 2006-Dec 2009) is analyzed. First, the monthly reference power curve is constructed from the historical wind turbine data. The best power curves obtained from 10 wind turbines over a period of 12 months are selected as candidates for reference curves. Additionally, the control limits corresponding to the reference power curves are established by applying Xbar and s chart.

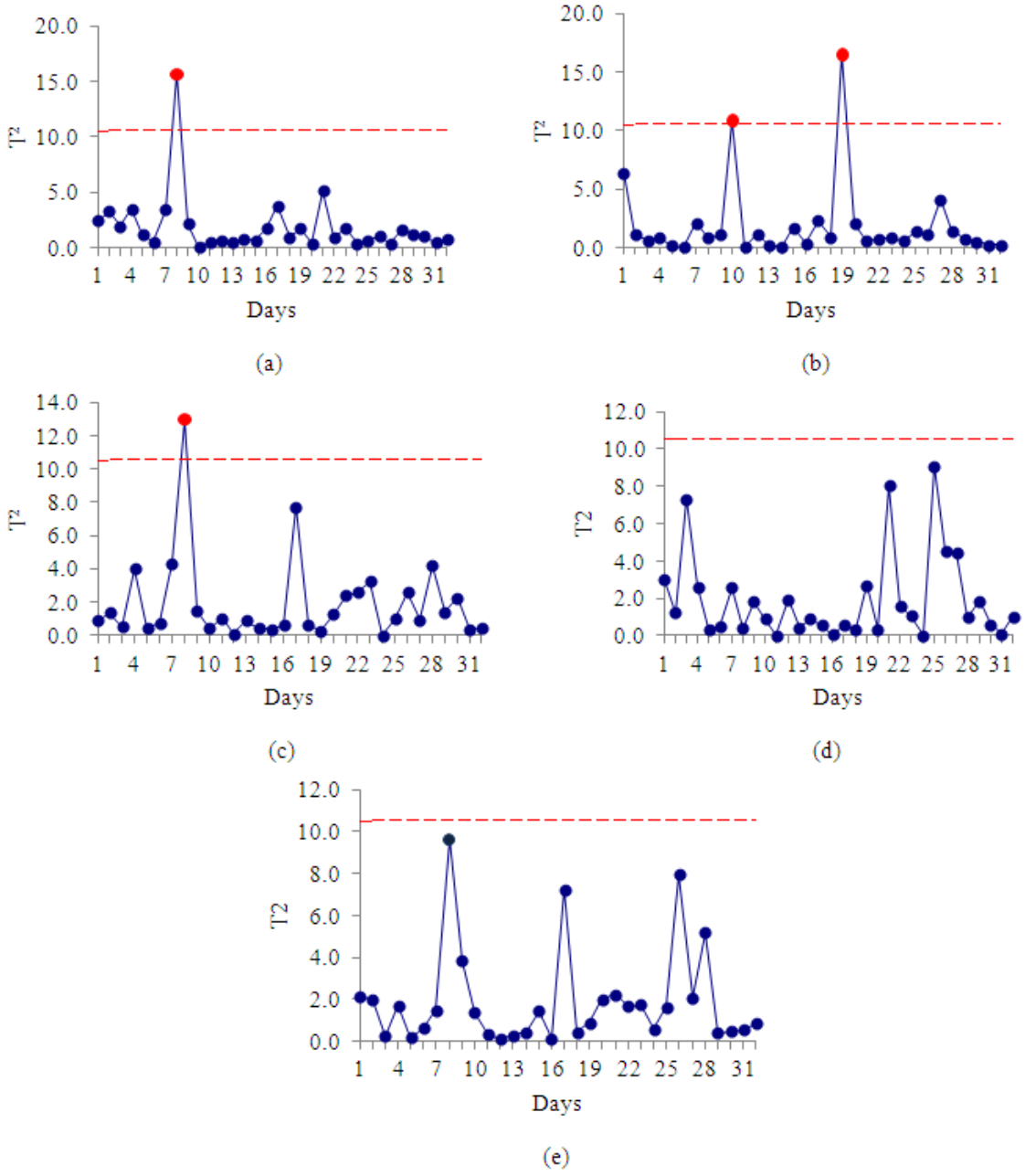


Figure 7.16. Hotelling's T^2 chart for the test data: (a) turbine 7, (b) turbine 10, (c) turbine 11, (d) turbine 12, and (e) turbine 15.

The mathematical description of the upper and lower control limit is given below (Equation 7.9-7.15)

$$\bar{x}_{skew} = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{n_m^2}\Sigma_i\Sigma_j\left((x_{im}-\bar{x}_m)^T.Cov_m{}^{-1}.(x_{jm}-\bar{x}_m)\right)^3 \quad (7.8)$$

$$s_{skew} = \sqrt{\sum_{i=1}^{M}\left(\frac{\left(\frac{1}{n_m^2}\sum_i\sum_j\left((x_{im}-\bar{x}_m)^T.Cov_m{}^{-1}.(x_{jm}-\bar{x}_m)\right)^3-\mu_{skew}\right)^2}{M-1}\right)} \quad (7.9)$$

$$UCL_{skew} = \bar{\bar{x}}_{skew} + c\times\bar{s}_{skew} \quad (7.10)$$

$$LCL_{skew} = Max\{0, \bar{\bar{x}}_{skew} - c\times\bar{s}_{skew}\} \quad (7.11)$$

$$\bar{x}_{kurt} = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{n_m^2}\sum_i\left((x_{im}-\bar{x}_m)^T.Cov_m^{-1}.(x_{im}-\bar{x}_m)\right)^2 \quad (7.12)$$

$$s_{kurt} = \sqrt{\sum_{i=1}^{M}\left(\frac{\left(\frac{1}{n_m^2}\left(\sum_i\left((x_{im}-\bar{x}_m)^T.Cov_m{}^{-1}.(x_{im}-\bar{x}_m)\right)^2\right)-\mu_{kurt}\right)^2}{M-1}\right)} \quad (7.13)$$

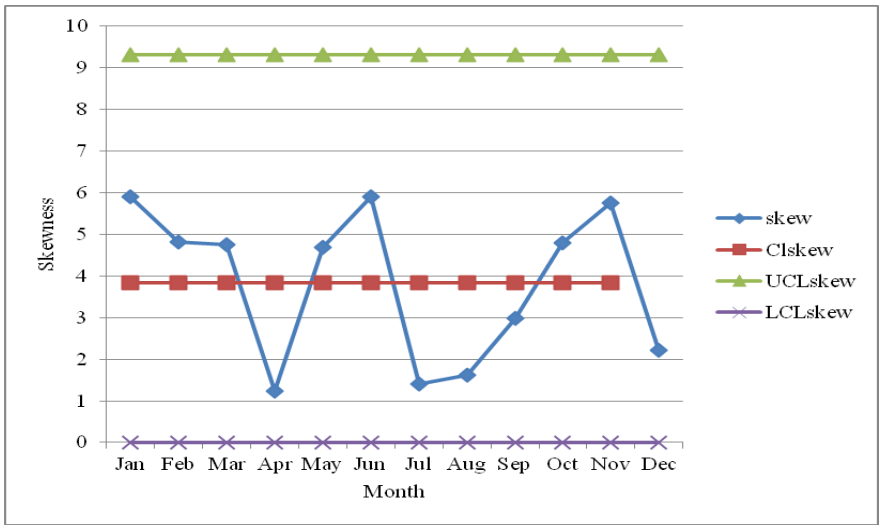
$$UCL_{kurt} = \bar{\bar{x}}_{kurt} + c\times\bar{s}_{kurt} \quad (7.14)$$

$$LCL_{kurt} = Max\{0, \bar{\bar{x}}_{kurt} - c\times\bar{s}_{kurt}\} \quad (7.15)$$

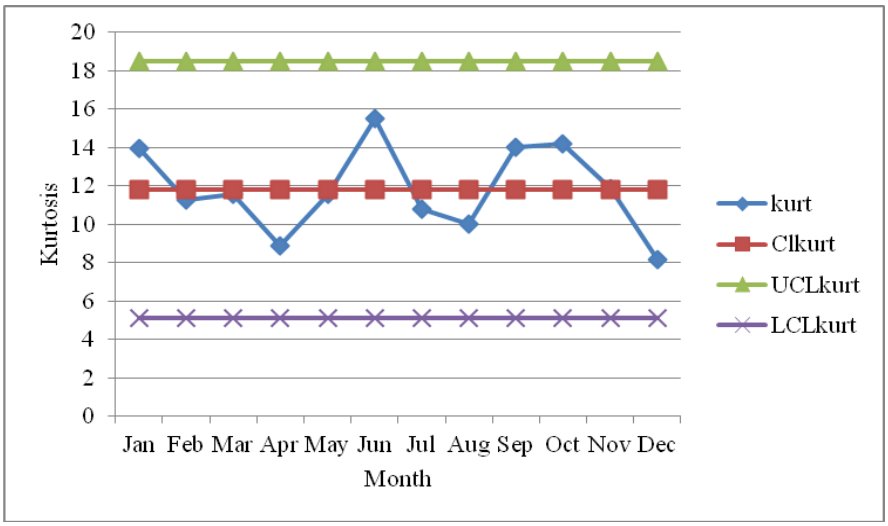
The identified control limits were obtained on the historical monthly data from Jan 2006-Dec 2008, and were tested on Jan 2009-Dec 2009. Figure 7.17 (a-b) represents the kurtosis and skewness values obtained from the reference power curves in year 2009. It can be seen that the identified control limits are well established.

Next, the monthly skewness and kurtosis is analyzed on individual wind turbines. Figure 7.18 (a-b) represent the normal and abnormal month identified from the analysis. In the event of abnormal month (Jun 2009), the skewness and kurtosis of turbine 4

amplified 10 times the usual control limit. However, other turbines were also poorly performed in Jun 2009.



(a)



(b)

Figure 7.17. Established control chart on testing data (Jan 2009-Dec 2009): (a) skewness, (b) kurtosis.

Figure 7.19 represents the power curve corresponding to the turbine 2 for January 2007 month. The high skewness and kurtosis values are corroborated by the underperformance in the wind turbine.

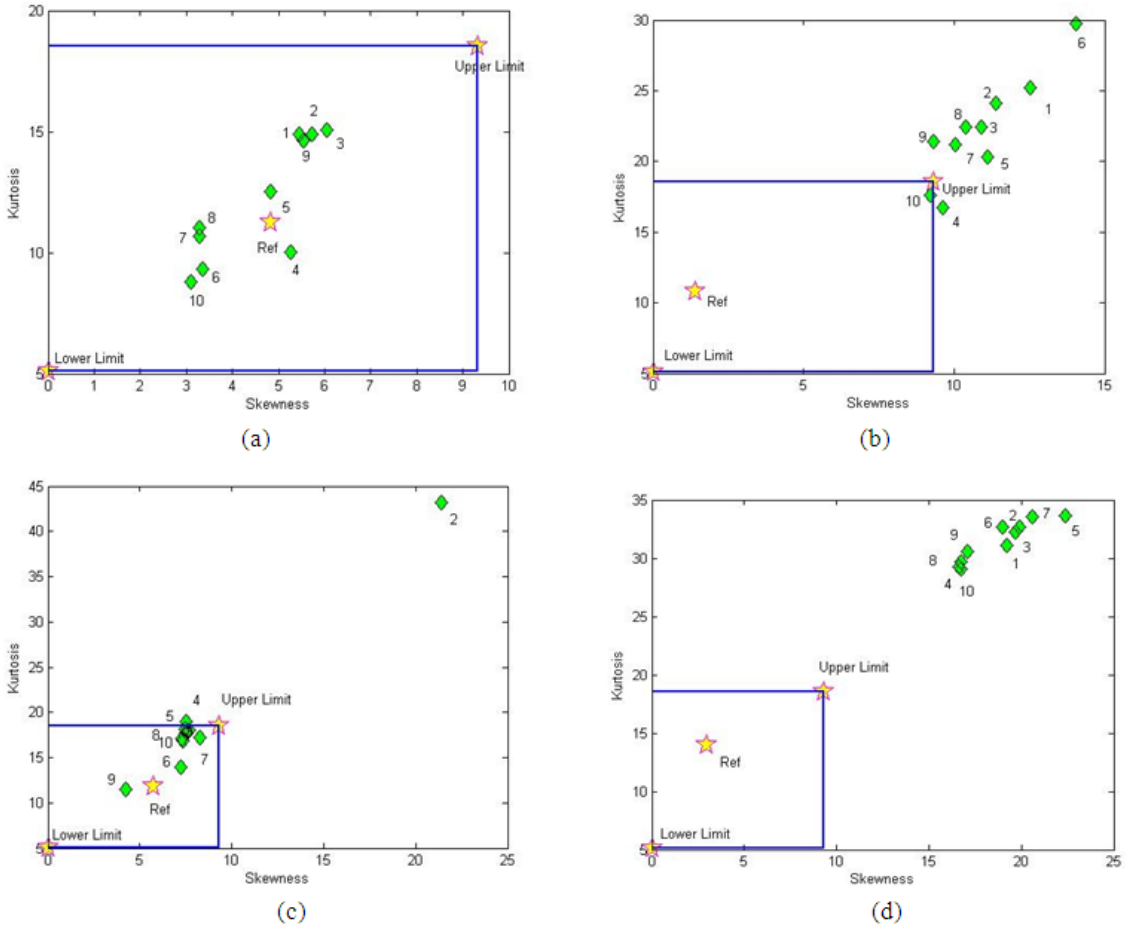


Figure 7.18. The performance of wind farm: (a) normal month (May 2009), (b) abnormal month due to curtailment (Jul 2009), (c) underperforming wind turbine due to system fault (Jan 2007), and (d) underperforming due to wind speed difference/faults (Sep 2009).

Issues associated with blade pitch controller, cable twisting, and curtailments were found in the fault logs. In order to validate the methodology developed in this section, the production data is analyzed for all wind turbines. Table 7.4-7.5 provides the details of wind farm production for May 2009 and January 2007 month. In addition to total energy production, the production data provides information about the wind turbine availability, time spent in the scheduled maintenance, and time available for production.

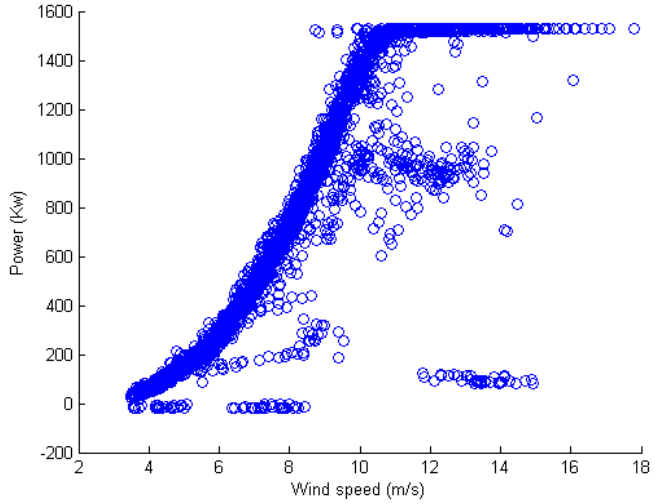


Figure 7.19. Power curve of wind turbine 2 for January 2007 month.

Based on the monthly production data as mentioned in Table 7.4-7.5, metric namely capacity factor and effective capacity of the wind turbines are evaluated. While capacity factor just depends on the total production output, the effective capacity represents the total energy produced by the wind turbine when it was operational. The two metric are evaluated as.

Table 7.4. Production data of 10 turbines for May 2009.

Turbine No.	Production (KWh)	Operation period (sec)	Turbine Ok (sec)	Downtime (sec)	Grid downtime (sec)
T1	543207	2311070	2506314	106663	140
T2	557928	2332950	2524422	3021	36
T4	535969	2416213	2599468	36066	7046
T3	534365	2441933	2564873	83077	350
T5	532819	2213722	2354300	21312	131139
T6	553524	2465115	2587407	381	0
T7	555759	2280846	2466814	12242	36
T8	550942	2505942	2627073	2050	36
T9	538602	2412925	2585400	316	0
T10	550486	2345475	2510419	9324	36

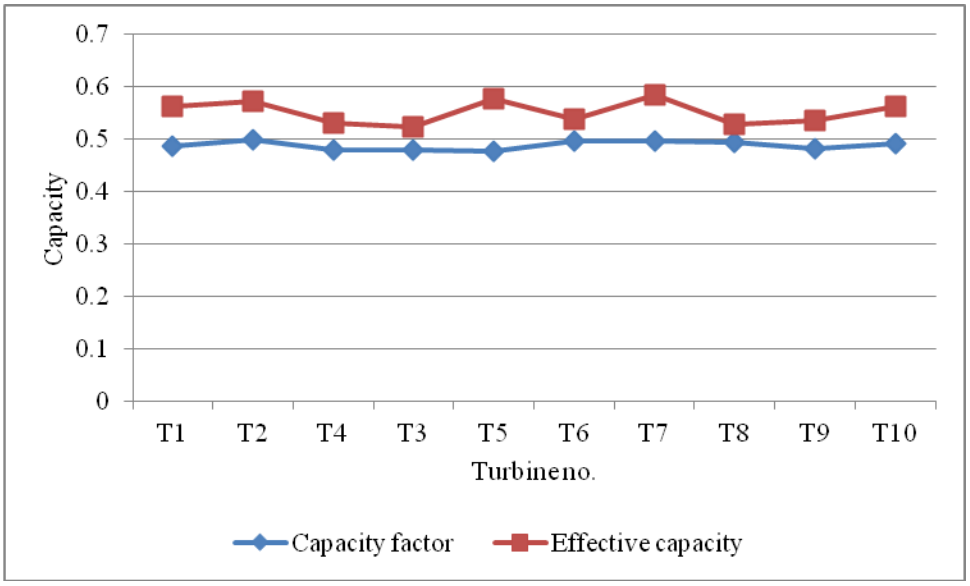
Table 7.5. Production data of 10 turbines for Jan 2007.

Turbine No	Production (KWh)	Operation Period (sec)	Turbine ok (sec)	Downtime (sec)	Grid Downtime (sec)
T1	387656	1818895	2195860	109295	0
T2	134298	794794	1010562	219474	5803
T3	396054	1940468	2220398	0	0
T4	390751	1870289	2246373	0	0
T5	385736	1772790	2170422	108103	0
T6	302623	1515413	2077813	64584	0
T7	365372	1899292	2181527	28598	0
T8	387741	1889839	2208120	0	0
T9	375200	1821935	2171730	6005	0
T10	320085	1474247	1784415	17962	0

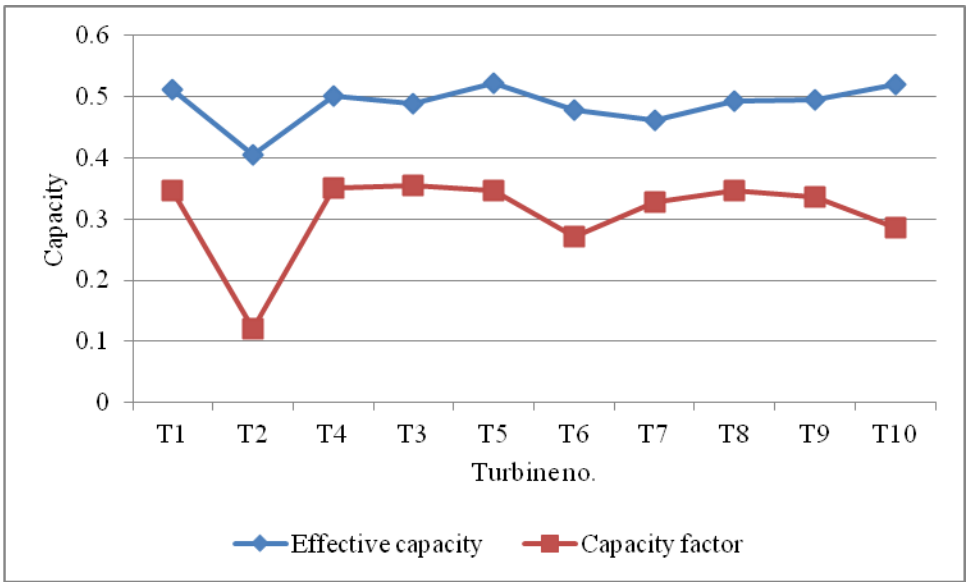
$$CF = \frac{\text{production(KWh)}}{\text{no.ofdays} \times \text{no.ofhours/day} \times \text{rated capacity(KWh)}} \tag{7.16}$$

$$EC = \frac{\text{production (KWh)}}{\text{Operation period (hr)} \times \text{rated capacity (KWh)}} \tag{7.17}$$

Figure 7.20 (a-b) provides the results obtained for the data given in Table 7.4-7.5. Clearly, the effective capacity and capacity factor are almost similar in the case when all the turbines were found to be within the specified control limit (see Figure 7.18 (a)). Whereas, capacity factor and effective capacity for the case with one turbine faults has the lower value for one out of control wind turbine (i.e. turbine 2) (see Figure 7.18 (c)).



(a)



(b)

Figure 7.20. Turbine capacity evaluation: (a) May 2009, (b) Janurary 2007.

The results obtained by analyzing the production data validates the applicability for multivariate skewness and kurtosis in performance monitoring and evalution of overall wind farm.

7.5. Summary

A systematic approach for monitoring performance of a wind farm was presented. Three performance curves, the power curve, the rotor curve, and the blade pitch curves were used. The Mahalanobis distance was calculated to identify outliers in the performance curves. The bivariate performance curve data was grouped into several clusters for better identification of outliers.

Using the skewness and kurtosis of bivariate data, the initial high frequency data was compressed to a single value. A control chart approach based on Hotelling's control chart was used for continuous monitoring of the data points in time. The transformed kurtosis-skewness graphs are better suited for monitoring than the high frequency performance curves.

CHAPTER 8

CONCLUSION

This dissertation was focused on developing prediction models for wind turbine faults. The research conducted in this dissertation was divided into parts: (1) Monitoring and prediction of wind turbines faults using states information, and (2) Monitoring the performance of overall wind farm. Within the first part, models using the turbine fault logs and operational data were constructed for prediction. Prediction models in the form of decision tree structure, neural network structure, association rules etc. were found. These models were derived from the large SCADA dataset. Models were used to predict the future state of the system, after the accuracy of the models was validated on the testing instances.

The second part of the research focused on the monitoring of overall wind farm. Monitoring the overall wind farm was based on the operational data only. Using data-mining and statistical approaches, the large volume of the data was transformed into two dimensions for performance assessment.

Eight data mining algorithms, neural network, neural network ensemble, support vector machine, boosting tree, random forests, classification and regression tree, genetic programming and *k* nearest neighbors, were applied to the development of data driven models. Although, 100% accurate models cannot be derived, still, data-mining algorithms were able to provide an acceptable accuracy of 90% or higher in most of the cases.

Modeling a single wind turbine and a wind farm were both discussed in this dissertation. In the single wind turbine research, data mining algorithms were utilized to monitor the performance of a single wind turbine. In doing so, the fault prone components of wind turbines were modeled and predicted in future. To study the wind farm monitoring, data from several wind turbines is analyzed and a unified model was developed for performance assessment. While the study on single wind turbines needs

fault information, monitoring the overall wind farm relies only on operational data, and therefore can be performed online.

This dissertation aimed to solve the monitoring and maintenance issues associated with wind turbines. Fault prediction approaches relying on the historical data were developed under the broad domain of performance monitoring.

The prediction models developed in this dissertation are effective yet inexpensive and can be easily integrated in real wind energy conversion systems (WECS). As wind industry is emerging, rare data driven research related to wind turbine performance monitoring has been performed.

An extensive study related with identification of critical status patterns of wind turbines was performed in Chapter 2. The aim was to identify the hidden patterns in the turbine statuses which could become a potential fault in the future. A component performance monitoring scheme was developed to generate alarm signals based on the predicted output. The results obtained in chapter 3 was further improved in chapter 3 through enhanced turbine monitoring where only operational wind turbine data was used. The efficacy of the models was validated on several unseen faults.

Chapter 4 presented detailed analysis on various fault prone components of wind turbines. Faults associated with wind turbine generators i.e. generator brush worn, and turbine blades, i.e. blade angle implausibility were analyzed. Due to nature of the faults, the inherited class imbalance in the dataset was resolved using advanced data-processing techniques. Tomek links, cost sensitive classification and genetic algorithm were exploited to make the dataset free from class imbalance. The prediction output was generated in the form of tree structure which could be easily analyzed and validated.

In chapter 5, another important issue of wind turbine rotating components i.e. bearing over temperature was resolved. Due to the nature of the process, anomaly detection based approaches were utilized. On an average, the presented approach was able to predict the bearing over-temperature 50 minutes ahead of its actual occurrence.

In chapter 6, much higher frequency (i.e. 42 MHz) of data was analyzed with an aim to identify and predict faults in wind turbine gearbox. The data was collected in a test environment. The analysis based on time and frequency domains were successfully able to identify the faults in turbine ring gear. Models based on data from a single sensor and multiple sensors were exploited to perform time-ahead prediction. In Chapter 7, a systematic approach for monitoring the performance of overall wind farm was presented. Wind turbines operational parameters namely wind speed; wind power, rotor speed, and blade pitch angle were used as input in performance assessment. The initial high frequency data was represented by a single kurtosis and skewness value to depict the status of wind turbines in a wind farm. The proposed approach was validated by comparing the power curves of wind turbines. Case study monitoring the behavior of several wind turbines over a period of four years is presented.

In the further research of wind farm monitoring, current work will be extended to validate the proposed approach with rotor and blade pitch curves. More data will be gathered to look for deterioration of rotor speed, and blade pitch curve. If required, abnormalities in the turbine rotor speed and blade pitch movement will be induced based on certain distribution for performance comparison. Also, the relationship between the underperforming wind turbines with their capacity factor will be investigated.

REFERENCES

Aha, D.W., Kibbler, D., Albert, M.K. Instance-based learning algorithms. *Machine learning*. Vol. 6, pp. 37-61, 1991.

Amirat, Y., Choqueuse, V., Benbouzid, M.E.H., and Charpentier, J.F. Bearing Fault Detection in DFIG-Based Wind Turbines Using the First Intrinsic Mode Function. *XIX International Conference on Electrical Machines - ICEM 2010*, Rome

Andrawus, J.A., Watson, J., Kishk, M. and Adam, A. The Selection of a Suitable Maintenance Strategy for Wind Turbines. *Wind Engineering*. Vol. 30 (6), pp. 471-486, 2006.

Annunzio, C. D, and Santoso, S. Wind power generation reliability analysis and modeling. *IEEE Power Engineering Society General Meeting*. pp. 35–39, 2005.

Arthur, N., and Dunn, M. Effective Condition Based Maintenance of reciprocating compressors on an offshore oil and gas installation. *IMechE International Conference on Compressor and their system.* 2001.

Ashley, F., Cipriano, R.J., Breckenridge, S., Briggs, G.A., Gross, L.E., Hinkson, J., and Lewis, P.A. *Bethany Wind Turbine Study Committee Report*, 2007 www.townofbethany.com

Avdelidis, N.P, Almond, D.P., Ibarra-Castanedo, C., Bendada, A., Kenny, S. and Maldague, X. Structural integrity assessment of materials by thermography. *In Conf. Damage in Composite Materials CDCM (Stuttgart, Germany)*, 2006.

Bakirtz, A.G. A probabilistic method for the evaluation of the reliability of stand-alone wind energy conversion systems. *IEEE Trans Energy Convers*. Vol. 7(1), pp. 530–536, 1992.

Barandela, R., Sanchez, J.S., Garcia, V., and Rangel, E. Strategies for learning in class imbalance problems. *Pattern Recognition*. Vol. 36, pp. 849-851, 2003.

Barberis, N. N, and Holmstrom, O. Aspects of relevance in offshore wind farm reliability assessment. *IEEE Trans Energy Convers*. Vol. 22(1), pp. 159–66, 2007.

Bastia, G.E.A.P.A., Prati, R.C., and Monard, M. C. A study on the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, Vol. 6(1), pp. 20-29, 2004.

Biegel, B., Juelsgaard, M., Kraning, M., Boyd, S., and Stoustrup, J. Wind Turbine Pitch Optimization. *2011 IEEE International Conference on Control Applications*

(CCA) Part of 2011 IEEE Multi-Conference on Systems and Control Denver, CO, USA. September 28-30, 2011.

Billinton, R. and Allan, R.N. Reliabiliv Evaluation of Large Electric Power System. *Kluwer Academic Publishers, Boston, Massachusetts*, 1988.
Billinton, R., and Chowdhury, A. A. Incorporation of wind energy conversion system in conventional generating capacity adequacy assessment. *In IEE Proceedings-C*. Vol. 139(1), pp. 47–56, 1992.

Billinton, R., and Gan, L. Wind power modeling and application in generating adequacy assessment. WESCANEX 93. Communications, Computers and Power in the Modern Environment. *In: Conference Proceedings, IEEE; Billinton, R. and Allan, R.N., Reliability Evaluation of Power Systems, Plenum Publishing (New York)*, 1996.

Breiman, L. Random Forest. *Machine Learning.* Vol. 45, pp. 5-32, 2001.

Byon, E., Ding, Y. and Ntaimo, L. Optimal maintenance strategies of wind turbine systems under stochastic weather conditions. *IEEE Transactions on Reliability*. Vol. 59(2), pp. 393-404, 2010.

Cano, J.R., Herrera, F., and Lozano, M. Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. *IEEE transactions on evolutionary computation*, Vol. 7(6), 561-576, 2003.

Caselitz, P., Giebhardt, J. Rotor Condition Monitoring for Improved Operational Safety of Offshore Wind Energy Converters. *Journal of Solar Energy Engineering*. Vol. 127, pp. 253-261, 2005

Caselitz, P., Giebhardt, J., and Mevenkamp, M. Application of condition monitoring systems in wind energy convertors. *In proceedings of the EWEC, Dublin*. 1997, pp. 1-4.

Caselitz, P., Giebhardt, J., and Mevenkamp, M. Application of condition monitoring systems in wind energy convertors. *Proceedings of the EWEC*, Dublin, pp. 1-4, 1997.

Caselitz, P., Giebhardt, J., and Mevenkamp, M. Development of a fault detection system for wind energy convertors. *In proceedings of the EUWEC, Goteborg*. 1996, pp. 1004-1007.

Caselitz, P., Giebhardt, J., and Mevenkamp, M. Online fault detection and prediction in wind energy convertors. *In proceedings of the EWEC*, Thessaloniki. 1994, pp. 623-627.

Chandler, H. Editor. Wind energy—the facts. *European Wind Energy Association*, 2003. Available at: /www.ewea.org06-05-02S.

Chands, P.K., and Tokekar, S.V. Expert-based maintenance: a study of its effectiveness. *IEEE Trans Reliab*. Vol. 47(1), pp. 95–97, 2008.

Chawla, N.V., Bowyer, K.W., Hall, L.O., and, Kegelmeyer. W.P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*. Vol. 16, pp. 321-357, 2002.

Chen, H. Generating system reliability optimization. PhD thesis. University of Saskatchewan. 2000. pp. 103–108.

Ciang, C.C., Lee, J.R., and Bang, H.J. Structural health monitoring for a wind turbine system: a review of damage detection methods. *Meas. Sci. Technol.* Vol. 19, pp. 122001-122021, 2008.

Cleofas, L., Valdovinos, R.M., Garcia, V., and Alejo, R. Use of ensemble based on GA for imbalance problem. *Lecture notes in computer science*. Vol. 5552, pp. 547-554, 2009.

Cotton, I., Jenkins, N., and Pandiaraj, K. Lightning protection for wind turbine blades and bearings. *Wind Energy*. Vol. 4, pp.23–37, 2001.

Cusido, J., Romeral, L., Ortega, J., Rosero, J., and Espinosa, A. Fault detection in induction machines using power spectral density in wavelet decomposition. *IEEE Transactions on Industrial Electronics*. Vol. 55(2), pp. 633–643, 2008.

Davidson, J., and Hunsley, C. The Reliability of Mechanical Systems. *In 2nd ed. Institute of Mechanical Engineers, Great Britain*. 1994.

Decade, V. S. and Modak, J. P. Maintenance strategy for tilting table of rolling mill based on reliability considerations. *Reliability Engineering and System Safety*. Vol. 80, pp. 1–18, 2003.

Dekker, R. Application of maintenance optimization models: a review and analysis. *Reliability Engineering and System Safety*. Vol. 51, pp. 229-240, 1996.

Dimitrovski, A., amd Tomsovic, K. Impact of wind generation uncertainty on generating capacity adequacy. *In 9th international conference on probabilistic methods applied to power systems*. pp. 1–6, 2006.

Ding, Y., Byon, E., Park, C., Tang, J., Lu, Y., and Wang, X. Dynamic Data-Driven Fault Diagnosis of Wind Turbine Systems. *Proceedings of the 7th international conference on Computational Science, Part I: ICCS*, Vol. 4487, pp. 1197-1204, 2007.

Dodd, C., McCalla, T., and Smith, J.G. How to protect a wind turbine from lightning. *National Aeronautics Space Admin, DOE/NASA 0007-1, NASA-CR-168229*, 1983.

Echavarria, E., Tomiyama, T., and Bussel, G. Fault diagnosis approach based on a model-based reasoner and a functional designer for a wind turbine: an approach towards self-maintenance. *Journal of Physics Conference Series*. Vol. 75(1), 2007.

Eecen, P.J., Braam, H., Rademakers, L.W.M.M., and Obdam, T.S. Estimating costs of operations and maintenance of offshore wind farms, *EWEC2007 conference*, pp. 7-10 May, Milan, Italy, 2007.

Farrar, C. R., and Worden, K. An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A (London: Royal Society Publishing)*. Vol. 365(1851), pp. 303–315, 2012.

Fonseca, I., Farinha, T., Barbosa, F.M. On-condition maintenance of wind generators - from prediction algorithms to hardware for data acquisition and transmission. *WSEAS transactions on circuits and systems*, Vol. 7(9), pp. 909-918, 2008.

Garcia, M.C., Sanz-Bobi, M.A., Del Pico J. SIMAP: intelligent system for predictive maintenance application to the health condition monitoring of a wind turbine gearbox. *Journal of Comput Ind*, Vol. 57(6), pp.552–68, 2006.

Giebhardt, J., Rouvillain, J., Lyrner, T., Bussler, C., Gutt, S., Hinrichs, H., Gram-Hanses, K., Wolter, N. and Giebel, G. Predictive condition monitoring for offshore wind energy converters with respect to the IEC61400-25 standard. *Germany Wind Energy Conf. (DEWEK)*, 2005.

Giorsetto, P., Utsurogi, K.F. Development of a new procedure for reliability modeling of wind turbine generators. *IEEE Trans Power Apparatus Syst*. Vol. 102(1), pp. 134–43, 1983.

Gu, J. Random Forest Based Imbalanced Data Cleaning and Classification. *PAKDD*, pp. 1-7, 2007

Hahn, B., Durstewitz, M., and Rohrig, K. Reliability of Wind Turbines Experiences of 15 years with 1500 W. pp. 1-4, http://www.iset.uni-kassel.de/abt/FB-I/publication/2006-02-09Reliability.pdf

Hameed, Z. ., Hong, Y.S.., Cho, Y.M., Ahn, S.H., Song, C.K. Condition monitoring and fault detection of wind turbines and related algorithms: A review. *Renewable and Sustainable energy reviews*, 13, pp. 1-39, 2009.

Hatch. C. Improved wind turbine condition monitoring using acceleration enveloping. *Orbit*. pp.58-61, 2001.

Hripcsak, G., and Wilcox A. Reference standards, judges, comparison subjects: roles for experts in evaluating system performance. *Journal of American Medicine Informatics Association*. Vol. 9, pp. 1-15, 2002.

Huyer, S. A., Simms, D., and Robinson, M.C. Unsteady Aerodynamics Associated with a Horizontal-Axis Wind Turbine. *American Association for Artificial Intelligence Journal*, Vol. 34(7), pp. 1410-1419, 1996.

Jeffries, W. Q, Chambers, J.A., and Infield, D.G. Experience with bi-coherence of electrical power for condition monitoring of wind turbine blades. *IEEE Proceedings on Vision Image & Signal Processing*, Vol. 145, pp. 141–148, 1998.

Jiang, W., Zheng, Y., and Donghan, F. A review on reliability assessment for wind power. *Renewable and Sustainable Energy Reviews*. Vol. 13 (9), pp. 2485-2494, 2009.

Karki, R., and Billinton, R. Reliability/cost implication of PV and wind energy utilization in small isolated power systems. *IEEE Trans Energy Conversion*. Vol. 16(4), pp. 368–73, 2001.

Kollo, T. Multivariate skewness and kurtosis measures with an application in ICA, *Journal of Multivariate Analysis*, Vol. 99, pp. 2328–2338, 2008.

Kramer, S.G.M., Leon, F.P., and Appert, B. Fiber optic sensor network for lightning impact localization and classification in wind turbines. *In Proceedings of IEEE ICMFIIS'06*. pp. 173–178, 2006.

Kusiak, A., and Zhang, Z. Adaptive Control of a Wind Turbine with Data Mining and Swarm Intelligence. *IEEE Transactions on Sustainable Energy*. Vol. 2(1), pp. 28-36, 2011.

Kusiak, A., and Li, W. Virtual Models for Prediction of Wind Turbine Parameters. *IEEE Transactions on Energy Conversion*. Vol. 25(1), pp. 245-252, 2010.

Kusiak, A., and Verma, A. A Data-Driven Approach for monitoring blade pitch faults in wind turbines. *IEEE Transactions on Sustainable Energy*. Vol. 2(1), pp. 87-96, 2011.

Kusiak, A., and Verma, A. The Future of Wind Turbine Diagnostics. *Wind System Magazine*. Vol. 2(8), pp. 66-71, 2010.

Kusiak, A., Zheng, H.Y., and Song, Z. Models for Monitoring Wind Farm Power. *Renewable Energy*. Vol. 34(3), pp. 583-590, 2009.

Kusiak, A., Zheng, H.Y., and Song, Z. Short-Term Prediction of Wind Farm Power: A Data-Mining Approach. *IEEE Transactions on Energy Conversion*, Vol. 24 (1), pp. 125-136, 2009.

Landberg, L. Short-term prediction of the power production from wind farms. *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 80(1), pp. 207–220, 1999.

Learney, V.C., Sharpe, D.J., and Infield, D. Condition monitoring technique for optimization of wind farm performance. *International Journal of COMADEM*. Vol. 2(1), pp. 5–13, 1999.

Li, J., Li, X., and Yao, X. Cost- sensitive classification with genetic programming. *Proceedings of the 2005 Congress on Evolutionary Computation*, Vol. 3, pp. 2114–2121, 2005.

Li, Q., Wang, Y., and Bryant, S.H. A novel method for mining highly imbalanced high-throughput screening data in PubChem. *Bioinformatics*. Vol. 25(24), pp. 3310-3316, 2009.

Lin, R., Zhu, S., Wu, H., and Zheng, J. Rolling bearings fault diagnosis based on energy operator demodulation approach. *In 4th World Congress on Intelligent Control and Automation, Shanghai, China*, 2002, pp. 2723–2727.

Lucentem M., Condition Monitoring System in Wind Turbine Gearbox. Master's degree report, Stockholm, Sweden, 2008.

Margineantu, D.D. On class probability estimates and cost-sensitive evaluation of classifiers. *In Workshop on Cost-Sensitive Learning at the 17th International Conference on Machine Learning*, 2000, pp. 1-3.

Marquez, F. P. G., Schmid, F., and Collado, J. C. A Reliability centered approach to remote condition monitoring- A railway points case study. *Reliability Engineering and System Safety*. Vol. 80, pp. 33–40, 2003.

McMillan, D., and Ault, G. W. Condition monitoring benefit for onshore wind turbines: Sensitivity to operational parameters. *IET Renewable Power Generation*. Vol. 2, pp. 60–72, 2008.

Mobley, K. K. An introduction to Predictive Maintenance. 2002. ISBN 0-7506-7531-4.

Orsagh, R. F., Lee, H., Watson, M., Byington, C.S., and Powers, J. Advanced Vibration Monitoring for Wind Turbine Health Management. Impact technologies, http://www.impact-tek.com/

Pacot, C., Hasting, D., and Baker, N. Wind farm operation and maintenance management. *In proceedings of the PowerGen Conference Asia, Ho Chi Minh City, Vietnam*. 2003, pp. 25–27.

Pan, F., Converse, T., Ahn, D., Salvetti, F., and Donato, G. Feature Selection for Ranking using Boosted Trees. *CIKM'09*, November 2–6, 2009, Hong Kong, China

Pedersen, K.O.H., and Havemann, H. An Alternative approach to power engineering. Power Engineering Society Summer Meeting, 2000. *IEEE 2000*; Vol. 4, pp. 2085–90.

Pitchford, C., Grisso, B. L., and Inman, D. J. Impedance-based structural health monitoring of wind turbine blades. *Proc. SPIE—Health Monitoring of Structural and Biological Systems (CA, USA)*. pp. 65321I-1–65321I-11, Publishing (New York), 1996.

Rademakers, L.W.M.M., Braam, H., Zaaijer, M.B., and Van Bussel, G.J.W. Assessment and optimization of operation and maintenance of offshore wind turbines, *ECN*, pp. 1-4, 2007.

Rao, B. K. N. Handbook of Condition Monitoring, Oxford, Elsevier, 1996.

Rausand, M. Reliability-Centred Maintenance. *Reliability Engineering and System Safety*, Vol. 60, pp. 121–132, 1998.
Rausand, M., Hoyland, A. Hoboken System Reliability Theory. John Wiley & Sons 2004, ISBN 0-471-47133-X.

Reeves, C.R., and Bush, D.R. Using genetic algorithms for training data selection in RBF networks, Instance selection and construction for data mining, H.Liu, and H.Motoda, Eds. Norwell, *MA: Kluwer*, 2001, pp. 339-356.

Ribrant, J. Reliability performance and maintenance - A survey of failures in wind power systems, Master's thesis, KTH School of Electrical Engineering, 2005.

Ribrant, J., and Bertling, L. Survey of failures in wind power systems with focus on Swedish wind power plants during 1997-2005. *IEEE transaction on energy conversion*, Vol. 22(1), pp. 167-173, 2007

Rodriguez, L., Garcia, E., Morant, F., Correcher, A., and Quiles, E. Application of latent nestling method using colored Petri nets for the fault diagnosis in the wind turbine subsets. *In Proc. 2008 IEEE Int. conf. emerging technologies and factory automation*. pp. 767-773.

Schlechtingen, M., and Santos, I.F. Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. *Mechanical Systems and signal processing*. Vol. 25(5), pp. 1849-1875, 2011.

Singh C, Lago-Gonzaloz A. Reliability modeling of generation systems including unconventional energy sources. *IEEE Trans Power Appl Syst*, 1985, PAS-104:1049–55.

Sorensen, J.S. Framework for Risk-based Planning of Operation and Maintenance for Offshore Wind Turbines. *Wind Energy*. Vol. 12(5), pp. 493-506, 2009.

Tandon, N., and Parey, A. Condition monitoring of rotatory machines. *Technical report*, 2002.

Tarek, H., Ehab, F., and Magdy. M. One day ahead prediction of wind speed and direction. *IEEE Transactions on Energy Conversion*. Vol. 23(1), pp. 191-201, 2009.

Tavner, P.J., Xiang, J., and Spinato, F. Reliability analysis for wind turbines. *Wind Energy*, Vol. 10(1), pp. 1-18, 2007.

Thresher, R., Robinson, M., Veers, P. The status and future of wind energy technology. *IEEE Power & Energy Magazine*, Vol. 5, pp. 34–46, 2006.

Ting, K. M. An instance-weighting method to induce cost sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*. Vol. 14(3), pp. 659–665, 2002.

Ting, K. M., and Zheng. Z. Boosting cost-sensitive trees. *In Proceedings of the First International Conference on Discovery Science*. pp. 244–255, 1998.
Tomek, I. Two modifications of CNN. *IEEE transcations on systems man and communications (SMC)*, Vol. 6, pp. 769-772, 1976.

Tsai, C.S., Hsieh, C.T., Huang, S.J. Enhancement of damage-detection of wind turbine blades via CWT-based approaches. *IEEE Transactions on Energy Conversion* Vol. 21, pp. 776–81, 2006.

Turney, P. D. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, Vol. 2, pp. 369– 409, 1995.

Verbruggen, T.W. Wind turbine operation & maintenance based on condition monitoring
WT-O, Final report. *ECN-C–03-047*, April 2003.

Walford, C.A. Wind turbine reliability: understanding and minimizing wind turbine operation and maintenance costs. Sandia Report, SAND2006-1100. Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550; 2006.

Wang, L., Singh, C. Adequacy assessment of power-generating systems using a modified simple genetic algorithm. *IEEE Trans. Power System*, Vol. 17, no. 4, pp. 974–981, 2002.

Wang, L., Singh, C. Population-Based Intelligent Search in Reliability Evaluation of Generation Systems with Wind Power Penetration. *IEEE Transactions on power systems*, Vol. 23, no. 3, pp. 1136-1345, 2008.

Wang, X., Dai, H., Thomas, R.J. Reliability modeling of large wind farms and electric utility interface systems. *IEEE Trans Power Apparatus Syst*, 1984.

Weiss, G.M. Mining with rarity: a unifying framework. *SIGKDD Explorer Newsletter*. Vol. 6, pp 7-10, 2004.

Wiggelinkhuizen, E., Verbruggen, T., Braam, H., Rademakers, L., Xiang, J., Watson S. Assessment of condition monitoring techniques for offshore wind farms. *Journal of Solar Energy Engineering-Transactions of the ASME* 2008, Vol. 130, pp. 0310049-1–0310049-9.

Wilkinson, M.R., Tavner, P.J. Extracting condition monitoring information from a wind turbine drive train. *In Proceedings of UPEC'04*, 2004, pp. 591–594.
Wilkinson, M.R., Tavner. P.J. Condition monitoring of wind turbine drive trains. *In: Proceedings of ICEM'06*, 2006. pp. 1–6.

Woodhouse, J. Combining the best bits of RCM, RBI, TQM, Six-Sigma and other solutions. *Institute of Asset Management publication*, 2002.

Zaher, A., and McArthur, S. A multi-agent faults detection system for wind turbine defect recognition and diagnosis. *In: Proc.2007 IEEE Lausanne POWERTECH*. pp. 22-27.

Zhang, A., Verma, A., and Kusiak, A. Fault Analysis of the Wind Turbine Gearbox. *IEEE Transactions on Energy Conversion*, Vol. 27, No. 2, 2012, pp. 526-535