Automated Fault Detection for Wind Farm Condition Monitoring

Ammar S. Zaher

A thesis submitted for the degree of Doctor of Philosophy

Department of Electronic and Electrical Engineering
University of Strathclyde
Glasgow G1 1XW

September 2010

Contents

Index of Figures & Tables

List of Abbreviations

ACL	Agent Communication Language
AMRM	Agent Management Reference Model
AmbientParser	Ambient Temperature SCADA Data parsing agent
ANN	Artificial Neural Network
API	Application Programming Interface
BFGS	Broyden–Fletcher–Goldfarb–Shanno
BMU	Best Matching Unit
CA	Collation Agent
CBR	Case Based Reasoning
CCL	Constraint Choice Language
CMS	Condition Monitoring System
COMMAS	Condition Monitoring Multi-Agent System
DataManagement	The Data Management Agent
DF	Directory Facilitator
EAA	Engineering Assistant Agent
FDD	Fault Detection and Diagnosis
FDS	Fault Detection System
FIPA	Foundation for Intelligent Physical Agents
FIPA-ACL	See ACL
FIPA-SL	See SL
FRI	Fault Record Interpretation
FRR	Fault Record Retrieval
GB	Gearbox
GBBearingParser	Gearbox Bearing SCADA data parsing agent
GBOilDataParser	Gearbox oil SCADA data parsing agent
GBOilProcessing Agent	Gearbox oil SCADA data processing agent
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IEI	Incident and Event Identification
IIOP	Internet Inter-Orb Protocol
IP	Interaction Protocol
JADE	Java Agent DEvelopment
JOONE	Java Object-Oriented Neural Engine
KIF	Knowledge Interchange Format
LWKD	Landwirtschaftskammer Deutschland
MAS	Multi-Agent System
MSE	Mean Square Error
NN	Neural Network
PDataParser	Power SCADA Data parsing agent
PEDA	Protection engineering Diagnostic Agents
PVD	Protection Validation and Diagnosis
RDF	Resource Description Framework
SCADA	Supervisory Control And Data Acquisition
SL	Semantic Language

SOM	Self-Organising Map
SVM	Support Vector Machine
UHF	Ultra High Frequency
WDataParser	Wind speed SCADA data parsing agent
Wffdagent	System initialisation agent, performs necessary start up routines.
WSD	Windstats Deutschland
WSDK	Windstats Denmark

List of Variables & Equations

(t)	Current time point
(t-1)	previous time point (10 minutes ago)
(t-2)	previous time point (20 minutes ago)
(t-3)	previous time point (30 minutes ago)

Euclidean distance metric (similarity measure):

$$Dist = \sqrt{\sum_{i=0}^{i=n} (V_i - W_i)^2}$$

Dist	Distance between two points (V_i, W_i) in feature space
V_i	Current input vector
W_i	Cluster centre

Radius of BMU:

$$\sigma(t) = \sigma_o \exp\left(-\frac{t}{\lambda}\right)$$

$\sigma(t)$	Radius at current iteration
σ_o	Initial width of lattice
$\left(-\frac{t}{\lambda}\right)$	Time constant at current iteration of loop (*t*)

Adjusted Weight of node based on varying distance from BMU:

$$W(t+1) = W(t) + \Theta(t)L(t)(V(t) - W(t))$$

$W(t+1)$	Adjusted weight of node
$W(t)$	Weight of node from previous iteration
$\Theta(t)$	Degree of influence based on distance from BMU
$L(t)$	Learning rate
$V(t)$	Input vector

Degree of Influence based on distance from BMU:

$$\Theta(t) = \exp\left(-\frac{dist^2}{2\sigma^2(t)}\right)$$

$\Theta(t)$	Degree of influence based on distance from BMU at current iteration
$dist$	Current distance of node from BMU
σ	Current width of lattice

Learning Rate:

$$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right)$$

$L(t)$	Learning rate
L_0	Constant learning rate value
$\left(-\frac{t}{\lambda}\right)$	Time constant at current iteration of loop (t)

Net input (summation of inputs into a NN):

$$net_j = \sum_{i=1}^{i=n} w_i x_i$$

net_j	Net input
w_i	Connection weights at node w connection $_i$
x_i	Input value

Activation Level based on transfer function used:

$$y_j = f(net_j)$$

y_j	Calculated activation level
net_j	Net input
f	Chosen transfer function

Sigmoid Transfer function:

$$y_j = f(net_j) = \frac{1}{(1+e^{-net_j})}$$

net_j	Net input

Back Propagated Error at Output Node:

$$\delta_j = (t_j - o_j) f^{'}(net_j)$$

δ_j	Error at current node
t_j	Target output at node
o_j	Actual output at node
net_j	Net input

Back Propagated Error at Hidden Node:

$$\delta_j = f^{'}(net_j)\sum_k \delta_k w_k$$

δ_j	Error at current node
δ_k	Error at k weight connection at current node
w_k	Weight at k connection at current node
net_j	Net input

Classification Plane in High Dimensional Space SVM:

$$f(x) = sign(w \cdot x - b)$$

sign	Signum function returns the sign of a real number
w.x	Vector determining orientation of plane
b	Scalar determining offset of plane from the origin

Data Set Normalisation Scaling:
Xn = (X-Min)/(Max-Min)

Xn	The new scaled value
X	Value to be scaled,
Min	Smallest value in the training data set,
Max	Largest value in the training data set.

Abstract

Wind turbines are becoming more established as an economically viable alternative to fossil-fuelled power generation. Recently wind farms consisting of hundreds of units are being built in various locations around the country adding a significant amount of electrical generating capacity. As the size of wind farms continues to increase, business economics dictate the need for effective condition monitoring systems that allow for careful asset management to minimise downtime and maximise availability and profits. Most modern turbines are built with integrated condition monitoring systems that acquire data and store this through Supervisory Control and Data Acquisition (SCADA) Systems. This data quickly becomes unmanageable and brings with it the problems of managing and interpreting it.

This thesis considers the development of an automated SCADA data analysis system that aims to interpret the large volumes of data that are generated, with the intention of identifying faults in their early stages before they manifest into more serious catastrophic failures. A number of different analysis techniques for interpreting the SCADA are considered and a methodology of identifying faults in their incipient stages in the gearbox and generator using basic SCADA temperature data is described. Most CM techniques in the research literature focus on one aspect of a wind turbine with regards to identifying faults that may manifest within it. This research also puts forward the development of a multi-agent platform capable of combing multiple data sources and analysis techniques into one system to improve the opportunity of extracting and interpreting interesting information found in the SCADA and present it through a single point of contact for the operator. This provides the possibility of developing a more complete condition monitoring system that can monitor all of the main components of each turbine across a complete wind farm using both, existing and future condition monitoring techniques developed for the interpretation of wind farm data.

Acknowledgements

First of all I would like to extend my warmest thanks to Prof. Stephen McArthur for his support, thoughtful guidance and positive words throughout the duration of my research. For this I am extremely grateful.

A great number of Individuals have also unknowingly helped me in my research, and therefore I would like to take this opportunity to thank all my fellow researchers and academics in the Electronic & Electrical Engineering Department for their insight and, experience. All of you have helped contribute towards the completion of this Thesis. I would not have been able to accomplish such a task without the fantastic research environment provided by the Institute of Energy and the Environment at the University of Strathclyde

And finally to my mum, dad and all my Family: thank you for your unconditional love and support throughout this period of my life. I love you too unconditionally.

Ammar S. Zaher Glasgow September 2010.

1. Introduction and Motive for Research

Wind energy is becoming the fastest growing renewable energy source around the world (EWEA 2009). An increasing number of wind farms are being built around the UK by utility companies with the aim of achieving their desired carbon emission targets, and providing alternative energy sources for consumer supply. Although modern wind turbines have reached high technical standards, there is still a strong potential for further development especially with large, megawatt size machines, where enhancements in availability, reliability and lifetime of the turbine are all viable factors for improvement.

The benefits that condition monitoring (CM) offers in traditional power plants is well documented (Leaney and Sharpe et al 1997) and the advantages they pose are arguably indispensable for the high value equipment being monitored. It is seen as a key requirement for electrical utilities to monitor the condition of their large plant items in order that they can maximise plant availability and minimise the number of unplanned outages, to ultimately capitalize on profits. Unplanned outages that occur at plants can result in significant losses in revenue due to the lengthy procedures involved in shutting down the plant, identifying and removing the damaged equipment and then finally replacing the necessary components. CM systems are essentially designed to avoid such undesirable scenarios by providing operators with information concerning the health of their machines, which in turn can help them improve the wind farm's operational efficiency by allowing more informed decisions to be taken regarding maintenance.

Most modern turbines are now manufactured with some form of integrated CM system that can monitor the main internal components. This monitored data is collated and stored via a supervisory control and data acquisition (SCADA) system that archives the data for the monitored components in a convenient manner. Ideally this SCADA data would then be analysed by a trained operator to deduce the health or state of the turbine's components, where this information would be used to counteract unplanned outages and help plan effective maintenance schemes. This ideal scenario is not always the case however, as practical difficulties and economic issues can impact the operation and maintenance of wind farms which can lead to CM systems not being exploited in an effective manner where their full potential is maximised.

1.1 Limitations affecting the efficiency of wind turbine CM systems

The first issue that can limit their efficacy arises from the data collected by the CM systems. It quickly accumulates to create large and unmanageable volumes making data analysis extremely difficult, impractical and more often than not an impossible task to be carried out manually. The second issue that limits their use originates from the complex operational characteristics of a turbine, mainly due to its close coupling with the weather conditions. This makes the analysis of its behaviour and hence its operation and maintenance (O&M) much more challenging due to the dynamic nature of these imposing factors (e.g. veer, shear). This leads to the need for trained experts who are able to interpret the data. This poses a problem since the number of engineers who would be able to use the output of the CM system is limited to those who have a thorough knowledge of the operational mechanics and characteristics of the machines and how this is reflected in the data, which is a very specific and rare skill set. The last issue is a matter of cost, there is of course an added expense of hiring such expert operators to carry out this tedious and time consuming task.

Turbine manufacturers are keen to exploit the commercial opportunity with wind farm operators by arguing from a theoretical perspective concerning the benefits associated with CM. Operators are more cautious however and tend to question the technical and economic value of CM systems until they are comfortable that the benefits have been demonstrated in the industry. Most wind farm operators in the UK (at the time of writing), experience this problem of information overload and the difficulty associated with the interpretation of their CM data (Patel 2007) and (Wind Energy Update 2009). Therefore more often than not they tend to ignore the output of their systems completely "unless some failure requires an analysis on a reactive basis" (Clive et al 2008). They are quite content in the meantime to carry on applying periodic maintenance to their sites until it can be demonstrated that the output of their CM systems can be understood and utilised in a more effective manner even though this approach may not be the most economical solution.

1.2 An Automated Solution

In order to overcome these problems and maximise the potential of CM systems to make their benefits more apparent, a system that can automate the process of analysing and interpreting the large volumes of data is needed. This would remove the burden from operators since generalised high level information regarding the condition of the plant would be available automatically, dramatically reducing the complexity of having to manually carry out the data analysis task and help the operators to make a more informed decision regarding the maintenance of their machines.

The aim of this research is to develop an online SCADA data analysis fault detection system (FDS). Online FDS have been developed for a number of applications but expertise in the domain of wind turbine CM, as will be shown by the literature review in the following chapter, is not yet established to the point which would enable a robust and effective FDS to be developed. FDS do not replace the operator but aim to provide them with a form of decision support. They can be seen as an addition to the already existing integrated CM systems as they enhance the usability of the CM output providing meaningful results which inherently exist in the data but may not be initially apparent through the use of data analysis algorithms. Therefore a complete CM system can be seen as being made up from the interaction of two main closely coupled components and one without the other can often render the system inefficient and, arguably, incomplete. These are:

- The monitoring (hardware/sensors) technologies dedicated to recording important measurands on the turbine and;
- The data analysis algorithms that make it possible to identify and diagnose failures and possibly identify warning signs before problems occur.

This thesis will illustrate the development of a FDS system using multi-agent system technology which uses machine learning techniques to analyse and interpret SCADA from a live operational wind farm. It has been identified from the literature that most of the techniques and research focuses on the monitoring of a specific problem or component of the turbine in isolation. This is mainly due to the complex nature of each individual problem. A key feature of this research is the use of multi-agent technology to address this gap and provide a platform for wind farm FDS. This, as is discussed in this thesis, will allow a number of independent analysis techniques to be brought together, processing data from multiple sources and therefore allow a more complete view of the turbines' condition in the wind farm. To the best of the author's knowledge there has been no other research found in the literature (at the time of writing) that addresses the automated SCADA data analysis to provide both a flexible and extensible platform for the monitoring of wind farms in the manner presented in this thesis.

1.3 Key Challenges & Research Direction

In order to manually analyse SCADA successfully, it is necessary that the analyst undertaking the task has the knowledge and understanding of the connection between actual defects that may occur in a wind turbine's components, and what they manifest in the data. The ideal scenario for the development of a FDS is that this information would be readily available to capture, and therefore build systems that would go through the same thinking process of a trained operator in order to interpret the data. This is not however the case and such information is not easily attainable as few operators have had the extensive experience required to acquire this knowledge. This is mainly due to the relatively recent introduction of wind turbine CM systems and the limited amount of historical data available.

In the case where this information is non-existent or unavailable, then detailed data mining activities (the process of analysing data from different perspectives and summarising it into useful information) (Usama 1996) can be used in an attempt to derive useful relationships. Successful interpretation of the data is therefore predicated upon either of the following pre-requisites (McArthur & Booth et al 2005):

1. Access to detailed knowledge of how defects manifest themselves in the monitored data i.e. human expertise; and or
2. The availability of extensive historical data along with records of the actual defects in order to allow data mining activities to be used to extract defect knowledge for fault identification.

Fault record information is viewed by wind farm operators as commercially sensitive, and gaining access to this kind of data was not possible within the scope of this research project. Gaining access to historical data was not so problematic however and permission was obtained to access almost 2 years worth of SCADA from a wind farm sited in Scotland comprising 26 Bonus 600kW stall regulated turbines commissioned in November 1995. While this is not an extensive data set, a sufficient number of interesting events were found in order to prove the novel concepts and contributions from this research.

The lack of access to defect records and expertise in how faults manifest themselves in the data is ultimately the main challenge which has to be overcome. This highlights the author’s main reason for using anomaly detection (also referred to as novelty detection). Such an approach places minimal dependence on the requirement of having access to this experience and knowledge whilst still achieving the objective of analysing the data to identify faults as early as possible. Anomaly detection can capture a model of the ‘normal’ behaviour of the data recorded through the sensors from an item of plant. This model captures how the data evolves and changes with respect to the factors that may influence it under normal circumstances. This model of ‘normality’ allows for the detection of anomalous behaviour, even when this type of behaviour has not been seen previously highlighting its strengths for effectively meeting the problem specification.

The key research questions which arise here are:

- What techniques can be used to achieve the desired novelty detection suited for application to the wind farm SCADA in order to identify problems early on in the components of the turbine?

- Once this process of detection is attained, how can the process of automating the whole procedure of data analysis from its inception to the interpreted results output from the system be realised into an online FDS?

This thesis will aim to discuss the main issues revolving around these two questions and present a suitable solution which can be used as a framework for wind farm FDS.

1.4 Thesis Overview

This thesis is organised into a number of chapters which take the following form: chapter 2 provides a summary of wind turbine technology, the typical failures experienced by a turbine and the traditional methods of CM systems that are used to monitor the various components in a turbine. An outline of the typical maintenance schemes in place in wind farms today is also detailed. It then goes on to explain existing key contributions to wind turbine SCADA data analysis.

Chapter 3 introduces some common techniques used for industrial fault detection. Applicable techniques such as neural networks, various clustering algorithms, support vector regression and self organising maps along with their applications and suitability are explored.

Chapter 4 provides information on the notion of automating the data analysis process. Multi-agent systems are explained detailing the level of control and flexibility they offer and their advantages over other programming methods for developing an automated online FDS are discussed.

Chapter 5 then goes on to outline the process that was used to develop the novelty detection models. An explanation of the data that was used along with the training, validation and testing methods are all presented as well as case studies detailing confirmed results of a gearbox failure and other problems detected in the SCADA. It also introduces the concept of how the output of the various models can be corroborated offering a more comprehensive view of the status of the turbines in a wind farm.

Chapter 6 details the development of the multi-agent FDS, discussing its implementation, architecture, its ontology design and its application to monitor the SCADA from a complete wind farm.

Finally chapter 7 provides discussion and a critical review of the results and system developed along with possible avenues for future work.

1.5 Dissemination of Research Contributions

Throughout this research an effort has been made by the author in order to help convey the ideas and methodologies proposed in this thesis to a variety of different audiences through both peer reviewed publications and presentations. The publications made throughout the duration of the author's research are listed below:

1.5.1 Publications

Zaher A and McArthur S.D.J (2007), "A Fault Detection System for Wind Turbine Defect Identification and Diagnosis", Science, Engineering and Technology Event (SET '07), Westminster, London, March 2007 (poster)

Zaher A and McArthur S.D.J (2007), "A Multi-Agent Fault Detection System for Wind Turbine Defect Recognition and Diagnosis" Proceedings of Power Tech Conference (Power Tech 2007), Lausanne, July 2007 (conference paper) available online at: http://www.prosen.org.uk/pub/powertech07-zaher.pdf

Catterson V.M, McArthur S.D.J, Judd M.D, Zaher A., (2008), "Managing Remote Online Partial Discharge Data" IEEE Transactions on Power Delivery, Vol. 23, No.4, October 2008 (Journal paper).

Zaher A, McArthur S.D.J, Infield D.G, Patel Y., (2009), "Online Wind Turbine Fault Detection through Automated SCADA Data Analysis" 2009, Wind Energy Volume 12, Issue 6, Pages 574-593, September 2009, Online ISSN: 1099-1824, Print ISSN: 1095-4244 Copyright 2009 John Wiley & Sons, Ltd, (Journal paper).

Zaher, A, Cruden A, Booth C, Liethead W. 2009. Database Management for High Resolution Condition Monitoring of Wind Turbines. Proceedings of the 44th International Universities Power Engineering Conference (conference paper).

1.6 Chapter References

Clive P.J.M, McLaughlin D., McKenzie J.H, (2008), "Making the Most of SCADA data: wind farm performance assessment", Accessed September 2009, Available at: http://www.taplondon.co.uk/bwea30/pdf/3_Dave%20mcLaughlin-web.pdf

European Wind Energy Association (EWEA). (2009), "2008 statistics: wind now leads EU power sector", Accessed September 2009. Available at: http://www.ewea.org/fileadmin/ewea_documents/documents/statistics/2008_wind_map.pdf

Leaney V., Sharpe D., Infield D., Twidell J., (1997), "Condition Monitoring of Wind Farms using 10 Minute Average SCADA Data", Proceedings of European Wind Energy Conference, October 1997. Pp. 234-237.

McArthur S.D.J, Booth C.D, McDonald J.R, McFadyen I.T., (2005), "An agent-based anomaly detection architecture for condition monitoring", IEEE Trans. Power Systems, Volume 20, Issue 4, Nov. 2005 pp:1675 – 1682

Usama M. Fayyad, (1996), "Advances in data mining and knowledge discovery", Published by MIT press, 1st edition, 1996, ISBN 0262560976

Wind Energy Update, (2009), Weekly Intelligence Brief, "IN-DEPTH: Understanding How Much Information Can be Extracted from Routine SCADA data", Available at: http://social.windenergyupdate.com/news/depth-understanding-how-much-information-can-be-extracted-routine-scada-data Accessed (November 2009)

Author Discussions

Yusuf Patel, (2007) Author discussion. Wind farm operator

2. Wind Turbine Monitoring

2.1 The Modern Wind Turbine

To put it simply, a wind turbine generates electricity by converting the force of the wind (kinetic energy) into torque by acting on the rotor blades. It then uses this rotational torque to drive a generator to produce electrical energy. There are two main types of wind turbine designs used to carry out this process of energy conversion, namely they are the horizontal axis and vertical axis turbines. Horizontal axis wind turbines also known as (HAWT) can be considered as the prototype design for the majority of wind turbines installed in wind farms all over the world today. Designed by Johannes Juul a student of Danish origin, then known as the Gedser wind turbine (Danish wind energy association 2008) it has now come to be referred to as the 'Danish concept' due to its place of origin. An example of a typical modern turbine based on the 'Danish concept' is shown in Figure 2.1.

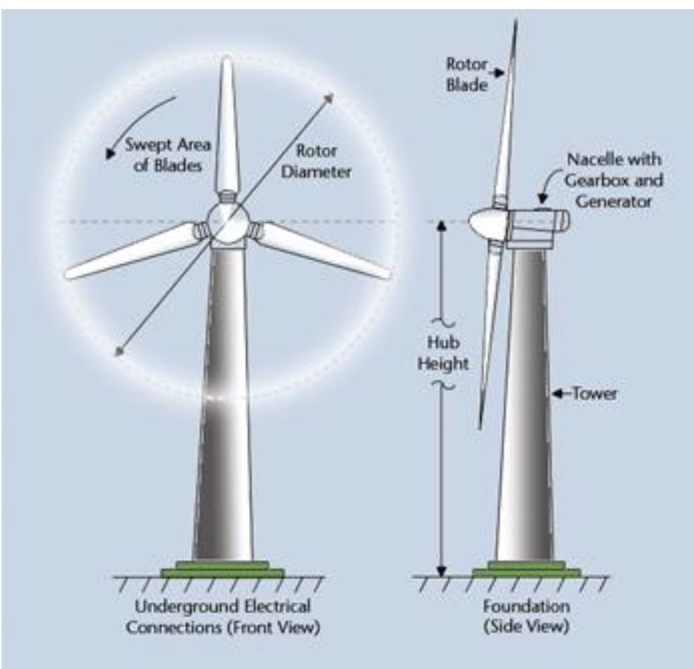

Figure 2.1: A three bladed 'Danish concept' turbine (Europa Energy Research 2010)

The modern Danish concept turbine has its 3-bladed rotor and hub placed upwind of the tower (i.e. facing the wind) in order to avoid the irregular and turbulent air current caused by the wind shade behind the tower. Electromechanical yaw systems monitor the current wind direction and rotate the turbine accordingly into the prevalent wind direction on its vertical axis in order to keep the turbine facing towards the wind.

Most modern designs yield maximum electrical energy at wind speeds of around 15 metres per second (around 30 - 33 knots) due to the fact that it is not ideal to design turbines that maximise their output at stronger winds which are too unusual and infrequent. In the case of stronger winds however, it becomes necessary that a form of power control is in place to dispose of the excess energy in the wind in order to avoid undesired damage to the turbine. Power control is achieved safely through two main methods. The first is through passive or stall control designed rotors which aerodynamically limit the lifting force acting on the rotor in excessive winds. The second method is achieved by actively varying the angle of the rotor blades to the wind using electronic or hydraulic mechanisms. These forms of power control are referred to as stall control and pitch control respectively.

An increasing number of larger wind turbines (typically 1MW and above) are developed with a modified form of power control known as the active stall power control mechanism. Technically this mechanism resembles that of pitch controlled machines since they also have pitched control blades. The machines will usually be programmed to pitch their blades much in the same way as a pitch controlled machine during periods of low wind speeds, however an important difference is in higher wind speeds when there is a risk of the generator being overloaded. Once the turbine has reached its rated power, the active stall control pitches the blades in the opposite direction from what a typical pitch controlled machine would do in order to increase the angle of attack on the blades, forcing them into a deeper stall therefore wasting the excess energy in the wind. This allows for a more accurate form of power output control which avoids the turbine overshooting the rated power in the event of a wind gust while also allowing the machine to be run almost exactly at rated power at all high wind speeds.

In the modern turbine, the generator is normally driven indirectly via a gearbox, both of which can be found in the nacelle of the turbine. The gearbox is used to step up the rotational speed of the rotor to a suitable level that can drive a relatively small light weight generator which is required in order to minimise the weight in the nacelle at the top of the tower. The gearbox must step up the rotation from tens of revolutions per minute to thousands at the output due to the relatively low number of poles found in the necessary smaller light weight generators.

The most common form of electrical configuration found in modern turbines is the doubly fed induction generator (DFIG), a form of variable slip generator. This is coupled to the electrical grid through an indirect connection (power converter interface). This allows the turbine to run at variable speed while the generator remains in synchronism with the electrical grid. There are a number of advantages associated with the ability to run a turbine at variable speed. Firstly in a pitch controlled machine the pitching mechanism is a mechanical process that has its associated response time. Having a variable slip generator means the generator can be run at half of its maximum slip when the turbine is operating near its rated power. In the event of a wind gust, the control mechanism can then increase generator slip to allow the rotor of the turbine to run at a faster rotational speed reducing stress and fatigue on the gearbox and generator while the pitching mechanism pitches the blades out of the wind to cope with the situation. Once the pitching mechanism has responded the slip can be reduced again. This process occurs in reverse if the wind speed suddenly drops. The second advantage is that reactive power can be controlled through power electronics in order to improve the power quality in the grid which is especially important if the turbine is running on a weak electrical grid.

The latest design concept in wind turbine technology is the direct drive turbine concept. These machines have no gearbox or drive train and use large synchronous generators driven directly by the rotor with the aim of circumventing the high failure rates of gearboxes in mind. Accommodating the larger diameter due to the number of required stator poles is still however an issue. Additionally, the full electrical converter necessary for such a configuration can result in a higher level of faults when compared to the induction generator in the more traditional configurations, as well as increasing the cost (Tavner et al 2006).

2.2 Component Failure Rates and Typical Downtime:

Large modern onshore wind turbine reliability is improving and the technology has matured over the years. A number of researchers have investigated the reliability of a variety of wind turbine configurations and models. Figure 2.2 shows the internals of a Danish Vestas V52-850kW wind turbine. From a condition monitoring outlook one can consider the mechanical wellbeing of all the major components in the turbine of paramount importance in order to ensure healthy operation.

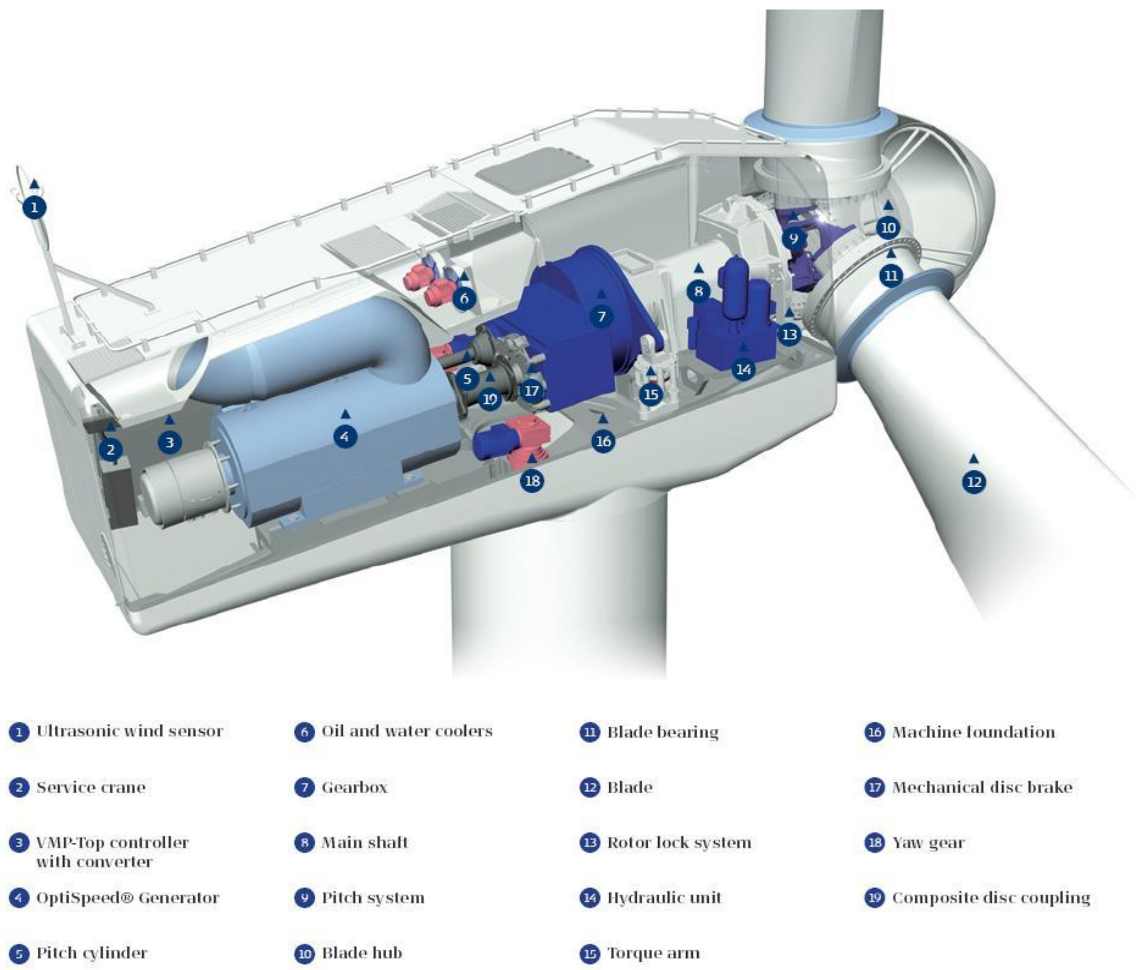

Figure 2.2: the internals of a Vestas V52-850kW turbine (Vestas 2009)

Naturally however some components are of greater significance than others mainly due to economic issues, such as their cost and replacement expenses, let alone the duration of downtime caused by their failure which will be discussed later. Consequently information available regarding the individual component failure rates is of particular interest and key to develop a good understanding of the requirements an online FDS should be tailored to. Using this information we can determine which components are of importance from an economic viewpoint and ideally should be kept in a healthy mechanical state of operation, in order to gain a better understanding of wind farm operation & maintenance (O&M).

There is a large spread of reliability figures which have been published in the literature. Some of these reliability figures are estimates based on authors opinions such as (Sayas & Allan, 1996) and others are based on expert judgement of wind farm operators (Van Bussel & Zaaijer's 2001) and (Negra 2007). The work of (Tavner et al 2006) and (Ribrant & Bertling 2007) represent more statistically plausible estimates of overall wind turbine failure rates since both of their work is based on actual wind turbine failure statistics data.

The work of (Tavner et al 2006) investigates the failure rates for the major internal components. This information is shown in Figure 2.3 below:

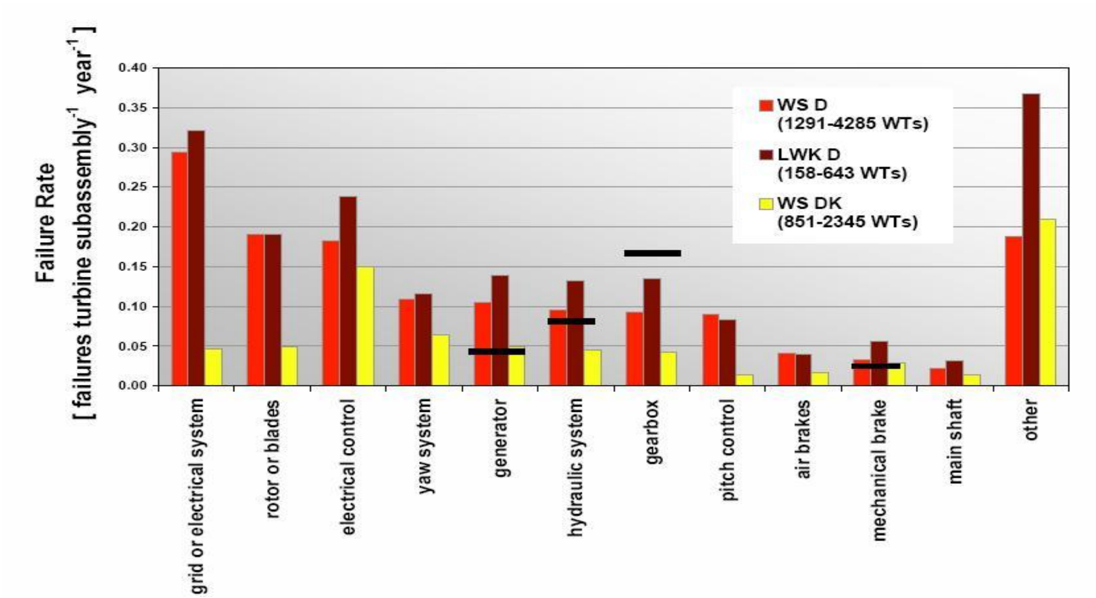

Figure 2.3: Distribution of failure rates of wind turbine components from 3 surveys (WSD, WSDK, and LWKD) averaged over 11 years from over more than 7000 turbines. (Tavner et al 2006)

Figure 2.3 shows the results from a total of three surveys. The surveys are based on data collected over a period of 11 years, from Windstats surveys in Denmark and Germany shown in figure 3 as WSDK and WSD respectively. LWKD is another survey carried out on a population of turbines installed in Schleswig Holstein in Germany also. The output of the studies shown is limited in the sense that the periods of the data collected differ for each population. WSDK was collected monthly, WSD quarterly and LWKD was collected annually. WSDK consists of a large mixed population decreasing in number from 2345 – 851 turbines of an average age of 15 years and predominantly based on stall regulated turbines. WSD is based on a larger mixed population growing in number from 1295 – 4285 turbines from a variety of turbine models with different control configurations of average age < 3 years. LWKD is based on a smaller segregated population again growing in number from 158 – 643 turbines of average age >3 years. Tavner et al refer to these limitations as having some effect on the results presented. Nevertheless their purpose is to offer an idea of the typical rates for a wide range of turbine types which proves a useful statistic to have in order that we can gain a better understanding of wind turbine O&M. From the failure rates presented in the diagram, it can be seen that the highest failure frequencies occur in: the electrical system & electric controls, rotor or blades (i.e. hub & blades), generator, yaw system, hydraulics and finally the gearbox.

The paper then goes on to detail how these failure rates reflect in the actual downtime hours caused by the specific failure modes. Downtime refers to the total time it takes from the point of component failure to when the turbine is brought back online for generation. This involves the process of diagnosing the problem (whether by manual observation of SCADA data or by a site visit) as well as the repair or replacement of the component required. In the case of more severe failures and a replacement is necessary, the time it takes to source the component as well as any special equipment needs to be considered since it can contribute significantly to the overall downtime caused. Another major factor for consideration is the appropriate weather windows that are required in order to carry out the maintenance tasks which again can often be a major constraint adding several days to turbine downtime. It is important that we understand how failure rates reflect in downtime. This is critical from a wind farm operator's perspective as it is the components that contribute significantly to a turbine's downtime that will be their main concern. This is also extremely important so that we can begin to appreciate what an operator requires from an online FDS. The downtimes of the various components identified from the study from the LWK population of turbines are shown in figure 2.4:

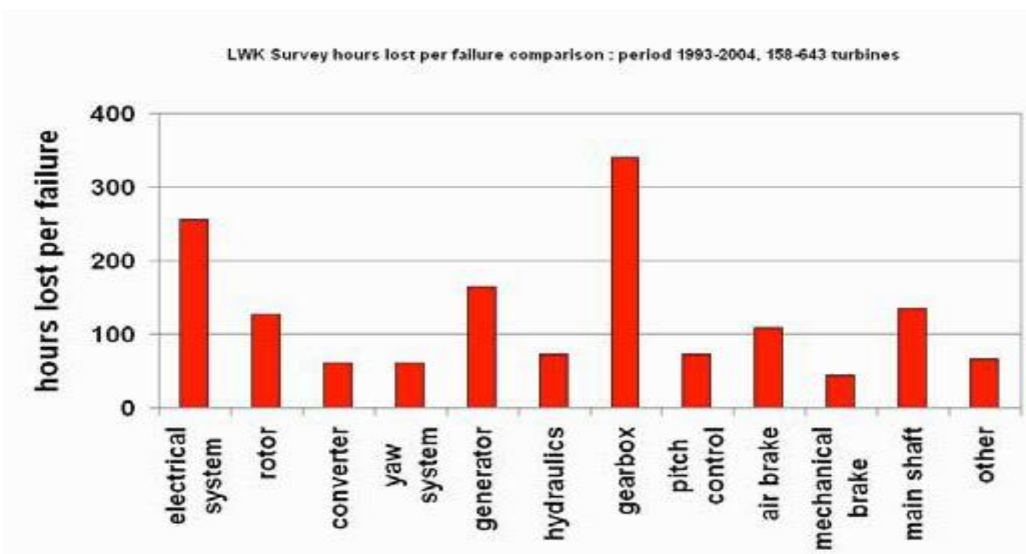

Figure 2.4: Downtime hours associated with various turbine components (Tavner et al 2006)

The various failure modes of the larger data sets in the study were not available. This is usually due to the commercial sensitivity of such data. Utility companies are not comfortable with releasing this information stemming mainly from the competitive nature of the market. Figure 4 shows the downtime in hours for the various components from the LWK study. From figure 2.4 it can be seen that the major downtime contributors in order of most significant first are the gearbox, electrical system, generator, main shaft and the rotor. The results from the studies carried out by (Ribrant & Bertling 2007) on Swedish wind power plants during 1997 – 2005 shows the most significant downtime contributors listed in the following order: gearbox, control system, electric system, rotor and the generator. Since downtime is affected by many factors, it is inevitable that various studies might vary considerably and possibly result in a different order of significance. Nevertheless both studies show that the components listed are the major contributors to overall wind turbine downtime.

2.3 Breakdown, Periodic & Condition Based Maintenance Policies

Maintenance policies are employed in most major power plants which contain high value machinery and equipment. These policies define how the overall operation and maintenance of the plant is handled and how failures or any problems that may arise are dealt with. Wind farms typically employ the same maintenance policies as other forms of power plant such as coal fired or oil and gas thermal plants. Maintenance can take the form of one of three different policies. These policies are referred to as: breakdown maintenance, periodic maintenance, and condition based maintenance.

The breakdown policy as the name suggests involves running equipment until it breaks down or ceases to function. Maintenance is then carried out at this point to return the equipment to a functioning state. This form of policy is only employed in plants whose failure is not seen as critical from both an economical and safety viewpoint. For most power systems assets this policy is not typically used for any items of high value (Schneider et al 2006).

Periodic maintenance is the predominant policy utilised in a large number of industries (Schneider et al 2006) e.g. transformers in high voltage electricity grids and the automotive industry. This policy involves a time based maintenance schedule where items of plant are taken offline and undergo inspection and maintenance actions at set intervals. The main reasons behind its popularity are the simplicity of its implementation. Some wind farm operators employ this policy mainly for this reason and perceive it as the logical choice to onshore wind farm maintenance based on the comfort associated with applying a scheme which they are familiar with. The main disadvantage with periodic maintenance however, is that it does not offer the most economic solution to O&M especially for offshore wind farms. The reason behind this is that it fails to take into consideration the operational state of the machinery as a deciding factor to performing the maintenance procedures. For example a turbine that does not require maintenance will still be taken offline, checked and then brought back online. The result of this is that most of the maintenance performed is unnecessary quite often leading to an adverse effect of adding substantial wear to the machinery. From an economic perspective this is not seen as a cost effective approach given that the time it takes to carry out such procedures will result in substantial downtime and cost, which otherwise, could have been avoided had the turbine been kept online generating more revenue. One can perceive the extent of the impact being further augmented as this procedure is replicated over the scale of the complete wind farm. Moreover, another negative implication of periodic maintenance can be perceived in the case where a turbine requires attention between maintenance schedules. For example if an incipient problem develops in the components, failure to respond to such an event within an appropriate time scale will often lead to a greater possibility of actual failure in the component. This can often occur before maintenance is next scheduled, meaning an outage which possibly could have been avoided results due to the late identification of the incipient problem. It is important to note at this point that both breakdown and periodic maintenance policies do not make use of Condition Monitoring (CM) systems.

Condition based maintenance (CBM) on the other hand attempts to overcome the deficiencies of periodic maintenance through the use of CM systems. A number of issues can however impact the degree of success of a CBM policy. The following section goes on to look at the concept of CBM, its implications and some of the specific problems that often arise with the implementation of this approach to maintenance.

2.3.1 The Concept of Condition Based Maintenance and Its Issues

The principle of CBM is that maintenance is carried out only when needed, therefore optimising the process of O&M. This process can be achieved through manual visual inspections of the machinery or through the use of condition based data which is acquired through CM systems that provide real time information regarding the state of the machinery. The availability of this kind of information theoretically gives operators the advantage of being able to schedule maintenance actions in an optimal manner taking into consideration the resources available and the associated costs of carrying out the work. In this way healthy turbines which do not require attention can be left to operate generating revenue while the resources and effort of personnel can be focused on the turbines with problems detected.

While the initial perceptions of such an approach would typically seem very positive, a number of issues arise with the implementation of a CBM policy which currently affects the success of its operation, therefore making its employment less encouraging. The first is the issue of usability, the aforementioned problems of information overload and data interpretation associated with CM systems can hinder the value of CM making the argument for their case unclear. (McMillan et al 2007) raise a number of points in their research which evaluates the techno-economic issues associated with the application of CM for wind turbines and attempts to quantify the benefit of its use. The negative aspects which the authors mention stem mainly from the economical perspective and can be summarised as follows:

1. The value of using CM systems for wind turbine monitoring is currently unclear due to the difficulty in accurately quantifying the benefit. This is even more

apparent in onshore wind which is considered a low-profit margin plant by operators. This therefore makes it difficult to justify an economic argument.

2. The initial investment in monitoring tools is high. The costs associated with the SCADA system, the sensors required and the expert personnel needed to constantly monitor the output data and extract meaningful information from the system, all add up resulting in a substantial cost which must be economically justifiable in the long run. Such costs are justifiable for single large rotating plant rated at hundreds of MW; however, the case is not quite as clear for wind turbines.

3. The final issue is a matter of accuracy regarding the CM systems themselves. False positives and negatives (false alarms and failure of an alarm activating respectively) can lead to unnecessary shutdowns which can neutralise the added benefits of their use.

While the economic points mentioned are important to consider, they would be less of a concern to a wind farm operator if the third point mentioned regarding the accuracy of the systems was less of an issue i.e. the CM systems actually provide accurate and useful output. Operators could pragmatically assess and identify if the use of a CBM policy would be beneficial for them in their wind farms. As an example, it has been said that the CM system at Horns Rev wind farm based in Denmark has caused more downtime than it has saved due to false positives. This is clearly undesirable; however it is aspects like these that further rationalize the reasons for looking into online FDS which can automate the analysis and interpretation process in a robust and effective manner consequently helping to prove their value. What is important is that the operators are aided by the output rather than being confused, which can lead to adverse decisions being made regarding maintenance. The research presented in this thesis attempts to address these issues through the development of accurate normal behaviour models embedded in a multi-agent framework which allows more robust CM platforms to be built. This will be discussed in greater detail in a later chapter.

Despite these issues surrounding wind turbine CM, it has been extensively researched over the years and is still an important topic of which a large range of information can be found in the literature. The next section provides an overview of the various monitoring techniques detailing the key research contributions that have influenced this research.

2.4 Review of Wind Turbine Condition Monitoring:

There are three main groups in which the literature regarding wind turbine CM can be split. The first is the literature which focuses on the techno-economic analysis of wind turbine CM systems and the reliability of the turbines. The important findings and how they tie in with the research presented in this thesis has already been discussed in section 2.2 of this chapter. The second section in the literature is devoted to the sensor technologies / CM approaches that are applicable to wind turbine monitoring, and finally the third section encompasses the data analysis and processing algorithms that are focused at interpreting the data generated. Naturally there is an overlap between the second and the third groups mentioned as the relation between the data acquisition and data processing technologies are closely coupled. A brief overview of the hardware and sensor technologies will be discussed first in the next section, where as the section which follows will present the advances and contributions in research tailored towards the CM of wind turbines including a section devoted to the key existing research in data analysis and interpretation influencing the anomaly detection models / system presented in this thesis.

2.4.1 Sensor Technologies Applicable to Wind Turbines:

A number of papers have been published which provide a review of the majority of the monitoring techniques applicable to wind turbines. The number and variety of sensor technologies applicable for turbine monitoring is somewhat diverse. Modern turbines typically utilise a number of different types of sensor to monitor the various components. The main monitoring technologies found in the literature concerning wind turbines are vibration, temperature, oil analysis, strain, pressure, acoustic, current transformer (CT) and voltage transformer (VT) sensors. A particularly comprehensive and insightful review of the key techniques as well as the techniques being researched was carried out by (Hameed et al 2007), (Verbruggen T.W. 2003) and (Lu et al 2009). The papers summarise the most common technologies applied indicating their suitability for the various components. A summary of their findings is provided in the following sub-sections.

Vibration Analysis & Acoustic Emission:

Vibration analysis is the most widely applied technology in condition monitoring for rotating machinery. Different types of vibration sensor are required to measure the different frequency ranges. Position transducers are used to measure low frequency vibrations, velocity sensors are for mid range and accelerometers are used to measure the high frequency range. Spectral emitted energy (SPEE) sensors are used for very high frequency ranges such as acoustic vibrations.

Acoustic monitoring has a strong similarity with vibration monitoring however the principal difference is in the higher orders of frequencies that they are typically applied to. Vibration sensors register local motion on the component while acoustic sensors “listen” for vibration; therefore they are often capable of giving an indication of defects in their developing stages.

Most incipient problems which occur in rotating mechanical systems will innately result in vibrations. This highlights the main reasons of the technology being widely applicable and well suited to detecting problems which develop in a number of components.
With respect to the monitoring of wind turbines, vibration monitoring has been found to be used for detecting gearbox bearing and gear wheel damage, the main bearing associated with the main shaft of the turbine, torsion and oscillation of the main tower and in some cases acoustic vibrations in the blades through the use of the mentioned SPEE very high frequency sensors.

While vibration monitoring has been extensively used in other traditional forms of rotating mechanical machinery, its application and working methods for wind turbines differs with respect to the low rotational speeds as well as the dynamic load characteristics associated with them. In other applications loads and speeds are characteristically constant for the most part which simplifies the required signal analysis. There is limited experience for dynamic applications especially as complex as wind turbines.

Another difference brought to light by (Hameed et al 2007) is the high investment costs in relation to production losses. For more traditional types of power plant, the investment costs in CM equipment are typically covered by the benefit of their utilisation which is seen through a reduction in production losses. For wind turbines, especially onshore, the production losses are already relatively low meaning the benefit gained is not so evident.

Oil Analysis:

Oil analysis is usually executed offline through taking samples and assessing their quality after specified lengths of operation. The application of online sensors is increasingly being used however in order to safeguard the quality of the lubrication oil internal to the components which can easily become contaminated by particles, dust and moisture. The prices of such sensors are now available at acceptable price points which allow for the detection of moisture and particle counting while the machine is running. The size and count of particles found in the oil gives an indication of the rate of wear in the components.

Online oil analysis is typically used in the gearbox of the turbine. It can give an indication of gear tooth damage as well as any contamination that may occur in the lubrication oil. Contamination of the lubrication oil can significantly contribute to wear in the gearbox internals. Moisture in the oil can also dramatically reduce the efficacy of its lubrication properties again leading to an increase in wear. A direct relationship can be seen between the working lifetime of the gearbox and the size and number of particles found in the oil. Optimal oil management can effectively reduce costs with the respective damage and oil replacement that may well be necessary, and therefore, can lead to a reduction in downtime. (Quail & Hamilton 2010) provide a comprehensive analysis of the current on-line / in-line oil analysis techniques with regards to wind turbine gearbox CM. Quail and Hamilton recommend the use of a combination of sensors that could be used to analyse different characteristics of the oil stating that individually the information resulting from the sensors would not be highly accurate, however their integration into one system is expected to obtain greater accuracy. Quail and Hamilton state that "since the elemental composition" of wind turbine gearboxes is known, a list of the potential particulates can be established allowing sensors to be developed to that can selectively identify each of them. Some of the suitable techniques mentioned for oil analysis in terms of cost, size, accuracy and development are: online ferrography, selective fluorescence spectroscopy, sold state viscometers and photo-acoustic spectroscopy.

Temperature Monitoring

Temperature monitoring is frequently used in wind turbines (Lu et al 2009). Excessive friction and wear between metal components in mechanical machinery will generally result in an undesirable and disproportionate amount of heat being given off. This heat will be in excess of the normal operating temperatures which result from the typical amount of stress and friction levels which occur under normal healthy operation. Monitoring the temperature of components can therefore also give an indication of excessive grinding and wear. With respect to turbines, parameters such as the gearbox oil, gearbox bearing, generator winding and ambient nacelle temperature as well as the temperature of other specific components depending on the sophistication of the CM system are typically monitored (Yusuf Patel 2007).

Strain & Pressure Monitoring:

Strain and pressure sensors are a common form of monitoring technique. Strain gauges attach to an object's surface, and as the object is deformed, the properties of the strain gauge are also changed. They are typically made from flexible electrically conducting material. Stretching or deforming an electrical conductor within its limits such that it does not break or permanently deform will change its electrical resistance. This resulting change in electrical resistance allows the level of stress on the object to be measured respectively. There has been interest in using strain gauges for monitoring the stress levels on the blades of wind turbines however methods are not yet well developed. The cost of implementing a robust sensor system suitable for the large scale of the turbine blades is often too high to be justified and is largely the reason that this type of monitoring is not common place. Optical fibre and acoustic methods are currently under research as they are applicable for detecting failures in the blades too.

Pressure sensors are used to measure the pressure of gasses or liquids, and as such are typically used in the hydraulic systems installed in the turbine. The hydraulics are primarily used in the pitch control systems of the blades. A failure in the pitch control system can lead to increased mechanical stress which in turn would lead to a reduced energy yield and therefore its monitoring is critical.

This section has summarised the main sensor technologies that are applicable to wind turbine monitoring. Research in wind turbine monitoring is an ongoing process which is still in its early stages. An important point found during the review process was that there is still no definitive standard for wind turbine CM systems and the majority of the systems applied in industry monitor different parameters using various techniques. This aspect is reviewed in the next section.

2.4.2 Review of Research Projects Undertaken in Wind Turbine CM Systems

A number of research projects undertaken have looked at attempting to define the technology and methods that should be used for particular aspects of wind turbine monitoring (one of which the author is also involved in (Zaher & Cruden et al 2009)). This is understandable, given that this increase in wind turbine technology is relatively recent, and yet to mature. The literature available on these research projects is now reviewed.

(Wilkinson et al 2006) explore the notion of building and installing a CM system for offshore wind farms. Their work looks at designing a CM scheme using reliability based information of existing onshore turbines through an analysis of the failure modes, and the consequent effects they have on the turbines. They investigate the reliability of the different turbine design concepts in particular the differences between direct and indirect drive machines. The results of their reliability analysis showed that indirect drive machines based on the typical Danish concept turbine appear more reliable despite the fact that they incorporate gearboxes. Direct drive machine reliability issues are dominated by electrical sub-assembly failures. The findings show that the slow speed synchronous generators exhibit a failure rate of double that of the higher speed indirect drive machines. The reason given is stated to be possibly linked with the larger diameter of the generator leading to sealing and insulation problems. The gearbox for indirect drive machines display higher mean time to repair (MTTR) however and so are deemed as being the major problem for the indirect drive turbine concept.

A failure mode effects analysis (FMEA) was carried out for each of the failure modes identified and a numerical value reflecting the risk of each mode was calculated by a subjective process of weighting the significance of the severity, frequency and the probability of detection associated with each failure mode. The results of this process are said to be incorporated into the CM test rig which consisted of a DC motor, a two stage gearbox and generator. Transducers which measure shaft speed, shaft torque and shaft vibration were fitted. The test rig was driven using real wind data for the purpose of simulating an actual wind turbine in a wind farm. The paper claims that the wind model driving the rotation of the drive train excites an array of harmonics, enabling the natural frequency of the main components (in the drive train) and faults in the generator to be detected. The techniques were not tested on a real turbine, however it is claimed that applying signal processing reduced the noise ratio which the authors suggest can be applied to monitoring the generator of a full size wind turbine.

(Amirat et al 2007) review well established condition monitoring techniques developed for induction motors. The emphasis of the work revolves around faults and detection methods that can be monitored using the wind turbine generator (DFIG) terminals. The authors comment on the fact that it seems possible to detect drive train faults through the terminals of the associated generator through their review of the literature. Imbalances and defects in small wind turbine blades can also be diagnosed by measuring the power spectrum density at the generator terminals. They state that wind turbine generator operations are predominantly transient and therefore the use of non-stationary techniques is required for fault detection.

(Popa & Jensen et al 2003) carried out an experimental investigation on the incipient fault detection methods found in the literature suitably adapted for use in wind turbine systems using the DFIG electrical configuration. Three main experiments were reported to have been carried out namely: one to detect stator phase unbalance, one for rotor phase unbalance and finally one for turn-to-turn faults in order to study the electrical behaviour of the DFIG. The experimental system was developed as a model of a wind turbine which consisted of a wound rotor DFIG with slip rings driven by a gearbox. An induction machine CM system was developed measuring the stator and rotor currents, stator voltages, rotor speed, and temperature of the windings. The faults were simulated using a variety of methods. The paper reports that the simulated faults can be detected through time and frequency domain analysis and that the frequency spectrum of the stator and rotor line currents was found to give the best results.

A recent trend emerging in wind turbine CM research is the focus on methods for rotor blade monitoring. (Burnham & Pierce 2007), (Dutton et al 2003) and (Rumsey et al 2008) all consider acoustic emission technology as a novel approach for blade monitoring. Each of these streams of research is at the early stages of development mainly due to the expense of adopting such a technique for blade monitoring. (Dutton et al 2003) developed interesting pattern recognition damage classification software based on cluster analysis to process the acoustic emission data. One particular cluster was found to emerge frequently in the data as a result of the simulated stress tests. It was identified to occur across a number of blade tests and originated only in blades with developing damage. The authors state that the pattern recognition software could form the centre of a wind farm acoustic emission monitoring system due to its potential to learn new damage cluster signatures. (Burnham & Pierce 2007) discuss the limiting aspects of acoustic based techniques based on the physics of acoustic wave propagation in typical structured components. They carry out a comparison between acoustic technology approaches and the conventional active ultrasound approach. (Rumsey et al 2008) implement a structural health monitoring system using an off the shelf acoustic emission Non-Destructive Testing (NDT) system. The experiment was carried out on a 9 meter glass-epoxy and carbon-epoxy wind turbine blade. The blade failed in fatigue after 4 million load cycles on the tensile side of the blade. The authors state that the system detected significant acoustic emission events early in the test providing a very informative diagnostic tool.

Other wind turbine rotating elements such as gearboxes and bearings seem to receive less attention in the literature than rotor blades. This may be due to the fact that algorithms for condition estimation of rotating elements have already been extensively researched. (Carden & Fanning 2004) claim that the majority of the research in this area however is for systems with operation at near constant rotational speed such as aero-engines and steam turbines. (Becker & Poste 2006) however, state that the variable speed operation of a wind turbine dramatically limits the efficacy of these existing algorithms.

While there are quite a number of techniques which are applicable to wind turbine CM it can be seen that much of the proposed research is at a low level of technical maturity with no definitive CM standard. A large number of the research projects are still investigating various instrumentation setups and how they can be used to portray the condition of the machines, which indicates that wind turbine CM is still in its early stages. One proactive step for CM in the German wind energy market is the introduction of a review clause in 2002 (see Becker & Poste 2006) by German insurance companies due to the high incidence of gearbox failures and damage found to occur after only a few thousand hours of operation. As a cost deterrent to encourage an improvement in machine operating life, insurers require that all roller bearings in a drive train be replaced after either 40,000 operating hours or five years (whichever is earlier), unless an appropriate CM system certified by them is installed. This approach rewards wind farm owners for the introducing CM systems into their wind farms encouraging the industry as a whole towards the notion of CBM policies.

The next section goes on to look at the research devoted to understanding which factors affect wind turbine performance and component degradation and how it is portrayed in the data collected from CM systems.

2.4.3 Review of Wind Turbine Condition Monitoring Data Analysis & Interpretation Systems

This section covers the research found in the literature that is more closely related to the data analysis and interpretation aspect of wind turbine CM. This area of literature is more closely related to the research presented in this thesis, particularly the SCADA data analysis methods since they are directly applicable to the systems that are already integrated into the turbine at the time of manufacture. The research efforts devoted to this area of the literature are therefore seen to be of paramount important since they present less concerns and issues in terms of the integration and cost of having to retrofit new CM systems to the turbine. This makes them a convenient and practical option for wind farm operators. This however does not disregard the research into new sensor and hardware technologies as data analysis and interpretation is based primarily on the output of such systems.

The Condition Monitoring for Offshore Wind farms (CONMOW) was a collaborative project carried out by a number of large and well established institutes in wind turbine research (see Wiggelinkhuizen et al 2007). Its purpose was to investigate the notion of a cost effective integral CM system for wind turbine monitoring with a specific focus on the development of data analysis algorithms. These algorithms were to be integrated in the SCADA systems to produce accurate information to aid O&M while also attempting to lower the cost of CM systems. The authors state that, at the time of drafting the state of the art CM techniques report, no successful applications of wind turbine CM examples were found in the literature. A number of turbines were instrumented, monitoring the drive train, gearbox and bearings using different vibration measurement systems. Load measurement systems for the blades and tower as well as weather data using the wind farm met mast and the operational SCADA data available from the turbines were also all used for the project.

The interesting points noted from this project was its tendency towards data analysis, particularly the SCADA data analysis, since it is in line with the ideology of the research carried out in this thesis. The authors explore the idea of 'normal' behaviour modelling for the generator bearing temperature SCADA parameter using a combination of statistical techniques. The model is built as a function of power output vs. generator bearing temperature which has been linearly corrected for ambient temperature over a small rotational speed range. The data is averaged and binned over 0.1m/s widths. The authors make use of the 'normal' behaviour model as a means to detect an increase in bearing temperature which could arise due to bearing wear or possibly shaft misalignment. In practice, the authors state that monitoring the real-time trend of this function allows comparisons to be made with the 'normal' bounds captured through historical data. Observations consistently above the maximum levels set by the bounds of the models can therefore easily be spotted allowing problems that manifest themselves in the generator bearing to be detected. Another model of ten minute averaged nacelle vibrations plotted as a function of the square of the wind speed was also developed in an attempt to detect abnormal vibration levels in the nacelle. Again the authors give examples of the kinds of abnormal behaviour that might be detected however they also state that the effectiveness of both the models developed could not be demonstrated due to no major failures occurring during the time period the turbines were instrumented for the project and hence the data sets only provided normal operation which conformed well with the models built. The majority of the data analysis therefore focused on fault simulations.

The objectives of the project were ambitious and very positive, regrettably however the project was hampered by mainly non-technical issues which led to less data being collected. Time was an issue meaning no online oil monitoring could be installed as well as no experiments with the SCADA processing algorithms were carried out resulting in little novel output from this work.

(Caselitz & Giebhardt 2005) developed a number of CM techniques specifically designed for the rotor. They summarise the rotor faults into three main types, blade surface roughness, rotor mass imbalance and aerodynamic asymmetries. The authors then go on to describe their methods of detecting each of these problems. They utilise statistic evaluation of the wind speed and power output of a wind turbine to monitor the overall rotor performance but more specifically for the detection of increased surface blade roughness. The process involves using data averaged over 5 minutes for both the wind speed and the power output binned into wind speed groups of 0.1m/s width. The associated average power output for each wind speed bin is calculated resulting in a power curve for the turbine which can be used to establish the normal bounds of operation. Alarm limits are also calculated using the standard deviation of the power values grouped in each wind speed class. Deviations from this learned model can then be flagged as alarms by the CM system. The model attempts to take into consideration fluctuations caused by external factors such as gusting by only flagging an alarm if a certain number (defined by the programmer) of consecutive readings fall out with the calculated bounds. However no case studies of the algorithm detecting failures are shown. For both rotor imbalances and aerodynamic asymmetries, the authors use vibration monitoring and spectral analysis to detect any manifesting problems

(Hart et al 1997), (Li et al 2001) and (Singh et al 2007) approach the problem of normal power models using artificial Neural Networks (NN) as a tool for nonlinear function estimation in an attempt to capture the complex relationship between the recorded wind speed and the actual power output of the turbine.

(Hart et al 1997) were the first to utilise this approach as part of the new and renewable energy programme managed by the energy technology support unit (ETSU). The approach seeks to use the neural network for the purposes of power performance analysis by training the network to output the averaged power generated by the turbine based on relevant SCADA parameters being fed as inputs to the network. In order to detect power performance anomalies, the network was initially trained on a data set comprising of ‘good’ turbine performance. The main difficulty with this approach is that it requires a data set which one can assume does not contain any defects or fault conditions. In order to counter this, the authors claim that by performing clustering analysis on the data it may be possible to detect anomalous values in the data set such as obvious outliers which could potentially reduce successful training. In practice however no such pre-processing was carried out before the training process. The input parameters used on the network were mean wind speed, mean shaft speed and mean nacelle direction to provide the associated output of mean active power. A second attempt at retraining a model was carried out after observing large errors in the estimation of power output making it difficult to utilise the model to track genuine subtle changes in turbine performance. In an attempt to reduce the effect of erroneous readings made by the anemometer the second model developed was trained using nacelle direction and the measured power output of adjacent turbines only. The authors claim that by doing so this removes the requirement for accurate wind speed measurements. In order to ensure the precision of this approach it was necessary to utilise data from adjacent turbines which were not subject to any operating restrictions, this restricted the amount of available data. When blind testing the models developed, power performance faults had to be simulated in the form of step changes, ramps, sinusoids and random spikes. The authors summarise that errors of power performance consistently greater than 4.5kW and nacelle angle orientation errors greater than 15˚ should be detectable. However they stress a shortcoming of their research in that their results are based on relatively few tests and that there is a trade off between the sensitivity of detection and the speed at which the system is able to detect a particular fault.

(Li et al 2001) examine and compare statistical based regression models and NNs for the same purpose of power performance modelling. The models developed however, are intended to be used for power generation forecasting only, and not the detection of turbine performance degradation. The input parameters used for the models built are averaged wind speed measurements and directions taken at a height of 40m, while the respective output is the estimated power generated. The regression models are shown to be function dependent where as the NN model obtains its function estimation through the typical training procedure using pre-processed data sets. The authors apply this pre-processing to speed up the training process and improve the outcome of the learned model. They apply normalisation to the wind speed measurements from a range of 0 – 50mph to a range of 0-4. Secondly they apply a transform function on wind direction from 0˚ - 360˚ into a more limited range which is not specified. The regression models are found to be more difficult to develop due to their requirement of defining an explicit function, while the NNs have a fairly simple training process which is constrained by the available data and consequently the effectiveness of the training algorithm on the data set. The outcome of the comparison of both models shows that the NN outperforms the regression model in its power estimation capability. While the authors do not use all of the following parameters as inputs to their models they do state that wind power generation can also be affected by other factors such as air density, the vertical wind profile, season and the time of day. Under the complicated influence of these numerous factors selection of an appropriate function for a regression model would be extremely difficult. This gives the NN the added advantage of practicality in implementation. Again no details of actual or even simulated faults are mentioned. The models built seem to be tested only on trying to replicate the normal operation of a turbine.

(Singh et al 2007) are the most recent group of researchers to utilise the NN approach for wind turbine power estimation again for power generation forecasting. They mention the various factors aside from the obvious wind speed and direction which can affect turbine power output. Factors such as air density, topography of the site such as hills, mountains etc which can cause the wind profile to deviate from the ideal case are noted. However since accurate measurement of wind speed is impractical and expensive and subject to topographical constraints the use of similar inputs is used to train a NN. A number of interesting training differences are used. Pre-processing of the input data is made use of to compress the input patterns in order to enable the NN to learn faster. More interestingly however is the use of pre-processing on the output data which is the generated power. The NN is trained to produce an output of the output power ratio (generated power against rated power). This can be easily converted to the actual power if required at any time. The authors do not explore the effect this has over modelling the generated power output directly. The paper concludes with a comparison between the traditional methods used for power estimation using the manufacturer's power curve based simply on a function of wind speed and the NN approach. The results show that the NN offered a much more accurate estimation (closer to the actual generated power) over a monthly period than the traditional method which produced a far more optimistic but considerably inaccurate estimation.

The work of (Garcia et al 2006) focuses on the development of a predictive maintenance support system which attempts to incorporate the integration of every task involved in a formal predictive maintenance strategy, from the detection of incipient faults to the actual scheduling of maintenance taking into consideration technical and economic criteria. The authors' state that the system developed, named SIMAP, is a general tool oriented to the diagnosis and maintenance of industrial processes. The system is split into a number of components: a fault detection module based on normal behaviour modelling utilising NN's tailored towards the gearbox of a wind turbine; a diagnosis module based on a simple fuzzy expert system consisting of three main rules; and, an automated maintenance scheduling calendar. The rules within the expert system as well as the suggested maintenance effectiveness metrics used appear to be highly subjective and turbine specific to be generalised. The result of the NN normal behaviour model shows its capability in detecting a gearbox fault 2 days before the actual failure which is an interesting and positive result. The approach used by the authors in this study is the closest piece of work carried out in the literature to date which relates to the study presented in this thesis. The main drawback of the system developed by the authors in this paper however is how they detach the operator's control by attempting to replicate the judgements that should typically be carried out by the operator. This is achieved using a series of NN and fuzzy estimation models which attempt to forecast the time to failure of the gearbox by utilising a limited set of turbine specific historic failures. This casts some doubt on the re-usability of the approach and the yielded results. In contrast, the notion behind the system developed in this thesis is to aid the operator in their decision making process by informing them of events that are important to them. In this way they can make the decision based on the evidence supplied by the system, rather than being detached from the decision making process completely.

(Hawker et al 2007) a group of UK based consultants describe a SCADA data based downtime analysis tool which attempts to improve wind farm operator decision making. The system makes use of 10 minute operational data, alarm event logs, technician reports and reference anemometry. Initially periods of downtime are identified from the operational data. Environmental conditions are then matched to the associated period of downtime using information reported in the alarm logs detailing wind speed, temperature or known grid status. Finally using a case base where downtimes have been previously analysed and classed by a human operator, the system makes use of pattern matching algorithms to match the frequency and duration of the specific alarms reported in the data. Finally the operator post processes the output of the system to cross check it against all available data and alter it accordingly to any technician notes from repairs or maintenance. While such a tool can be useful for learning the cause behind specific turbine failures, it does not provide a means to identifying failures before they occur. Therefore from the perspective of implementing a CBM scheme, such a system would not aid an operator in their decision making regarding maintenance of their farm.

(Christensen & Anderson 2009) present a remote condition monitoring system which uses the vibro remote CM system of Bruel & Kjaer. The system monitors the vibration levels across the main bearing, the gearbox and the generator. In order to provide a reliable measurement of vibration severity, the vibration levels are trended into active power bins to accommodate the vibration response of the machine components as it varies with external loading. A set of alarms are then defined for each power bin group indicating a severity level. Two examples of real bearing faults as well as one instance of shaft misalignment detected by the vibration system were presented by Christensen & Anderson, proving the success of their approach. An alarm processing program filters the multitude of alarms that can be generated by all of the turbines and transforms them into a single alarm for each part of the turbine for every turbine monitored. This severity level produced by the system is then verified by diagnostic experts.

(Gray & Watson 2009) present a methodology for damage calculation applied to a typical 3 stage wind turbine gearbox design based on the concept of physics of failure. Gray & Watson state that damage is generally accumulated due to an "irreversible change that takes place in the microstructure of a component subjected to certain loading or environmental conditions". A detailed assessment of the system and its potential failure modes as well as the definition of acting loads and modelling of the damage kinetics was carried out to identify critical operating conditions. Damage models were built on the basis of this assessment that were then used to estimate the accumulated bearing damage for a number of wind turbines. The proposed methodology by Gray & Watson is extremely positive. It is clear however that the methodology requires an in depth understanding of the dynamics of the gearbox under all kinds of conditions and loads. Gray & Watson state themselves that the development of highly accurate damage models should generally be viewed as an iterative long term process where "all failure modes that have a significant influence on the wind turbine reliability" need to be understood in full detail to allow for accurate model development. A possible limitation of the proposed methodology however when compared to nonlinear function capture methods such as NNs is the difficulty involved and effort required to undergo the analysis process for different types of gearbox models. Different gearboxes will undoubtedly have different operational characteristics which will require yet another full analysis to understand its operation. Non-linear function capture methods will only require simple retraining procedure which will automatically fit models to the relations found in the data.

(Garlick et al 2009) present a model based approach for the CM of wind turbine generator bearings. They make use of a least squares algorithm coupled with an AutoRegresive with eXogeneous input (ARX) model structure applied to raw SCADA data. Before training the model on fault free data, the data undergoes a correlation check where the generator winding temperature parameter was found to exhibit the highest correlation with generator bearing temperature. Using this outcome only the generator winding temperature parameter was used as an input to the model which would then go on to provide an estimation of the generator bearing temperature. Garlick et al state that the results obtained show that significant discrepancies between model estimates and actual bearing temperature can be used to identify a problem exists in the bearing. As a comparison of the NN models developed for this thesis published in (Zaher et al 2009) however the ARX model does not perform as well. The NN model conveniently allows for multi-variable input without complicating the model development process anymore than for a single input variable model allowing higher levels of accuracy to be obtained. The ARX model development process however becomes far more complex as described in (Li et al 2001) and therefore is more difficult to achieve the same level of accuracy. Another comparison in the model development process is that the NN model automatically fits itself to the data supplied in the training data sets. With the ARX model, as shown in the paper by (Garlick et al 2009), the structure of the equation used to fit the model to the data must be chosen by the developer. (Li et al 2001) also state that this can introduce a potential difficulty in the model development process since an incorrect model function will not be capable of fitting itself to the supplied data. This is especially true if the relationship being modelled is not well understood also. The results presented by (Li et al 2001) in their comparison of NN and ARX regression models shows that the NN outperforms the ARX model.

Finally the work of (Leaney et al 1999) describes the development of a methodology to analyse wind farm SCADA data with the aim of enhancing early detection of problems which develop in wind turbines. The authors comment on the fact that while there is a lot of ongoing research that makes use of vibration and high frequency power output measurements to detect specific failures, they typically require a large amount of additional monitoring equipment and sophisticated data analysis. They also point out that there is a lack of techniques that are able to predict when components are likely to wear out and fail.

Their approach makes use of basic 10 minute averaged SCADA parameters which are widely available across most turbines to highlight general and specific turbine problems. The concept behind their approach is that problems in the turbine can be detected by general turbine performance degradation. One of the methods developed is a long term wind speed prediction model for all wind directions which is used as an input into a wind farm performance prediction package to measure long term turbine performance. The wind speed prediction model uses one year's worth of meteorological data so that long term wind speeds for all wind directions can be deduced. The meteorological and external physical factors such as roughness of the site terrain, obstacles, topography, location of the turbines, wake effects and each turbine's associated manufacturer power curve are all input into the wind farm performance package along with wind speed prediction model. This information enables a prediction for each of the individual turbines and the whole wind farm.

An interesting aspect highlighted in the paper is the trends which were observed in component performance prior to failure. Power fluctuation data analysis based on experimental data by calculating power turbulence and power variability was used. The authors state that noticeable changes in the power variability curve (plotted against wind speed) were observed prior to a gearbox bearing failure. A small decrease at the same time was also observed in the power curve. The results of this analysis formed the basis of the authors going on to develop normal behaviour models for anemometer degradation, wind direction misalignment, pitch settings, general power deterioration, gearbox faults and clutch faults. The respective data sets used to build the models were mainly based on averaged data over a period of time. What is interesting is the normalising process that is applied to the data sets to build the normal behaviour models. Two techniques are compared, one based on linear regression and the other is known as Kriging, a linear least-squares based interpolation algorithm. The authors demonstrate that Kriging gives lower errors for long term estimates (i.e. longer than 3 months). They state that any departure in the data sets outside of the error range should trigger an alarm but no cases of fault detection are shown. The analysis is done offline however the authors indicate that an online system is possible since the calculations are not prohibitively time consuming.

It is interesting to note that quite a large proportion of the work carried out by researchers in wind turbine CM and data analysis is based on simulation and is quite often heavily influenced by assumptions. This is most likely due to the lack of extensive historical data sets and fault records which are available due to the relatively young nature of wind turbine CM due to its recent adoption. This contributes to the lack of experience and knowledge in how faults manifest themselves in the data. Another issue worth mentioning is the difficulty in actually acquiring data which exhibits actual component failures due to its commercially sensitive nature. This in itself can also pose boundaries which can significantly restrict the valuable outcome of research projects that are undertaken.

2.5 Chapter Summary

It is evident that although CM systems for wind farms are widely deployed, wind farm operators in industry are still unsure as to whether condition-based maintenance is an appropriate operational policy for their assets. This is based on a combination of factors which originate mainly from the perception of CBM as a complex maintenance concept when compared to the more traditional periodic maintenance policies. Condition based maintenance of machines and plants encompasses measuring and evaluating actual machine conditions, detecting problems which arise in their incipient stages and determining the remaining service life in order to allow effective operation and maintenance decisions to be made. The initial costs and the complexity of the hardware and software systems required to provide such a complete CBM solution can easily be perceived as being the more complicated approach and can be off-putting to operators. When coupled also with the difficulty of developing algorithms adapted and suited to the interpretation and analysis of new wind turbine technology the current lack of enthusiasm expressed by operators for the CBM approach can be understood.

The theoretical benefits of CBM shown nevertheless offer persuasive arguments for its use in wind farms, not least the probable future growth of offshore sites. The development of flexible, extensible and robust systems must be accelerated in order that these theoretical benefits can be achieved in reality.

It can be identified from the literature that most of the techniques and research focus is on a specific problem or component of the turbine in isolation. This is mainly due to the complex nature of each individual problem. A key feature of the research described in this thesis is to address this gap and provide a platform for wind farm CM systems using multi-agent technology. This framework will allow a number of independent analysis techniques to be brought together, processing data from multiple sources and focus on a wider range of components or problems. In this way, a more complete view of the current status of the turbines in the wind farm can be achieved. This also provides the opportunity for corroborating the output of the various techniques which can allow users of the system to gain an understanding of how faults and problems are reflected in the various SCADA parameters.

The normal behaviour models proposed and developed in this thesis are based on typical data parameters collected by commercial wind turbine SCADA systems. This means the approach developed here can be widely applied by wind farm operators on older existing machines with less sophisticated CM systems. While the models developed in this thesis model the parameters available in the SCADA data acquired, the development of the agent platform contributes to the novelty and archival value of the system in that future researchers can build upon the framework easily integrating new techniques for interpreting and analysing the data that may become apparent as research in this area continues over the coming years. New sensor based technologies may also emerge and become adopted for the monitoring of specific components such as the recent research in acoustic emission technology for detecting problems that occur in the blades. The platform developed can easily be extended to analyse new data sets and incorporate the findings in the output of the system.

The methodology presented in this thesis is designed with the notion of supporting wind farm operators to make more informed maintenance policy decisions. These systems are becoming especially pertinent with the view that in the coming decades a large number of wind turbines will come out of their warranty period and the responsibility for O&M will (generally) be transferred from the manufacturers to the utilities themselves.

The need for the research presented in this thesis has been established in this chapter, with reference to the gap found in the literature in this area of study. This thesis is concerned with the development of an extensible and flexible online FDS for wind farms based on the principle of anomaly detection models. The models developed therefore must be capable of capturing the complex relationships between SCADA parameters accurately in order that the normal behaviour can be correctly estimated under varying and dynamic operational range of turbine conditions. The techniques suited for developing such models are discussed and described in chapter 3.

2.6 Chapter References

Amirat Y., Benbouzid M.E.H., Bensaker B., Wamkeue R., (2007), "Condition Monitoring and Fault Diagnosis in Wind Energy Conversion Systems: A Review", Electric Machines & Drives Conference, 2007, IEMDC 07, IEE International, Volume 2, pp.1434-1439

Becker, E. and Poste, P. (2006) 'Keeping the blades turning. Condition monitoring of Wind Turbine gears', Refocus, v7, n2, March 2006, p26-32

Burnham, K. and Pierce, S.G. (2007) 'Acoustic techniques for wind turbine blade monitoring', Key Engineering Materials, v 347, 2007, p 639-44

British Wind Energy Association, (BWEA 2008), 'Wind Energy Technology', Accessed February 2009, Available at: http://www.bwea.com/ref/tech.html.

Carden, P. and Fanning, P. (2004) 'Vibration Based Condition Monitoring: A Review', Structural Health Monitoring, Volume 3, Issue 4, 2004, pp.355-377

Caselitz, P. and Giebhardt, J. (2005) 'Rotor Condition Monitoring for Improved Operational Safety of Offshore Wind Energy Converters', Transactions of the ASME: Journal of Solar Energy Engineering, volume 127, issue 2, May 2005, pp.253-61

Danish Wind Energy Association. (2008) "The Wind Energy Pioneers: The Gedser Wind Turbine", Accessed February 2009. http://www.windpower.org/en/pictures/juul.htm

Dutton A.G., Blanch M.J., Vionis P., Lekou D., van Delft D.R.V., Joosse P.A., Anastassopoulos A., Kouroussis D., Kossivas T., Philippidis T.P., Assimakopoulou T.T., Fernando G., Doyle C., Proust A., (2003), "Acoustic Emission Condition monitoring of Wind Turbine Rotor Blades: Laboratory Certification Testing to Large Scale In-Service Deployment", proceedings of European Wind Energy Conference 2003.

Europa Energy Research, (2010), European Commission Research Website, Accessed June 2010, Available at: http://ec.europa.eu/research/energy

Hart D.J., Baines N., Riddle A.F., (1997), "Intelligent data management system for wind farm monitoring and operation optimisation", 1997 *to be published under ETSU contract* W/36/00281/29/00

Hameed, Z., Hong, Y.S., Cho, Y.M., Ahn, S.H. and Song, C.K (2007) 'Condition monitoring and fault detection of wind turbines and related algorithms: A review', Renewable and Sustainable Energy Reviews, Available online August 2007, doi:10.1016/j.rser.2007.05.008

Hawker G., Hall S., Douglas N., (2007), "Downtime Analysis for Improved Decision Making", proceedings of European Wind Energy Conference (EWEC 2007).

Li S., Wunsch D.C., O'Hair E., Giesselmann M.G., (2001), "Comparative Analysis of regression and Artificial Neural Network Models for Wind Turbine Power Curve Estimation", Journal of Solar Energy Engineering, 2001, Volume 123, pp. 327-332, Transactions of the ASME.

Leaney, V.C., Sharpe, D.J. and Infield, D. (1999) 'Condition Monitoring Techniques for Optimisation of Wind Farm Performance', International Journal of COMADEM, 1999

Lu Bin, Li Yaoyu, Wu Xin, Yang Zhongzhou, (2009), “A Review of Recent Advances in Wind Turbine Condition Monitoring and Fault Diagnosis”, Power Electronics and Machines in Wind Applications, PEMWA June 2009, pp1-7.

McMillan D., (2008), “Techno-Economic Evaluation of Condition Monitoring and its Utilisation for Operation and Maintenance of Wind Turbines using Probabilistic Simulation Modelling”, Thesis submitted July 2008, Institute for energy and Environment, University of Strathclyde.

Negra, N.B., Holestrom, O., Bak-Jensen, B. and Sorensen, P. (2007) ‘Aspects of Relevance in Offshore Wind Farm Reliability Assessment’, IEEE Transactions on Energy Conversion, March 2007, 22, (1), pp. 159-166

Popa L.M., Jensen B., Ritchie E., Boldea I., (2003), “Condition Monitoring of Wind Generators”, Industry Applications Conference, 2003. 38th IAS Annual Meeting. Conference Record of the, Volume 3, Issue 1, 12-16 October 2003, pp.1839-1846.

Ribrant, J. and Bertling, L. (2007) ‘Survey of Failures in Wind Power Systems with Focus on Swedish Wind Power Plants During 1997-2005’, IEEE Trans. Energy Conversion, March 2007, 22, (1), pp. 167-173

Rumsey, M. A. and Paquette, J. A. (2008) ‘Structural health monitoring of wind turbine blades’, Proceedings of SPIE - The International Society for Optical Engineering, v 6933, Smart Sensor Phenomena, Technology, Networks, and Systems 2008, p 69330E

Sanz-Bobi, M.A., Garcia, M.C. and del Pico, J. (2006) ‘SIMAP: Intelligent System for Predictive Maintenance. Application to the health condition monitoring of a wind turbine gearbox’, Computers in Industry, Aug 2006, Volume 57, Issue 6, pp. 552-568

Sayas F.C., and Allan R.N., (1996) ‘Generation availability assessment of wind farms’, IEE Proceedings- Generation, Transmission, Distribution, volume 143, issue 5, September 1996, pp.507-518

Schneider, J., Gaul, A.J., Neumann, C., Hografer, J., Wellssow, W., Schwan, M. And Schnettler, A. (2006) ‘Asset Management Techniques’, Electrical Power and Energy Systems, Nov 2006, 28, (9), pp. 643-654

Singh S., Bhatti T.S., Kothari D.P., (2007), “Wind Power Estimation Using Neural Network”, Journal of Energy Engineering, ASCE, pp.46-52

Tavner, P.J., van Bussel, G.J.W. and Spinato, F. (2006) ‘Machine and Converter Reliabilities in Wind Turbines’, of IEEE Power Electronics Machines and Drives Conference, April
2006, pp. 127-130

van Bussel G.J.W., and Zaaijer M.B., (2001) ‘Reliability, Availability and Maintenance aspects of large-scale offshore wind farms’, Proceedings of MAREC, Newcastle, March 2001

Verbruggen T.W., (2003), “Wind Turbine Operation & Maintenance based on Condition Monitoring WT-OMEGA; Final Report”, ECN Windenergie 01-04-2003, ECN-C—03-047 ECN Report.

Vestas, (2009), “Vestas Wind Power Solutions v52-850kW the turbine that goes anywhere”, product brochure, accessed February 2009, http://www.vestas.com/en/wind-power-solutions/wind-turbines/kw.aspx

Wiggelinkhuizen, E.J., Rademakers, L.W.M.M., Verbruggen, T.W., Watson, S.J., Xiang, J., Giebel, G., Norton, E.J., Tipluica, M.C., Christensen, A.J., and Becker, E. (2007) 'Conmow Final Report', ECN Report, June 2007. Accessed February 2009 http://www.ecn.nl/docs/library/report/2007/e07044.pdf

Wilkinson, M. Spianto, F. Knowles, M. , (2006), "Towards the Zero Maintenance Wind Turbine", Universities Power Engineering Conference (UPEC 2006) proceedings of the 41st International 6-8 Sept 2006, Volume 1 pp.74-78

Zaher, A, Cruden A, Booth C, Liethead W. (2009). Database Management for High Resolution Condition Monitoring of Wind Turbines. Proceedings of the 44th International Universities Power Engineering Conference (conference paper).

Author Discussions

Yusuf Patel, (2007) Author discussion. Wind farm operator

3. Wind Turbine Fault Detection through SCADA Data Analysis

The purpose of this chapter is to identify a suitable technique that can provide useful and powerful interpretation capabilities for the SCADA data sets provided by the project's industrial partner, with the purpose of achieving early wind turbine fault detection. The literature reviewed in chapter 2 (section 4.3), has demonstrated that minimal research has been undertaken in the area of fault detection and diagnosis of wind turbines, utilising the SCADA data collated through the integrated CM systems installed in modern turbines.

The chapter begins with an examination of the SCADA data available to determine some of its basic characteristics and the meaningful information that can be extracted from it through possible correlations that may exist between the parameters for the purpose of fault identification. It then proceeds by reviewing some of the typical techniques utilised in the domain of data analysis for Fault Detection and Diagnostics (FDD) in industrial systems, in an attempt to explore typical approaches to the problem, examining their suitability for analysing wind farm SCADA. The challenges associated with SCADA data analysis and interpretation are analysed, and an explanation of why the anomaly detection approach adopted is required for the development of a successful fault detection mechanism for this particular problem. Finally the reasoning behind the chosen technique is summarised defining its selection for the models developed for this research.

3.1 Initial Analysis of SCADA Data

There are a number of parameters that are included in the SCADA data. Some integrated CM systems are more sophisticated than others, and this reflects in the range of parameters that are monitored. The parameters existing in the data sets acquired from the industrial partners involved in this research project are listed below:

- Active power output (10 min average and standard deviation (SD) over 10 min interval)

- Anemometer measured wind speed (10 min average and SD over 10 min interval)
- Nacelle temperature (1 hour average)
- Gearbox bearing temperature (10 min average)
- Gearbox lubricant oil temperature (10 min average)
- Generator winding temperature (10 min average)
- Power factor (10 min average)
- Reactive power (10 min average)
- Phase currents (10 min averages)

21 months worth of data consisting of the above parameters from April 05 to December 06 was acquired from a wind farm consisting of 26 Bonus 600kW stall regulated wind turbines. The parameters listed above are typical of data collected by commercial wind turbine SCADA systems, meaning an approach which can successfully utilise this data to provide early detection of faults as they occur in the components of the turbine would be widely applicable by wind farm operators. It is important to note that no failure or fault data records were supplied meaning the data is unlabelled and that the parameters cannot be correlated with a specific failure scenario.

In order to achieve the fault detection objectives of this research, it is necessary to first determine the parameters from the provided data sets that will be used to provide early fault detection information, before defining the nature of the fault detection mechanisms themselves. The importance of the gearbox and generator components has already been established from the review carried out in Section 2 of Chapter 2. Failures in both these components leads to a significant loss in revenue due to the considerable cost and downtime associated with obtaining replacements (Yusuf Patel 2007) and (Polinder H. 2006).

From the SCADA data made available for this project, it was decided that three SCADA parameters could be modelled and utilised for the identification of faults manifesting themselves in the turbine's components based on simple *physics*. The three parameters are the gearbox oil, bearing temperature and the generator winding parameter. As has already been demonstrated from the review in Chapter 2 (section 2), a number of research groups have investigated techniques for estimating the power performance of a turbine, or range of turbines across a wind farm for the purposes detecting performance degradation. Because some power and wind speed data is provided in the SCADA data sets acquired, there is potential therefore for the development of a power performance model to be used in conjunction with the three SCADA parameters identified earlier in an attempt to corroborate any faults identified. Estimating power performance is an extensive research problem in its own right, however a model was replicated from the literature and built for corroboration purposes to investigate the potential benefit that can be gained from combining power performance estimation with fault detection modules. The power performance model developed and the corroboration results are detailed at a later stage in chapter 5 section 7. The remainder of this chapter focuses on the development of the fault detection aspect for the three component oriented SCADA parameters.

3.1.2 Correlations between SCADA Parameters

The two gearbox parameters which can be found in the SCADA data, namely the gearbox main bearing and lubrication oil temperature, give an indication of how hot the gearbox is running, and therefore, modelling the normal behaviour of these parameters offers the possibility of detecting gearbox overheating. While a straightforward threshold check could be used to flag up temperatures exceeding a certain limit, this might well be too late to avoid significant damage to the gearbox. The desired functionality should take into consideration any relevant aspects of turbine operation. This approach would allow temperatures to be detected that are too high in the context of the concurrent level of power generation, leading to a quicker and more effective identification of anomalous behaviour.

In order to capture the normal behaviour of the two gearbox temperature parameters, the variables that can affect those temperatures must be taken into consideration so that an accurate model can be built. Wind turbines can only aerodynamically capture a proportion of the energy in the incident wind (Danish wind energy association 2008).

This energy is converted by the rotor blades into mechanical power and is transmitted to the gearbox by the low speed shaft. The gearbox then steps up the rotation rate to around 1500 rpm on the high speed shaft that drives the electrical generator to produce electrical power. This means that the mechanical load and stress that the gearbox undergoes, and therefore its temperature, is closely related to the amount of power generated by the turbine. The temperature of the gearbox will also be affected by the ambient temperature inside the nacelle of the turbine and so this also should be taken into consideration. Similarly the generator's temperature will be proportional to the active power i.e. the power generated by the generator, as well as the ambient temperature inside the nacelle. It can therefore be seen that the SCADA parameters which bear the most significant impact on the gearbox oil, bearing and generator winding parameters, are the active power output and the ambient temperature. Hence the gearbox oil, bearing and winding temperature must be related to the power and ambient temperature values via some form of function which affects each of the three parameters to some degree.

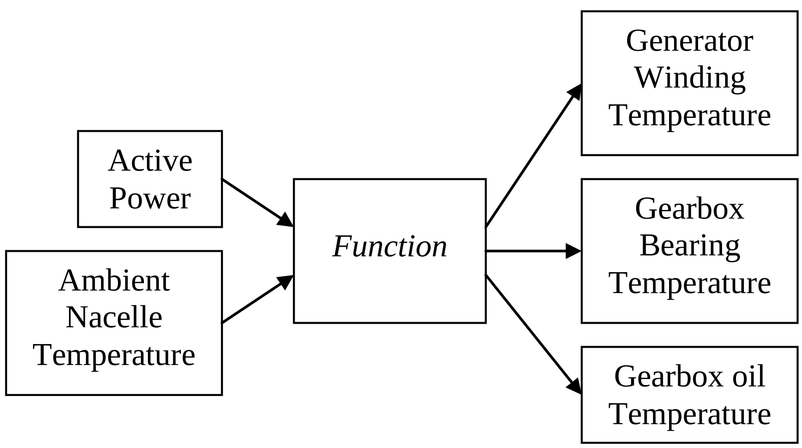

Figure 3.1: SCADA Data parameters relations

The nature of the data should also be taken into consideration when assessing suitable data analysis techniques. According to (Chandola et al 2009), data can be grouped into the following categories: Binary, Categorical, Continuous or a combination of these. Wind farm SCADA data is continuous in nature; it is seen as measurements recorded continuously over time, and as such the relationship among the data instances (each measurement) is seen as being sequential, with a temporal aspect relating the previous measurement with the next recorded measurement.

The multivariate nature of the data mentioned above also inherently imposes the requirement of techniques that can exploit multiple input data streams without difficulty, providing the opportunity to more accurately capture the non-linearity that exists between the parameters.

The next section goes on to explore a variety of techniques that exist in the area of fault detection and diagnosis applied to sensor data, monitoring different types of industrial equipment. The aim is to explore their characteristics and to identify the kinds of data and problems they are most suited to solving.

3.2 Artificial Intelligence and Statistical Based Techniques for Fault Detection and Diagnosis of Industrial Mechanical Systems

Artificial intelligence (AI) and statistical based techniques have become increasingly utilised in the field of Fault Detection and Diagnosis (FDD) of industrial systems. In the online fault detection review article published by (C. Angeli & A. Chatzininkolaou 2004), the authors' state that "in the case of very complex time-varying and non-linear systems, where reliable measurements are very complicated and valid mathematical models do not exist" techniques in the field of artificial intelligence "allow for the development of new approaches to fault detection in dynamical systems". The reasons behind their use is that they provide the necessary association, reasoning and decision making processes that mimic the thought process of a human brain when solving diagnostic problems. At the heart of most problems in this field of research, a large amount of historical data collected from some item of plant or machinery monitored through sensor technologies exists. The dynamic nature of the environment in which a turbine operates and is constantly subjected to, makes the development of mathematical models that can capture and interpret the operational behaviour of the turbine extremely difficult. These AI and statistical based techniques can be used in a variety of ways which can transform this data into knowledge for the purpose of achieving a suitable mechanism for fault detection or diagnosis. These reasons formulated the basis for the author's decision to investigate this particular domain of techniques.

A review of literature in the area of AI and statistical based FDD for commercial industrial systems was undertaken throughout the duration of this research. The aim of this exercise was to gain a detailed understanding of techniques which had been applied to tasks of a similar nature, and identify an appropriate approach for the application of early fault identification for wind turbines. The techniques which have been found to be utilised in this field are very diverse, often borrowing from the principles of anomaly detection and data mining disciplines with solutions specifically formulated for their particular application. Some of the common techniques which have been found to be utilised in this area of research can be categorised into knowledge (or expert) based systems (KBS), classification based approaches such as neural networks and support vector machines and statistical based techniques such as clustering based algorithms. All of these techniques can be used for the purpose of extracting or inferring knowledge from large volumes of sensor data. The following sections provide an outline of the named methodologies along with some examples from the literature detailing applications of how they have been utilised while highlighting some of the influential contributions. The review covered in the next section is by no means an extensive and complete review of all possible techniques that can be applied to the data, but rather serves as an insight into identifying some of the commonly applied techniques in the area of FDD.

3.2.1 Knowledge (Expert) Based Systems in Fault Detection

Expert systems have been defined by (Feigenbaum 1982) as "an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution". What distinguishes an expert system from a conventional program, are the facts that an expert system provides a simulation of human reasoning regarding a problem domain, where the main focus is the expert's ability of problem solving and choosing how to perform relevant tasks under the constraints of a specific scenario. Some of the first diagnostic expert based systems developed for technical fault diagnosis were built as early as the 1970's at the Massachusetts Institute of Technology (C. Angeli & A. Chatzininkolaou 2004)**.**

Derivatives of Knowledge-Based System architectures

The architecture of a KBS generally consists of two main components namely, a knowledge-base and a reasoning mechanism. The knowledge-base holds the facts, procedural rules and heuristics of the system which essentially determine the expert capability of the system. The knowledge representation schemes for a knowledge-base can typically take the form of one of three main types, rule based, case based or model based. The selected scheme can influence the design of the reasoning mechanism.

Rule based knowledge-bases are the most common form of KBS. They are typically focused on a very specific problem which is well understood by a number of domain experts. The process of capturing this knowledge which can be either explicit (well defined) or tacit (generally experiential) (Luger & Stubblefield 1997) is important for the success of this approach.

Model based systems on the other hand utilise a model that describes a particular engineering plant function that is derived from design studies. These models simulate how a device should be functioning and any deviation from the expected behaviour would imply that the device has malfunctioned. The mechanism involves an initial phase which generates a set of hypothesis for a given symptom (Davidson et al 2005). The derived set of hypotheses is then reduced to a set which appropriately covers all of the current observed behaviours of the device. The final phase involves selecting candidate hypotheses as explanations of the discrepancy between the observed and expected behaviour on the foundation of additionally acquired discriminating information. This approach differs from the rule based approach in that it assesses the possible explanations for an observed symptom in terms of the device model rather than the captured probable causes in a rule base.

Finally, Case based reasoning systems exploit solutions to previous problems of a similar nature to solve current problems (Watson & Marir 1994) and (Luger & Stubblefield 1997). It essentially functions by measuring the similarity of the currently observed case, to the cases present in the "case base". This measure of similarity is used to distinguish between potentially matching cases. Therefore rather than a chain of inferences explaining the causes of a problem as with rule based and model based systems, a more detailed solution of a similar case can be presented along with the supporting evidence of why this particular case was selected. This approach is highly dependent on the availability of suitable cases to provide experienced diagnostic knowledge of an engineering plant or device.

Reasoning Approach

This part of the system is responsible for the reasoning strategy of the system. It enables it to respond to varying situations and to infer new knowledge from the existing knowledge in the knowledge-base (Davidson et al 2005). The reasoning mechanisms for the three derivatives of KBS differ in their method of how they infer information from the knowledge stored in the rules, models or case libraries. Storing all possible combinations of knowledge in a particular problem domain would be largely impractical as the number of elements would increase excessively reaching a *combinatorial explosion* (Rolston 1988). Because of this high level knowledge can be captured for use in general applications, or, detailed knowledge regarding a very specific and well defined application as shown in (Strachan et al 2008) can be used in KBS.

Example Knowledge-Based Systems

A good example of a knowledge-based system developed for the purpose of fault detection and diagnosis is that developed by (Strachan et al 2008) which presents a system which provides a comprehensive diagnosis of the defects responsible for partial discharge activity detected in oil-filled power transformers. The authors propose a means to abstracting the important features which characterise the observed phase resolved partial discharge patterns. The system then makes use of captured knowledge which describes the visual interpretation of these patterns to provide both the location and diagnosis of the defect. The system essentially mimics an expert's visual interpretation of these phase resolved partial discharge patterns. The particular system developed by Strachan et al in this paper uses a number of rule bases. Each of the rule bases encapsulates knowledge which is associated with a specific stage of the partial discharge interpretation process. The inference engine utilises input data to decide which rules in the rule base to invoke. If the conditions of a rule are satisfied, then it is invoked, and the rule concludes that some predefined action should be performed, for example, to report a diagnosis or to invoke another rule. The particular inference engine utilised by Strachan et al is a forward chaining reasoning mechanism across the different rule bases which underpins the incremental approach of knowledge-based diagnosis to the possible causes of partial discharge.

An interesting point which (Strachan et al 2008) stress, are the advantages of the KBS approach over other successfully applied machine learning and pattern recognition techniques for the same application. The authors comment on the fact that while the trained classifier approach can be successful for fault detection and diagnosis, it remains specific to the transformer data set from which the model was initially derived from. This highlights the fact that the classifier approach requires historical data for training, further leading to another limitation where the classifier will only be capable of recognising defects for which it has been trained. The diagnostic rules for the paper presented by (Strachan et al 2008) were said to be elicited from partial discharge experts.

(McArthur et al 1996) developed a model based reasoning system applied to protection system performance analysis in the electricity supply industry. In a Similar fashion to the more recent wind farm power plants, the use of SCADA systems result in an increased amount of data becoming available, overwhelming utility engineers. The authors state the need for intelligent data interpretation systems that can convert data into appropriate information for engineers. A decision support system is presented which aims to perform three main tasks, alarm processing, fault diagnosis and comprehensive validation of protection performance. The third task employs a model-based reasoning paradigm where as the other tasks are developed as rule-based systems.

A consistency based reasoning mechanism is adopted where the correct behaviour of individual components and their interconnections is modelled in a logical or mathematical fashion. The authors utilise the diagnostic approach for current techniques for feeder protection and protection relays. For these components knowledge of how each of these components should operate is built into the models, for example the operation of the protection relays are modelled using mathematical equations which define the relay response to fault currents. The authors state that the advantages of the adopted model based diagnostic approach only require a single model of correct behaviour to diagnose all scenarios of problems assuming the component is modelled correctly. They can also be utilised to predict system states given specific conditions which can function as a useful tool for training engineers.

What is clear from this model based approach is that a robust definition of how the monitored item of plant or machinery operates is necessary to build the models. This approach is very effective when the components and their characteristics are completely understood. It can therefore be concluded that it is un-applicable in conditions where the operation of the components is not well understood or defined through logical or mathematical expressions.

For the application of case based reasoning (CBR), the authors (Olsson et al 2004) propose a CBR system for fault diagnosis of industrial robots at ABB Robotics. The methodology proposes a combination of signal processing to filter out noise from received sensor signals which are then forwarded to a case-based classification component to recommend a fault class. The main strengths mentioned regarding the case-based approach are its direct reuse of concrete examples in history. The method provides the opportunity for learning from experiences without the need for data training which omits the problem of over-fitting which can arise with classifier techniques explained at a later stage in the chapter. They also provide the ability for incremental learning if new useful cases are properly inserted into the case library. The author's state that a complete case library is not required for the system to function properly. This is not entirely accurate however as a case-based system diagnosis capability is only as good as the scenarios in the case-base which have been encountered and if such scenarios can be generalised to previously unseen but similar scenarios. (McDonald et al 1997) state that where suitable cases exist, CBR provides the option of rapidly developing a KBS. The authors also comment that it remains to be seen "*how many cases, particularly in complex engineering domains are sufficient to realise a KBS capable of providing sensible conclusions for the range of problems which might occur*". Because of this CBR systems are said to be likely to remain as part of a larger hybrid system which augments the abilities of the system rather than being the main diagnosis module.

Summary of Knowledge Based System Characteristics

In conclusion, it is clear that knowledge based systems are built around highly specific areas of expertise where detailed knowledge is available, allowing for the development of rule-based, case-based or model-based systems. Each of the three systems is dependent on the availability of specific types of information. Rule-based systems require rules which can be elicited from experts in the domain or from documentation. Model-based systems require information which allows the development of accurate models that can completely define the operation of the monitored system. Finally case-based systems require historic cases with solutions to similar situations which might be faced.

While the advantages of the KBS approach are evidently favourable, it is immediately apparent that it is not suitable for the SCADA data application presented in this thesis. The main reason for this is the lack of available domain knowledge or access to previously recorded cases of failure which can suitably aid fault detection and diagnosis through the analysis of SCADA data through rule, model or case type knowledge-based systems. Through collaboration with the industrial partners (Scottish Power) it was explained that examples of such domain knowledge with respect to the faults which can develop in a wind turbine was unavailable at the time of writing and only the SCADA data supplied was available.

3.2.2 Clustering Based Fault Detection

The process of clustering is used to group *similar* data instances or vectors consisting of *n* features into groups known as *clusters* in *n-dimensional* space (Tan et al 2005). These clusters apparently reflect a mechanism that causes individual instances to bear a stronger resemblance to each other than they do to the remaining instances. From the data mining perspective (extracting knowledge from data (Witten & Frank 2005), clustering can be used to identify groupings within the data that correspond to specific fault instances for example (Strachan 2005) where specific groupings in the data correspond to a specific fault type**.** Equally however from an anomaly detection outlook these groupings can be used to highlight normal classes of behaviour while instances of data that do not conform to any of the clusters or that which conform to a specific cluster represent an anomaly.

In fact, in the comprehensive literature survey on anomaly detection, the authors (Chandola et al 2009) split clustering based anomaly detection into three categories based on the following assumptions:

1. *Normal data instances belong to a cluster in the data, while anomalies do not belong to any cluster.*
2. *Normal data instances lie close to their closest cluster centroid, while anomalies are far away from their closest cluster centroid.*
3. *Normal data instances belong to large and dense clusters, while anomalies either belong to small or sparse clusters.*

These two different types of fault detection, namely fault classification and anomaly detection based fault detection give rise to the concepts of supervised and un-supervised training. It is important that the difference between these training modes is explained so that the '*learning'* mechanisms of the remainder of the techniques reviewed in this chapter can be put into perspective.

Supervised and Un-supervised Training Modes

Machine learning techniques such as statistical based clustering algorithms and classifiers are typically used in one of two main modes, namely: supervised (and semi supervised), and unsupervised anomaly detection. These modes refer to the training procedure and how its requirements can affect successful model capture. Typically supervised training requires the data to be fully ‘labelled’ in the sense that each instance of data is associated with either a normal or anomalous class. Any unseen data is compared against the trained model to determine which class it belongs to. In real world applications this is rarely the case as labelled data is normally unavailable.

Semi-supervised training is essentially the same as supervised training. The main difference however is the assumption that the training data instances used are labelled for only the normal class. That is only normal “looking” data is used for training. This assumption should be fairly accurate and based on compelling evidence however as if abnormal data is used to train the model, an inaccurate representation of normal behaviour will be captured. Its applicability is therefore dependent on whether this assumption can be made regarding the training data. Semi-supervised training is considered to be applicable to a larger range of applications since there is no requirement for labels of the fault classes which are not always present in data sets.

Finally unsupervised training operates in a manner which does not require labelled data. This therefore makes techniques which can carry out unsupervised training the most widely applicable approach out of the two main training modes since most data sets fall into this category. Techniques which operate in this mode make the implicit assumption that normal instances of data are far more frequent than abnormal instances in data sets. Both self organising maps, as well as the expectation maximisation clustering algorithms operate in this manner and are explained within this section.

When considering wind farm SCADA, the data is not “labelled” in classes of anomalous and normal behaviour. This limits the scope of the applicable techniques which are compatible with the semi-supervised and unsupervised training modes, making the training process substantially more complex. While the SCADA data is not labelled into any groups, techniques which operate in semi-supervised mode can be applied through assumptions regarding which portions of data constitute normal turbine operation. These assumptions can however often lead to the possibility of inaccuracies being built into the model during the training process. Therefore, it is particularly important that the selection of data for training purposes accurately represents the behaviour which is to be modelled.

Example Clustering Algorithms

While there are a large variety of clustering algorithms available, they typically revolve around the unchanged principle of grouping similar instances into groups. Modifications can be made to aspects such as the similarity measure used, how the number of clusters is identified, as well as constraints placed on a cluster's shape and size, which essentially allows for a diverse range of clustering mechanisms. Two powerful clustering techniques are described in the following section along with example applications in the area of fault detection and diagnosis.

K Means Clustering

The most classic form of the clustering technique is known as the k-means algorithm. The algorithm's clustering process operates by an initial specification of the number of clusters that are being sought after known as the parameter K. K points are then chosen at random by the algorithm to be used as cluster centres in the *n-dimensional* feature space (where *n* is determined by the number of features an instance of data has i.e. a vector's features). Each data instance in the data set is then assigned to the closest cluster centre according to the Euclidean distance metric shown in equation 3.1:

$$Dist = \sqrt{\sum_{i=0}^{i=n}(V_i - W_i)^2} \qquad \text{(eq3.1)}$$

Once all of the instances have been assigned, the centroid (mean) of the instances in each cluster is calculated and taken to be the new centre values for the respective cluster group. The entire process is then repeated with the new centre values in an iterative manner until the same instances are assigned to the same cluster centres i.e. the cluster centres no longer change. At this point the centres are stabilised and will not change for that particular training data set. New previously unseen input data instances (or vectors) can then be assigned to the cluster which closest matches its position in the *n-dimensional* feature space.

(Strachan 2005) makes use of K means clustering as one of a number of techniques used for the classification of six particular kinds of defect which occur as a result of partial discharge events in oil filled transformers. The data set used to train the K-means classifier model is labelled for each partial discharge event, noting its associated cause (one of six defined defect types). The data is pre-processed using a feature extraction process which calculates a number of statistical features that characterise a 3-dimensional phase resolved pattern, where the pattern represents the recorded partial discharge event. In total, 101 features were deduced per feature vector to characterise the phase resolved pattern, based on the work of (Gulski 1991). The work of Gulski has demonstrated its relation to the causing defect rather than processing the data in its raw format. These parameters are basic, deduced and statistical features such as mean pulse height, variance, standard deviation, kurtosis and skewness to name a few. A dimensionality reduction process known as Pearson’s correlation coefficient (Godfrey et al 1988) was then used to reduce the size of the feature vector by identifying correlations between the features. The coefficient varies between +1 and -1 indicating a positive or negative linear correlation respectively, and zero indicates no correlation. This process reduced the feature set to approximately half the size in view of the fact that the removed features were seen as offering no new discriminatory information from a data mining perspective. This emphasises the need for the selected features of the data instances present within the data set to exhibit some form of spatial relation when mapped to the feature space which allows them to be differentiated in order to ensure the success of the K-means clustering process. This also stresses the fact that the data in its raw format, might not be correctly suited to the clustering process and that some form of feature extraction is a prerequisite to the techniques success.

The training process involved specifying and setting the ‘number of clusters’ parameter K, which incidentally is the only variable parameter used in the development of the K-means model. It is initially set prior to the training process and then adjusted accordingly throughout the development process in tandem with the evaluation and refinement phases.

The specification of the K parameter is considered a “trial and error” process as opposed to one of scientific judgement. This is considered the main shortcoming of the K-means model development process. A ‘rule of thumb’ proposed by (Dasarathy 1991) does however exist which states that the number of K-means clusters should lie in the range indicated by the following rule:

Number of Classes <= ***Number of Clusters*** *<= SQRT (Number of training Data Records)*

Based on this assumption, (Strachan 2005) incremented the ‘number of clusters’ parameter iteratively through the calculated range until the classifier model performed satisfactorily.

While there maybe a specific set of desired classifications for a particular application, typically the number of clusters which are found to be formed in the *n-dimensional* space after the clustering process do not equate to the number of classifications. There is also a large possibility that each cluster of data does not entirely consist of data instances belonging to one class, which would be the desired case, but instead contain instances from multiple classes. In order to extract meaning from the output of the clustering process once the clusters have been established, it is necessary to assign labels to the clusters found. Strachan overcomes this through assessing the proportion of each class represented within each cluster. Labels are then assigned corresponding to the dominant class of instances in the cluster. A confidence rating is also assigned to the cluster where this is calculated as a percentage of data instances associated with the dominant class present in the corresponding cluster against the total number of instances present in the cluster.

In conclusion from this application it is apparent that in order to classify the various defects which can result from partial discharge events using the K-means clustering algorithm, the data must be labelled accordingly. While this is an implication on the data rather than the technique itself, this implication is necessary for its success as it allows the technique to be trained to recognise the characteristics of each of the defects to successfully classify them.

The previous feature extraction research carried out by (Gulski 1991) which allowed the important characteristics of the data instances to be accurately summarised, formed the basis for applying the clustering technique with success.

Self Organising Maps

The Self Organising Map (SOM) is another popular clustering algorithm which has been developed. It uses a different mechanism to achieve the basic clustering process. The SOM (also known as the Kohonen Map) uses a neural network to discover the underlying structure of the data. It provides a means of representing multi-dimensional data into lower dimensional spaces, usually one or two dimensions. It can therefore function as a powerful visualisation tool as it makes use of a set of neurons mapped to a two dimensional Cartesian plane making them easy to inspect.

If we consider a two dimensional SOM consisting of a 2D lattice of connected nodes as shown below in figure 3.2, the workings of their clustering process can be better understood. Each neuron is fully connected to the input layer; (in the case of figure 3.2, a 2 dimensional vector) which is presented to the network. Each of these nodes (neurons) has a specific topological position in the lattice and contains a vector of weights of the same dimension as the input vectors. For example, if the training data presented to the network consists of a set of vectors of *n dimensions,* then each node will also contain a corresponding weight vector of *n dimensions*.

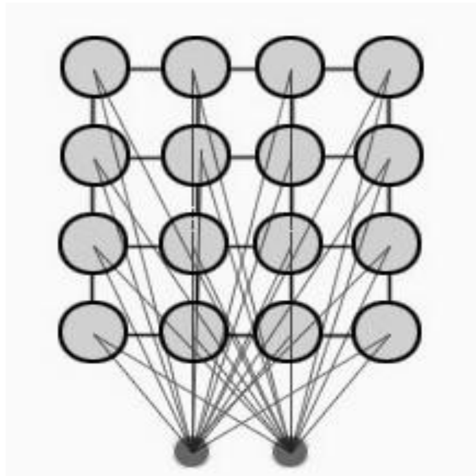

Figure 3.2: A 4 x 4 Node SOM with a 2 input vector

A SOM does not require a target output to be specified like other classic multi-layer neural networks during their training phase. This allows it to operate in an unsupervised manner.

The initial distribution of weights for each node is randomly set. The input vector is then presented to the network. Where the node weights match the input vector, that area of the lattice is selectively optimized to more closely resemble the data for the particular class the input vector belongs to. Therefore, from an initial distribution of random weights, and over multiple iterations, the SOM eventually settles into a map of stable zones. Each zone can be thought of as a feature classifier, allowing the graphical output of the network to be envisaged as a type of feature map of the input space. The algorithm operates the following steps in the order listed and occurs over multiple iterations:

1. The weights of each node are initialized to random values.
2. An input vector is then chosen randomly from the set of training data instances and presented to the lattice.

3. Each node in the lattice is examined in order to assess which node's weights are most similar to the input vector. The resulting node found to most closely resemble the input vector is commonly known as the Best Matching Unit (BMU). This is calculated through a similarity metric, typically the Euclidean distance shown in equation 3.1 above.

4. The radius of the neighbourhood of the BMU is then calculated. This is a value that starts large, typically set to the 'radius' of the lattice, but diminishes each time-step for example through the use of an exponential decay function expressed in equation 3.2. Using Pythagoras theorem, any nodes found within this radius are deemed to be inside the BMU's neighbourhood.

$$\sigma(t) = \sigma_o \exp\left(-\frac{t}{\lambda}\right) \qquad t=1,2,3\ldots \tag{eq3.2}$$

5. The BMU as well as each neighbouring node's (the nodes found in the previous step) weights are adjusted to make them more like the input vector. The closer a node is to the BMU; the more its weights get altered. This is carried out according to the following equation

$$W(t+1) = W(t) + \Theta(t)L(t)(V(t) - W(t)) \tag{eq3.3}$$

Where Θ and L are calculated using the exponential decay function using the respective equations:

$$\Theta(t) = \exp\left(-\frac{dist^2}{2\sigma^2(t)}\right) \qquad \text{t=1,2,3\ldots} \tag{eq3.4}$$

$$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right) \qquad t=1,2,3\ldots \qquad \text{(eq3.5)}$$

6. The process from step 2 is then repeated for N iterations.

Through this process it can be observed that the specification of the parameter K i.e. the number of clusters, is automatically identified based on the data set through the training process without the need for its specification as with other clustering mechanisms.

(Germen et al 2007) make use of a SOM for the classification of mechanical faults in induction motors. They are used to classify broken rotor bars and misalignment type faults in the motors. The research was conducted in a laboratory environment with a total of four induction motors each displaying a different mechanical fault. Two of the motors exhibit a different number of broken rotor bars 3 and 5 respectively, the third is used for misalignment tests and the last is a healthy motor to draw comparisons. In order to classify the possible faults of the motors using a SOM, An important step which the authors' state significantly influences the success of the SOM for classification is to determine adequate features for the clustering process. The authors make use of knowledge that exists from previous related research which proves that broken rotor bars give rise to certain side-band frequencies in the line current spectrum, while misalignment fault frequencies fall anywhere around the supply frequency. (Germen et al 2007) encapsulate this information as a feature vector through signal processing which offers discriminatory properties for the faults considered.

Germen et al's objective was to use SOM to discriminate data from the defective induction motors from the healthy one while also classifying the type of defect. A data set of 80 experiments was utilised where 20 were obtained from the misaligned motor, a further 20 obtained each from both the motors with 3 and 5 broken bars, and the last set of 20 were obtained from the healthy motor. Each of the 80 experiments was then converted into its corresponding feature vector. The results of the clustering process produced 3 different cluster sets. The first cluster formed the data from the healthy motor;

the second formed the data from the misaligned motor, while the last cluster combined both the broken rotor bar data from the two motors.

This application of SOM demonstrates that, in-line with most clustering techniques, its success is again determined on the ability to extract discriminative feature vectors from the data which requires knowledge of the data or the domain to form a data set which is compatible with the clustering process. This is perhaps considered the most challenging aspect of the clustering process and heavily influences its success rate.

Summary of Clustering Techniques for Fault Detection

The following conclusions are reached after considering the three clustering algorithms described in this section. The various applications reviewed demonstrate that clustering can be applied to both fault classification and for anomaly detection based fault detection.

- For classification based fault detection, where specific faults are diagnosed, the data instances must be labelled into their respective fault classes for successful training and diagnosis of previously unseen data. This requirement is not specific to the clustering technique but rather is considered a prerequisite for fault classification in general.
- For anomaly detection based fault detection with unlabelled data, labelling the clusters which are formed in terms of anomalous or normal classes can be an issue especially in data sets which contain larger numbers of clusters.
- Several clustering based techniques are effective only when anomalies do not form significant clusters among themselves as this makes it more difficult to infer useful information from the clustering process.
- The success of the clustering process is highly dependent on the structure of the data used for training. Most applications tend to pre-process the data carrying out some form of feature extraction which offers discriminative features suitable for

clustering the data into their appropriate groups. Data in its raw format is rarely suited for clustering.

- The testing phase for clustering algorithms is relatively fast due to the typically small number of clusters which every test instance must be compared to. The number of clusters is normally fixed once the training phase is complete and remains unchanged for the majority of applications. This again simplifies the testing phase.

The most important points that emerge from this analysis of clustering techniques and their applicability to data processing for FDD is that they rely heavily on the structure of the data, and the extent of the differentiable features which can be extracted from the data. The significance of the feature extraction process is apparent in the range of applications reviewed where they were all built on previous substantial research that focused on the identification of distinguishable features in the data.

Through consultation with industrial wind farm operators, at the time of writing, it was advised that wind turbine SCADA data was not well understood (Yusuf Patel 2007) and hence such discriminative features were not readily available. Techniques which are therefore able to process the data in its raw format would be deemed more suited and less complicated to apply.

3.2.3 Artificial Neural Network Based Models for Fault Detection

The principle of an artificial neural network is a biologically inspired technique which is based on the parallel architecture of the human brain. The human brain consists of vast numbers of neurons which have been estimated to be in the order of 10-500 billion (Haykin 1994). Each of these neurons is interconnected with branches known as dendrites (inputs) and axons (output) shown in figure 3.3. These branches form a highly complex biological neural network. The axon of a neuron is connected to the dendrites of multiple other neurons through a junction known as the synapse. When the combined signals received by a neuron from connected neurons exceeds a particular threshold, the neuron fires passing an impulse signal down its axon. This signal is then input at the dendrites of other connected neurons through their synaptic connections. The synapse can have an excitatory or inhibitory influence on the signal passed between the neurons, modifying the amount of signal transferred. It is this synaptic efficiency (or strength of connection) that is modified as the brain *learns* and *memorises*.

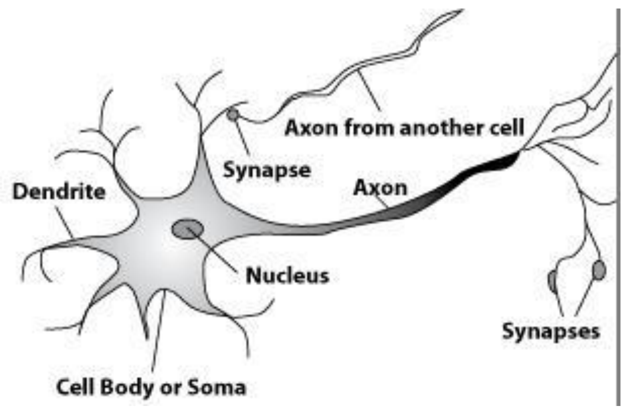

Figure 3.3: The Biological Neuron

In an artificial Neural Network (NN), the neuron, often referred to as the perceptron, is mathematically modelled to replicate the functionality of a biological neuron. The NN works in an identical manner where the weighted connection between the perceptrons is analogous to the memory effectively stored in the synaptic weights of the biological neuron.

Figure 3.4, illustrates the composition of a perceptron processing element. Each neuron input is multiplied by its corresponding weight which signifies the strength of the connection. The output of the neuron fires if the summation of the inputs modified by the weights, totals to a value greater than some threshold value.

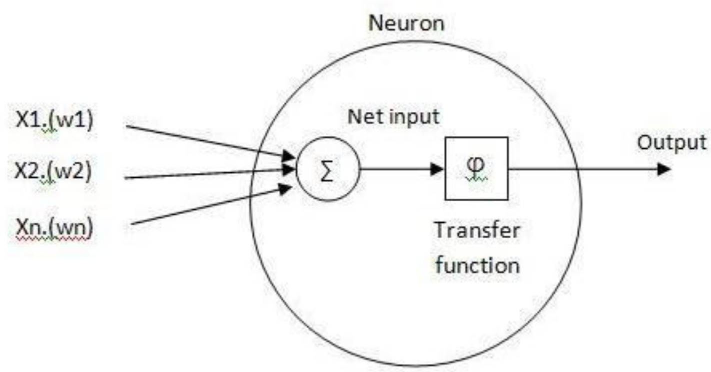

Figure 3.4: The Perceptron processing element

All of the neurons combined in an NN effectively provide a powerful non-linear data-processing structure. This gives the NN the ability to perform tasks similar to that of the biological neural network such as pattern matching and learning. These abilities work well in tasks where it is difficult to formulate a step by step algorithmic solution, but have access to many examples of the kind of behaviour required; they have therefore been proven useful for data analysis, capturing patterns and relations between multiple parameters in data sets. The neurons are arranged into a number of layers which consists of an input layer, output layer and any number of intermediate hidden layers in between, linking the input layer to the output layer. The hidden layers are essentially responsible for performing the activation function between the input and output data. NN fall into the category of classifier techniques, where the name is derived from the techniques ability to learn a model (classifier) from a set of (labelled) instances, and then classify a test instance into one of the classes using the learnt model. Their applications are not however limited to classification only as will be explained in the following sections.

Neural Network Operation

A neural network (NN) operates in one of two modes, the training mode, where the network attempts to learn some function, or the testing mode where the network uses what it has learned in the training phase to provide some output to previously unseen input data. The training and testing data samples are presented to the network through the input layer. The training mode can operate in any of the two main modes supervised (and semi-supervised), or unsupervised depending on the data available. The SOM's detailed earlier are an example of a neural network operating in an unsupervised manner performing clustering which displays their flexibility. The supervised training process is achieved through splitting the data into training and testing data sets. The supervised process entails the training data consisting of inputs and their associated outputs, which the network attempts to relate through a non-linear function by exposure to a sufficient number of examples. The weights in each neuron are initially set at random values. The input values are then propagated through the network yielding an output with an error between the initial value and the desired output. This error is then fed back through the network and used to adjust the weights accordingly with the aim of minimising the magnitude of the error term. This process of adjusting the weights is determined by the algorithm used to train the network. This process is then iterated until the weights no longer change, or the set number of training cycles is complete and the network is considered '*trained*'. The network can then be tested on the second data set which deliberately comprises data that the network has not been trained on or previously exposed to. Its success at estimating values that are as close as possible to the desired output values determines how well the network has learned or captured the relation between the inputs and outputs. (Rumelhart et al 1986) compares the operation of a neural network to a *"parallel computer that can program itself to compute some function, given suitable exemplars specifying that function".* He describes it as a system where we do not require the knowledge of how to write the program in order to get the system to carry out the desired task.

This description serves to elucidate the mysterious self-learning nature of NNs. The fact that we do not have to know how to write the program to achieve a particular task implies that we do not know how the NN arrives at its output. NNs have often been referred to as 'black-boxes' as they effectively learn by example, where their inputs and outputs are clearly defined, but their internal functionality remains hidden and unexplained. They are therefore heavily dependent on the quality of data used for training, as accurate and representative data is required for an effective learning procedure. The ‘black-box’ nature of NNs is considered their main limitation as this leads to the inability to provide an explanation or justification for the output provided. Despite this, NNs have been extensively used and successfully applied to a wide range of domains requiring pattern recognition, estimation and prediction as well as fault classification.

Back Propagation Algorithm

The back propagation algorithm is one of a number of variations of NN training algorithms. These variations mainly differ in topology (i.e. interconnections between nodes), weight adjustment and activation function used. The back propagation algorithm is the most popular supervised learning strategy used in neural networks. It was considered a significant breakthrough in NN research when the algorithm also known as the ‘multi-layer feed-forward network’ was first developed (Rumelhart et al 1986). This specific derivation of neural network has been used to address the problem of FDD across a large number of fields and applications for both fault classification and anomaly detection. Its ability for pattern recognition gives it the necessary mechanic to perform prediction and estimation allowing it to lend itself well to applications such as, speech recognition, image classification, medical diagnosis and condition monitoring to name a few. This pattern recognition ability is achieved through their capacity for effectively characterising non-linear relationships.

Figure 3.5 shows the topology of a back propagation network consisting of three layers an input layer, one hidden layer and the output layer. The number of input nodes is typically determined by the number of inputs fed into the input of the network. The number of hidden nodes as well as layers is determined empirically, while the number of output nodes is dependent on the output requirements of the application.

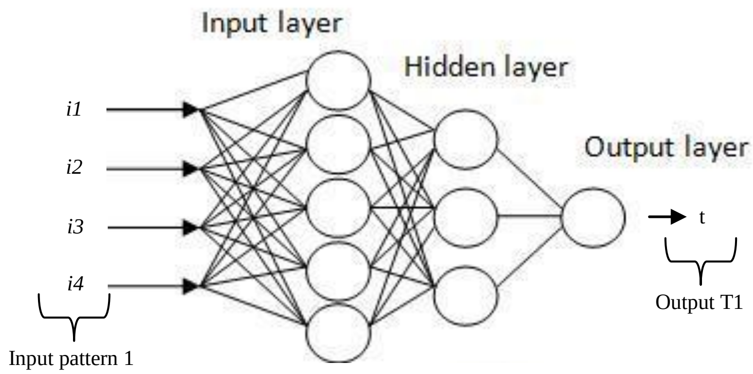

Figure 3.5: Back Propagation Neural Network Architecture

Each of the nodes in each layer is connected to the nodes in the adjacent layer as shown in figure 3.5. Preparing the training data set consists of pairing each of the training input patterns presented to the network with its corresponding target pattern e.g. (input pattern 1) is paired with (*T1*) in the case of the diagram.

The back propagation algorithm consists of two main steps, the forward propagation step and a backward propagation step. This is repeated for each input pattern presented during training. The algorithm has been defined by (Tarassenko 1998) as follows:

1. Feed-forward step

    a. Select a training pair (inputs and associated target output) from the training data set and apply it to the network input

    b. Calculate the actual network output for the given input pattern

    c. Calculate the error between the actual network output and the target output.

2. Back-Propagation step

    a. Modify the weights of the network in a manner which minimises the calculated error. This is achieved by propagating the calculated error back through the network from the output layer to the input layer.

3. Repeat steps 1 and 2 for each of the training patterns in the data set until the error for the entire set is acceptably low.

Effectively, this algorithm can be viewed as a gradient descent method (Rumelhart et al 1986), tasked with finding the weights for a multi-layer feed forward network with non-linear processing elements. In this way it presents a procedure for learning, where in principle it evolves a set of weights to produce an arbitrary mapping between the input and output (i.e. mapping a relation between the training pairs).

A total of three calculations are carried out at each node for every iteration of the training procedure. Two of these calculations are performed during the feed-forward step, where each node determines whether it will fire to provide an output. The third calculation is performed to calculate the error at the network node where a slight difference in this calculation exists depending on the type of node i.e. a hidden or output node.
The first calculation involves calculating the net input. This is essentially the summation of the products of the inputs *(i1, i2, i3…)* multiplied by the connection weights (*w1, w2*, *w3…*) summarised by equation 3.6.

$$net_j = \sum_{i=1}^{i=n} w_i x_i \qquad \text{(eq3.6)}$$

The second calculation involves determining the activation level of each node by assessing the value of net j computed in the first step to the activation function used equation 3.7.

$$y_j = f(net_j) \qquad \text{(eq3.7)}$$

There are a variety of activation functions that can be used in a neuron to determine whether it fires or not. The most typical function used for the back-propagation NN however is the sigmoid shown in figure 3.6. The value of net input is represented along the *x-axis* and the output of the neuron is shown on the *y-axis* also illustrated by equation 3.8

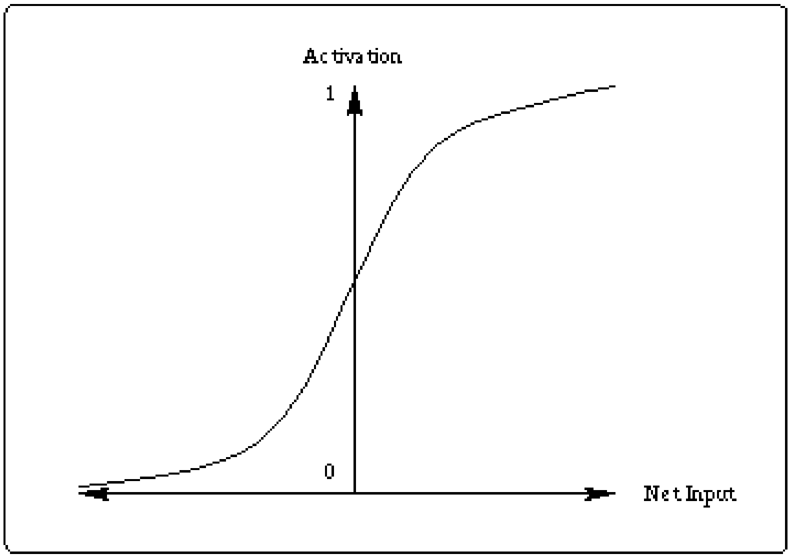

Figure 3.6: Sigmoid activation function

The sigmoid function often referred to as the squashing function due to its shape, effectively compresses the range net j such that it lies between a value of zero and one. The shape of the function inherently provides a form of automatic gain control where a small increase in net j close to the zero results in a higher gain due to the steeper nature of the curve at lower net j values. This allows for the accommodation of larger signals without saturation and small signals without excessive attenuation. (Chapter 5 Section 2.2 details other commonly used activation functions and how networks can include layers consisting of more than one type of transfer function).

$$y_j = f(net_j) = \frac{1}{(1+e^{-net_j})} \qquad \text{(eq3.8)}$$

Once the actual output of each node is established, the third calculation, the error between the actual output and the target output, must be determined. If the node is an output node, the back-propagated error is calculated using equation 3.9. For a hidden node the error is calculated using equation 3.10.

$$\delta_j = (t_j - o_j) f^{'}(net_j) \qquad \text{(eq3.9)}$$

$$\delta_j = f^{'}(net_j) \sum_k \delta_k w_k \qquad \text{(eq3.10)}$$

Where δ is the error at the node, t is the target output, *o* is the actual output and *w* is the node weighting. From these equations it can be seen that the amount by which a given weight will change is proportional to the derivative of the nonlinear activation function. Consequently the largest change in a weight will occur for neurons that have an output mapping near the mid range of the function i.e. neurons that have not yet committed to being on or off. (Rumelhart et al 1986) believes that this characteristic contributes to the stability of the learning system.

These non-linear capabilities of the back-propagation NN allows it to effectively model non-linear activation functions between several input variables and one or more output variables, making it well suited to pattern recognition, classification, parameter estimation, forecasting and prediction applications.

Overtraining and Generalisation

An important aspect which should not be overlooked during the training process is the issue of overtraining a network. When training a neural network with examples, the purpose of the training process is to capture a relationship between the inputs and outputs presented to the network, which will allow it to generalise to patterns out with the training data set. Overtraining can occur when the number of training samples is relatively small in comparison to the complexity of the network (i.e. too many nodes with a large number of connections (weights) which can be used to capture relations). In such cases the network can essentially replicate the training data set with great accuracy since it effectively memorises it. Initially the network appears to perform well but when tested on previously unseen patterns which deviate slightly from those used in training, it exhibits a high generalisation error since the system has the freedom to take advantage of spurious correlations in the data. Constraining this degree of freedom is therefore important in order that the network adapts to only the dominant regularities as opposed to the spurious irregularities that might exist in the training data, effectively generalising the underlying correlation which exists to provide reasonable solutions to new previously unseen inputs.

Determining the size of the network for a specific application is considered the main challenge of the training process, and is often referred to as something of a 'black art'. Typical approaches include training smaller networks, incrementing the size with each iteration until the smallest one is found that will learn from the data and generalise successfully on a predefined data set. Another approach is to train a network larger than necessary and then employ one of a number of pruning techniques to remove or penalise network weights (Reed 1993).

Example applications

The comprehensive review by (Zhang 2000) covers an extremely wide range of applications which utilise neural networks for classification. Zhang states that since neural networks are non-linear models, this makes them flexible in modelling real world complex applications. The vast range of literature reviewed in the paper on neural networks for classification indeed supports this statement.

(Tarassenko et al 2000) investigate the principle of novelty detection (also referred to as anomaly detection) as an approach to the problem of fault detection. The paper introduces the concept of a neural network predictor as a model of normality. The approach only requires normal class data to be defined, where a model of this normality is learnt by including only normal examples in the training data. Abnormalities are then identified by testing for novel instances against this description. The neural network was applied to detect subtle changes in the temperature profile of aero-derivative gas turbine engines. The data was acquired via 17 thermocouples and several thousand hours of normal operation was available to build models of normality. No data on any occurrence of engine malfunction or failure was available however and so the anomaly detection approach was deemed suitable. The compressed air heated up in the engines nine combustors swirls outwards towards the 17 thermocouples giving rise to a characteristic thermocouple signature. The signature varies with engine speed where all of the thermocouple outputs increase by similar amounts when the engine speed is increased respectively. A secondary nonlinear distortion of the signature caused by minute changes in the swirl angle also results with an increase in speed.

The authors argue that when a fault develops in one of the combustion chambers in the engine, there is a local effect on the temperature profile. Only a small number of contiguous thermocouples are significantly affected while the remainder remain unchanged. The models of normality were therefore built around learning the function relating the temperature values which are affected so that a thermocouple reading can be predicted based on these readings. The nonlinear effect caused by the change in speed was also required to be captured and so a multi-layer perceptron model was used as a nonlinear predictor using speed as an input along with the temperatures values of the contiguous thermocouples. These values were used to predict the output of a thermocouple on the opposite side of the turbine casing. The models of normality were engine specific and so testing involved constructing models for each of the engines and testing on unseen patterns of data for the corresponding engine. A real fault detected by the model is presented in the paper demonstrating the value of novelty detection conclusively and proving the success of the modelling methodology capturing the nonlinear relations required between the inputs and outputs.

The work of (Garcia et al 2006), reviewed in chapter 2 section 2.4.3 is the most similar work found in the literature and considered an influential contribution to the fault detection aspect of the work presented in this thesis. The authors present a system developed for the purpose of health condition monitoring of a wind turbine gearbox. The part of the system abbreviated as SIMAP which is of interest here, is the fault detection module focused on identifying incipient faults in a wind turbine gearbox. The models are NN based models, utilised to capture the normal behaviour of the gearbox oil and bearing temperature, as well as the thermal difference between the gearbox oil before and after it is cooled by the heat exchanger. The authors do not state if the data used is SCADA data, however the parameters utilised to train the NN normal behaviour models are similar to the monitored parameters in the data sets acquired for this thesis. The parameters used for the models consist of the two regressive inputs for the parameter modelled, as well as the generated power, the nacelle temperature, and cooler fan slow run or fast run inputs detailing the state of the cooling mechanism. No detailed information of the training process or the kind of NN used is discussed in the paper however. The authors present a case study in which the models detect a gearbox problem 2 days in advance of its failure. While 2 days may not provide sufficient time to schedule an appropriate maintenance action, it does display promising results for the analysis of SCADA parameters. An in depth comparison between the models / results of their work and the workings presented in chapter 5 of this thesis is published in the journal of Wind Energy by (Zaher et al 2009).

Related to the theme of wind energy, (Moghaddas-Tafreshi et al 2007) make use of two NN models to forecast the power to be generated by a wind turbine one hour in advance. They make use of multi-layer networks for both models, trained using the back-propagation algorithm for its prediction and estimation capabilities. One model makes use of the last eight hourly wind speed inputs to predict the next time step of wind speed (one hour in advance) at a number of different wind masts in the wind farm and hub-height for the turbine in question. The outputs of this model are then used as inputs to the second model which is used to provide an estimation of the expected power output.

Summary of Neural Network Characteristics

The following conclusions are reached after considering neural networks for the use of FDD through the analysis of its mechanics and the various applications reviewed.

- Their non-linear nature allows complex non-explicit relations to be learned making them well suited for forecasting, estimation and pattern recognition

- They are able to handle multi-variate data without adding complexity to the development process as inputs and outputs are automatically mapped during the training phase.

- They are useful for situations where no mathematical algorithm exists, but plentiful examples of the kind of behaviour required is available.

- They have high data requirements for successful training

- They can be utilised for both, classification of faults as well as anomaly detection modules through normal behaviour capture as shown in the literature reviewed.

It is interesting to note the large range of applications which neural networks have been applied. Their main limitation as discussed is the black box nature they possess where no reasoning is provided for how they arrive at a particular result. This limitation can affect certain types of applications more so than others, where perhaps a classification based application might benefit from some form of explanation as to how it arrives at its result. The normal behaviour applications reviewed however have been shown to be extremely useful at capturing the relations between the data parameters while the emphasis of how the NN arrives at its results is not considered to be as important.

The characteristics which they possess lend themselves very well to the wind farm SCADA data, due to the fact that they can incorporate multiple variables and capture the undefined, non-linear relations that exist between them where no mathematical functions exist to define this non-linearity. The data which exists also provides a large number of examples of the desired behaviour. Their application by (Garcia et al 2006) to the wind farm gearbox oil and bearing data along with the results obtained suggest that they are a suitable technique for analysing the SCADA data to provide the capability of fault detection through anomaly detection. It would therefore be interesting to explore their ability to capture the relations between similar parameters as well as the generator winding temperature parameter from a different wind farm consisting of different wind turbine models to achieve early fault detection.

3.2.4 Support Vector Machines

Support Vector Machines (SVM) are a relatively new computational learning technique based on statistical learning theory developed by (Vapnik Chervonenkis 1995). They have recently emerged as a general mathematical framework for estimating dependencies between a finite number of samples. Similarly to Neural Networks, they fall within the classifier group of techniques and demonstrate the capability to perform pattern recognition and prediction. The difference between them however is the method used for minimising the error for the training data set during the training process. Neural networks utilise empirical risk minimisation while SVMs make use of structural risk minimisation which is stated to provide better generalisation abilities achieved through the minimisation of the upper bound of the generalisation error (Vapnik 1995).

In order to better understand the SVM's principle of operation (Bennet et al 2000) explain that three key concepts must be grasped, these are *margins, duality* and *kernels.* Consider the example of a simple case of binary classification (Bennett et al 2000) where a set of data points X_i (i=1,….., m), have the corresponding labels of $Y_i = \pm 1$. Each of the data points is represented in N dimensional attribute space. Initially let us imagine the scenario where the data points belonging to each of the two sets are linearly separable in the N dimensional space i.e. a plane exists that can correctly classify all of the data points in both sets. If we let the classification function be:

$$f(x) = sign(w \cdot x - b) \qquad \text{(eq3.11)}$$

where w is a vector that determines the orientation of the discriminant plane, *sign* represents the signum function which returns the "sign" of a real number, and the scalar b determines the offset of the plane from the origin. There are an infinite number of possible separating planes that correctly classify the training data. A plane which maximises the margin between the classes however would be the preferred discriminant, since minor perturbations of any data point would not introduce misclassification errors as shown in Figure 3.7. Generalisations on previously unseen data are also more likely to classify correctly with such a discriminant without the need for additional information.

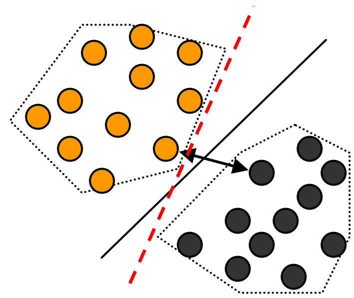

Figure 3.7: Two possible linear discriminant planes

In order to construct the optimal hyper-plane which maximises the distance, we can examine the *convex hull* of each class' training data (indicated by the dotted lines surrounding each class of data), and then locate the closest points in the two *convex hulls* indicated by the arrow as shown in Figure 3.6. These two points from each class are incidentally named the support vectors. There are a number of mathematical methods that can be used to determine the optimal discriminant hyper-plane. One method is to find these support vectors by solving a quadratic problem as shown in (Bennet et al 2000). The other is to maximise the margin between two parallel supporting planes also discussed by (Bennet et al 2000). Effectively both methods result in the same solution and the choice can be made regarding which method is used which highlights the mathematical concept of duality.

(Bennet et al 2000) state that from a statistical learning theory perspective, the quadratic program formulations required to solve these problems are well founded. Statistical learning theory is said to *"prove that the bounds on the generalisation error on future points not in the training set can be obtained. These bounds are a function of the misclassification error on the training data and terms that measure the complexity or capacity of classification function."*

For the linear case briefly described above, maximising the margin between both classes reduces the function complexity. Therefore, by explicitly maximising this margin, the bounds on the generalisation error are also being minimised leading to better generalisation with high probability.

For non-linear function capture, SVM's utilise *Kernels*. In the typical case where the data classes are not linearly separable and no linear discriminant function will work, methods are needed to convert the linear classification algorithm into a nonlinear algorithm. A classic method is to add additional attributes to the data that are nonlinear functions of the original data. In this way existing linear classification algorithms can be applied to the now expanded data set in feature space to produce nonlinear functions in the original input space. A quadratic classification example is provided in (Bennet et al 2000).
Some of the more popular kernels are the *degree d polynomial, radial basis function* and the *two layer sigmoid neural network*. Research into new Kernels is ongoing with specific domain requirements in mind. Therefore, in order to change from linear to non-linear classification, only the kernel method should be substituted instead of the original linear function described earlier with no algorithmic changes required from the linear case. SVM's have been applied successfully to a number of applications, ranging from text categorisation (Dumais et al 1998), Particle identification (Barabino N. et al 1999) and engine knock detection (Rychetsky M. et al 1999).

An interesting application of SVM is that of (Assuncao et al 2006). The authors' make use of the least squares SVM approach to develop a model for the estimation of transformer top oil temperature. What is interesting about this application is the comparison of the results against an NN model developed for the same purpose. Both the SVM and NN models are used as tools for the purpose of building normal behaviour temperature regression models that can be used to predict future values of oil temperature. The NN model was formed as a two layer feed-forward structure using ambient temperature and transformer loading as inputs. The number of hidden nodes was varied between 2 and 20 during the training process in an attempt to find the optimal network architecture. The data was also normalised between the range of -1 and +1 minimising the range of values the network must accommodate in its output in order to yield better training results.

The least squares SVM was trained on the same data using an optimisation algorithm to tune the hyper parameters σ (specifying the width of the kernel used) and φ (regularisation penalty parameter, determines trade off between the error minimisation and smoothness of the estimated function). In order to avoid over fitting, the authors state that a small data set was required, as larger sets resulted in poor generalisation performance. The final results showed that the SVM estimation performance was marginally better than the neural network, with both models however performing better than the IEEE model standard proposed in Annex G (IEEE Transformers Guide 1996) cited by (Assuncao et al 2006).

(Hao & Lewin et al 2008) make use of a SVM to improve the detection of partial discharge monitoring in high voltage oil filled power transformers. The authors compare a passive hardware filter and the SVM technique. Their results from a laboratory experiment are stated to indicate that the SVM approach provides better performance than the passive hardware filter, reliably detecting discharge signals.

(Widodo et al 2007) published a review article surveying the application of SVMs to the diagnosis of rolling element bearings, induction motors, diagnosis of machine tools and a number of other industrial CM based scenarios. The authors state that while SVM provides good performance for classification, their use is not as established as other classical approaches such as neural networks, expert systems and case based systems for condition monitoring and fault diagnosis.

Summary of Support Vector Machine Characteristics

The following can be concluded regarding the use of Support vector machines for data analysis tailored towards the application of fault detection and diagnosis.

- Like NN's they provide a general methodology for a range of problems, with the ability to perform similar tasks such as classification, regression and anomaly detection providing the necessary characteristics for powerful data mining.
- (Bennet et al 2000) state that problems such as local minima which can occur in neural networks are eliminated due to their statistical learning mechanisms. The results which they produce are stable and reproducible independent of the specific algorithm used to optimise the SVM model. In comparison the results of an NN are dependent on the particular algorithm and the initial values of weights of the network connections and so results may vary with each training process.
- There is no need to be an SVM expert and understand the statistical theory behind their learning mechanism making the method relatively simple to use.
- Model selection parameters such as the type of kernel to use (including its associated parameters) and penalty regularisation are still present in the training process however and can highly influence the success of the approach.
- Similarly to neural networks incorporation of domain knowledge is achieved mainly through the preparation of the data sets used for training, and the interpretation of the results such as the support vectors found by the algorithm do not offer much information meaning they too exhibit a black box nature.

While some of the reviewed applications have shown that SVM's provide optimistic classification results often outperforming neural networks, they have not been applied as extensively as neural networks have. More literature, tutorials and therefore examples exist of how to successfully apply neural networks to a variety of applications. The similar abilities which they share with neural networks would also allow for suitable fault detection models to be developed from the wind farm SCADA data parameters.

3.3 Utilising the Anomaly Detection Approach for SCADA Data Interpretation

After assessing the techniques reviewed and considering the unlabelled SCADA data available, it quickly becomes evident that the anomaly detection approach is the method most suited for achieving fault detection. The reasoning behind this is the absence of domain related knowledge regarding the relation with how incipient component faults are represented in the SCADA data parameters. A lack of resources, as well as the time consuming and tedious nature of analysing arduous volumes of data is regarded as the primary reason why current wind farm owners do not analyse the data as was previously discussed in the introduction. The limited documented knowledge of wind turbine fault detection available in the literature however (discussed and reviewed in chapter 2) prohibits the development of expert or knowledge-based systems making them currently an unsuitable solution to the wind turbine application that would not offer much value to wind farm operators. The lack of extensive historical data with access to fault records also rules out the development of case based system solutions that would typically be trained on specific fault signatures that can be used to offer useful diagnostic information based on previous failure scenarios. Clustering based classification has also been shown to be highly dependent on the availability of labelled data which corresponds to specific fault scenarios. As labelled data is not available, they too cannot be utilised as viable solutions for processing the SCADA data. These limitations ultimately led towards the decision of utilising anomaly detection as a means to achieving fault detection through the observation of abnormal behaviour.

The concept of anomaly detection corresponds to the discovery of events that typically do not conform to expected behaviour. Anomalies, by definition, are infrequent; however their importance is quite high when compared to other events, making their detection extremely important. The process consists mainly of an attempt to capture an accurate model of the normal behaviour of some process or machine in operation with the aim of being able to differentiate between normal and otherwise abnormal behaviour. With regards to wind turbine SCADA data, a model of the 'normal' behaviour would be captured in the form of sensor data, recorded from the various sensors mounted on the monitored components of the turbine. The objective of this model is to capture how the data evolves and changes with respect to factors that may influence it under normal circumstances. This would therefore allow for the detection of anomalous behaviour, even if this type of behaviour has not been seen previously. The model is used to provide an estimate of the sensor output based on the inputs that can affect the values recorded by the sensor. This estimation can then be compared to the real value recorded by the sensor, where a significant deviation from the estimated value would be viewed as an abnormality. In this way incipient faults can be highlighted and presented to the operator, dramatically reducing the complexity of their task since only significant information of relevance to the health of the turbine is presented to them.

While it can be argued that anomaly detection cannot be used to accurately classify faults given that no knowledge of the different types of faults is included in the models. Its use in this research (where knowledge of faults in SCADA data is unavailable) however, is to provide the initial stage of the fault identification process. Early detection of failures and problems would allow operators to schedule maintenance schemes appropriately, optimising the efficiency of their resources and hence the potential of their wind farms. Once the anomalies are detected, the opportunity of 'labelling' them according to specific fault classes can be achieved through the help of an experienced operator as the industry and knowledge of wind turbine CM progresses. The operator's knowledge could be captured as rules and then used to classify various instances of failure. This however is out with the scope of this piece of research and can be regarded as a viable solution to be implemented in the future once this kind of knowledge and information becomes available.

The success of the anomaly detection approach is determined by the accuracy of the developed models. An inaccurate model might produce erroneous estimates causing the system to flag up false alarms. It is therefore imperative that the accuracy of the model captured does in fact represent 'normal operation' of the modelled item of plant as closely as possible to ensure the development of robust fault identification mechanisms.

This factor can often pose difficulties with the anomaly detection approach to fault detection, as a large data set with fault free regions is obligatory for its success. The most effective process to testing how accurately the models represent 'normal behaviour' is through testing how the output of the model compares with different instances of previously unseen operational behaviour which can be classed as normal. The next section goes on to look at the associated challenges of anomaly detection and the difficulties which are faced when employing this approach.

3.3.1 Anomaly Detection Challenges and Issues

The key challenges associated with anomaly detection have been summarised by the comprehensive review article on anomaly detection carried out by (Chandola et al 2009)**.** Figure 3.8 summarises the scope of their review while the following set of points summarise the challenges:

- Defining a representative normal region in the data is challenging
- The boundary between normal and outlying behaviour is often not precise
- Availability of labelled data for training and validation
- The exact notion of an outlier is different for different applications
- Data might contain noise
- Normal behaviour keeps evolving and a current notion of normal behaviour might not represent normal behaviour in the future possibly requiring retraining
- Appropriate selection of relevant features

Techniques	Classification Based Clustering Based Nearest Neighbor Based Statistical Information Theoretic Spectral
Applications	Cyber-Intrusion Detection Fraud Detection Medical Anomaly Detection Industrial Damage Detection Image Processing Textual Anomaly Detection Sensor Networks

Figure 3.8: Range of techniques surveyed by (Chandola et al 2009)

These issues are generic in nature to achieving anomaly detection regardless of the application. The authors (Chandola et al 2009) state that due to these challenges "the anomaly detection problem, in its most general form, is not easy to solve". It is important to understand that each instance of data in any data set can be described by a set of attributes. The nature of these attributes determines the applicability of anomaly detection techniques since most of the techniques "solve a specific formulation" of the anomaly detection problem. By assessing the data available, its nature and the required output from the anomaly detection mechanisms, a formulation can be induced and an improved sense for the most suited technique(s) can be achieved. These factors are often determined by the application domain in which the anomalies are to be detected. Figure 3.9 shown below portrays how the application domain imposes its requirements on the chosen anomaly detection technique.

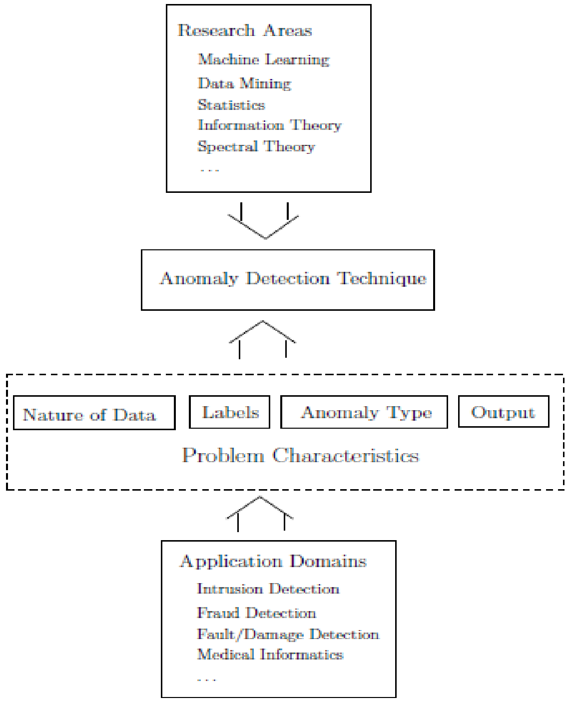

Figure 3.9 Key components associated with an anomaly detection technique (Chandola et al 2009)

It is necessary that the techniques used have the ability to capture the dynamic nature of the operational behaviour of a turbine. It is also important that the normal behaviour modelling solutions are capable of utilising multiple parameters in order to provide an accurate estimate of the evolution of the SCADA parameters, whilst also taking into consideration the current operating conditions.

3.3.2 Defining SCADA Anomalies

Attaining an understanding of the application domain is important in order that a context for the anomaly is defined based on domain knowledge. In this way we can be sure that the anomalies detected offer some useful information regarding the monitored item of plant or machinery making it easier to interpret the output of the anomaly detection mechanism developed. This section describes the different types of anomaly classes that can be identified in data sets and then goes on to identify which particular class SCADA data anomalies would fall under.

Anomaly Classes

Anomalies can be classed as one of three main types, namely point, contextual and collective anomalies. A point anomaly describes a single instance of data which is "anomalous" with respect to the remainder of its data set. The majority of the anomaly detection research which can be found in the literature is focused on identifying this type of anomaly (Chandola et al 2009). Contextual anomalies on the other hand, are characterised by an instance of data which is anomalous in a specific context and would not be considered abnormal unless viewed within this context. The concept of a context is evoked by an inherent structure which can be found within the data set and must be specified as a part of the problem formulation. Each instance of data is defined using the following two sets of attributes.

1. Contextual attributes: these are used to determine the context for a specific instance. For example in time series data, time is a contextual attribute which verifies the position of an instance with respect to the entire sequence of data. Some examples of the typical forms of contextual attributes which can be defined are listed by (Chandola et al 2009)**:**
    - Spatial attributes: these define the location of a data instance and therefore a spatial neighbourhood.
    - Graph attributes: edges that connect data instances define the neighbourhood for each instance.
    - Sequential attributes: the attributes of an instance of data are its position in a sequence.
    - Profile attributes: when data does not have an explicit spatial or sequential structure but nevertheless can be grouped into components using a set of contextual attributes. For example grouping users in activity monitoring systems (e.g. cell-phone fraud detection).

2. Behavioural attributes: define the non-contextual aspects of an instance of data. For example in a data set where the instances exhibit a spatial relationship such as

the average rainfall of the entire world, the calculated average rainfall at any location can be considered a behavioural attribute.

Finally collective anomalies, as the name suggests relate to the notion of an anomaly arising from a collection of related data instances being anomalous with respect to the data set. Each individual data instance might not necessarily be anomalous in its own sense, but rather the occurrence of these related instances together is regarded anomalous. An important point to be noted here is that while a point anomaly is not confined to any particular type of data set and can occur in any data set, collective anomalies are only possible in a data set whose instances are related in some manner. Contextual anomalies, on the other hand, are distinguished through their occurrence entailing the availability of context attributes in the data.

When considering the required output from an anomaly detection mechanism with regards to the SCADA data parameters, anomalies occur mostly because of an unusual observation for a given set of operational conditions. With the parameter relationships identified at the beginning of the chapter it can be seen that SCADA data anomalies would best fall within the class of collective anomalies, since a collection of related data instances are anomalous with respect to the entire data set (i.e. across multiple parameters). The following section provides a summary of the characteristics which define the nature of the acquired wind turbine SCADA data.

3.2.6 Summary of SCADA Data Modelling Requirements

From this preliminary analysis of the SCADA data, a number of points emerge that can be used to summarise its key characteristics.

- The data is multivariate and continuous in its nature (time series).
- The data is unlabeled which requires techniques which can operate in an unsupervised or semi-supervised manner.

- Anomalies are best viewed as being collective anomalies. As described in the previous section an anomalous reading in one parameter can be considered abnormal with respect to other related readings from different parameters. A preference towards the collective anomaly approach is prompted mainly due to the following favourable advantage regarding the nature of the wind turbine application:
    - A clearer understanding of how the turbine is operating can be achieved through a technique which can utilise multiple related parameters to support its evidence of an abnormal temperature instance. Therefore a more accurate model of normality can be captured as the turbine's current operating conditions are taken into consideration. This limits the chances of flagging false anomalies, leading to the development of a more accurate fault detection mechanism.

Taking these modelling requirements into consideration, the next section details the reasoning behind the technique selected for testing from the set of reviewed methods.

3.3 Technique Selection

Considering the techniques reviewed and the SCADA data available, there are two techniques which offer practical characteristics that would allow for the development of anomaly detection mechanisms given the requirements described in the previous section. These are the back propagation neural network and support vector machines.

While both techniques provide the necessary characteristics required to capture normal behaviour models of the SCADA parameters, and exploring which one would provide the best results would prove interesting, a decision was taken to utilise NN's as the method to model the relationships between the SCADA parameters. They were chosen as the approach to be used to model the SCADA data parameters due to their extensive application to similar problems, coupled with the fact that resources and tools required were more readily available to build NN models. Also their application to similar data by (Garcia et al 2006) also offered positive results with the detection of a gearbox failure two days in advance. From a wind turbine maintenance perspective however, the detection two days in advance would not be considered practical, as scheduling maintenance activity or arranging for the replacement of a gearbox in this time frame would not be considered plausible. Despite this, the positive aspect of this result is the fact that the non-linear function capture ability of the NN shows potential for successfully capturing the relations between the parameters modelled. Therefore through their application to another data set with plentiful examples of the desired behaviour from another turbine model, there is potential for improving the result through a more refined data preparation and training process.

3.4 Chapter Summary

This chapter served the purpose of reviewing a number of techniques commonly utilised in the area of industrial fault detection and diagnosis where little to sometimes no domain knowledge exists regarding the data collected through condition monitoring systems. For this particular application, the anomaly detection approach was shown to be the necessary approach with two of the reviewed techniques offering the characteristics to satisfy the requirements for developing fault detection mechanisms. The decision to initially test neural networks to develop normal behaviour models is summarised through the following points:

1. They lend themselves well to this particular application where plentiful data exists of the desired behaviour which is to be captured and have been extensively applied to similar applications.

2. The SCADA parameters exhibit undefined non-linear relations between one another. Therefore the non-linear function capture ability of NN’s makes them suited to such a problem.

3. The resources and tools required to build the NN models were readily available in comparison with the other techniques.

4. The work of (Garcia et al 2006) explored NN models tested on gearbox oil and bearing data but from a different turbine model with positive results without detailing the steps behind the training and testing process. Investigating whether or not the method lends itself well to other wind turbine models and the generator winding parameter along with the possibility of improving model accuracy would prove interesting.

5. The outcome of their models (Garcia et al 2006) which detected a gearbox failure 2 days in advance could serve as a benchmark of performance which could be used to determine the effectiveness of the data pre-processing and training procedures undertaken to see if the sensitivity to faults propagating in the components could be improved.

Chapter 5 goes on to detail the training and testing of the normal behaviour models developed along with case studies of their application to unseen SCADA data. Before this however, chapter 4 describes the technology suited to achieving a flexible and extensible automated framework for carrying out the data analysis process.

3.5 Chapter References

Angeli C., Chatzinikolauo A., (2004), "On-line Fault Detection Techniques for Technical Systems: A Survey", International Journal of Computer Science & Applications, 2004, Vol. 1, No.1, pp.12-30.

Assuncao T.C.B.N., Silvino J.L., Resende P., (2006), "Transformer Top Oil Temperature Modelling and Simulation", Proceedings of World Academy of Science, Engineering and Technology, Vol. 1, October 2006, ISSN 1307-6884.

Barabino N., Pallavicini M., Petrolini A., Pontil M., Verri A., (1999), "Support Vector Machines vs Multi-Layer Perceptrons in Particle Identification", Proceedings of the European Symposium on Artificial Neural Networks, 1999, pp. 257-262.

Bennett K.P., Campbell C., (2000), "Support Vector Machines: Hype or Hallelujah?", SIGKDD Explorations. Copyright 2000, ACM SIGKDD, December 2000, Vol. 2, Issue 2, pp1-13.

Chandola V., Banerjee A., Kumar V., (2009), "Anomaly Detection: A Survey", ACM Computing Surveys, September 2009, pp. 1-72

Danish Wind Energy Association. (2008) "The Wind Energy Pioneers: The Gedser Wind Turbine", Accessed February 2009. http://www.windpower.org/en/pictures/juul.htm

Dumais S., Platt J., Heckerman D., Sahami M., (1998), "Inductive Learning Algorithms and Representations for Text Categorisation", 7th International Conference on Information and Knowledge Management, 1998

Dasarathy B.V., (1991), "Nearest Neighbour (NN) Norms: NN pattern classification techniques", IEEE computer Society press tutorial, 1991

Davidson, E.G. and McArthur, S.D.J. and McDonald, J.R. (2005), “Model-based reasoning techniques for the analysis of protection operation.”, International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, 13 (2). pp. 97-104. ISSN 0969-1170

Feigenbaum E., (1982), Knowledge Engineering in 1980’s, Dept. of Computer Science, Stanford University, Stanford CA.

Germen E., Ece D.G., Gerek Ö.N., (2007), “Self Organising Map (SOM) Approach for Classification of Mechanical Faults in Induction Motors”, International Work Conference on Artificial Neural Networks 2007, pp.855-861.

Godfrey M., Roebuck E.M., Sherlock A.J., (1988), “Concise Statistics”, Editor Arnold Edward, ISBN: 0 7131 3591 3, 1988.

Gulski E., (1991), “Computer-Aided Recognition of Partial Discharges using Statistical Tools,” P.h.D Thesis submitted 1991, Delft University, Delft, The Netherlands.

Haykin S., (1994), “Neural Networks: A Comprehensive Foundation”, Published by Upper Saddle River, N.J. : Prentice Hall, ISBN: ISBN0132733501

IEEE Transformers Guide, (1996), “IEEE, *Guide for Loading Mineral-Oil-Immersed Transformers”*, ISBN: 1-55937-569-8, June 2002.

Luger G, Stubblefield W, (1997), “Artificial Intelligence: Structures and Strategies for Complex Problem Solving”, third edition, published by Addison Wesley Longman Inc. (chapter 6).

McDonald J.R, Burt G.M., Zielinski J.S., McArthur S.D.J, (1997), Intelligent Knowledge Based Systems in Electrical Power Engineering”, Chapman & Hall London Publishers

McArthur S.D.J., Dysko A., McDonald J.R., Bell S.C., (1996), The Application of model Based Reasoning Within A Decision Support System for Protection Engineers", IEEE Transactions on Power Delivery, October !996, Vol. 11, No.4

Moghaddas-Tafreshi S.M., Panahi D., (2007), "One-Hour Ahead Forecasting of Wind Turbine Power Generation Using Artificial Neural Networks", proceedings of Internationale Energiewirtschaftstagung IEWT 2007. Accessed September 2009, Available at:
http://eeg.tuwien.ac.at/events/iewt/iewt2007/papers/abstracts/Poster/PosterTechnologien/Tafreshi_WindForecastingNeuralNetworks.pdf

Olsson E., Funk P., Xiong N., (2004), "Fault Diagnosis in Industry Using Sensor Readings and Case-Based Reasoning", Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, January 2004, Vol.15, Issue. 1

Polinder H., Frank F.A. van der Pijl, Gert-Jan de Vilder, Tavner P., (2006), "Comparison of Direct-Drive and Geared Generator Concepts for Wind Turbines", IEEE Transactions on Energy Conversion September 2006, Vol.21, Issue 3, pp.725-733.

Reed R., (1993), "Pruning Algorithms: A Survey", IEEE Transactions. Neural Networks, July 1993, Vol. 4, pp740-747.

Rolston D.W., (1988), "Principles of Artificial Intelligence and Expert Systems Development", McGraw-Hill Book Company

Rumelhart D.E., Hinton G.E., Williams R.J., (1986), "Learning Internal Representations by error propagation", Parallel Distributed Processing: Explorations in Microstructure of Cognition, 1986, Vol. 1, pp.319-362.

Rychetsky M., Ortmann S., Glesner M., (1999), „Support Vector Approaches for Engine Knock Detection“, Proceedings of the International Joint Conference on Neural Networks, July 1999, Washington USA.

Sanz-Bobi, M.A., Garcia, M.C. and del Pico, J. (2006) ‘SIMAP: Intelligent System for Predictive Maintenance. Application to the health condition monitoring of a wind turbine gearbox’, Computers in Industry, Aug 2006, Volume 57, Issue 6, pp. 552-568

Strachan S.M., (2005), “Decision Support System Development for Plant Condition Monitoring and Protection Design Reuse using Structured Methods of Knowledge Engineering and Knowledge Discovery”, PhD Thesis submitted June 2005, Institute for Energy and the Environment, University of Strathclyde.

Strachan S.M., Rudd S., McArthur S.D.J, Judd M.D., (2008), “Knowledge-Based Diagnosis of Partial Discharges in Power Transformers”, IEEE Transactions on Dielectrics and Electrical Insulation, February 2008, Vol. 15, No1

Tan P.N., Steinbach M., Kumar V., (2005), “Introduction to Data Mining”, Addison-Wesley.

Tarassenko L. (1998), “A Guide To Nueral Computing Applications”, ISBN 0 340 70580 2, Arnold, 1998

Tarassenko L., Nairac A., Townsend N., Buxton I., Cowley P., (2000), “Novelty Detection for The Identification of Abnormalities”, International Journal of Systems Science, 2000, Vol. 31, issue 11, pp.1427-1439.

Vapnik V.N., (1995), “The Nature of Statistical Learning Theory”, Springer, New York, 1995.

Watson I., Marir Farhi, (1994), "Case Based Reasoning: A Review", The knowledge Engineering Review, Vol.9, Issue. 4, December (1994) pp.327-354, Available at: http://www.ai-cbr.org/classroom/cbr-review.html (Accessed June 2010).

Widodo A., Yang B., (2007), "Support Vector Machine in Machine Condition Monitoring and Fault Diagnosis", Mechanical Systems and Signal Processing 21 pp2560-257.

Witten I.H., Frank E., (2005), "Data Mining Practical Machine Learning Tools and Techniques", Morgan Kaufmann Publications.

Zaher A, McArthur S.D.J, Infield D.G, Patel Y., (2009), "Online Wind Turbine Fault Detection through Automated SCADA Data Analysis" 2009, Wind Energy Journal Wiley & Sons Ltd. (Journal paper).

Zhang G.P., (2000), "Neural Networks for Classification: A Survey", IEEE transactions on Systems, Man and Cybernetics-Part C Applications and Reviews, November 2000, Vol. 30, Issue No.4.

Author Discussions

Yusuf Patel, (2007) Author discussion. Wind farm operator.

4. Multi-Agent Systems for Online Condition Monitoring

This chapter of the thesis aims to describe Multi-Agent technology and assess its suitability for the development of an online automated data analysis CM system. It will begin by providing the relevant definitions for the field and propose the reasoning behind why the technology is seen as being of value for developing such systems. It will take a look into technologies that are closest in functionality to MAS and discuss how they compare against one another. It will then proceed to describe the workings of the various components that constitute MAS and how they can be tailored to support processes that are specific to the developer's problem domain. It aims to highlight how the framework can give a developer the flexibility to build autonomous systems which can split their problem into sub-components (agents) which can negotiate with one another to achieve the overall desired objectives. Examples of a number of systems showing how the framework has been used to build similar CM data analysis systems found in the literature will also be discussed.

4.1 Overview

In order to put MAS's into perspective, a convenient and logical place to start is by addressing the question of "*what is an agent?*" (McArthur & Davidson et al 2007). Having posed this question, it seems inherently difficult to answer, since there is no single straight forward definition that can be found. The most referenced definition in the literature is that proposed by (Wooldridge 1999). He provides some high level definitions through two concepts, namely an *agent* and an *intelligent agent*. In its simplest sense, an agent is described as being a software or hardware entity that is capable of autonomously reacting to changes in its environment. Its means to doing so is by monitoring its environment, through sensors or other data and information sources. This definition offers a blurred description however as a wide range of existing systems such as protection relays, or even a thermostat, can fall under this description.

The concept of an *intelligent agent* builds further on the *agent* definition by taking into consideration how it reasons about its reaction to the environment. An intelligent agent must be capable of displaying flexible autonomy which is characterised by the mixture of a number of features namely: reactivity, pro-activeness and social ability. Reactivity can be best described as the agent's ability to react to changes in its environment, in a timely and appropriate manner, where some action based on the changes and the function it is designed to achieve, is carried out. Pro-activeness is the agent's ability to take responsibility for how to achieve its own goals, without explicitly being hard coded to do so. Social ability determines the agent's ability to interact with other agents, in terms of the various forms and complexity of interaction that it supports. The degree, to which an agent exhibits these characteristics, is determined by the level to which the developer builds them into the software. These three properties can be present in varying quantities, which directly affects the extent an agent demonstrates the characteristic of flexible autonomy.

A Multi-Agent system consists of more than one agent and intelligent agent combined within a co-operative system. The result is a system which is capable of dynamic re-organisation of its overall function, modifying its actions over time as it needs, in order to ensure that the overall desired objective is met. These dynamic and flexible properties are realised through the inherent capabilities that intelligent agents offer.

These specific characteristics allow agent technology to offer new methods of developing solutions to issues and problems in power applications through their offered suite of techniques and abilities. (McArthur & Davidson et al 2007) cite a number of different research areas where the technology has been applied including diagnostics (Davidson et al 2004), condition monitoring (McArthur & Strachan et al 2004), power system restoration (Nagata & Sasaki 2002), market simulation (Widergren et al 2004) and (Koesrindartoto et al 2005), and finally automation (Buse et al 2003).

While the white paper by (McArthur & Davidson et al 2007) cites a diverse range of applications, the system proposed in this thesis is concerned with the development of an automated CM system. This chapter will therefore focus mainly on how MAS characteristics can be used to benefit the development of such CM systems. In light of this, an important aspect which should be considered is to determine the challenges that are posed by CM systems. This in turn will allow us to understand the requirements of such an application as well as assess the suitability of Multi-Agent System technology as a platform for wind turbine CM.

4.2 Requirements of Data Analysis for CM Systems

Condition monitoring of equipment and plant items offers a number of challenges (McArthur & Davidson et al 2007). These challenges can be generalised since they are issues which are associated with the management and interpretation of data regardless of the specific plant type being monitored. The authors (McArthur & Davidson et al 2007) state that the challenges can be summarised as follows:

- Gathering data from a variety of sensors;
- Interpreting the data to extract meaningful information. This often requires the use of multiple algorithmic and intelligent system based approaches;
- Combining the evidence and information from different interpretation algorithms to generate an overall diagnostic conclusion;
- Delivering diagnostic information in the correct format, to relevant engineers; and
- Automatically altering power system and plant settings based on the conditions of the plant.

It can be seen from the list of points that these issues also concur with the data analysis aspects of wind turbine monitoring, just as they do for any plant type since they are general issues of data management and interpretation, and not specific to any one CM plant application.

Before exploring the potential benefits of MAS technology for CM applications, it is important at this point that methods which are comparable to Agent technology are explored in order to identify if there are other approaches that also lend themselves well to CM applications. While there are no direct alternatives that offer what the MAS framework offers, they have been compared to web services and grid computing. The similarities and differences are discussed in the following section.

4.3 Grid computing and Web services, Alternatives to Multi-Agent Systems?

The two other paradigms that are compared to Agent technology are Web services and grid computing. Grid computing is the application of several computers to a single problem at the same time. Grid computing depends on software, to divide and apportion pieces of a program among several computers. The primary advantage of grid computing is that each node can be purchased as commodity hardware, which when combined can produce similar computing resources to a multiprocessor super computer, but at a much lower cost. While a grid may be constructed with a specialised application in mind, they are often developed with the aid of general purpose grid software libraries and middleware, making them capable of accommodating the execution of a different range of problems.

Web services on the other hand are defined by the World Wide Web Consortium (W3C) as “a software system designed to support interoperable machine-to-machine interaction over a network”. They are often just web Application Programming Interfaces (API’s), that contain sets of functions, procedures and methods that can be accessed over the internet (or a network) and executed on a remote system hosting the requested services.

The commonality between the three technologies is described by (McArthur & Davidson et al 2007) as each of the technologies offering a perspective on the problems associated with distributed computing. They harness distributed hardware and software resources to complete a specific task, while supporting some form of messaging between their component parts. The differences between them on the other hand are prominent due to a number of factors. The first noted difference is in the scope of the application that each technology is focused on solving. Grid computing primarily tends to focus on harnessing the computational power available in the hardware resources of the grid, to solve computationally complex problems. Web services are designed to offer interoperability between software systems by providing the necessary mechanisms required for both the discovery of those systems and their communication across a network.

Initially, web services and MAS can seem similar. Often similar styles of interaction diagrams are used to describe both web services and agent interactions. The notion of modules providing a "service" acting as intermediaries for other modules to utilise and accomplish their objectives is common to both the technologies. However a distinct dissimilarity is noticed in the standards and support of a richer set of interactions, which allow negotiation to occur between modules that can be found in MAS. So while web services and MAS both share the functional ability to support interoperability between software systems, the interoperability is considered to be more restricted than that of MAS. These interactions will be discussed in greater detail in a later section in the chapter.

The principal difference that (McArthur & Davidson et al 2007) state exists between MAS, grid computing and web services, is the aforementioned notion of autonomy. The authors cite the reference for current standards (Huhns 2002) which they state has no provision for autonomy in web services. Similarly there is no requirement for the nodes in computational grids to exhibit autonomy. It is the cooperative and proactive nature of agents that set them apart from grid computing and web services. MAS technology has even been debated as a mechanism suited to deliver improved web and grid computing services (Huhns 2002).

Now that we have established that there is no direct alternative to the MAS framework, the remainder of the chapter takes a more in depth look at MAS technology in order to establish its suitability for the requirements described in the previous section.

4.4 The FIPA Platform, Multi-Agent System Standards

FIPA, the Foundation for Intelligent Physical Agents was established in 1996 as an international non-profit association to develop a collection of standards relating to software agent technology (Bellifemine F. 2007a). At that time software agents were already very well known in the academic community but even to this date they have only received limited attention from commercial enterprises beyond an exploratory perspective. The purpose of the FIPA consortium was to produce standards that would form the foundations of a new industry, by being usable across a vast number of applications.

At the core of FIPA the following set of principles are stated to exist by (Bellifemine 2007a):

1. Agent technologies provide a new paradigm to solve old and new problems;
2. Some agent technologies have reached a considerable degree of maturity ;
3. To be of use some agent technologies require standardisation ;
4. Standardisation of generic technologies has been shown to be possible and to provide effective results by other standardisation forums ;
5. The standardisation of the internal mechanics of agents themselves is not the primary concern, but rather the infrastructure and language required for open interoperation.

The evolution of FIPA led to the proposition of many agent-related ideas, which lead to the definition of standards revolving around the concepts of agent management and agent communication. The FIPA Agent Management Reference Model (shown in Figure 4.1) defines the framework in which FIPA compliant agents exist and operate. It provides the necessary reference model required for the creation, registration, location, communication, migration and retirement of agents (FIPA 2004).

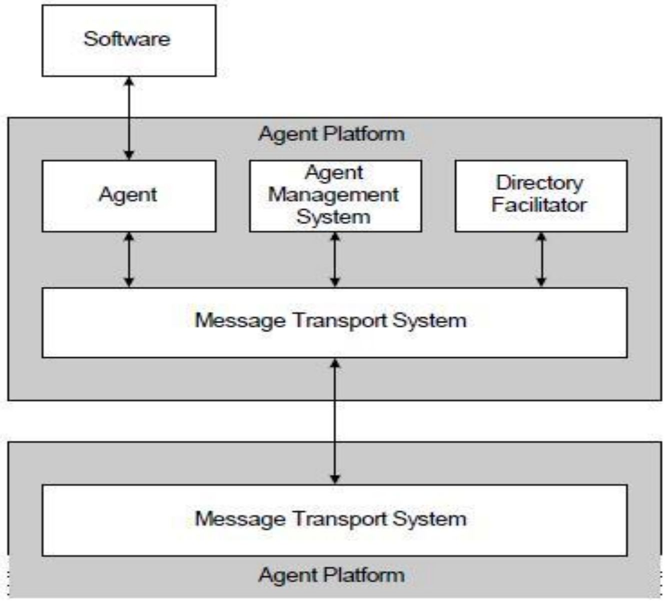

Figure 4.1: The FIPA Agent Management Reference Model (AMRM) (FIPA 2004).

The Agent Management Reference Model (AMRM) consists of a number of logical components namely two utility agents: the Agent Management System (AMS), and the Directory Facilitator (DF) as well as a message transport system. The AMS is a mandatory component of the agent platform. It provides supervisory control over access and use of the agent platform. Only one AMS can exist within a single platform. It also allocates agent identifiers (AIDs) to each agent that registers with it and provides a "white pages" like service, where an agent can ask for the address for another, whilst also monitoring the life cycle of each agent within the platform. The DF is an optional component within the agent platform. It provides a "yellow pages" service which allows each agent to register with it the services (tasks / information) it provides, along with the information or services it is interested in. In this way agents within the platform can query the DF, to find out what services are offered by other agents. The message transport service is the software component that supports the exchange of all messages within a platform, including messages to and from remote platforms as shown in figure 4.1. Agent communication will be discussed in further detail in a later section of the chapter.

4.4.1 The Java Agent DEvelopment framework (JADE)

JADE is an open source FIPA compliant agent platform which implements this agent abstraction over the object oriented Java programming language. It treats agents as collections of behaviours, where a behaviour, can be regarded as one element of functionality within a task (Bellifimine et al 2007b). Every JADE agent is composed of a single execution thread, and the behaviours run in a sequential manner, where one executes completely before another can run. This means more than two agents can execute their behaviours at the same time, but each agent can only have one behaviour running at any time. The behaviours are scheduled on a simple queuing mechanism, and it remains up to the agent designer to implement the behaviours in such a way that the correct one is run at the correct time. This round robin non pre-emptive scheduling policy is carried out on all behaviours that are in their ready state. This scheduling is hidden from the programmer, keeping agent management simple and efficient. Behaviours can therefore begin execution based on a check whether certain conditions have been met. If the condition has not been met, then the behaviour goes into a queue of blocked behaviours, allowing another behaviour in the ready queue to begin execution.

As a simple example, if we consider the scenario of an elevator in a building with two floors. The elevator itself as well as the user can be modelled as separate agents. For simplicity the elevator can have two behaviours, one that opens and closes its door, and the other moves the lift between floors. The user agent has only a single behaviour which is simply to make a call to the elevator to request its use. If the lift is on the top floor, and a user agent requests it on the ground floor, the elevator agent would execute its close door behaviour and once this is complete, execute its move behaviour in response to the user's request. During the process of the open/close door behaviour, the move behaviour would be in the ready queue waiting for the open/close door behaviour to complete since all behaviours run to completion without pre-emption. Once the move behaviour is in execution, the open and close behaviour cannot run, i.e. it will be in its blocked state. Only once the elevator is stationary, and at either floor one or two, is the open and close door behaviour available for execution in the ready queue once more. These preconditions essentially represent the conditions which must be satisfied, before a behaviour can execute.

Another important aspect to note which simplifies the development task, is how the JADE platform hides certain low-level detail from the developer (Catterson 2006). Two examples of this can be seen in the code snippets shown in figures 4.2 and 4.3. The first (figure 4.2) is the process of the agent registering with the DF. It can be seen that this is handled by simply creating a `ServiceDescription` object and calling the `register()` method. The second (figure 4.3), is the sending and receiving of messages is handled simply by calling send() and receive() methods. In this way the JADE platform ensures that it conforms to the FIPA reference model specification, whilst making the job of the engineer as convenient and simple as possible. This is indeed an important aspect of any engineering system, since conforming to standards plays an important role in aiding others' understanding of the system, allowing interoperability between independently developed systems. On the other hand, the specific detail of how an agent performs behind the scenes processing of tasks, (such as sending and receiving messages), is of no real interest to the developer, who is concerned only with the final result. This allows engineers to focus on the challenges of their application, designing the higher level actions and abilities of the agent, while keeping the need for knowledge of unnecessary detail to a minimum.

While this abstraction of agent messaging is present within the JADE platform, it does not remove the fundamental design aspects of agent communication from an agent designer. The following section will go on to describe the level of details of agent communication an agent developer is required to consider, when designing their system.

4.5 Agent Communication

Agent communication can be perceived as being split over three levels: the message transport protocol, the agent communication language (ACL), and the content language and ontology.

```
public class MyAgent extends Agent
{
    private Codec codec;      // These are described in greater
    private Ontology onto;    // detail later in this Chapter
    private DataStore store; // The agent's local data store

    protected void setup ()  // This method creates the agent
    {
        codec = new SLCodec ();
        onto = MyOntology.getInstance ();
        store = new DataStore ();

        /* Create a service description to send to the DF */
        ServiceDescription sd = new ServiceDescription ();
        sd.setType ("my_service");
        sd.setName (getLocalName ());
        DFAgentDescription dfd = new DFAgentDescription ();
        dfd.setName (getAID ());
        dfd.addServices (sd);

        /* Register this service with the DF */
        try {
            DFService.register (this, dfd);
        } catch (FIPAException fe) {
            fe.printStackTrace ();
        }

        /* Add a behaviour to do something */
        addBehaviour (
           new SampleBehaviour (this, codec, onto, store);
    }

    protected void takeDown () // This method destroys the agent
    {
        try {
            DFService.deregister (this);
        } catch (Exception e) {
            e.printStackTrace ();
        }
    }
}
```

Figure 4.2: An example of a JADE Agent created by extending the Agent Class (Catterson 2006)

```
public class SampleBehaviour extends CyclicBehaviour
{
    private Codec codec;

    /* The behaviour's constructor method */
    public SampleBehaviour (Agent a, Codec c, Ontology o, DataStore d)
    {
        super (a);
        codec = c;
        this.setDataStore (d);
    }

    public void action () // This is called when the behaviour is run
    {
        MessageTemplate mt = MessageTemplate.and (
             MessageTemplate.MatchPerformative (ACLMessage.QUERY_REF),
             MessageTemplate.MatchLanguage (codec.getName ()));
        ACLMessage msg = myAgent.receive (mt);

        if (msg == null) // There was no message matching mt
        {
            block ();
            return;
        }

        /* Create a response message and send */
        ACLMessage response = msg.createReply (); // set up headers
        response.setPerformative (ACLMessage.INFORM);
        response.setContent ("Load priority 1");
        myAgent.send (response);
    }
}
```

Figure 4.3: An example of a JADE Agent's behaviour. This particular behaviour extends the cyclic behaviour class which allows for repeated triggering (Catterson 2006).

The message transport protocol is a service provided by an agent platform, to transport FIPA-ACL messages between agents on any given agent platform and between agents on different agent platforms. Messages provide a transport envelope that comprises the set of parameters, detailing, for example the sender and receiver of the message. The general structure of a FIPA compliant message is depicted in figure 4.4 below (Bellifimine et al 2007a).

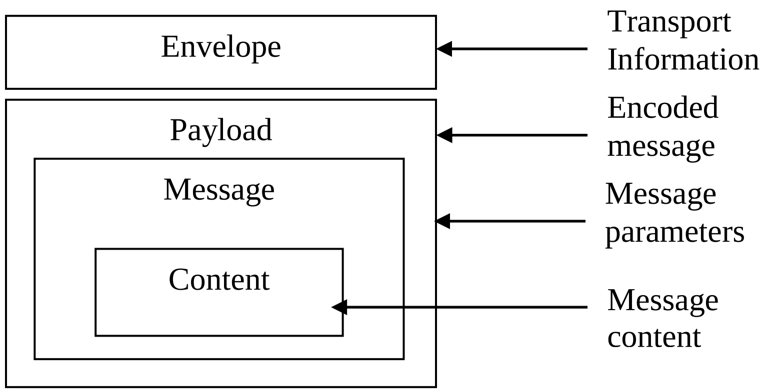

Figure 4.4: FIPA message structure (Bellifimine et al 2007a)

FIPA specify a number of message transport protocols, including the likes of HTTP and IIOP (FIPA 2002a and FIPA 2002b). FIPA specifies how these protocols can be used to transfer agent messages. This low level detail of messaging is taken care of by the agent platform, and it is not imperative that the developer is aware of the specific protocol being used and the precise details of its internal workings.

4.5.1 FIPA Agent Communication Language (ACL)

A FIPA-ACL message contains a set of one or more message parameters. These parameters are shown in table 4.1. FIPA defines communication in terms of a function or action called the communicative act or the performative. The only mandatory parameter in all ACL messages is the performative. This indicates the type of communication that is being attempted with the message. While the remainder of the other parameters in an ACL message are optional, it is expected that most will contain information regarding the sender, receiver and content parameters.

Parameter	Description	Category
Performative	Type of the communicative act of the message	Required
sender	Identity of the sender	Participant
receiver	Identity of the intended recipients	
reply-to	Which agent to direct subsequent messages to within a conversation thread	
content	Content of the message	Content
language	Language in which the content parameter is expressed	Content descriptor
Encoding	Specific encoding of the message content	
Ontology	Reference to an ontology to give meaning to symbols in the message content	
Protocol	Interaction protocol used to structure a conversation	Conversation control
Conversation-id	Unique identity of a conversation thread	
Reply-with	An expression to be used by a responding agent to identify the message	
In-reply-to	Reference to an earlier action to which the message is a reply	
Reply-by	A time/date indicating by when a reply should be received	

Table 4.1: ACL message parameters

This level of communication protocol is the second level in this three layer model presented, but incidentally, the first with which the developer is concerned with. The use of performatives allows the designer to decide the types of messages that can be sent between the agents, and determine how they interact (FIPA 2002e). Examples of some of the most common performatives and the communicative act they represent are shown in table 4.2. An idea of the type of conversations that can arise between agents can be garnered by glancing at the various communicative acts. As an example, the FIPA request interaction protocol as defined in FIPA 2002c allows one agent to request another to perform some action (see figure 4.5). The participant processes the request, and makes a decision whether to accept or refuse the request.

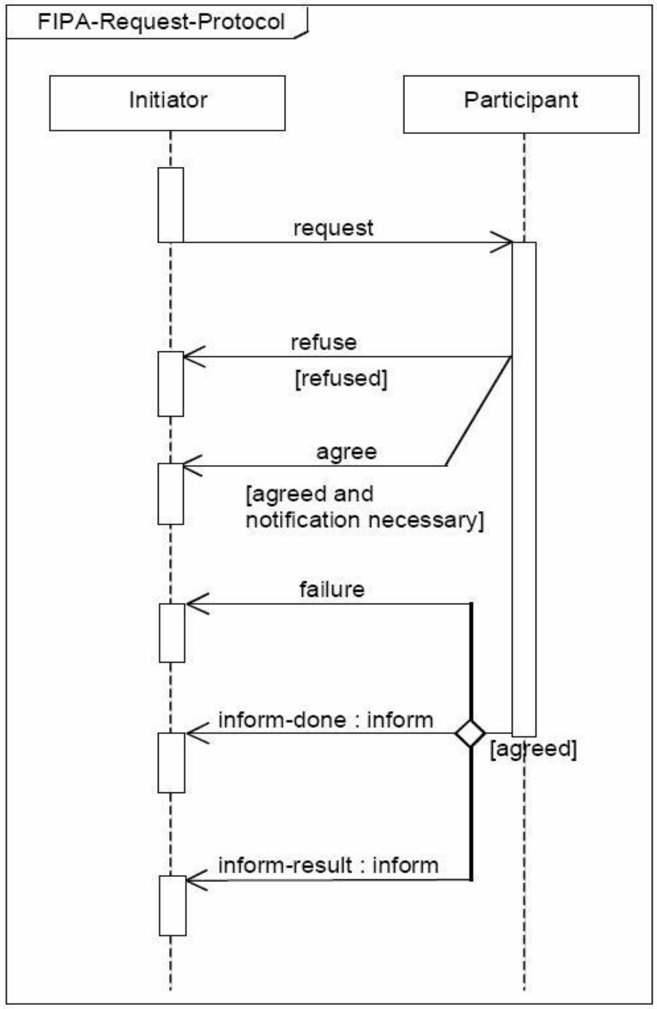

Figure 4.5: The FIPA Request Interaction Protocol (FIPA 2002c)

FIPA specify a number of interaction protocols that deal with pre-agreed message exchange protocols for ACL messages (FIPA IP Spec). The use of these interaction protocols is indicated by setting the **Protocol** ACL message parameter.

Multiple conversations between agents can take place at any one time. These are handled by the conversation control parameters specified in table 4.1. For example if an agent sets the **reply-with** parameter, the immediate reply message in the conversation copies this value into the **in-reply-to** parameter, while the **conversation-id** remains consistent throughout the conversation. When designing agents, it is up to the developer to decide which ACL parameters are required for their messages, and what they should contain in order for them to conform to the FIPA interaction protocols. While this level of communication will allow agents to interact carrying out conversations on a basic level, it is the specification of the ontology that dictates the topic and concepts which the conversations revolve around. This is discussed in the next section.

FIPA Communicative Act	Description
Agree	Agreeing to perform some action, e.g., in response to a Request
Refuse	Refusing to perform an action, with an explanation
Call for Proposal	Calling for proposals to perform some action
Failure	Informing an agent that an action was attempted, but failed
Inform	Informing an agent that a given proposition is true
Not Understood	An agent does not understand what another did, e.g., a message was sent but not understood
Propose	Submitting a proposal to perform an action
Query Ref	Asking another agent for the object referred to by a referential expression
Request	Requesting that an agent perform some action
Subscribe	Requesting persistent notification when a reference object changes

Table 4.2: Commonly used ACL communicative acts and the type of communication they indicate (FIPA 2002e)

4.5.2 Adapting MAS to a Problem Domain through Ontologies

An important aspect of building systems with the MAS framework is the flexibility they offer in allowing a developer to adapt them to deal with problems specific to their domain. This is essentially achieved through the third layer of the communication protocol model, (the second level that an agent designer is concerned with). This level involves the design of the concepts and terms that the agents utilise, known as the ontology. By defining the language and concepts that an agent system is associated with, the MAS framework can be tailored to solving problems that are specific to the developer's requirements, since the agents can begin to understand the significance of each piece of information and its context within the problem domain they are dealing with.

Ontologies were primarily developed "*to bridge the gap between what exists and the languages, both natural and artificial, for talking and reasoning about what exists*" (Sowa 1999). Essentially they were used to show how concepts were related to one another by categorising like entities and listing the common properties of each category**.**

Naturally when developing MAS's, a more practical approach is to design an ontology that encompasses only the general concepts which are immediately related to the developer's problem domain. There is no need for an agent ontology to feature a complete classification of concepts. The lack of a general ontology that can be used for agent messaging means that there is no FIPA standard for governing ontologies. The vocabulary used by any agent is not generic enough to define a general specification for all agent ontologies, therefore, developing a specialised ontology at the system specification stage of development, is a more convenient approach for an agent designer.
The ontology development 101 tutorial (Noy & McGuiness), defines some fundamental ideologies for ontology design, which essentially revolve around the notion that there is no single correct way to model a domain. The best solutions tend to depend on the application that the developer has in mind, and the extensions and concepts that can be anticipated. The other general rule stated which should be observed, is that concepts in the ontology should be based closely to objects both physical and logical in the field being modelled. Therefore it is apparent that in order to design an ontology suited to a particular domain, thorough knowledge of the domain as well as the sort of messages and information that will be exchanged between the agents, is necessary.

4.5.2.1 FIPA Content Languages, Specifications for Defining an Ontology

The ACL specifications discussed in the previous sections do not confine what should be in the content parameter of a message, but rather dictate the format of the content. The content descriptor parameters in a message namely: the content language, ontology and encoding, entirely define the format of a message's content. The encoding simply provides a standard for how characters in a message string are represented; whereas the content language and ontology provide the syntax and semantics of the content respectively. FIPA supports a total of four content languages (FIPA 2003) namely: FIPA-SL (Semantic Language), KIF (Knowledge Interchange Format), RDF (Resource Description Framework) and CCL (Constraint Choice Language). Each of these languages is developed with different purposes in mind. KIF is largely tailored towards data transfer without the capability of representing actions that agents can perform. CCL is focused on representing constraint satisfaction problems, by providing the ability to describe variables, the domain of possible values for those variables, and the constraints placed on pairs of variables. RDF and SL were both developed with a more general purpose in mind. They provide general grammars for expressing data and actions, making them suitable for communication in a complex and dynamic multi-agent system. Only FIPA-SL however, has reached a stable standard (FIPA 2002d), while the other three content languages are still in an experimental phase. It is largely due to this and its more generic nature, that this thesis will adopt the use of FIPA-SL as the chosen content language.

The chosen content language influences and shapes the structure of the ontology. The FIPA-SL content language allows the developer to define an ontology through the use of a list of concepts, predicates and actions. The concepts model domain concepts, for example in the case of a wind farm typical concepts might be a *wind turbine*, a *gearbox*, a *generator*, or even intangible concepts such as a type of *data measurement*, a *fault*, or type of *alarm*. Predicates specify concept relationships, and can always be evaluated to true or false. An action is a special type of concept, specifically developed for communicative acts such as Request and Call for Proposal, where agents discuss an event happening for example.

It can be seen that the MAS development environment is fairly extensive, offering new and interesting solutions to solving both new and existing problems. The JADE API provides a level of control to the developer which allows them to build complex systems, without the inconveniences of having to provide solutions to the lower level aspects of MAS design and development. The next section provides a review of some applications where MAS have been applied to in the recent years in the area of CM and automation.

4.6 A Review of Multi-Agent Systems CM solutions

This section provides a review of how agents and their properties are being used in the research literature. While they have been utilised for a wide range of applications as is described in the white paper by (McArthur & Davidson et al 2007). This section will focus on how their characteristics can benefit the development of automated data analysis for CM systems.

Agent based solutions in the area of monitoring and diagnostics is mostly concerned with assisting engineers with data management and interpretation of plant data, through simplified data gathering, reducing overall data flow, and automatically extracting useful information from the raw data collected from the CM systems. An example of this is the Protection Engineering Diagnostic Agents system (Hossack et al 2003). It uses agent technology to integrate a legacy SCADA interpretation system and new data retrieval systems to automatically collate and analyse power system data, relating to protection operation. This reduces the extent of manual effort required to determine the cause of certain incidents, by prioritising digital fault recorder data based on analysis of live SCADA data, and presenting the underlying links between various system events. Rather than simply managing this flow of data by filtering, PEDA uses multiple data sources to extract a higher level of meaning, presenting interpreted information to the user, instead of just raw data.

The system comprises of an Incident and Event Identification (IEI) agent, a Fault Record Retrieval (FRR) agent, a Fault Record Interpretation agent (FRI), the Protection Validation and Diagnosis (PVD) agent, the Collation (CA) Agent and finally the Engineering Assistant (EAA) Agent. Each of these agents has its own specific skill set and forms part of the overall diagnosis and interpretation process. By splitting the system into a number of different independent agents, the developer can utilise the benefits of the agent framework allowing multiple data sources to be independently analysed through one system and their results easily combined. This feature provides the potential for identifying matching trends in multiple parameters to strengthen evidence for fault detection or diagnosis using a number of different processing techniques. This is particularly useful for most CM applications since items of plant or machinery are typically monitored using a number of different sensing technologies leading to multiple data streams. For example in the PEDA system the IEI agent wraps an expert system which uses the rules within its rule base, to identify and classify power system incidents such as disturbances and switching operations through SCADA data. The FRR agent connects and retrieves the utility's fault record database, while the FRI agent also makes use of a rule-based expert system for classifying and interpreting digital fault recorder data. The EAA agent provides the information from the remainder of the agents, via a customisable user interface to the engineers.

It can be seen that the agents within the system all work collectively, carrying out their own part of the diagnostic process to provide useful, interpreted information, from the raw data to the end user. An interesting point to note is that PEDA was designed to conform to FIPA standards for agent messaging, using both FIPA-ACL and FIPA-SL for the communication language and content language respectively. By adhering to the standards, this served as the main driver for successfully interfacing the legacy systems within its framework.

Another example of an agent system designed for data interpretation is the Condition Monitoring Multi-Agent System (COMMAS), developed for use in transformer monitoring (McArthur & Strachan et al 2004). The paper describes how a multi-agent system was designed to employ the data generated by Ultra High Frequency (UHF) monitoring of partial discharge activity. The authors include an interesting discussion on the functional requirements of a UHF monitoring system. A number of requirements are derived from the issues which are associated with UHF monitoring and are stated in the following manner:

- Automatic capture and conditioning/formatting of relevant data;
- Automatic interpretation of the conditioned/formatted data to identify incipient and serious defects;
- Discrimination between a sensor failure and an actual plant failure. (Achieved through corroborating the interpretation results and sensor data with other relevant data sources.
- Provision of clear and concise defect information and remedial advice to the operational engineers;
- Finally extensibility and flexibility to include further interpretation techniques and monitoring technologies.

It can be seen that the requirements discussed in section 4.2 have been slightly tailored to accommodate the specifics of the UHF transformer data interpretation application. The guidelines described by McArthur et al can in fact be applied to the general data interpretation CM application, as will also be shown in the application presented in this thesis. The slight dissimilarities arise in the specific data flow and processing required by each different application only. The authors of COMMAS detail the design process, describing how the agent system was split into separate tasks and subtasks, assigned to specific agents and their behaviours.

Another point which should be noted is how the agent designers split the agent architecture over 4 layers, namely a data monitoring layer, an interpretation layer, a corroboration layer and finally the information layer. The raw data can therefore be seen to take a path flowing from the data monitoring layer, passing through each of the layers, where each of the agents present in every layer process the data in some way, so that at the information layer, only useful information is presented to the user. The use of agents in this CM system benefit the solution in that a number of different interpretation processes can be easily adopted throughout the processing of the data, which is implemented by the agents in the interpretation layer. This layer allows the bringing together of a number of artificial intelligence based diagnostic techniques that interpret the data in their own specific way leading to a particular diagnosis. In this way a more accurate diagnosis is achieved especially if the differing techniques produce the same result. The corroboration layer then concludes the final result of the system so that only one diagnosis (the most likely, based on the evidence supplied by the data) is presented to the user.

Another system developed by (McArthur & Booth et al 2005) is an agent based anomaly detection architecture designed for the purpose of condition monitoring. The paper describes how MAS technology can be used as the underpinning platform for a system which can learn the characteristic behaviour of an item of plant over time and alert engineers to unusual behaviour instances, even in the absence of knowledge of plant failure modes. The agent based approach allows for the addition of different data processing techniques to be applied in an online manner while the system is operating.

The notion that new interpretation techniques can be introduced at any time and incorporated into the existing system, is seen as a strong advantage for CM / data interpretation systems. This particular characteristic also strengthens the motives behind employing the technology for the system presented in this thesis.

From this review it can be seen that there are a number of features which the MAS framework employs that are favourable for the development of CM data analysis systems. These points are summarised by the two points below:

- Their ability to allow for the analysis and interpretation of multiple data sources using independent processing techniques combining all the results and evidence found in one system.

- And the ability to introduce new interpretation techniques on the fly at runtime to the system as agents into the system automatically forming themselves as the part of the overall system without the need to modify any already existing agents within the system.

If we consider the monitoring of wind turbines we can also see that these features would prove extremely useful for a CM system developed for wind farms. There are various sensors mounted on the internal components of wind turbines used to monitor them. For example the aforementioned sensor and measurement technologies in chapter 2 such as oil analysis sensors, temperature sensing, wind speeds and active power output etc. The use of MAS technology allows individual sensors and information sources to be combined in the condition monitoring/data interpretation and diagnosis process. Importantly it allows information to be used when it is available or relevant. For example the monitoring of a specific component might require a number of measurement readings from different parameters taken at different time points to determine the evolution of the component with respect to the factors influencing it. This data must be collected as and when it is created and then formatted in a way in which the data processing agent that is passed this information can understand. Once this information is ready, it can be sent to a data processing agent, which then decides when a significant event needs to be relayed to the engineer or operator. The agent determines when such information should be communicated and to whom (other agents or the user). This approach allows the flexible integration of as much diagnostic data, information and knowledge as is available. It also permits new sensors and data analysis / interpretation algorithms to be introduced seamlessly into the overall system, since there is no higher level central systems integration control.

It can be seen that MAS technology is capable of manipulating distributed information and knowledge which lends itself particularly well for a wind farm based CM application.

4.7 Chapter Summary

While the multi-agent framework has not been adopted widely for CM applications in the literature, the features they provide allow for the development of advanced automated data processing paradigms, which are capable of processing multiple streams of data from independent data sources to provide robust data interpretation solutions. The two key issues hindering the development and implementation of condition-based maintenance schemes with regards to wind farms and transformer monitoring (as has been showcased by the COMMAS system (McArthur & Strachan et al 2004) discussed in section 4.6), are issues concerning the overwhelming volume of data acquired by CM systems, and the second is an issue of robust and accurate data interpretation. The author's choice of adopting the multi-agent framework as an advanced multi-data processing paradigm is an attempt at solving the volume of data issue. The key to such a problem would clearly benefit from automation in a manner which is both robust and can adapt to a changing environment without the need for user intervention. The use of agents allows developers to achieve this notion of autonomy within their systems by automating the data processing procedure. This removes the tedious and practically "impossible" nature of analysing these vast streams of data collected manually.

The ability to develop agents which are solely focused at interpreting a specific stream of data lends itself well to the multiple-sensor nature of current CM systems installed in wind turbines.

The information presented in this chapter has illustrated the requirements of the data analysis aspect associated with CM applications. It therefore elucidates why the author perceives MAS as a suitable solution for an online fault detection system. The next chapter explains the normal behaviour models developed for the purpose of detecting abnormal behaviour in the SCADA parameters. The chapter details the subject of data selection, data normalisation, training, validating and testing processes of the models developed and how each of these phases has a bearing on model performance.

4.8 Chapter References

Bellifemine F., Caire G., Greenwood D., (2007a), “Developing Multi-agent systems with JADE”, John Wiley & Sons Ltd, 2007, pp. 10-27.

Bellifimine F., Caire G., Trucco T., Rimassa G., “Jade Programmers Guide”, (2007b), copyright Telecom Italia 2007. pp.24

Buse D.P., Sun P., Wu Q.H., Fitch J., (2003), “Agent-based substation automation”, IEEE Power and Energy Magazine, Volume 1, Issue 2, Mar-April 2003, pp.50-55.

Catterson V., (2006), “Engineering Robustness, Flexibility, and Accuracy into a Multi-Agent System for Transformer Condition Monitoring”, Thesis submitted September 2006, Institute for Energy and Environment, University of Strathclyde.

Davidson E.M., McArthur S.D.J., McDonald J.R., Cumming T., Watt I. (2004), “Automating the analysis and management of Power System Data using Multi-agent Systems Technology” Applications and Innovations in Intelligent Systems XII(AI2004), Springer 2004, pp.151-164

Dimeas A.L., Hatziargyriou N.D., (2005), “Operation of a Multi-agent System for Microgrid Control”, IEEE Transactions on Power Systems, Volume 20, Issue 3, Aug 2005, pp.1447-1455

Foundation for Intelligent Physical Agents, (FIPA 2002a), “FIPA Agent Message Transport Protocol for HTTP Specification”, 2002. Available from http://fipa.org/specs/fipa00084 (cited on page 29)

Foundation for Intelligent Physical Agents, (FIPA 2002b), "FIPA Agent Message Transport Protocol for IIOP Specification", 2002. Available from http://fipa.org/specs/fipa00075 (cited on page 29)

Foundation for Intelligent Physical Agents, (FIPA 2002c), "FIPA Request Interaction Protocol Specification", 2002. Available from http://www.fipa.org/specs/fipa00026/SC00026H.pdf (cited on page 1).

Foundation for Intelligent Physical Agents, (FIPA 2002d), "FIPA SL Content Language Specification", 2002. Available from http://fipa.org/specs/fipa00008 (cited on pages 33, 104)

Foundation for Intelligent Physical Agents, (FIPA 2002e), "FIPA Communicative Act Library Specification", 2002. Available from http://fipa.org/specs/fipa00037 (cited on pages 20, 30)

Foundation for Intelligent Physical Agents, (FIPA 2003), "FIPA Content Language Specifications", 2003. Available from http://fipa.org/repository/cls.php3 (cited on pages 20, 32)

Foundation for Intelligent Physical Agents, (FIPA 2004), "FIPA Agent management Specification, 2004", Available from http://fipa.org/specs/fipa00023 (cited on pages 20,24,69)

Foundation for Intelligent Physical Agents, (FIPA IP Spec), "FIPA Interaction Protocol Specifications". Collection of Specifications Available from the following address: http://209.85.229.132/search?q=cache:http://fipa.org/repository/ips.html

Hossack J. A., Menal J., McArthur S. D. J., and McDonald J. R., (2003), “A Multi-Agent Architecture for Protection Engineering Diagnostic Assistance”, IEEE Transactions on Power Systems, Volume 18, Issue 2, pp.639-647, May 2003. (Cited on pages 2, 38, 40)

Huhns M.N., (2002), “Agents as Web services”, IEEE Internet Computing, Volume 6, Issue 4, July-Aug 2002, pp.93-95.

Koesrindartoto D., Junjie S., Tesfatsion L., (2005), “An agent-based computational laboratory for testing the economic reliability of wholesale power market designs”, IEEE Power Engineering Society General Meeting, Volume 1, June 12-16, 2005, pp.931-936

Koritarov. V. S., (2004) “Real-World Market Representation with Agents”, IEEE Power and Energy Magazine, pp. 39-46, July/August 2004. (cited on page 37)

Korbik A., McArthur S.D.J, Ault G.W, Burt G.M., McDonalr J.R., (2005), “Enabling Active Distribution Networks Through Decentralised, Autonomous Network Management”, Proc. 18th International Conference on Electricity Distribution (CIRED), Turin June 2005.

Lenat. D. B., (1995), “CYC: A Large-Scale Investment in Knowledge Infrastructure”, Communications of the ACM, Volume 38, Issue 11, pp.33-38, November 1995. (cited on page 34)

McArthur S.D.J, Strachan S.M., Jahn G., (2004) “The Design of a multi-agent transformer condition monitoring system”, IEE Transactions on Power systems, Volume 19, Issue 4, Nov. 2004, pp.1845-1852

McArthur S. D. J., Booth C. D., McDonald J. R. and McFadyen I. T., (2005), “An Agent-Based Anomaly Detection Architecture for Condition Monitoring”, IEEE Transactions on Power Systems, Volume 20, Issue 4, pp.1675-1682, November 2005. (Cited on page 38)

McArthur S.D.J., Davidson E.M., Catterson V.M., Dimeas A.L., Hatziargyriou N.D., Ponci F., Funabashi T. (2007), “Multi-Agent Systems for Power Engineering Applications—Part I: Concepts, Approaches, and Technical Challenges “, T. Power Systems, IEEE Transactions on Volume 22, Issue 4, Nov. 2007 pp.1743 – 1752

Motto A. L. and Galiana F. D., (2002), “Equilibrium of Auction Markets with Unit Commitment: the Need for Augmented Pricing”, IEEE Transactions on Power Systems, Volume 17, Issue 3, pp.798-805, August 2002. (Cited on page 37)

Nagata T., Sasaki H., (2002), “A multi-agent approach to power system restoration”, IEEE Transactions on Power Systems, Volume 17, Issue 2, May 2002, pp.457-462

Nagata T., Tao Y., Sasaki H., Fujita H., (2004), “Decentralised approach to power system restoration by means of a multi-agent approach”, Bulk Power system Dynamics –VI, August 22-27, 2004, pp. 673-678, Cortina d’Ampezzo, Italy.

Noy N. F., McGuiness D. L., “Ontology Development 101: A Guide to Creating Your First Ontology”, Tutorial written at Stanford University, Stanford CA, 94305, Available at:http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

Sheng S., Li K. K., Chan W. L., Xiangjun Z. and Xianzhong D., (2006), “Agent-Based Self-Healing Protection System”, IEEE Transactions on Power Delivery, Volume 21, Issue 2, pp610-618, April 2006. (Cited on page 36)

Staszesky D. M., Craig D., and Befus C., (2005), “Advanced Feeder Automation is Here”, IEEE Power and Energy Magazine, Volume 3, Issue 5, pp.56-63, September/October 2005. (Cited on pages 3, 39, 42)

Sowa. J. F., (1999), “Signs, Processes, and Language Games: Foundations for Ontology”, In International Conference on the Challenge of Pragmatic Process Philosophy, University of Nijmegen, Netherlands, May 1999. Invited paper Extended version available from http://www.jfsowa.com/pubs/signproc.htm (cited on page 33)

Tomita Y., Fukui C., Kudo H., Koda J., and Yabe K., (1998), “A Cooperative Protection System with an Agent Model”, IEEE Transactions on Power Delivery, Volume 13, Issue 4, pp.1060-1066, October 1998. (Cited on page 37)

Wei P., Yan Y., Ni Y., Yen J., and Wu F. F., (2001), “A Decentralized Approach for Optimal Wholesale Cross-Border Trade Planning Using Multi-Agent Technology”, IEEE Transactions on Power Systems, Volume 16, Issue 4, pp.833-838, November 2001. (Cited on page 37)

Widergren S.E, Roop J.M, Guttromson R.T, Huang Z., (2004), “Simulating the dynamic coupling and market physical system operations”, IEEE Power Engineering Society General Meeting, Volume 1, 6-10 June 2004, pp.748-753

Wooldridge M. (1999) “Intelligent Agents”, In G. Weiss, editor: Multi-agent Systems, The MIT Press, April 1999, pp.3-51.

Yokoi T., (1995) "The EDR electronic dictionary," Communications of the ACM 38:11, pp.42-44.

5. Methods and Applications I: Training, Testing and Validating Normal Behaviour SCADA Models

The previous chapters have described the rationale behind the MAS technology and NN approaches selected to develop an automated fault detection system capable of analysing wind farm SCADA data to provide early component fault identification. This chapter aims to describe the detailed process involved in training and testing the NN-based normal behaviour models for the selected SCADA parameters, namely the gearbox oil temperature, gearbox bearing temperature and the generator winding temperature for fault detection. A power estimation model will also be detailed for the purposes of corroboration with the fault detection models to be used within the MAS framework developed, described in chapter 6.

The importance of the 3 parameters to be used for fault detection and their associated components has already been determined. Successfully identifying faults in their incipient stages within these components while providing sufficient time for a maintenance decision to be made would prove extremely helpful to wind farm operators. This requires a model which is capable of accurately capturing the normal operational behaviour of the SCADA parameters in order to have the ability to identify subtle deviations early on. The application of NN to similar wind turbine gearbox bearing and oil temperature data by (Garcia et al 2006), is to the best of the authors knowledge, the only application of NN to this problem found in the literature at the time of writing. As mentioned in chapter 3 previously however, (Garcia et al 2006) do not describe the model training procedure in any way. It is therefore important to note that this thesis does not claim the use of NN as a novel means to normal behaviour modelling of gearbox SCADA parameters. Rather it illustrates novelty in how the gearbox oil and bearing models were developed, tested and evaluated noting the improvement in the results presented in the case studies of this chapter and published in (Zaher et al 2009). It also demonstrates novelty in its application to the generator winding temperature parameter to explore the success of this approach to capturing its normal behaviour.

There are a number of stages associated with training and testing NN models, these involve data preparation and selection, Training the models, validating the model inputs and finally testing on previously unseen data. This chapter provides the details for each of these stages clearly setting out the normal behaviour model development process along with the results produced.

5.1 Data Preparation and Selection

The first step involved in developing the normal behaviour models is the selection of data to be used to train the models. It is important that the data used is representative of the expected normal behaviour of the turbine under its complete operational range. The turbines modelled were the Bonus fixed-speed stall-regulated machines (Bonus Energy 2009), which theoretically, at rated power should produce a maximum of 600kW.

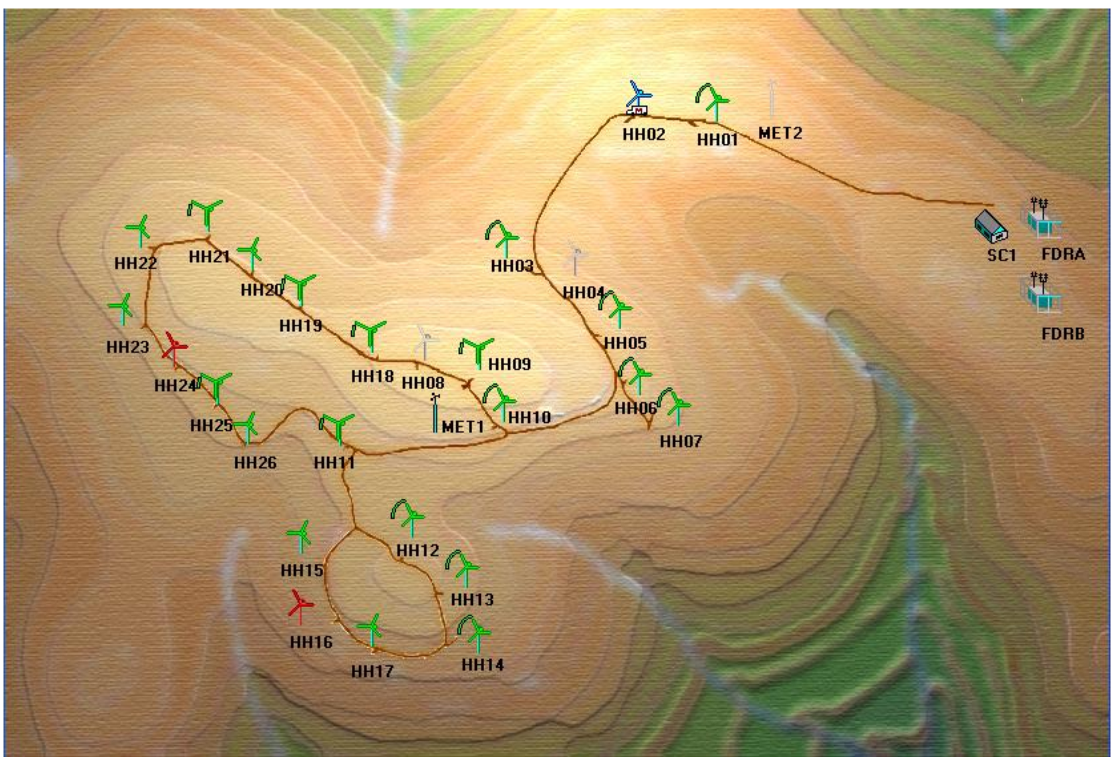

Figure 5.1: Hag Shaw Hill Wind Farm Turbine layout

The 26 turbines at the wind farm as shown in figure 5.1 are spread out over a large area of land, each experiencing their own specific conditions due to the differing locations of each turbine. These site specific external factors include: roughness of the site terrain, topography of the land, obstacles and wake effects caused by neighbouring turbines (Leany et al 1997). The terrain surface roughness reflecting vegetation and buildings in the proximity for example can influence the wind incident on the turbine. The lower the roughness (i.e. smoother surfaces such as sand, water and snow), the less the wind speed is impeded and the greater the output of the turbine and vice versa. In addition orographic effects such as hills and cliffs, exert a marked influence on the wind speed. The wind speed increases or decreases near the crest or foot of these features respectively, therefore the physical location of the turbine in relation to the hill, cliff or valley will also affect the wind profile a turbine experiences. Wake effects from neighbouring turbines can also distort the stream of wind passing through a turbine, increasing turbulence and decreasing the mean wind speed.

Because of these imposing complex external conditions, it was decided that both turbine specific models as well as a generic model would be developed and tested. The specific model would be trained on turbine specific data used to capture the specific imposing environmental factors affecting each individual turbine, where such characteristics should be reflected through each turbine's data. The generic model would be trained to determine whether or not these specific characteristics would affect the performance of fault detection if a generic data set, not specific to any particular turbine, was used. The benefit of using a generic model would be to considerably reduce the overall volume of data requiring processing given that the system would have to incorporate only 3 fault detection models for all 26 turbines against 78 (3 x 26) models if the turbine specific approach is used. Such a large number of models would be considered taxing for a centralised online fault detection system to process this volume of data in real-time. When considering applying such a system to more newly developed wind farms of a larger size, the amount of processing could become prohibitive for the typical centralised architecture of the data processing systems currently installed in wind farms (see figure

6.9 chapter 6). The results of both a turbine specific model and the generic model will be compared against the same data set to evaluate their performance. By comparing the results of the two models, the degree to which the generic model affects fault detection performance can be evaluated and a decision towards its suitability for accurate fault detection can be determined.

The Bonus 600kW turbine has a theoretical (manufacturer rated) cut in speed of 3-4m/s wind speed, achieves rated power at 16-17m/s and cuts out at approximately 25m/s (Bonus Energy 2009) shown in figure 5.2. By plotting the active power against the corresponding wind speed for all of the turbines the actual operational range could be compared with the theoretical manufacturer power curve to identify the actual bounds of operation. Figure 5.3 shows the power characteristic (power against wind speed) of T3.

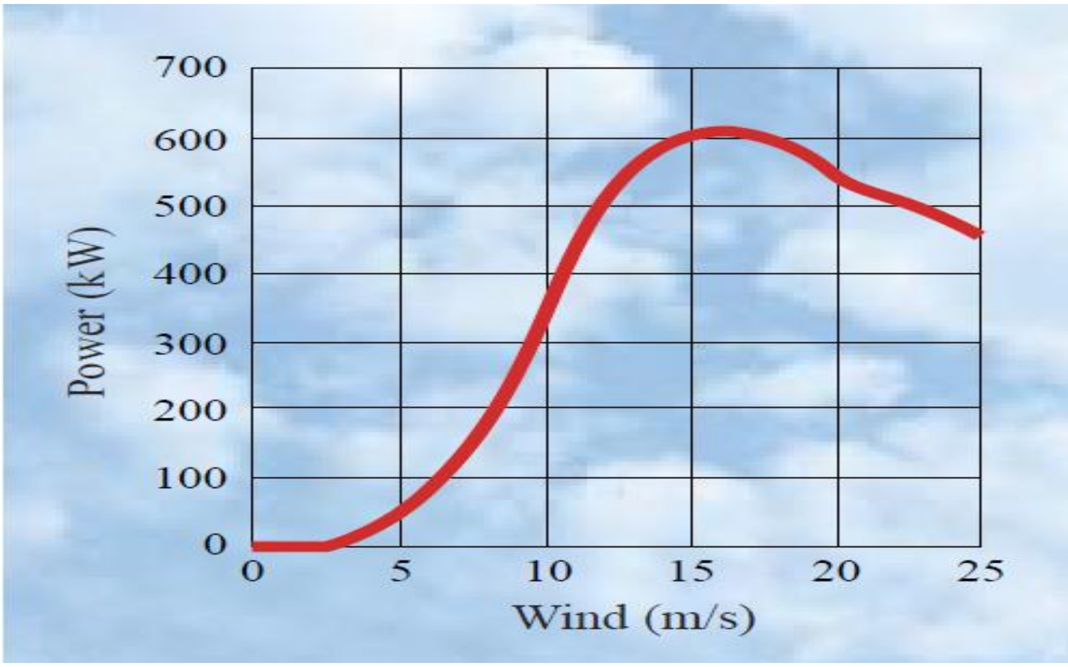

Figure 5.2: Bonus 600kW Manufacturers Power Curve

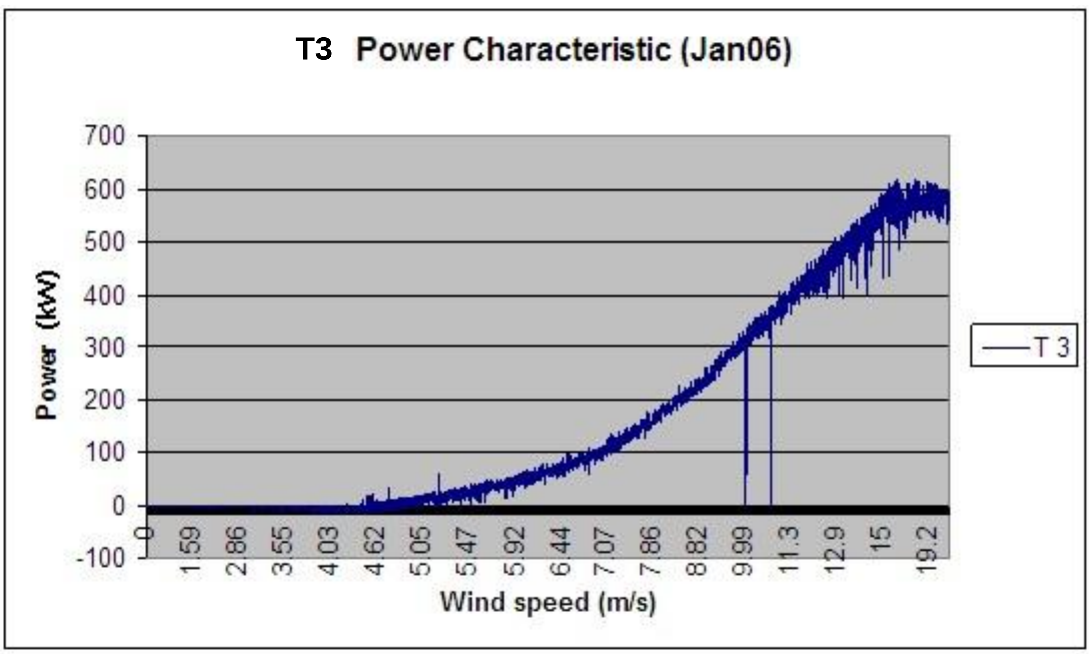

Figure 5.3: T3's Power Characteristic for the month of January 06 (Data is sorted in ascending order for the month)

As depicted in figure 5.3, it can be seen that the manufacturer's theoretical power curve (figure 5.2) is a relatively good match to the actual performance of the turbine.

When visually inspecting the data, a number of features were found which had to be considered before data from any particular turbine was used for training.

- Feature 1: The first aspect apparent in the data is periods where the SCADA telemetry is switched off i.e. no data is recorded. Throughout these durations, all parameters recorded for the particular turbine are zero in value. Through consultation with the project's industrial partners (Yusuf Patel, 2007) it was said that this could be interpreted as one of two possible scenarios. The first interpretation could be the turbines going under scheduled maintenance where the turbines are switched off before such an operation. The second case could be due to the turbine experiencing isolation from the electrical grid where it becomes temporarily disconnected; this is referred to as "islanding". This results in all power and communication to and from the machines being cut off, resulting in no data being recorded for the related duration.

- Feature 2: The second aspect found in the data is periods where the active power output is zero, but the remainder of the parameters are not. Through consultation the author was informed that this could be attributed to the turbine experiencing a failure, where the telemetry continues to record data up to and after the point of failure. Another alternative could also be that the turbine has been taken offline for unscheduled maintenance due to some problem but the telemetry system has not been switched off for the associated duration.

Processing the complete data set to identify any "interesting" instances of failure as described in feature 2 above requires an accurate anomaly detection mechanism that can provide a means to discover faulty scenarios.

When constructing the data set for both the specific turbine and generic model it was considered important to attempt to avoid using data from months which exhibited instances which fell under feature 2. By avoiding using data which leads up to instances of failure (which would almost certainly represent faulty operation) to train the normal behaviour models, a more accurately developed model of “normality” could be anticipated.

At the same time however, a turbine which experienced some form of failure would prove essential for test purposes. Because of this it was important to identify if any of the turbines experienced such a failure, and then utilise data from earlier months before the incident occurred which appear to be normal when compared to other operational neighbouring turbines for training. Again by trending all parameters for all turbines, visual inspection allowed for the identification of a potential problem in turbine 16 in the month of January 06 which conformed to the characteristics of feature 2 shown in figure 5.4 (a and b) below.

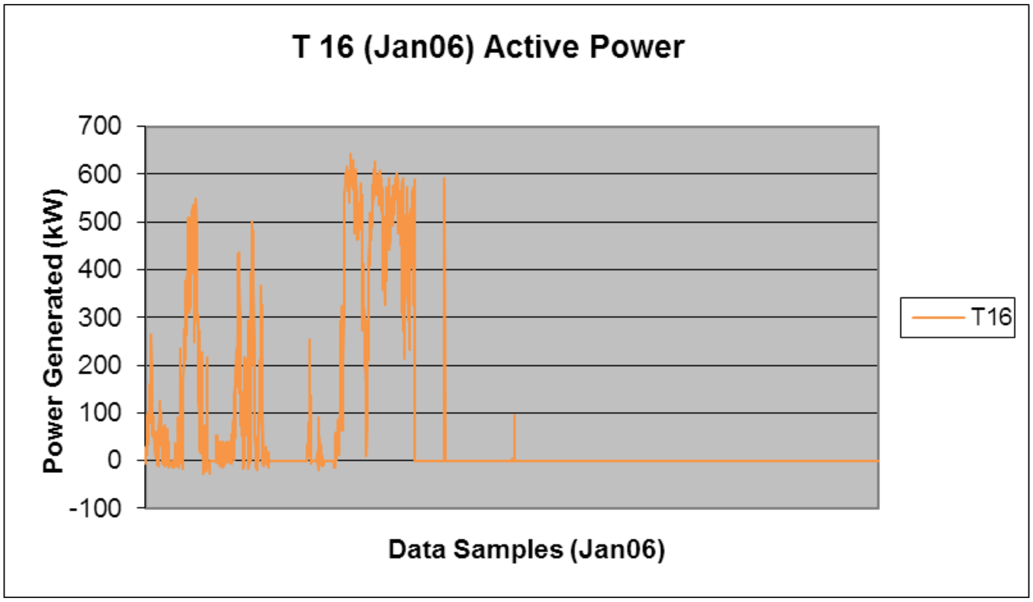

Figure 5.4a: Active power falls to zero once fault occurs

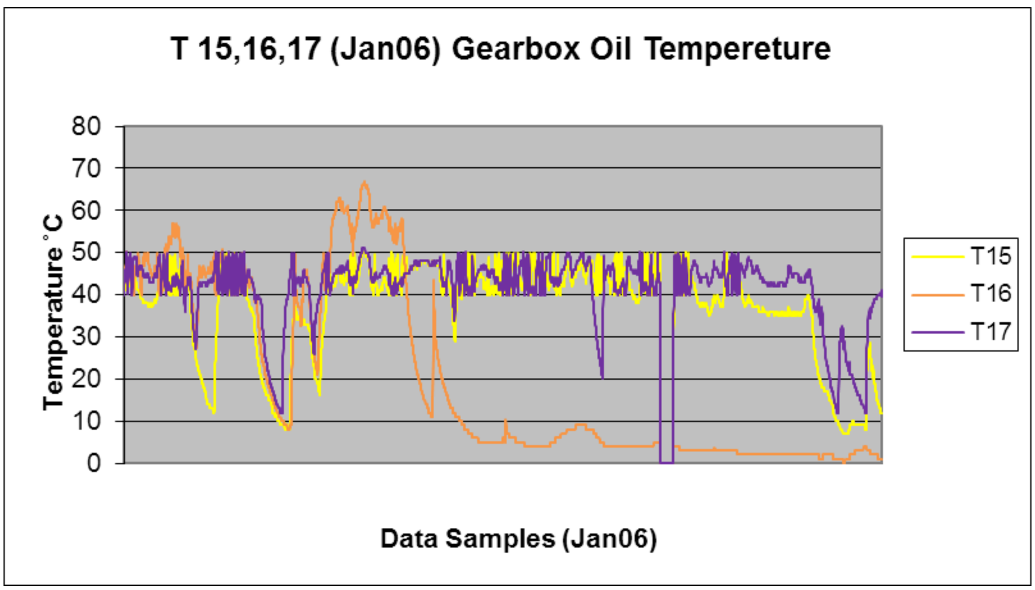

Figure 5.4b: gearbox oil temperature of T16 compared to neighbouring turbines

The trend in figure 5.4a above shows that the turbine is producing power where activity is shown in the first half of the figure. It abruptly drops to zero however for the remainder of the month. From figure 5.4b below, it can be seen that the gearbox oil temperature for T16 begins to decrease at the same point in time as the power drops to zero.

The trend for T16 is also noticeably running excessively higher than that of T15 and T17 under the same conditions (where T15 and T17 are neighbouring turbines see figure 5.1). This suggests that this is in fact a gearbox problem which turbine 16 is experiencing. The trend for T16 does not go to zero immediately as with the active power output (as described in feature 1), but gradually decreases as the turbine is experiencing no activity (no power generation) during its failed state, therefore leading to the temperature decreasing.

Another point worth noting is the operating temperature of T15 and T17. Both of the turbines' gearboxes are operating between the range of 40 and 50°C for the majority of the month. When compared with the remainder of the turbines in the data set, they too were found to reveal operating temperatures within the same range. This suggests that a normal healthy gearbox should operate around such temperatures when the turbine is generating power. When observing the active power output of turbine 16 in figure 5.4a, it is also clear that the power output is extremely variable in nature. This is understandable since it can be attributed to the constant wind speed and environmental changes the turbine experiences. The corresponding gearbox temperatures however are much less varying in nature thus making for a more predictable parameter for function approximation.

The process of selecting data from an *unlabelled* data set can be considered an undefined process. It mainly consists of a visual analysis and interpretation of the data which can lead to an element of subjectivity. There is no analytical method associated with such a process and the outcome is only apparent in the testing phases of NN development. The following sections describe how features 1 & 2 were used to attempt to deduce "normal behaviour" for both the generic and turbine specific models.

5.1.2 Data Selection for Turbine Specific Model

Since turbine 16 appears to have experienced a gearbox failure, it can be used as a means for testing both the generic and turbine specific models. For an unlabelled data set, the main method which can be used to class data as normal behaviour is through a direct visual comparison with multiple turbines which operate within the same vicinity. The SCADA data set acquired began from the month of April 05 and ended at December 06. Since turbine 16 experienced the failure during the month of January 06, the months April – June 05 were inspected to view its operation during those months in an attempt to avoid utilising data from months close to the time of failure. This period of operation is shown in figure 5.5.

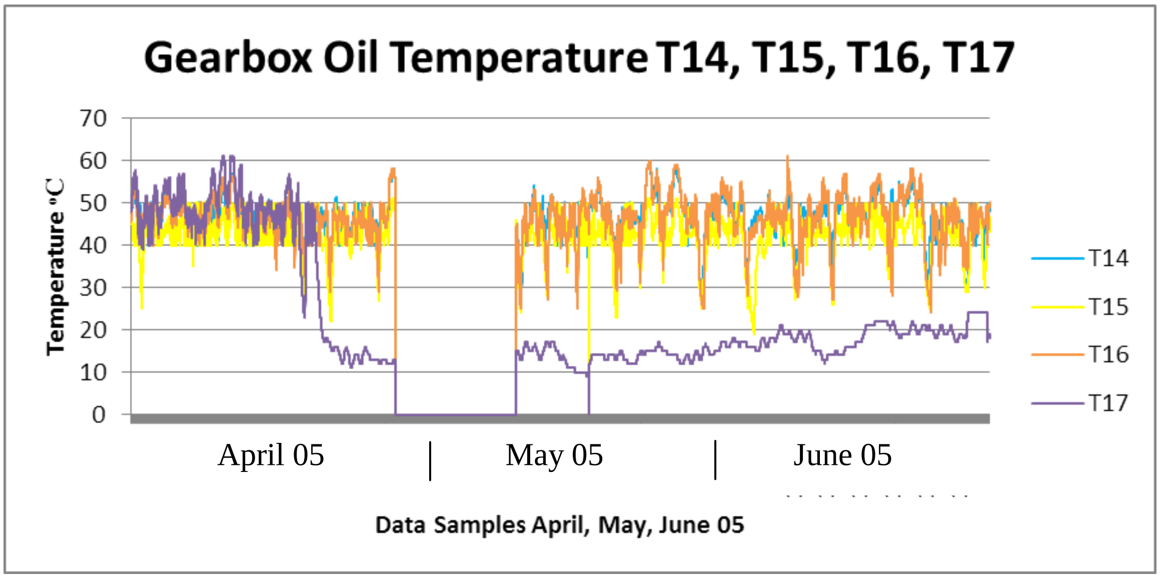

Figure 5.5: T16 with neighbouring turbines' operation during April – June 05

From figure 5.5, it can be seen that T17 seemed to experience a problem during the month of April 05. This is apparent since the trend for T17 exhibits the characteristics where the active power drops to zero but the SCADA Telemetry is not switched off (apart from sample 4201 to approx 6000 where all turbines are switched off) thus falling within the category of feature 2. For the latter part of the data set shown in figure 5.5 T14 and T16 seem to be operating at a higher temperature than T15 with peaking temperatures that approach 60˚C. This is higher than the 40-50°C range which the remainder of the turbines operate at, possibly suggesting that the failure may be in its incipient stage and yet to manifest completely. Because of this data from after the failure was investigated, as this ideally would provide a more accurate representation of a healthy turbine gearbox.

Another issue considered to be of importance is the element of seasonality and how it affects the data. The data had to be examined to identify if the weather season at different times of the year has an effect on the temperatures of the various components within the turbine. Figure 5.6 displays the temperature trends for T3 at different months of the year representing summer, spring, autumn and winter. From the graphs it can be seen that the temperature at different times of the year does not appear to affect the temperature of the components, with the temperature fluctuating between the same ranges (40 -50˚C) for an operational turbine throughout the year.

Therefore in order to avoid segmenting the data set further for the testing phase it was decided that the months directly after the gearbox replacement in turbine 16 (April 06 – December 06) could be considered suitable for training purposes. Three months were used (April, May and June 06) in order to leave a sufficient amount of data for testing purposes. In this way the remainder of the months could be used to create a coherent test data set allowing any interesting events occurring within the data to be tracked sequentially as they develop. The complete data set used with all of the input parameters sorted in ascending order is shown in figure 5.7**.** This depicts the range of "normal" operation values used to train the model.

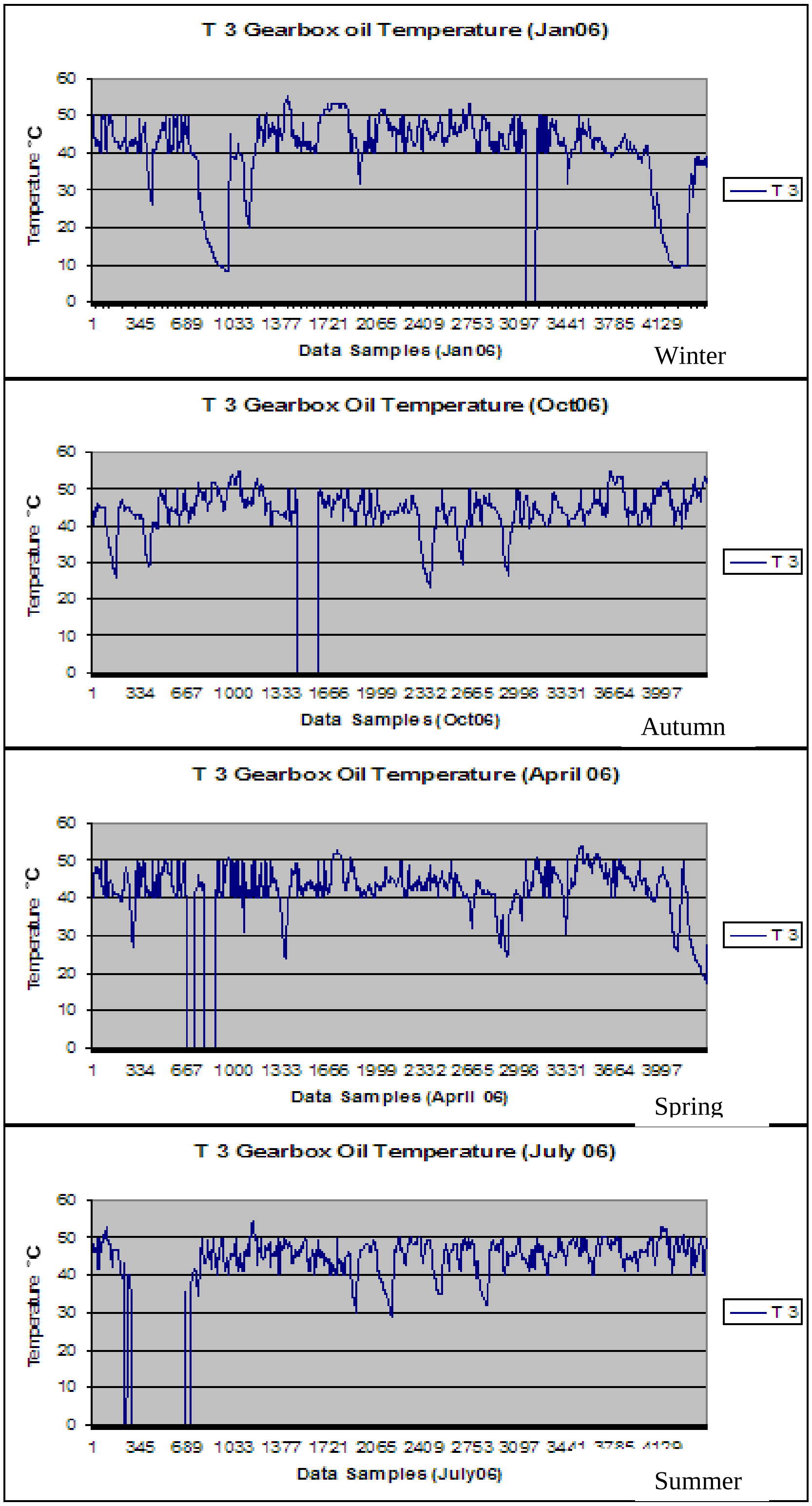

Figure 5.6: Seasonal data from T3

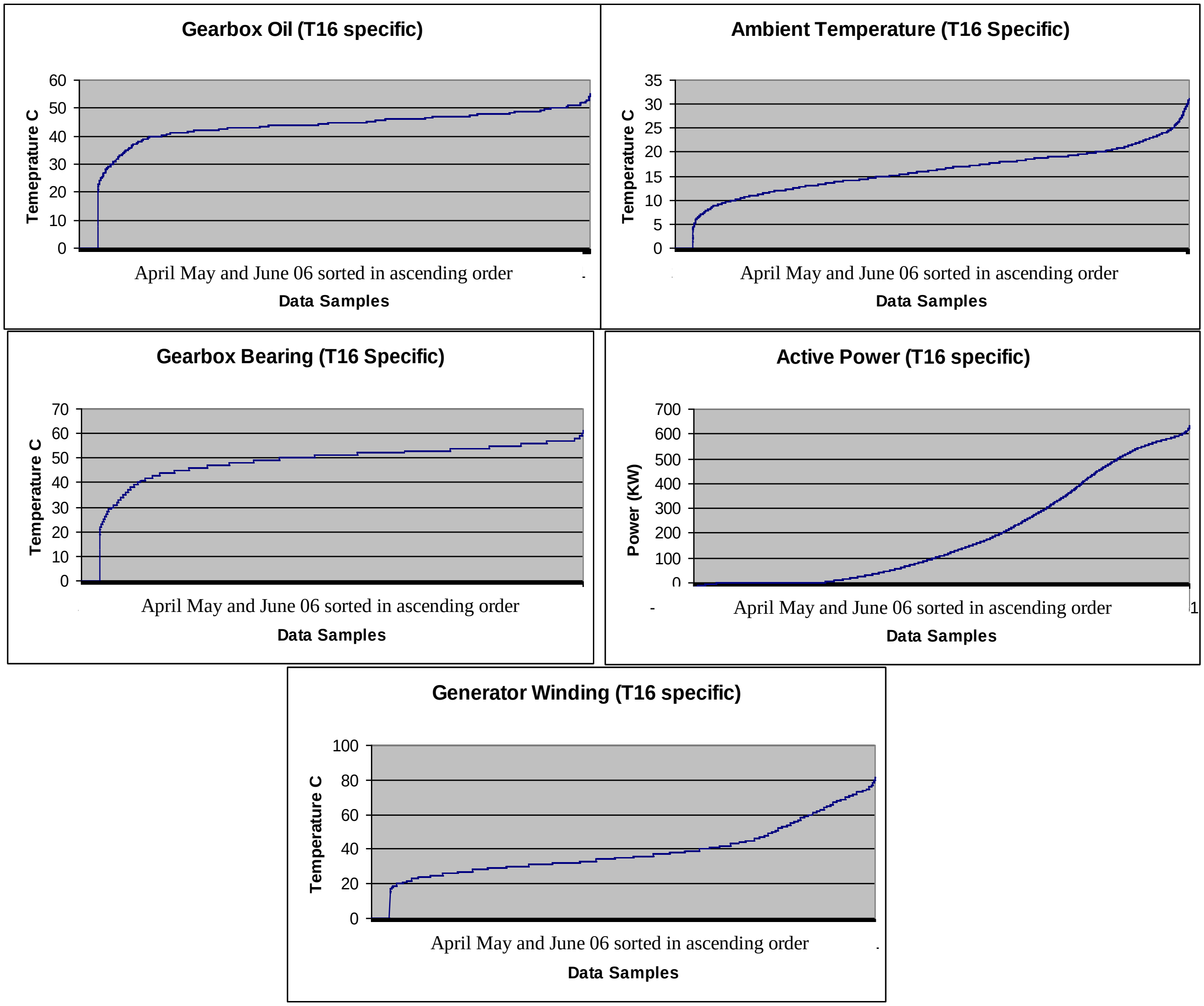

Figure 5.7: T16 Specific training data set

5.1.3 Data Selection for Generic Model

The benefit of considerably reducing data processing from training a generic model has already been established. Aside from this, training a generic model would also serve the purpose of identifying whether turbine location specific factors at a wind farm site have a perceptible effect on the operation of a turbine's internal components. If so, then this would be reflected as reduced accuracy in the model output.

The process used to select data for the generic model was the same approach adopted for the turbine specific model described earlier. It is important that the range of each of the parameters is as varied as possible while still ensuring that they are still within the bounds of normal operation. In order to achieve this, the trends for all 26 turbines were plotted across April, May and June 05. A large number of operating turbines were found to exhibit similar trend patterns within the same range of temperatures during this period of operation. The data trends for each of the parameters from these turbines were selected and their values averaged to produce one set of training data for each parameter. The resulting data set is shown in figure 5.8 whereas table 5.1 below summarises the information regarding both generic and specific data sets created for training.

Data set	Data period selected	Turbine (s) used	Comments	Shown in Figure
Specific	April – June 06	T16	Healthy Data from after the gearbox failure	Figure 5.6
Generic	April – June 05	Multiple T3-T7, T9,T10	Data from a collection of turbines all averaged	Figure 5.7

Table 5.1: Summary of training data sets created

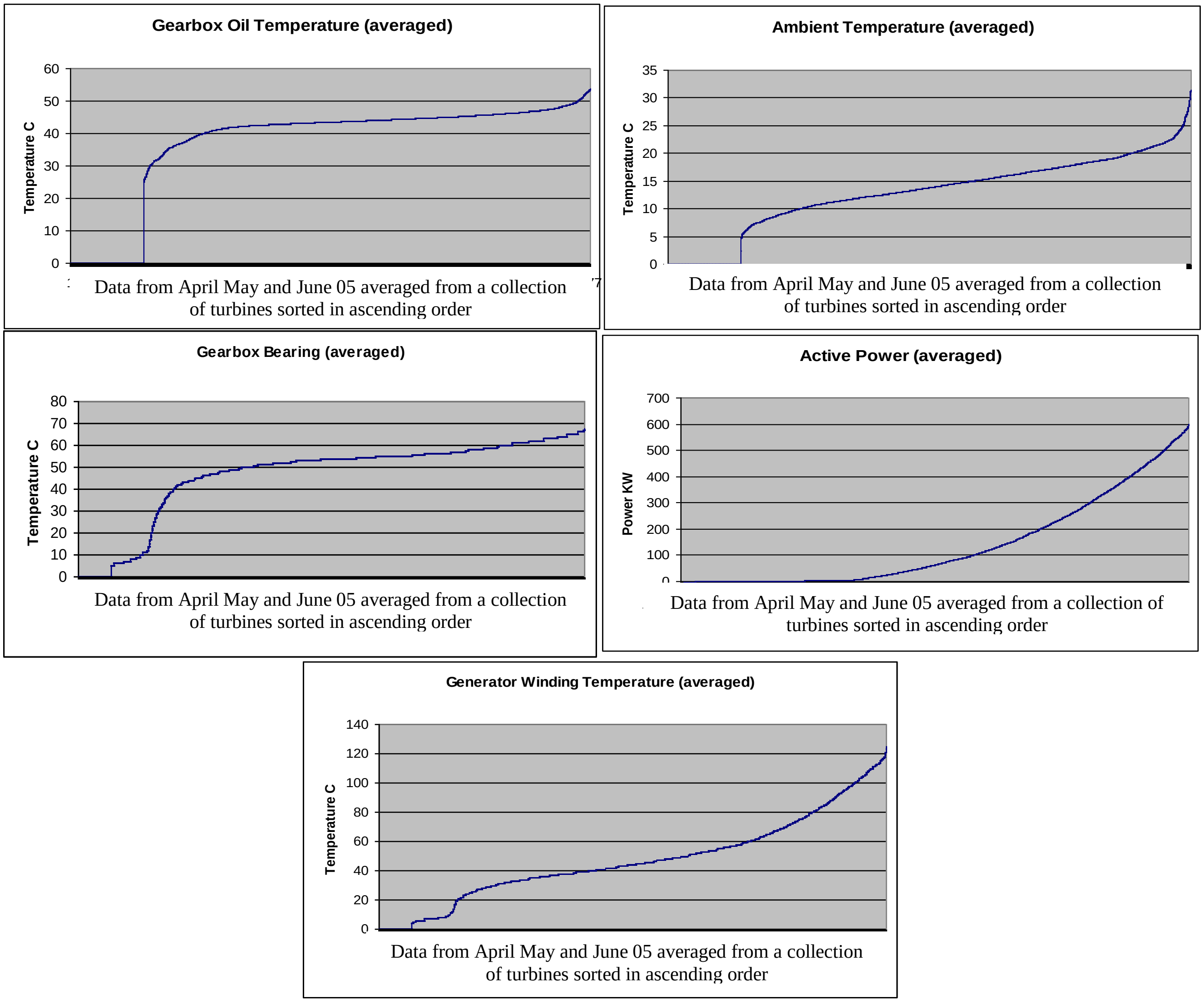

Figure 5.8: Training data set for generic models

What Figure 5.8 shows is the typical spread of values under normal healthy turbine operation sorted in ascending order. (Hong et al 2001), states that one of the best ways to avoid over-fitting when training NN models is to use “lots of representative data”. Therefore both the turbine specific and generic data sets were made to consist of a large number of data samples (approx 13000 data points). The following section describes the stages involved in the training process.

5.2 Neural Network Training Parameters

Model training is the point when the training data set built is presented as inputs to the NN. There are different software packages which are commercially available which enable the training of these models. The software utilised for this thesis is the MATLAB 7.01 neural network toolbox. This package allows for the modification of a number of parameters that allow the user to refine the training process for their particular data set in order to yield the best model output. This process of refinement is generally achieved over a trial and error process. The parameters which need to be defined through the MATLAB NN software package before training can commence are:

- Determining the network topology (i.e. the size and structure of the network, the number of neurons in each of the layers and the number of layers)
- Selection of the neuron transfer function
- Selection of the error performance function minimisation algorithm
- Setting the number of training epochs (i.e. the number of times the data set is presented to the NN) and the target Mean Square Error (MSE) (error performance function) to be achieved across the training data set.

5.2.1 Network Topology

In general the number of inputs nodes in the input layer of the neural network is defined by the number inputs for the application. Similarly the number of output nodes in the output layer is defined by the required number of outputs from the network. The number of hidden layers and the number of nodes within them are left at the discretion of the network developer. Typically an arbitrary number of hidden layers and nodes are initially selected and then refined until an acceptable level of generalisation (determined by the developer during testing) is achieved for their particular application. It is the hidden layer that is used to facilitate generalisation of the learning algorithm (Luger 2002).

A number of authors have demonstrated that neural networks which contain more than one hidden layer are more prone to falling into a local minimum (Hong et al 2001) (Gulski et al 1993). (Hong et al 2001) also states that by limiting the number of layers and neurons, the number of weights can be reduced therefore less data is required to train and test the network. It is therefore important to consider the size of the training data set when determining the size of the hidden layers of the neural network. Perhaps the most useful definition found in the literature is that of (Tarassenko 1998) who defines a general rule of thumb which can be used as an initial starting point for the number of hidden nodes to be used in the hidden layer. This rule is based on the number of inputs and outputs required for the application. This is defined below in equation 5.1:

$$J = \sqrt{IK} \qquad \text{eq5.1}$$

Where J is the number of hidden nodes to be used, I the number of network inputs and K the number of network outputs. Using this equation and the models defined in chapter 3 section 3.3.2, the number of hidden nodes for the input layer would be sqrt (4 inputs x 1 output) = 2 hidden nodes in the hidden layer. Based on this the architecture of 4-2-1 (4 inputs, 2 hidden nodes, 1 output) is formed. While this node count seems somewhat low considering the size of the training data sets (13000 input vectors) it was used as a starting point for training and testing purposes.

5.2.2 Neuron Transfer Function

As detailed in chapter 3 the neuron transfer function (or activation function) determines if a neuron fires to provide some output which either feeds into the input of another neuron or provides the end result if it is an output layer neuron. The MATLAB 7.01 NN toolbox offers the choice of three different types of transfer function. The log-sigmoid transfer function, the tan-sigmoid function and the linear function displayed below in figure 5.9:

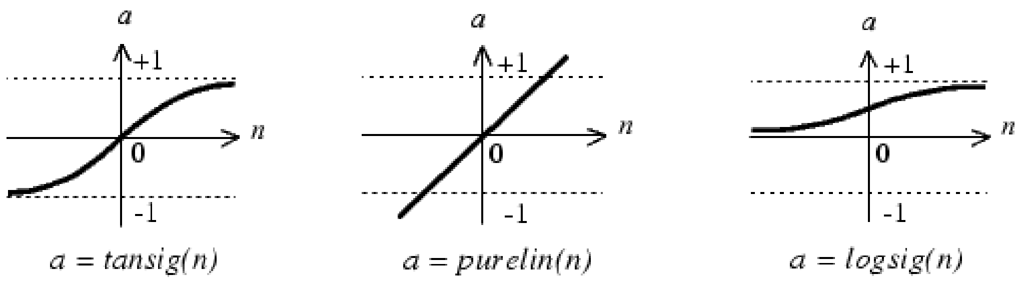

Figure 5.9: The different neuron transfer functions available in the MATLAB toolbox (MATLAB NN 1984-2007)

Each of the layers in the neural network can utilise any of the three transfer functions. What is important to note is the effect the neuron transfer function has on its output. For the SCADA models developed here it is necessary that the model output can support a large range of values as shown in figure 5.7 and 5.8. A common configuration is to utilise either the log-sigmoid or tan-sigmoid transfer functions in the input and hidden layers, and then utilise the linear transfer function in the output layer. Using this configuration the model output can support a large range of values (MATLAB NN 1984-2007). An alternative configuration stated in the MATLAB neural network help file (MATLAB NN 1984-2007) is to utilise non-linear transfer functions for all three layers, but scale the target output range of the model between a value of 0 and 1**.** The reason for this is if sigmoid layers are used for the output layer (MATLAB NN 1984-2007) the output of the trained model is limited to a small range (between 0 and 1). This method of scaling was also made use of by (Hong et al 2001) where the authors state that the benefits to be gained lead to "rapid training", a reduced possibility of the training algorithm "getting stuck" in local minimum (as will be explained at a later stage) and an improved pattern recognition ability. Based on this information, the second configuration where all three layers make use of sigmoid functions and the output is scaled was chosen to be implemented.

This scaling process can be applied to either the inputs or outputs (Hong et al 2001). As the inputs for the SCADA models used in this thesis were to be used in an online fault detection system which requires real-time processing, there is no way of knowing before-hand the complete range of input values that can arise during wind turbine operation. Therefore placing a bound on the input values presented to the model would introduce an element of inaccuracy. Because of this only the outputs were chosen to be scaled. This was helpful as with a ratio of inputs to outputs being 4:1; the amount of extra data processing would be limited during system use. Only one de-scaling operation for 1 output would be required as opposed to 4 scaling operations (for each input) would be required to provide a real temperature value at the model output. The equation used for the scaling process is defined by equation 5.2 below:

$$Xn = (X\text{-}Min)/(Max\text{-}Min) \qquad \text{eq 5.2}$$

Where *Xn* is the new scaled value,

X is the actual value to be scaled,

Min is the smallest value in the training data set,

Max is the largest value in the training data set.

A range of 0-70˚C (0 being min and 70 being max) was selected for the gearbox oil model; a range of 0-100˚C was used for the gearbox bearing model and finally a range of 0-160˚C to support the allowed range of generator winding temperatures. These ranges were definitively selected through trending and observation of each parameter’s data sets. They were chosen to be larger than the typical operational ranges for each parameter in order so the ranges would be large enough to accommodate the typical operational ranges of each parameter whilst also providing a degree of latitude for the models to avoid saturating the model outputs. This way the models can produce values which are considered to be within the ranges of faulty operation in the case that the input values (namely the power generated and ambient temperatures) are correspondingly high enough to warrant such undesirable output values. This ensures that the output of the models is not constrained to only the healthy normal operating range and can handle an increased range of input values.

5.2.3 Performance Function Minimisation Training Algorithms

The next parameter which needs to be set through the MATLAB NN toolbox is the training algorithm. The training process uses the training data set created which features network inputs and target outputs. During the training process the weights and biases present in each neuron of the network is iteratively adjusted to minimise the network performance function (the MSE). The MSE is defined as “the averaged squared error between the network outputs and the target outputs” in the training data set (MATLAB NN 1984-2007). The training process can be better explained if we picture an error surface which is derived during network training as shown in figure 5.10 (Strachan 2005).

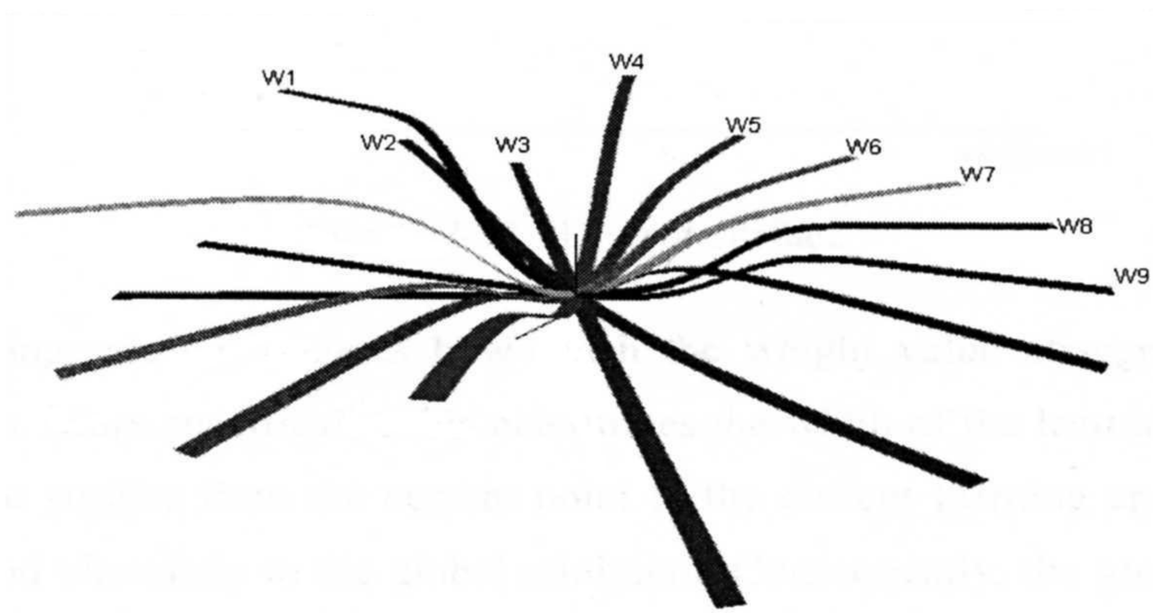

Figure 5.10: Network Weights Error Surface

This error surface is formed from the combination of weights and biases from all nodes in the network where each point on this surface characterises a unique weight configuration.

This surface consists of local minima and a global minimum. The global minimum represents the weight configuration which provides the lowest mean squared error between the network and target outputs across the complete training data set. This global minimum can be found using a gradient descent based search algorithm (Nielsen 2004) which in essence constitutes the model learning process**.**

Before this can occur however it is necessary that the weights are randomly initialised (MATLAB NN 1984-2007). This is an important initial step since equal valued weights would prevent the system from learning. This would occur due to the nodes in the hidden layer receiving matching error signals resulting in identical weight adjustments as the error is propagated back through the network nodes in proportion to the weights.

The MATLAB toolbox offers a number of training algorithms presented in table 5.2 which all make use of the gradient of the performance function to determine how to adjust the weights to minimise the MSE.

Acronym	Algorithm
LM	Levenberg-Marquardt
BFG	BFGS Quasi-Newton
RP	Resilient Backpropagation
SCG	Scaled Conjugate Gradient
CGB	Conjugate Gradient with Powell/Beale Restarts
CGF	Fletcher-Powell Conjugate Gradient
CGP	Polak-Ribiére Conjugate Gradient
OSS	One Step Secant
GDX	Variable Learning Rate Backpropagation

Table 5.2: Error Minimisation training algorithms provided in the MATLAB NN toolbox

It is difficult to know without trial which algorithm will provide the best performance in terms of speed of training and model output accuracy. However a speed and memory comparison of each of the available training algorithms is provided in the MATLAB NN help file (MATLAB NN 1984-2007). A comparison which tests all of the algorithms for two different types of problems, namely function approximation and pattern recognition with three different applications for each type of problem is presented. The SCADA models to be developed for this thesis fall under the category of function approximation, where the inputs to the model provide information regarding the turbine operation levels and the output required is an estimation of component temperatures based on these operation levels. Therefore the results from the function approximation tests were of interest, in particular the SIN and ENGINE applications shown in tables 5.3 and 5.4 respectively.

The SIN application is described as a simple function approximation problem which makes use of a 1-5-1 network architecture (1 input, 5 nodes in the hidden layer and 1 output) to approximate a single period of a sine wave. The results for this application are presented below:

Algorithm	Mean Time (s)	Ratio	Min. Time (s)	Max. Time (s)	Std. (s)
LM	1.14	1.00	0.65	1.83	0.38
BFG	5.22	4.58	3.17	14.38	2.08
RP	5.67	4.97	2.66	17.24	3.72
SCG	6.09	5.34	3.18	23.64	3.81
CGB	6.61	5.80	2.99	23.65	3.67
CGF	7.86	6.89	3.57	31.23	4.76
CGP	8.24	7.23	4.07	32.32	5.03
OSS	9.64	8.46	3.97	59.63	9.79
GDX	27.69	24.29	17.21	258.15	43.65

Table 5.3: Speed comparison of various training algorithms for SIN problem (all trained to an MSE of 0.002)

The fastest algorithm for this function approximation problem is the Levenberg-Marquardt algorithm (Roweis). From the training times it can be seen that it is over four times as fast as the next fastest algorithm. Incidentally the (MATLAB NN 1984-2007) states that the LM algorithm is best suited for function approximation problems where the network has fewer than 100 weights in total across all layers and where it is necessary that the approximation is very accurate.

The ENGINE function approximation problem also reinforces these findings. This application can be considered to be the most similar in nature to the models presented in this thesis. It utilises data obtained from an engine where the inputs to the model are engine speed and fuelling levels and the required outputs are the resulting torque and emission levels. The network architecture used for this problem is a 2-30-2 network. The results are shown in the table below (5.4 (MATLAB NN 1984-2007)).

Algorithm	Mean Time (s)	Ratio	Min. Time (s)	Max. Time (s)	Std. (s)
LM	18.45	1.00	12.01	30.03	4.27
BFG	27.12	1.47	16.42	47.36	5.95
SCG	36.02	1.95	19.39	52.45	7.78
CGF	37.93	2.06	18.89	50.34	6.12
CGB	39.93	2.16	23.33	55.42	7.50
CGP	44.30	2.40	24.99	71.55	9.89
OSS	48.71	2.64	23.51	80.90	12.33
RP	65.91	3.57	31.83	134.31	34.24
GDX	188.50	10.22	81.59	279.90	66.67

Table 5.4: Speed comparison of various training algorithms for ENGINE problem (all trained to an MSE of 0.005) (MATLAB NN 1984-2007)

The LM algorithm again performs the fastest with the Broyden–Fletcher–Goldfarb–Shanno (BFGS) Quasi Newton algorithm only performing slightly slower. This is due to the larger number of weights in the larger sized network, making its speed less apparent when compared to the other algorithms.

Because the initial NN model which will be used for training the SCADA models is a 4-3-1 architecture as described previously, the total number of weights will fall significantly under the 100 weight mark. Because of this the LM algorithm would be the best suited algorithm for training based on the comparison of results presented.

5.2.4 Training Epochs and Mean Square Error target

The final parameters which must be defined before training can commence are the number of training epochs and the target MSE. Ideally the target MSE should be set at a value which is considered acceptable for the application. For the SCADA models developed in this thesis, in order to yield models that are as accurate as possible, an excessively low target MSE of 0.00001 was used. The lowest MSE value which can be achieved is dictated by the global minimum in the weight space. Therefore by setting the target MSE to a value as small as 0.00001 ensures that the training algorithm will attempt to achieve the global minimum.

The number of Epochs defines the number of times the complete data set is presented to the NN. Again this value is left at the discretion of the developer but should not be set too high in order to avoid network over-fitting., According to (Hong et al 2001) the method of "early stopping" can be used to avoid this. Using this method, training is considered to be complete when the MSE error made on the training data set stops decreasing. This method was chosen to be adopted and an epoch of 1000 was selected.

5.3 Training and Testing Results for SCADA Fault Detection Models

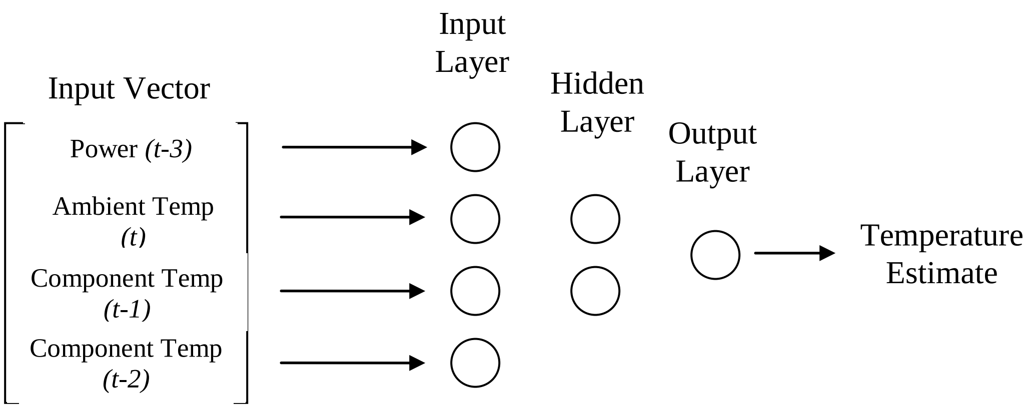

Figure 5.11: Initial NN architecture

Figure 5.11 shows the initial topology used for training the SCADA models. Each input vector from the training set consisting of:

- Power generated *(t-3)* i.e. three time steps before current time step *(t)*
- Ambient temperature *(t)*
- Two regressive temperature inputs *(t-1)* and *(t-2)* for the modelled parameter (gearbox oil, gearbox bearing and generator winding temperatures) *(t-1)*,*(t-2)* one and two time steps before current time step respectively.

is presented to the input layer of the neural network. For both the generic and specific model training data sets there are approximately 13000 input vectors. With an epoch of 1000 this results in a total of 13000 input vectors * 1000 epochs = 13000000 input trials to the network.

A number of network architectures were tried and tested during this stage. The number of neurons present in the hidden layer was varied as well as the number of hidden layers. The results of the initial architecture and two other architectures which were found to yield relatively accurate estimations for the generic and specific gearbox oil models are presented in table 5.5 and 5.6 respectively. The tests are based on three months worth of previously unseen data for turbine 16 i.e. data which the models have not been trained on.

Generic Gearbox oil Network Architecture	Training MSE	Mean / RMS error T16 Oct 06	Mean / RMS error T16 Nov 06	Mean / RMS error T16 Dec 06
4-2-1	5.684e-5	0.047 / 1.034	0.128 / 0.6913	0.1329 / 0.87
4-3-1	4.419e-5	0.024 / 1.061	-0.283 / 1.109	-0.055 / 0.844
4-5-3-1	1.359e-5	0.0106 / 0.92	1.182 / 9.336	-0.085 / 2.72

Table 5.5: Gearbox oil *Generic* model, training and testing results for 3 different network architectures tested on previously 3 months of unseen data

Specific Gearbox oil Network Architecture	Training MSE	Mean / RMS error T16 Oct 06	Mean / RMS error T16 Nov 06	Mean / RMS error T16 Dec 06
4-2-1	1.9e-4	-0.233 / 1.24	-0.27 / 0.722	-0.277 / 1.04
4-3-1	6.1793e-5	-0.09 / 2.55	-1.323 / 4.62	-0.34 / 1.38
4-5-3-1	5.27165e-5	-0.169 / 3.12	-6.17 / 18.7	-2.97 / 11.79

Table 5.6: Gearbox oil *Specific* model, training and testing results for 3 different network architectures tested on previously 3 months of unseen data

For each month worth of data (which amounts to approximately 4500 data points) the model estimations were compared with the actual temperature for T16. A direct subtraction between the model estimate and the actual temperature was used to obtain an error signal across all data points for each month's data set. The differences between each instance of data where then used to calculate the mean and RMS (Root Mean Square) errors.

The months October, November and December 06 were chosen for testing as they offered the best representation of healthy gearbox operation assuming the gearbox replacement procedure identified in section 5.1 did in fact take place. As can be seen from the numbers presented in tables 5.5 and 5.6, for both the generic and specific models, the 4-2-1 network model offered the best generalisation ability with consistently low error values for T16 across all three months despite the higher training MSE errors obtained during the training process. Another interesting point to note between the generic and specific model is the consistently lower error values produced by the generic model across all three model architectures trained. This suggests that the generic training data set used was a more accurate representation of normal operating behaviour therefore leading to more accurate model estimations.

For both the generic and specific models, the 4-layer network (4-5-3-1) produces the largest RMS errors for the month of November and December 06 in particular for the specific model. When viewing the model output trends (figure 5.12) the model shows signs of over-fitting which are evident in its mostly accurate estimations between the ranges of 40-50C but its incapability to generalise estimations for temperatures out-with this range therefore producing poor estimates. These are highlighted in figure 5.12.

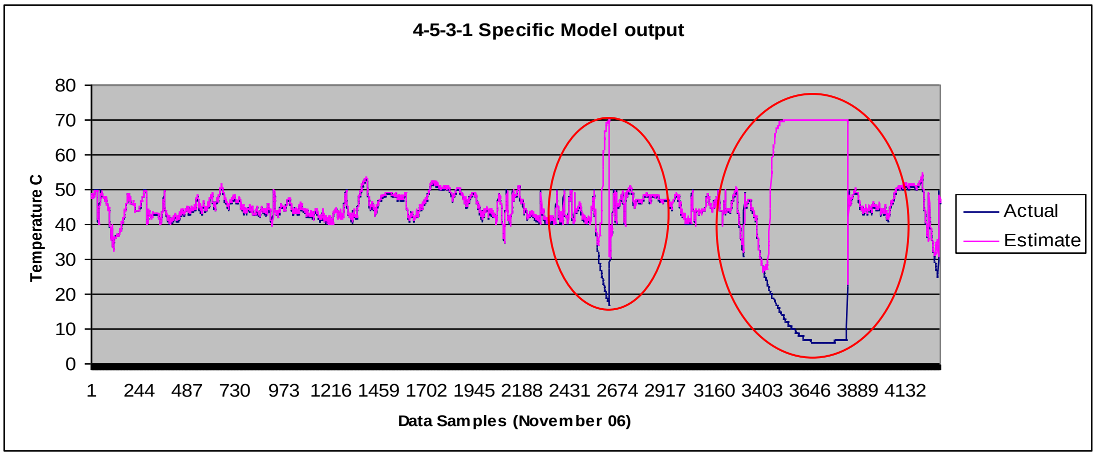

Figure 5.12: 4-5-3-1 Gearbox oil Specific Model showing signs of over-fitting when tested against the data for the month of November 06

The same training and testing process was adopted for both the gearbox bearing and the generator winding temperature models. Again both generic and turbine specific models were trained and the results are presented in tables 5.7-5.10.

Generic Gearbox bearing Network Architecture	Training MSE	Mean / RMS error T16 Oct 06	Mean / RMS error T16 Nov 06	Mean / RMS error T16 Dec 06
4-2-1	0.000158	-0.175/ 1.378	-0.058 / 0.768	-0.0675 / 0.929
4-3-1	7.58e-5	-0.076 / 1.15	-0.104 / 0.784	-0.08 / 0.952
4-5-3-1	6.13e-5	0.09 / 1.08	-0.098 / 0.751	-0.15 / 0.951

Table 5.7: Gearbox bearing *Generic* model, training and testing results for 3 different network architectures tested on previously 3 months of unseen data

Specific Gearbox bearing Network Architecture	Training MSE	Mean / RMS error T16 Oct 06	Mean / RMS error T16 Nov 06	Mean / RMS error T16 Dec 06
4-2-1	0.00027	-0.075/ 1.235	-0.074 / 0.8072	0.01707 / 0.92
4-3-1	0.00029	-0.005 / 1.29	-0.253 / 1.185	-0.05 / 0.995
4-5-3-1	0.000264	-0.015 / 1.26	-0.084 / 0.859	-0.01 / 0.914

Table 5.8: Gearbox bearing *Specific* model, training and testing results for 3 different network architectures tested on previously 3 months of unseen data

Generic Generator Winding Network Architecture	Training MSE	Mean / RMS error T16 Oct 06	Mean / RMS error T16 Nov 06	Mean / RMS error T16 Dec 06
4-2-1	0.000188	-0.54 / 1.2507	-0.644 / 1.1688	-0.604 / 1.31
4-3-1	0.000189	-0.578 / 1.2784	-0.691 / 1.2179	-0.635 / 1.348
4-5-3-1	0.000179	0.141 / 1.475	-0.073 / 1.303	-0142/ 1.384

Table 5.9: Generator Winding *Generic* model, training and testing results for 3 different network architectures tested on previously 3 months of unseen data

Specific Generator Winding Network Architecture	Training MSE	Mean / RMS error T16 Oct 06	Mean / RMS error T16 Nov 06	Mean / RMS error T16 Dec 06
4-2-1	8.66e-5	-2.232/ 2.913	-3.377 / 3.9184	-3.22 / 3.862
4-3-1	8.48e-5	-2.315 / 3.0333	-3.587 / 4.1227	--3.41 / 4.162
4-5-3-1	5.3343e-5	0.0102 / 2.042	1.052 / 2.217	1.0012 / 2.5007

Table 5.10: Generator Winding *Specific* model, training and testing results for 3 different network architectures tested on previously 3 months of unseen data

The results obtained from the three trained models for both the generic and specific data sets show that the generic models consistently offer better results over the specific models. The 4-2-1 *generic* model for the most part offered the best generalisation capability with the lowest RMS errors over the 4-3-1 and 4-5-3-1 generic models. For the generic gearbox bearing model the 4-5-3-1 architecture did perform slightly better than the 4-2-1 architecture model, however to avoid the possibility of over-fitting, the 4-2-1 generic model was still considered the most appropriate.

The error values produced by the trained models were deemed at an acceptable level for the diagnostics required by the project industrial partner for successful fault detection. The 4-2-1 generic model was therefore chosen as the model architecture to use for all of the trained fault detection models. The low errors produced by each model means that the output can be used directly as a comparison with the actual temperature trend to assess whether an anomaly is present. A straightforward difference of the two signals (actual and estimate) can be used. If the difference between the estimated value produced by the model and the actual value increases for a continuous number of instances i.e. a prolonged period of time and not a minor fluctuation, then this would flag as an anomaly. This is to exclude false identification of anomalies as a result of erroneous but transient data.

5.4 Validating the Model Inputs

In order to ensure that the use of each input parameter to the models is valid and is not used unnecessarily, a process of cross correlation is undertaken. The significance of each input is determined by assessing the model accuracy as each input is omitted one by one. The model accuracy can be quantified by the ability of the model to provide an estimation that is as close to the real value as possible. In this way the significance of each input can be understood by assessing the impact on the models' accuracy when omitting each parameter. The same set of data was used for training and testing the models in order to ensure fair results. The results of this cross correlation process are listed in table 5.10. The numbers show that each model performs best with all 4 inputs being used where the lowest mean and RMS errors are produced across all three component models. Both the gearbox oil and gearbox bearing models follow a similar trend with the RMS errors increasing in size with the ambient nacelle temperature input omitted first followed by the regressive inputs and finally, the power omitted last. This shows that the importance of the input variables for both of the models starts with power being the most important, followed by the regressive inputs and lastly the ambient nacelle temperature making the least impact on model accuracy. The generator winding model on the other hand does not follow the exact same trend on the period of data tested. The results show that the regressive inputs are the most important followed by ambient temperature and the power input comes in giving the least impact on model performance. The outcome from this process allows us to conclude that each of the input parameters used is of significant importance, each improving the model accuracy by adding information that aids the models to produce realistic estimates therefore justifying their use as an input.

Model	Variations	Mean error	RMS error
Gearbox Oil Model	All inputs present	0.05	2.21
	Ambient temp omitted	-0.37	2.59
	Regressive temp (t-1) (t-2) omitted	-1.35	8.06
	Power omitted	-1.03	8.47
Gearbox Bearing Model	All inputs present	-0.04	1.23
	Ambient temp omitted	-0.019	1.44
	Regressive temp (t-1) (t-2) omitted	-3.04	6.64
	Power omitted	-3.5	18.58
Generator Winding Model	All inputs present	-0.15	2.03
	Ambient temp omitted	0.094	5.77
	Regressive temp (t-1) (t-2) omitted	-0.41	8.67
	Power omitted	-0.0085	2.19

Table 5.11: Mean and RMS errors for each model input configuration for input validation purposes.

5.5 Gearbox Bearing and Oil Model Results with corroboration

Once the normal behaviour models were trained, the complete 2 years worth of data acquired was processed in order to see how well the model estimates agreed with the remainder of the data set to which they hadn't previously been exposed. Figure 5.13 shows how T16, during its normal period of healthy operation conformed to the model estimates during the month of October 06.

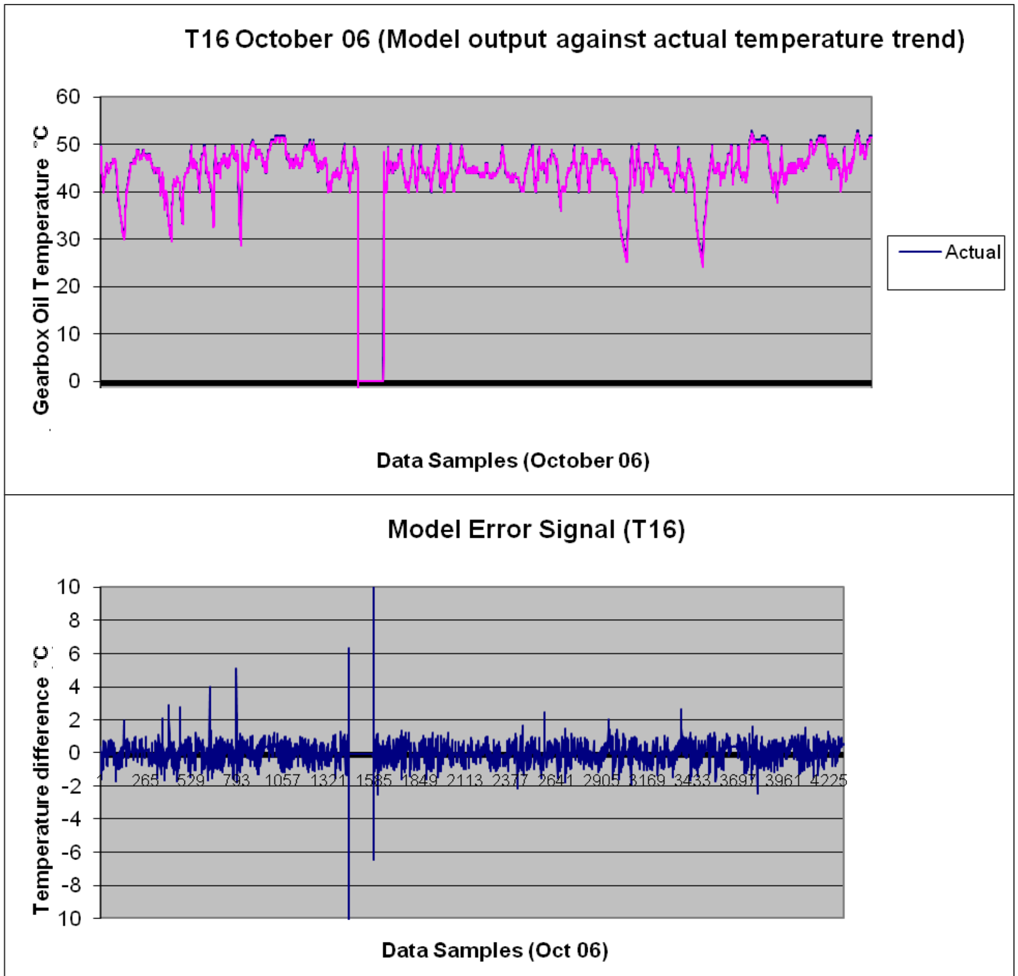

Figure 5.13: T16 conforming to model estimates signifying healthy gearbox operation.

As can be seen from the error signal obtained the majority of the differences lie between the ranges of -2 to 2°C. There is no consistent positive or negative divergence between the two signals for a sustained period of time.

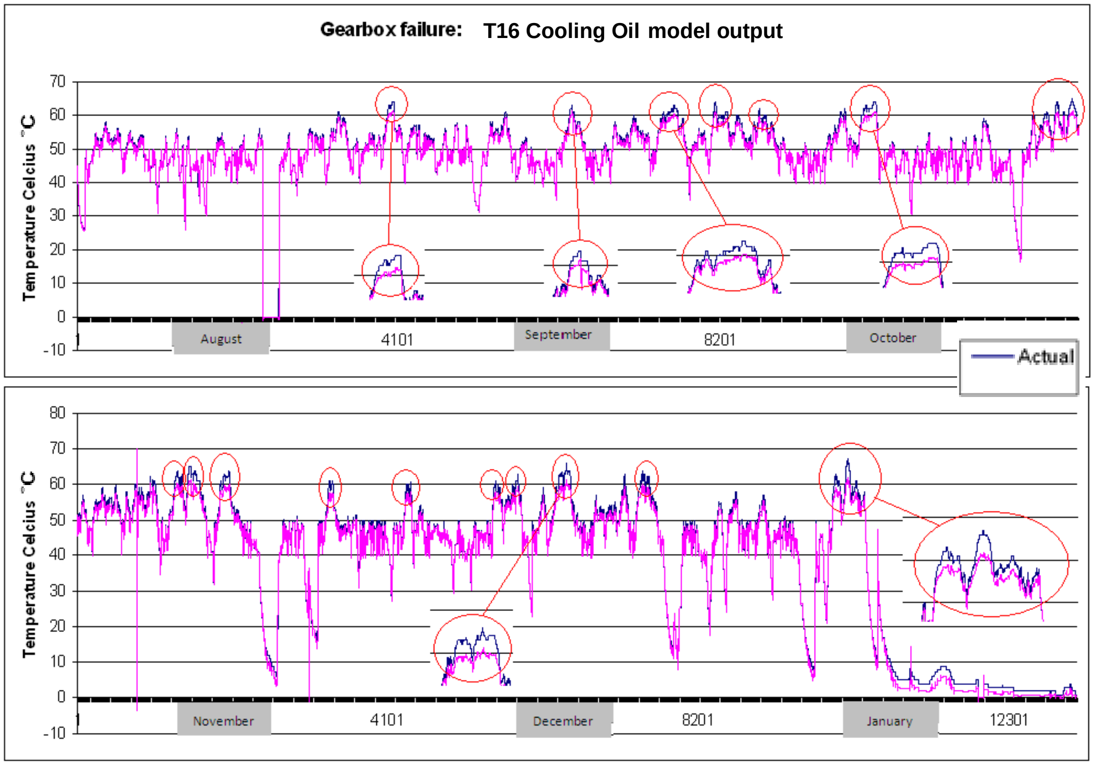

Figure 5.14a: Gearbox failure detected T16: gearbox cooling oil model output, anomalies detected from aug05-jan06.

Figure 5.14a shows the evolution of the gearbox cooling oil temperature trend for T16 from the period of *August 2005* to *January 2006* where it eventually fails. Figure 5.14b shows the corresponding error signal which depicts the difference between both the estimated output trend from the model as well as the actual temperature. The first significant deviation from the model estimates occurred towards the end of *August 2005*. From that point onwards the frequency of deviations and their duration increased. In this specific example it can be seen that the cooling oil model built here detected incipient problems in the form of overheating almost 6 months in advance of the actual failure. For comparison, the model built by Garcia et al was capable of detecting an incipient problem only 2 days before the actual failure.

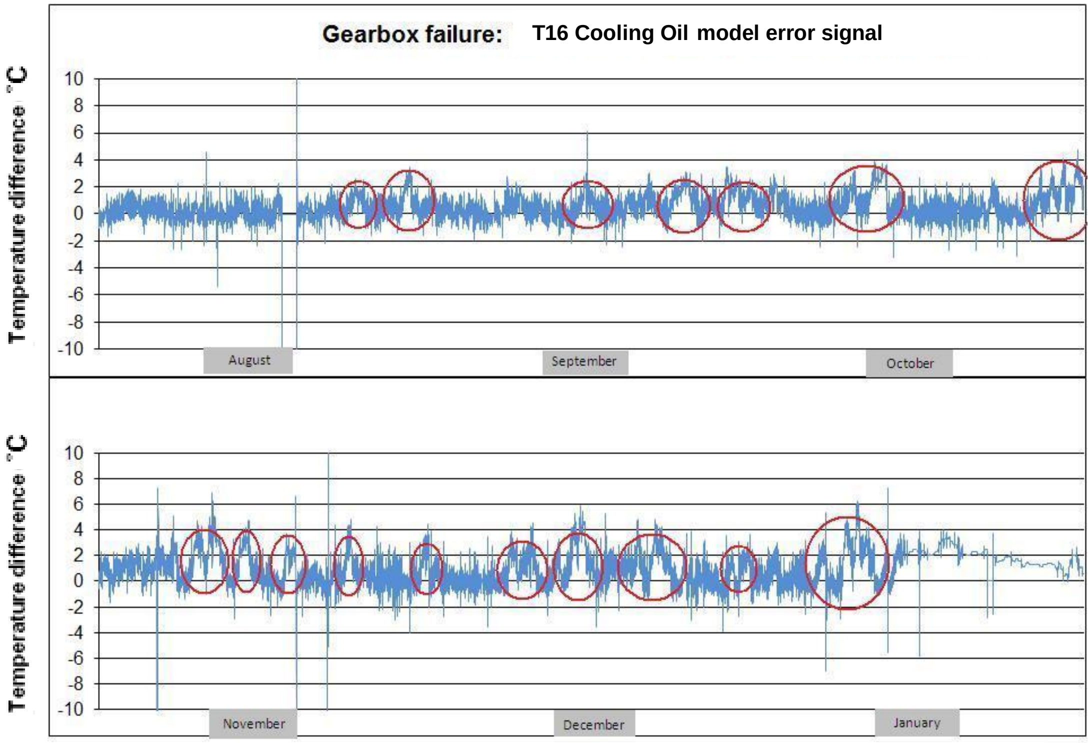

Figure 5.14b: Gearbox failure detected: gearbox cooling oil model error signal, anomalies detected from aug05-jan06.

Figure 5.15a and 5.15b show the gearbox bearing model temperature output for the same time period and the error signal respectively. Throughout this period the deviations from the model estimations were minimal and not significant enough to be classed as gearbox bearing anomalies when compared to other anomalous temperatures detected by the model. An interesting point to note from the results was that the minimal deviations occurred at the same time the cooling oil deviations were spotted which depicts the expected heat transfer between the gearbox components. According to these results, overheating was detected in the gearbox cooling oil while the gearbox bearing temperature conformed to the normal behaviour model. Corroborating the output from both models suggests that the failure was in fact a problem internal to the gearbox and not the gearbox bearing or cooling mechanism. This failure has been confirmed with the industrial contacts who supplied the data.

Figure 5.15a: Gearbox failure detected: Corresponding gearbox bearing model output detected during aug05-jan06.

Figure 5.16a and 5.16b show another example of gearbox problems detected by abnormally high temperatures in T17's gearbox oil parameter. In this example the T17 experienced problems towards the end of April 05 where it was taken offline as depicted by the trends. Data from before April 05 was not available however to identify when the first model deviations occurred. Nevertheless deviations were detected at the start of the month. These results highlight the importance of model accuracy, which in turn is achieved through good model selection and the effective training used to capture the normal behaviour of each parameter.

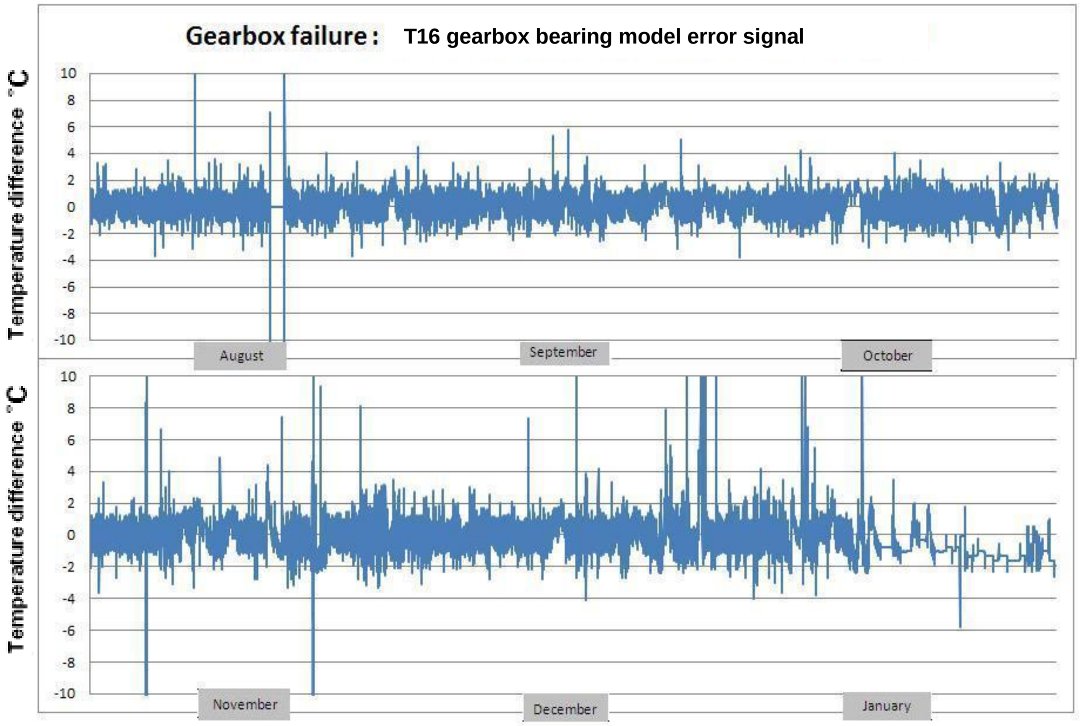

Figure 5.15b: Gearbox failure detected T16: gearbox bearing model error signal detected during aug05-jan06 showing no significant or prolonged periods of deviation.

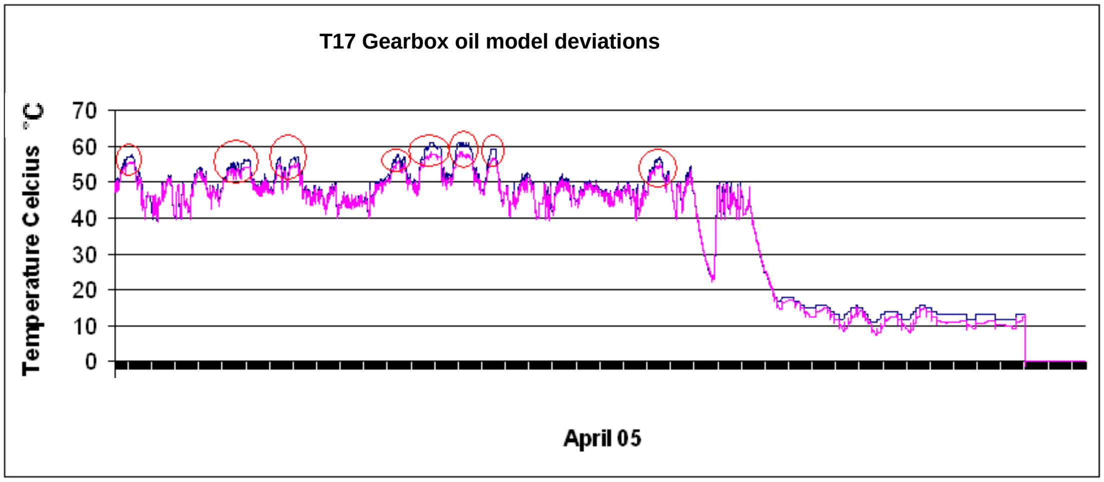

Figure 5.16a: Gearbox problems: detected by gearbox cooling oil model in T17 April 05.

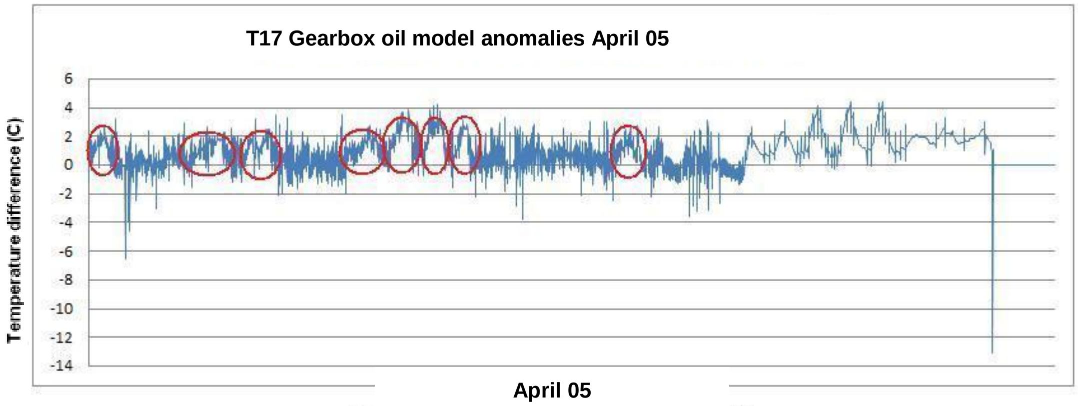

Figure 5.16b: Gearbox problems: cooling oil model error signal anomalies detected in T17 April 05.

5.6 Generator Winding Model Results

The method used for temperature anomaly detection for the gearbox was also adopted for the generator winding parameter available in the SCADA data. The approach used to train and test the model was identical to that of the gearbox. A malfunctioning generator was detected using the trained generator model within the 2 year data set. The last month of its operation before failure or it being brought offline is shown in figure 5.17a and b. The first noticeable deviations were detected in August 05 and continued to increase in size and frequency until its failure in November 06. A generator typically has a safety mechanism which cuts off power generation for a period of time whenever the windings overheat (Yusuf Patel 2007). This safety feature attempts to prolong the lifetime of the generator by preventing continued operation of the generator at a dangerously high temperature. The repetitive spikes seen in figure 5.17a are the effects on the temperature seen when the safety mechanism activates, interrupting normal power generation.

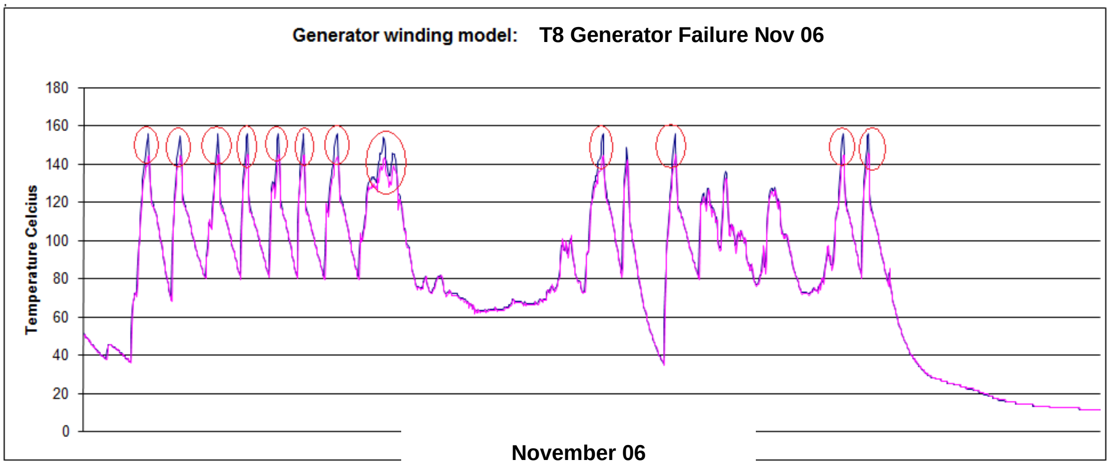

Figure 5.17a: Generator Failure or brought offline (November 06) detected in Generator Winding temperature parameter.

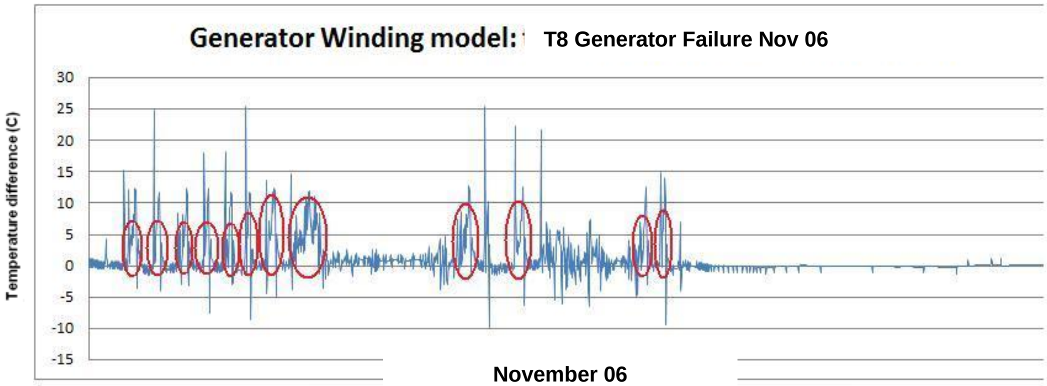

Figure 5.17b: Generator Failure: corresponding error signal

5.7 Wind Turbine Power Output Modelling

The significance of modelling the power output of a turbine for the purpose of monitoring its condition has been discussed by a number of papers published in the literature, namely (Caselitz et al 2005) and (Li et al 2001), both of which have been reviewed in chapter 2 section 4.3. As described previously the power characteristic of a turbine essentially represents the typical power output values plotted against the entire range of wind speeds which its operation covers. By keeping track of both wind speed and power output parameters, (Caselitz et al 2005) states that the overall health of the turbine can be supervised. Every manufactured turbine model design has a set power output performance level which it is expected to sustain in its everyday normal operation. By comparison of the actual power output and the expected output, an indicator of the turbines' generation performance is possible. Any degradation in this value will automatically reflect degradation in the turbines' condition.

Ensuring that a wind turbine's performance is optimal for a given set of operating conditions is an important factor to be taken into consideration if the operation and maintenance (O&M) costs are to be minimised. Assessing wind turbine performance is a relatively complex procedure since each turbine in a wind farm can be affected by many site specific factors (mentioned previously in section 5.1) that can greatly impact its operation and produce significant variations in its power performance, (Leaney et al 1997). Each of these aforementioned aspects makes it impractical to record ideal wind data. Most SCADA systems record only basic wind information such as mean wind speed over 10 minute intervals, the standard deviation over that 10 minute interval and in some more recent systems the wind direction (Yusuf Patel 2007). The estimation of power generation for diagnostic purposes is currently accomplished by comparing generated power to the manufacturers ratings for a given wind speed. A turbine will follow the power generation of this curve if the following conditions are met (Li et al 2001).

1) The wind speed is measured at the height of the turbine's nacelle;

2) The wind speed is uniform horizontally across the face of the turbine;

3) The vertical wind speed profile is the same as that experienced during the power performance testing of the turbine; and

4) The air density is the same as that during the calibration.

These conditions are rarely met at a wind farm for each turbine especially with the effect of site specific factors influencing the wind. Therefore deviations from the manufacturer's power curve cannot on their own be used for diagnostic purposes.

The data available in the research presented in this thesis is the averaged 10 minute intervals and standard deviation of both wind speed and active power available in the SCADA. The rationale behind the development of a mechanism for power performance estimation is primarily for the purpose of attaining an insight into the overall health and condition of a turbine, while possibly detecting performance degradation through pollution building up on the rotor blades (Caselitz et al 2005). Therefore its main purpose for the Fault Detection System (FDS) being developed in this thesis is to provide the user with an indication of how well the turbine is performing at a glance and also for corroboration purposes with the temperature fault detection models. Power performance estimation used in conjunction with the temperature anomaly detection models described earlier, will also serve the purpose of determining the time it takes from when an anomaly is detected in one of the turbines' internal components (i.e. the generator or gearbox) to the time it takes for this to be reflected in the turbines' overall performance. This will give an indication as to how long it takes faults to manifest themselves in the turbine to the point where they actually begin to impair the turbines generation ability.

The information and applicability (to the acquired SCADA Data) of the method used for the detection of blade surface roughness from the approach used by (Caselitz et al 2005) and the NN power estimation research found in the literature (Li et al 2001**)** stimulated the decision to build and test both of these mechanisms for power performance estimation. In this way the potential benefit that can be gained from corroborating a particular turbines' power performance with the temperature anomaly detection models built for the gearbox and generator can be more readily examined. Also, by building and testing both of these applicable methods to the data it was thought that this could lead to a more informative set of results that would allow the author to examine which methodology of the two yields the better output for the data sets available. The following section explains both of their internal workings in greater detail.

5.7.1 Model Input Data and Training Methodology

Method 1 (NN): Neural Network Power Estimation

The NN training and testing procedure closely followed the same methodology used for both the gearbox and generator anomaly detection models described in the earlier sections of this chapter. The reason behind this is due to the fact that this problem also falls under function approximation, where the NN attempts to approximate the relation between the current wind speed and the corresponding power generated by the turbine i.e. its power curve. A number of NN models were trained on some of the data available. The mean wind speed data available for each turbine is recorded from the nacelle mounted anemometer and therefore these readings give a better indication (ignoring anemometer sensor errors) of the turbine's experienced wind profile than a meteorological tower mounted some distance away from the turbine. An ideal case would be to have access to both sets of data since nacelle mounted anemometers also face their share of disturbances that affect the validity of the recorded measurements (Smith et al 2002). In this way the nacelle mounted anemometer readings could be verified against the tower readings in order to help eliminate anemometer inaccuracies, however this data was not available to provide this verification.

The averaged readings from the anemometer are taken over a period of 10 minutes meaning that the data loses detail in how the wind varied and fluctuated throughout that period. This means that there is a strong possibility that the power generated could easily be misrepresented since the power available in the wind is proportional to the cube of the wind speed (Danish Wind Energy Association 2008). Because of this, only problems that build up over a long period of time such as dirt on the blade and increasing blade surface roughness due to erosion mentioned by (Caselitz *et al* 2005) are the only form of fault that can be detected with this relatively low resolution data. In order to improve the quality of the data, the turbulence intensity was calculated to gain an insight into how varied the wind was during each 10 minute period. The turbulence intensity I_u gives an indication of the fluctuation of the wind speed over the averaged period (Infield Wind Resource notes) and is calculated by dividing the standard deviation by the mean recorded wind speed *U*.

$$I_u = \frac{\sigma_u}{\overline{U}} \qquad \text{eq5.3}$$

A month's worth of calculated turbulence intensity data was used along with the mean wind speed as inputs to train a NN on the associated power output for each turbine in an attempt to aid the NN approximate the relation between power and wind. In order to help further improve the overall accuracy of the model, the corresponding power output data was normalised between a range of -100 to 700, a large enough scale to support the range of output from a Bonus 600kW turbine. As each turbine experiences its own specific wind profile due to the many external factors mentioned previously, only turbine specific models were created i.e. using only turbine specific data to train the models as opposed to generic data averaged from a collection of turbines. Models were created for T8, T15 and T16 from the wind farm, where T8 and T16 experienced failures in the generator and gearbox respectively, and T15 was used for benchmark purposes since it experienced no failures. T15 is also situated next to T16 in the wind farm and so will have experienced a similar wind profile to that of T16.

The issue of data selection for training the power models was less defined than for the case with the temperature models. The reason for this is that there is no information regarding the condition of each turbine's rotor blades. Therefore the purpose behind training the power models was to capture the current condition of the turbine's rotor blades and determine if its performance degrades from that point in time and onwards, rather than attempting to capture the best example of 'normal' healthy operation. The only condition that would have to be satisfied is to ensure that data from a period where the turbine was experiencing faults in its other components namely the gearbox and generator was not used to train the models. This would allow for the investigation of whether the power output of a turbine is affected by the manifestation of faults in its gearbox and generator.

The model architecture selected for the power models was calculated using the same formula (eq5.1) used for the temperature fault detection models. The results produced by the models using this formula provided the better accuracy as shown previously and therefore it was also adopted as the method for selecting the appropriate model architecture for the power model. With 2 inputs and 1 output required from the NN, a hidden layer consisting of 2 nodes was used resulting in a 2-2-1 architecture. The model training results are shown below in table 5.12:

Power Network Architecture	Training MSE T8 Apr 06	Training MSE T15 Apr 06	Training MSE T16 Apr 06
2-2-1	0.00057	0.00106	0.000609

Table 5.12: Power model training results for T 8, 15 and 16

Figure 5.18a shows the output from a NN model trained for T15 against actual data. Figure 5.18b shows the corresponding error signal between the two signals where the actual turbine output (blue) is subtracted from the model estimate (red). When compared to the accuracy of the temperature fault detection models described earlier, it is clear that the output of the model is not sufficiently accurate to detect whether any one particular power measurement can be classed as an anomalous reading or not. From the error signal, the predominantly negative values shown in figure 5.18b also demonstrate that the model underestimates the turbine output more so than it over estimates. Therefore for performance assessment purposes, cases where the model underestimates the actual recorded power output of the turbine can be disregarded since the turbine has actually met and exceeded the estimate of the model. Calculations showed that typically over 90% of the estimations made by the trained model in any given month fall within a 30kW envelope of the actual recorded power output values for the healthy normal operation of T15. These calculations are presented in section 5.7.2.

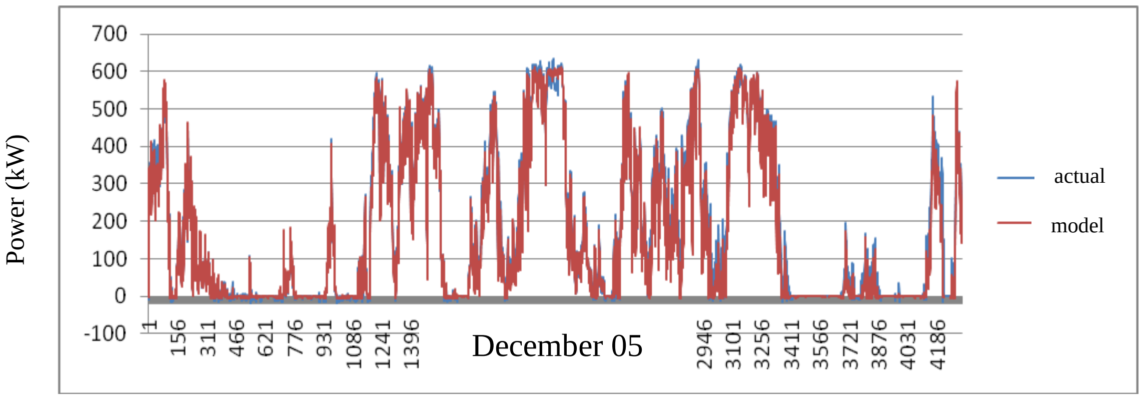

Figure 5.18a: Power model output for a healthy turbine (T15) during December 05.

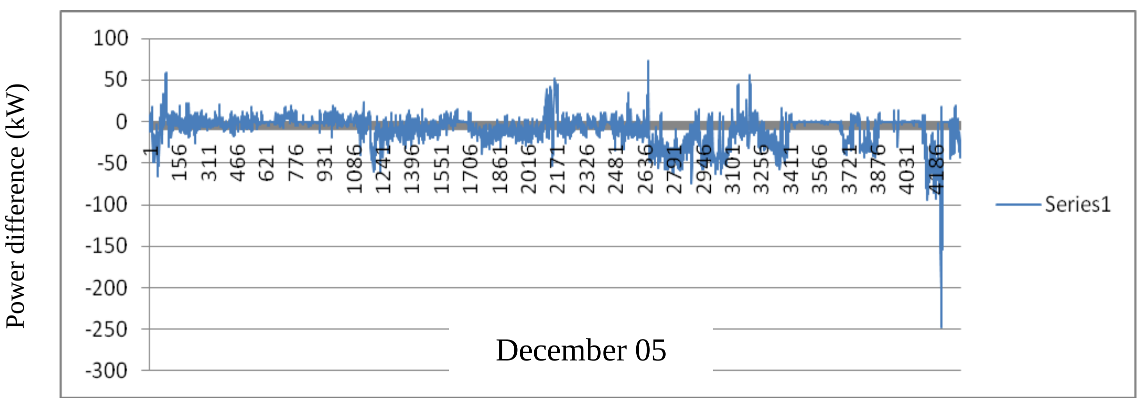

Figure 5.18b: The corresponding error signal between actual data and the model estimation during December 05.

Method 2 (Statistical): Learning the Averaged Power Curve

This method of analysis is based mostly on the research of (Caselitz et al 2005). A slight difference is that Caselitz used 5 minute averaged data instances of both wind and power values to learn the power characteristic of the turbine as opposed to the 10min averaged values used for the research in this thesis.

Learning the averaged power curves for each turbine was carried out in the following manner. The wind speeds are initially split into bins of 0.5m/s. The associated power values are then sorted into these groups. The average power value for each group is then calculated to give the learned power curve. Figures 5.19a, b and c show the learned power curve as well as the inner and outer alarm limits. These diagrams are screen captures from the output of the power performance agent software developed. The software learns the curves from data in an online manner and calculates the curves automatically once enough data has been gathered. The specific details of how the agent carries out this process is described in chapter 6. The inner alarm limits are calculated through the standard deviation of each of the groups and then added to either side of the averaged power curve. The outer alarm limits are chosen by the developer through the study of a number of turbines operating under normal conditions. A number of outer and inner alarm limits were tested in order to observe a strategy that attempts to minimise the number of false alarms generated. The resulting trained curves for T8, T15 and T16 are shown in figures 5.19 a, b and c respectively.

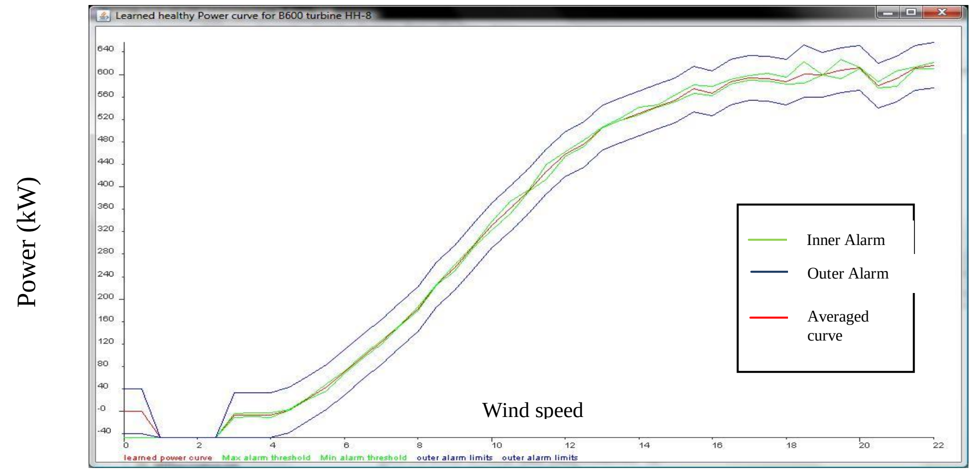

Figure 5.19a: Learned curve for T8 from healthy period of operation before generator failure

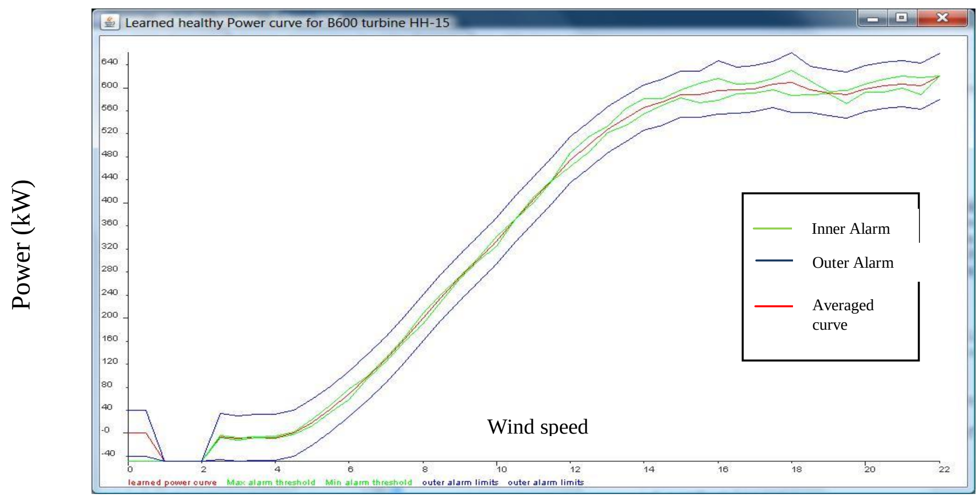

Figure 5.19b: Learned curve for T15 from healthy period of operation.

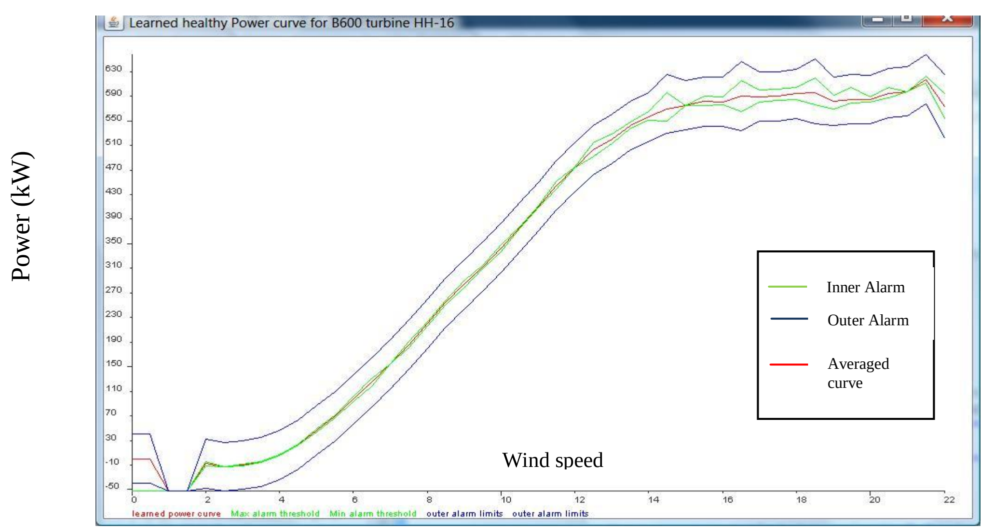

Figure 5.19c: Learned curve for T16 from healthy period of operation before gearbox failure.

Once the power curves for the turbines are learnt the power performance agent automatically switches to monitor mode. In monitor mode subsequent wind and power pairs can then be classified against the alarm limits calculated and consequently each given a performance classification. Figure 5.20 shows an example of a mock curve and how the classifications are assigned to pairs of points depending on where they fall within the calculated alarm limits by the *Power performance* agent described in chapter 6. While the classifications shown below are self-explanatory, it is important to note that the downtime classification refers to periods of unscheduled maintenance (feature 2 of the data see section 5.1).

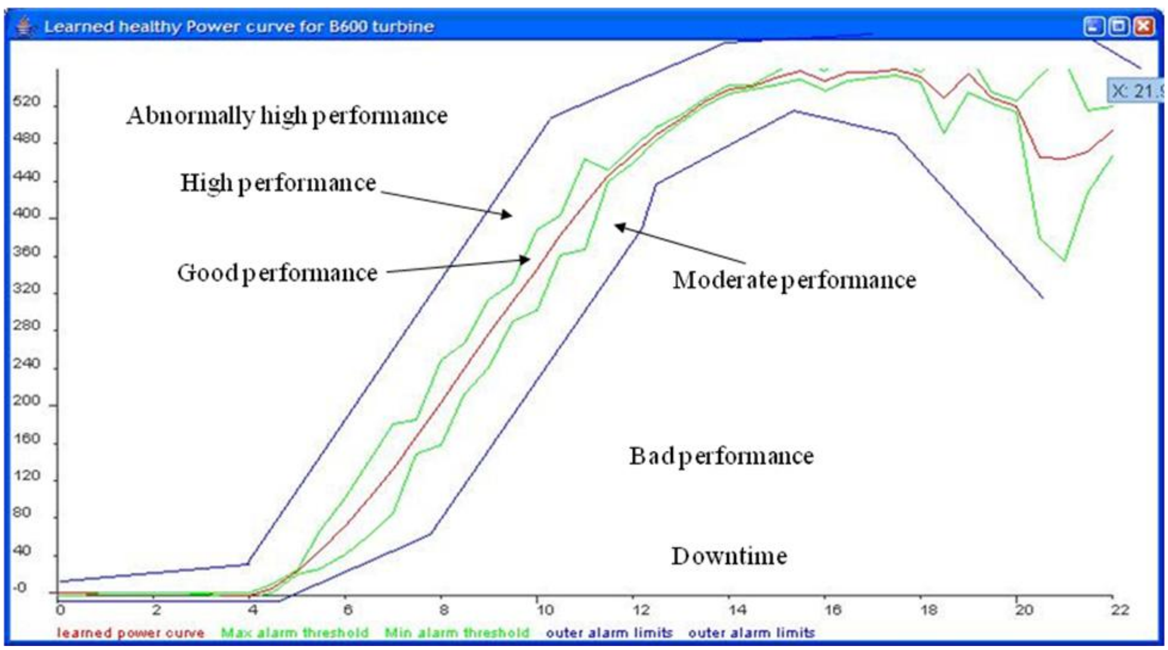

Figure 5.20: Power and wind data pairs are classified into the shown groups according to which alarm limits they fall within.

The total number of performance classifications can be totalled at the end of the month to give an overall indication of how well the turbine performed. Observing the trend of this performance indicator over a number of months can give an indication of turbine performance degradation.

In order to calculate this performance figure for a particular month the sum of all of the pairs which fall above the moderate alarm limit is calculated. Points which fall above this line are assumed to represent "healthy" performance from the turbine and are therefore taken as a percentage of the total number of data points recorded for that specific month. The resulting percentage provides an indication of how well the turbine performed for any particular month. The next section goes on to provide an analysis of the output of both techniques as well as exploring the possibility of corroborating the power module output with the temperature fault detection model outputs.

5.7.2 Analysis of power performance results and corroboration with temperature model outputs

As mentioned previously the learned models from both technques were trained using the same data from T 8, T15 and T16 to allow for a fair comparison of the results. It is important to also note that both methods do not count periods where the turbine is switched off (scheduled maintenance) or islanding (described earlier as feature 1 of the data, see section 5.1) as durations of bad performance. This allows for a fair evaluation of performance between the two methods since periods of (scheduled) maintenance as well as islanding both do not reflect poor power performance on the turbine's behalf, but rather occur as a result of external circumstances outwith the turbine's own control. Unshceduled maintenance (feature 2 of the data see section 5.1 also) however is assumed to affect the performance figures for both methods since they are periods where the turbine should be generating electricity but is not.

This inclusion of feature 1 and omission of feature 2 for the performance figures is done automatically with the NN model since it is exposed to the conditions of feature 1 in the training data but not feature 2. In order to replicate this behaviour for the statistical averaged curve method, *islanding / scheduled maintenance* classifications are added to the totals of the "healthy" groups (*moderate, good, high* and *abnormally high*) described in the previous section. The number of *Downtime* and *Bad performance* classifications however are not. Summing these together therefore gives an indication of the duration of poor performance for the turbine under study in any given month. The totals for each classification from the statistical method output are given in Appendix I. The performance figures from both methods are shown in Table 5.13 below.

Turbine	Analysis Method (see section 5.7.1)	Aug 05	Sep 05	Oct 05	Nov 05	Dec 05	Jan 06	Feb 06	Mar 06	Apr 06	May 06	June 06	July 06	Aug 06	Sep 06	Oct 06	Nov 06	Dec 06
T8	Method 1 NN	99%	91.6%	99.6%	95.7%	99.4%	90.8%	98.4%	99%	96%	77.2%	74.4%	96.3%	97.4%	71.9%	87.4%	64.9% Generator failure	Na Offline
	Method 2 Statistical	90.8%	82.3%	91.5%	90.2%	95%	87.1%	93.7%	92.7%	91.9%	71.7%	71.3%	91.5%	89.6%	64.9%	81.1%	57.9%	Na offline
T15	Method 1 NN	97.6%	95.2%	96.7%	97.7%	98.9%	97.2%	99.5%	97.6%	98.1%	96.6%	96%	96.2%	84.5%	82.4%	87.8%	86.5%	85%
	Method 2 Statistical	87.3%	85.9%	91.8%	94.3%	97%	96.9%	94.5%	93.9%	93.6%	88.8%	93.2%	86.4%	75.6%	74.9%	81%	76.2%	78.1%
T16	Method 1 NN	96.5%	93.3%	95.9%	96.8%	98%	61.8% Gearbox failure	Na offline	78.7%	97.5%	96%	92.7%	95%	92.3%	82.7%	93.9%	84.6%	93%
	Method 2 Statistical	87.4%	86.2%	89.8%	90.4%	87.8%	56.3%	Na offline	71.1%	90.5%	84.3%	79.1%	79.7%	75.7%	68.6%	81.5%	74.6%	85.5%

Table 5.13: Power performance efficiency for T8 (blue), T15 (white) and T16 (orange) from both trained power models

From the numbers shown in Table 5.13 the first thing which can be noticed is that both sets of numbers from the models display the same fluctuations in turbine performance. Another point which is noticeable is the higher efficiency figures given across the complete range from the NN models for all three turbines. This indicates that the NN model captures a closer representation of each turbine's actual power performance curve over the statistical model.

T16 and T8 experienced failures in the gearbox and generator respectively. They were used here for power performance assessment for corroboration purposes in order to inspect if these problems were reflected in the turbine's power output. T15's data was processed for both gearbox and generator anomalies and the results showed no sign of problems in either component and so it was used here as a reference turbine with healthy internals. From the figures shown in the table it does not appear that the gearbox problems of T16 are reflected in the turbine's ability to produce power. T8's generator problems however are evident in the power performance figures in the later months when the generator windings overheat and the safety mechanism interrupts power generation. This overheating is shown in the generator model outputs presented in section 5.6 earlier.

T8 experiences a generator failure in November 06 where the performance figures for both models is very low. Examining the figures leading up to this point reveals that T8 initially starts off with high efficiency between Aug 05 – April 06. It then drops in the months of May, June and September 06 due to the generator cut-offs resulting from excessively high temperatures. Examining the breakdown of the performance classifications presented in Appendix I shows that in the month of May06, T8 experienced a large proportion of unscheduled downtime which is mainly responsible for this relatively low efficiency figure. The output from the generator temperature model shown in figure 5.21 also corroborates this information depicting a few cases of high temperature which caused the generator to cut off, as well as a large duration where the temperature was below typical power generation temperature. This behaviour is repeated (shown in Appendix I) in the months of June and September before the generator finally fails in Novmeber 06.

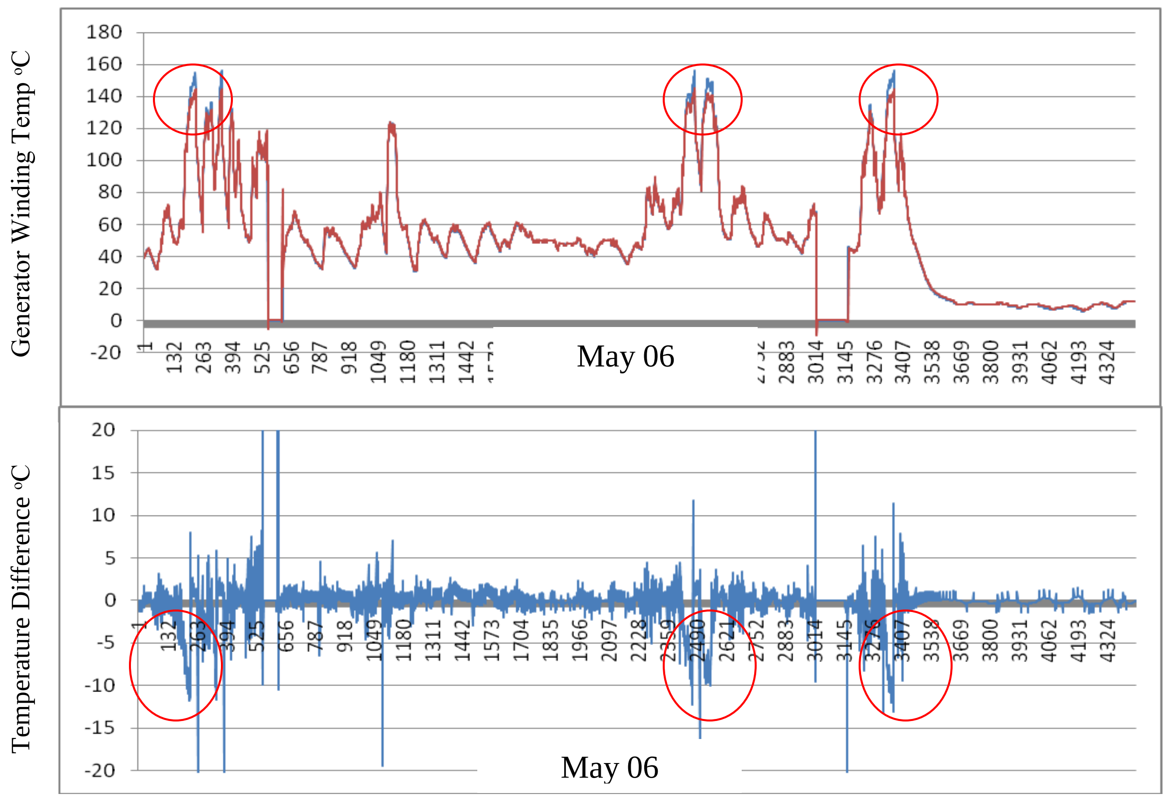

Figure 5.21: Generator Temperature anomalies for T8 during the month of May06.

What is interesting to note from the results presented is that both the power models developed showcase different strengths. The NN model provides a more accurate representation of the turbine's actual power performance characteristics. It doesn't however provide the breakdown of activity in any given month which provides a clearer picture of a specific turbine's performance. Therefore their use in conjunction with one another results in a more informative output to the wind farm operator.

The numbers from table 5.13 show that the performance of T15 decreased from August 06 onwards. Since no other problems were reported for T15, one might assume that this reduction in performance could be down to rotor blade problems. The main problem however hindering a proper analysis of these results is that the state of the rotor blades is not monitored for any of the turbines. Therefore gaining access to information which would allow the author to correlate the drop in performance to a problem in the rotor blades was not possible.

5.9 Model Results conclusion

The results produced by the models developed for this research are very positive. They provide an early warning of problems developing in the gearbox and generator that become apparent through abnormal temperatures. The anomalies detected by the models reduce the volume of data that must be analysed considerably, making the task of the operator much more practical. The models also supply this information in good time allowing for a more informed decision regarding the appropriate action to be made. Also the concept of the methods used here for power performance assessment has one significant advantage. It provides a performance metric in order to measure how well a turbine is performing while removing the dependencies on effects of site specific factors. In this way a turbine's performance is judged according to its own circumstances which makes the performance metric fair across all turbines in a wind farm. While the results presented do not show an immediate use for diagnostics purposes, they do however provide a means to monitoring the efficiency of all of the turbines across a complete wind farm. This serves to provide a key element of decision support for wind farm operators by helping to provide a more complete view of the current status of the turbines in the wind farm.

It can be seen that all of the factors involved in developing and training the NN models can have a significant impact on the accuracy of the relationship captured. Factors such as the training algorithm, the architecture of the network as well as the data selected for training the model all affect the output. The research carried out by (Garcia et al 2006) was the most advanced research to date found in the literature in this area of study. However, the results of the models developed and described provide earlier fault identification on the data set they were applied to.

5.10 Chapter 5 summary

This chapter has described the means used to interpret the SCADA data, converting it from raw data to information which can be of benefit to a wind farm operator. The methodology used to develop the fault detection (gearbox and generator) and power estimation models was detailed. A number of different neural network architectures were trained and tested for both a turbine specific and a turbine generic training data set. The results of the fault detection models of both the generic and turbine specific models were compared and the generic models were found to consistently provide better accuracy across all of the models developed. The models were then applied to the remainder of the data set and a confirmed gearbox fault was detected in its incipient stages 6 months before complete gearbox failure. When compared with the result of 2 days for the model developed by (Garcia et al 2006) this improvement is considered substantial as the 6 month window offers a much more practical time-frame for wind farm operators to determine a maintenance plan.

The results of the generator winding model also provided early detection of overheating 16 months before failure again giving considerable time for maintenance decisions.

The results of the power estimation model were not as accurate as the fault detection models due to the imposing external factors described which can dramatically affect the accuracy of sensors measurements. This coupled with the limitations of the data available recorded by the SCADA system made real-time estimations less reliable. By introducing error bounds on the model estimations, an indication of turbine performance over the complete month could be used to provide an insight into the overall health of the turbine.

It has been shown that corroborating the output between the gearbox based models (oil and bearing) as well as both power estimation models and generator model can provide useful information that allows for a more informative and holistic view of a turbine's current condition.

The next chapter will detail how the models developed in this chapter will be encapsulated into a system which can automate the complete data analysis process dealing with all aspects of data management and processing to monitor a complete wind farm. In this way the combined information extracted from the data by the models can be provided through one convenient point of contact for the operator.

5.11 References

Bonus energy (2009), "Bonus Energy MKIV Wind turbine brochure", accessed September 2009, available at: http://gebrauchtewindkraftanlagen.com/usedwindturbinesdatabase/bonus/Bonus-600kW-product%20information.pdf..pdf

Caselitz, P. and Giebhardt, J. (2005) 'Rotor Condition Monitoring for Improved Operational Safety of Offshore Wind Energy Converters', Transactions of the ASME: Journal of Solar Energy Engineering, volume 127, issue 2, May 2005, pp.253-61

Danish Wind Energy Association. (2008) "The Wind Energy Pioneers: The Gedser Wind Turbine", Accessed February 2009. http://www.windpower.org/en/pictures/juul.htm

Gulski E., Krivda A., (1993), "Neural Networks as a Tool for Recognition of Partial Discharges", IEEE Transactions on Electrical Insulation, Vol.28, Part 6, pp.984-1001, 1993

Hong T., Fang T.C., (2001), "Detection and Classification of Partial Discharge Using a feature Decomposition-Based Modular Neural Network", IEEE Transactions on Instrumentation and Measurement, Vol.50, Part 5, pp.1349-1354, 2001

Infield D., (Wind Resource notes), PROF. DAVID INFIELD, 'Wind resource characteristics' lecture notes, CREST, Dept. Of Electronic and Electrical Engineering, Loughborough University, UK

Leany V., Sharpe D.J., Infield D.G., (1997) 'Condition monitoring of wind farms using 10 minute average SCADA data', European wind energy conference, 1997

Li S., Member, Wunsch D.C., IEEE, O'Hair E.A., Giesselmann M.G., (2001), 'Using neural networks to estimate wind turbine power generation', IEEE Transactions on energy conversion, vol. 16, no. 3, September 2001

Luger G.F, (2002), "Artificial Intelligence – Structures and Strategies for Complex Problem Solving", Addison Wesley Fourth Edition ISBN 0 201 64866 0, 2002

MATLAB NN, (1984-2007), "Matlab Neural Network ToolBoxProduct Help", available with Matlab version 7.0 – R2008b.

Nielsen T.D., (2004), "A Very Brief note about gradient descent search", accessed September 2009, available at: http://www.idi.ntnu.no/emner/tdt4171/files/gradient.pdf

Roweis S., "Levenburg–Marquardt Optimisation", University of Toronto lecture notes, accessed September 2009, available at: http://www.cs.toronto.edu/~roweis/notes/lm.pdf

Sanz-Bobi, M.A., Garcia, M.C. and del Pico, J. (2006) 'SIMAP: Intelligent System for Predictive Maintenance. Application to the health condition monitoring of a wind turbine gearbox', Computers in Industry, Aug 2006, Volume 57, Issue 6, pp. 552-568

Smith B, Link H, Randall G, McCoy T., (2002), "Applicablity of Nacelle Anemometer Measurements for Use in Turbine Power Performance Tests", Published in American Wind Energy Association (AWEA) Wind Power Conference (2002)

Strachan S.M., (2005), “Decision Support System Development for Plant Condition Monitoring and Protection Design Reuse using Structured Methods of Knowledge Engineering and Knowledge Discovery”, PhD Thesis submitted June 2005, Institute for Energy and the Environment, University of Strathclyde.

Tarassenko L. (1998), “A Guide To Nueral Computing Applications”, ISBN 0 340 70580 2, Arnold, 1998

Zaher A, McArthur S.D.J, Infield D.G, Patel Y., (2009), “Online Wind Turbine Fault Detection through Automated SCADA Data Analysis” 2009, Wind Energy Volume 12, Issue 6, Pages 574-593, September 2009, Online ISSN: 1099-1824, Print ISSN: 1095-4244 Copyright 2009 John Wiley & Sons, Ltd

Author Discussions

Yusuf Patel, (2007) Author discussion. Wind farm operator.

6 Methods and Applications II: The Design of a flexible Multi-Agent System Architecture for the CM of wind farms

In order to integrate all of the analysis techniques described in the previous chapter into one convenient point of contact for an operator, a framework that is capable of integrating multiple data sources and multiple independent processes is needed. Multi-agent systems (MAS) as detailed in chapter 4, provide the necessary framework for designing such systems, allowing different tasks to be encapsulated into separate modules (agents) with independent objectives.

The deficiencies of the scheduled maintenance approach currently used for the condition monitoring of wind farms have already been outlined in chapter 2. The integrated monitoring system installed on the wind turbine produces large volumes of data which requires interpretation by an expert that can extract meaningful information from it. Currently the data is collated and left unused due to a limitation in resources of experienced personnel that can analyse the data. Because of this, unexpected failures which are not discovered during scheduled maintenance operations will only become apparent once the failure has actually occurred. A maintenance decision of whether to replace or fix the failed component is then decided based on an inspection of the plant. As has been discussed in chapter 2, this procedure is time consuming, unnecessarily lengthening turbine downtime leading to an un-optimised operating wind farm. These are considered problems which are apparent in all conventional monitoring systems which employ breakdown and scheduled maintenance policies.

The purpose of this chapter is to describe the architecture of the MAS software developed using the JADE API (mentioned in chapter 4) for this thesis with the intention of attempting to realise and employ a condition based maintenance policy for wind farm O&M. In order to be of benefit to wind farm operators, the large volumes of SCADA data acquired must be automatically analysed in order so the operator is relieved of the process of manually analysing and interpreting the data. This chapter will describe how this automation process is realised through a delegation of tasks and responsibilities to a number of different software agents working in conjunction with one another to achieve the overall system objective of automating the data analysis process. Each agent's responsibilities and processes will be detailed along with the ontology designed and developed to allow the agents within the system to communicate with one another. The system is designed to be applicable to wind farms consisting of any number of turbines and model, where the data management and data processing required can be modified to accommodate such differences between wind farm sites.

6.1 System Design and Architecture

The initial stage of the system design is to decide how the overall task of automating the data analysis process should be split into sub-tasks and delegated to independent agents. Each agent is then responsible for processing and carrying out the necessary negotiations with the other agents within the system to achieve its own objective. In this way the overall system objective can be met.

The main factor found to affect this design choice was the extent of the flexibility required by the application. For this thesis it was necessary to develop a system which is adaptable to any wind farm. This includes being able to process data stored in any format, carry out the appropriate data management to prepare the data for processing, route the data to the appropriate data processing agent, and finally present the processing output to the user.

This stage of the development was iterative in nature and the final proposed system design was not reached instantly. The final proposed system architecture is shown in figure 6.1 below.

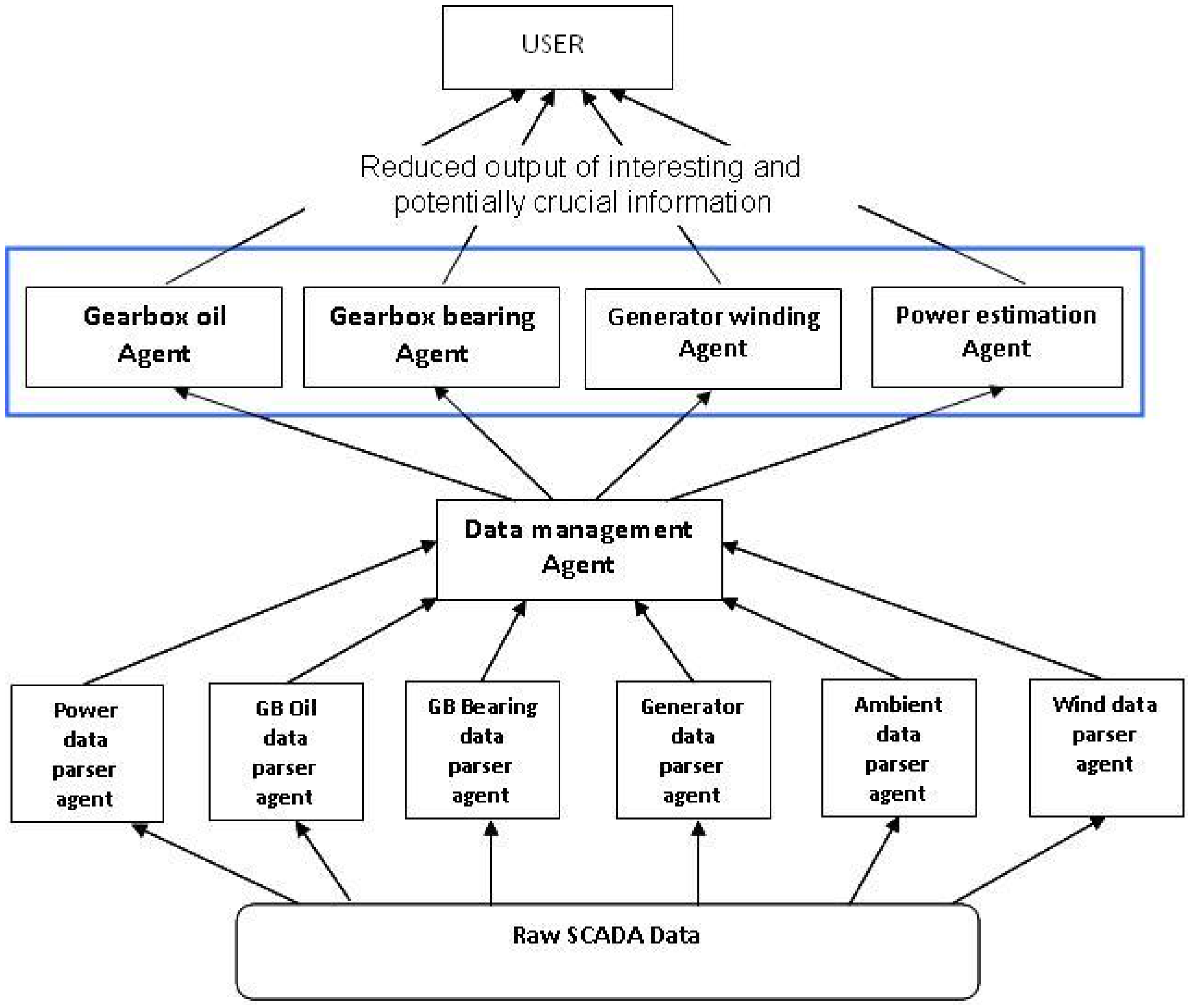

Figure 6.1: The Proposed System Architecture

The system is composed of 3 different kinds of agents consisting of four independent interpretation modules, data parser agents to handle and prepare the data for processing, and, finally the data management agent. The data management agent is responsible for collecting and formatting the required parsed data reducing the overall volume of data into feature vectors which can then be sent ready for processing by the appropriate interpretation module. By simply adding in a new data parser agent, this gives the system the capability of reading data files of different formats from different wind farms, while the data interpretation agents can be easily retrained and updated to detect faults for different turbine models. The data management agent can also easily be given a new behaviour that can accommodate the new data type and incorporate this in its feature vector creation and sending process. This architecture gives the system the flexibility to be applied to any wind farm of any size.

For a final revised system, the output of the data processing agents would ideally be presented through a clear and concise user interface. This was however considered out-with the scope of this research as the main focus is to prove the concept of an extensible and flexible automated fault detection system which provides the necessary framework for automating the data analysis process and offering the capability of simultaneously corroborating the output between a number of different analysis techniques

6.1.2 The Ontology Design

An integral part of the system design is the design of the ontology which allows the agents within the MAS to communicate with one another (see chapter 4). The ontology must be defined when the agents are being developed so that an understanding of the necessary communication between agents can be envisaged. The development of an ontology can often impede the flexibility and extensibility of the system in the future. As these are both important features necessary for the development of the system it is therefore vital to consider how an ontology is likely to extend over time rather than attempt to define a complete ontology from the beginning. Attempting to define a complete ontology from the start can potentially lead to the development of a closed ontology which in turn can lead to complications when upgrading the system. By keeping the ontology as simple as possible, covering only the modes of necessary interaction envisaged at the time of development (Noy & McGuiness) without trying to anticipate future extensions, an open ontology can be anticipated. Ironically by not anticipating future extensions, future additions to the system can be more easily implemented without the need for any of the existing agents already developed requiring modification.

As described in chapter 4 (see section 4.5.2.1) an ontology can be defined through the use of concepts and predicates. A concept is used to represent objects and a predicate is a statement which evaluates to true or false. For the system designed for this thesis, a concept only based ontology was used as the need for application specific predicates was not necessary. The FIPA-SL content language described in chapter 4 provided the basic predicates such *equals*, *and*, and *implies* which were found to be sufficient for the communication requirements necessary for this application.

Each concept contains attributes (or 'slots' to which they are sometimes referred), which help define the information it holds. For example a power data concept used to represent one SCADA data instance from the active power measurement recorded is shown below in figure 6.2

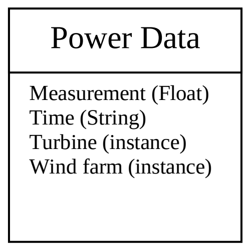

Figure 6.2: Power Data Ontology Concept

As shown from figure 6.2, the attributes can be different data types such as float, string or even an instance of another concept within the ontology. For example the use of the *turbine* slot would mean the power data measurement read from the data files would be associated with the turbine from which the measurement was taken. In this way the power data concept holds more useful information.

For future modifications to an ontology, an easily upgradable change would include for example introducing a new concept or a new slot which provides more information regarding a particular concept. However modifications which require the removal of slots or whole concepts would require amendments being made to the existing agents and therefore this should be avoided at all costs.

Bearing these points in mind the final ontology designed for the system developed was modelled to capture all aspects of the data and any concepts necessary for its processing as wells as any information that may be generated as a result of this processing. The ontology designed is shown in figure 6.3. The ontology was developed through the protégé v3.3 software. This software allows the user to define the ontology, and then once finished automatically create the ontology as a collection of classes (in JAVA) which can easily be imported and used within the intended piece of software.

The ontology (shown in figure 6.3) consists of four parent concepts, namely Wind-farm, Turbine, Component and Data. The ontology developed aims to split the concepts into two main types, real world objects and intangible software objects necessary for data processing. The wind-farm, turbine and component concepts all model real world objects which allow them to be individually identified by the Agent within the system while the Data concept represents the items being processed and generated by the system. The wind-farm concept holds basic information such as the string based site name and location as well as the integer based no. of turbines. It also holds multiple instances of the turbine concept one for each turbine installed in the wind farm which allows each one to be individually identified. Each turbine concept consists of a string based turbine-name attribute, a gearbox and generator both of which are instances of the component concept. As the system evolves with new data processing techniques dedicated to process data associated with different turbine components, new component concepts can be added under the component hierarchy and included as slot instances within the turbine concept.
The Data concept represents all forms of intangible objects prepared (Parameter), processed (Feature Vector) and generated (Generated Output) by the system. Any instance of the Data concept can take one of these forms. Each of these 3 different forms of data can further go on to take different forms which define exactly what object they represent to give further meaning to the various data forms used by the system.

As can be seen from figure 6.3, only the necessary concepts which will allow communication to occur between the agents regarding the SCADA data, the expected processing and the generated system output taking place between the agents for the current level of system processing was included into the ontology. The parent concepts where defined to be of a general nature, in this way any future addition to the system can easily be added under the appropriate hierarchical heading. For example if new forms of data are acquired and are to be included in the system's processing, then this can be added under the DATA concept. Each agent within the system is responsible for creating, sending and or processing particular concepts from the ontology. The following section describes the functionality of each of the system agents detailing how the ontology concepts are used by the system for inter-agent communication.

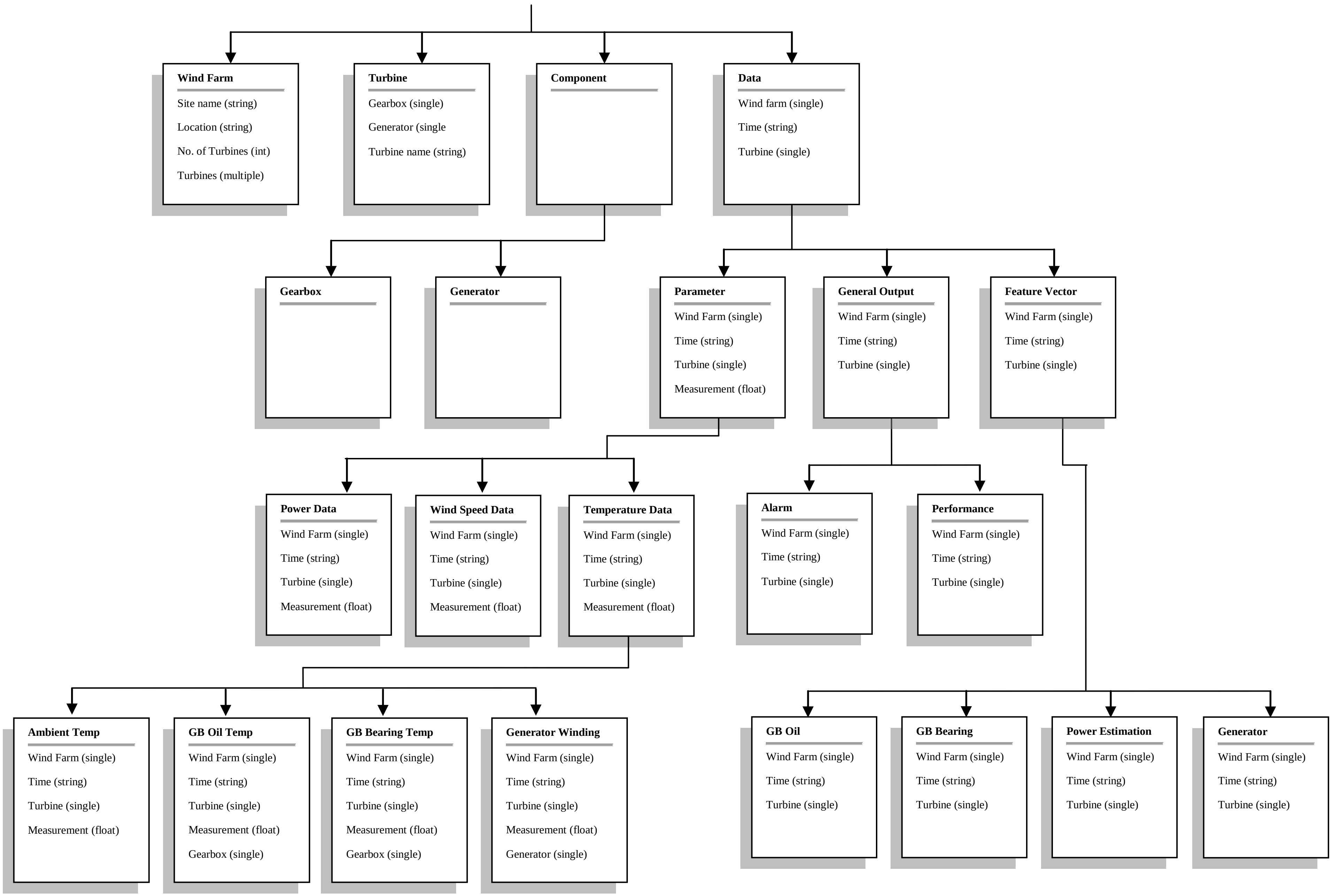

Figure 6.3: The Designed System Ontology

6.1.3 The System's Agents

As was briefly described in chapter 4 (see section 4.4.1) agents are built around what is known as a behavioural architecture. Each agent's responsibilities are split into behaviours that are executed and coordinated in such a way that enable it to complete its task. When developing an agent, the programmer is given the flexibility to produce the desired functionality using a number of different behaviour types. There are generic behaviours that can be called at any point in time, reactive behaviours that can react to a specific situation and timer-based behaviours that execute in a timely fashion at set time intervals. More details on the different types of behaviours available for use are documented by Bellifimine et al 2007. Regardless of the type of behaviour, they each must implement two main methods, namely the action() method and the done() method. The action method is the segment of code which defines the behaviours' actions and the done method defines the state or conditions where the behaviour is considered to be completed. The Agent scheduler (hidden from the programmer) which is implemented by the base agent class controls the behavioural operation of each agent using these methods. It runs the action methods of all the behaviours in the ready queue and then checks the done method to determine if the behaviour has been completed. Once the conditions set in the done method are met, the behaviour is removed from the ready queue by the scheduler and this process is repeated as the agents add and remove behaviours as necessary to complete their objective.

The following sections describe the functionality of the different system agents, the behaviours they implement and the ontology concepts they utilise for communication.

The Data-Parser Agents

The data parser has the responsibility of reading the data files, parsing them for the appropriate data and then sending this data in an understandable format to the processing agents. It does so by reading the SCADA data files and creating the appropriate Data Concepts for the SCADA parameter read. This allows the 'parameter type' of each data point read by the parsers to be identified by the remainder of the agents within the system. For example, the power data parser agent accesses the power data files, reads one measurement for all of the turbines within the wind farm at the current time stamp, creates a power data concept for each measurement read, and then completes the attributes with the appropriate information. In this format the data is ready to be put into the appropriate input vectors required for each model detailed in chapter 5, encapsulated in the interpretation agents.

There is a data parser agent for every data type used by the system as this allows for more flexibility. New data types can easily be added to the system by introducing a new parser agent and similarly data types no longer used by the system can easily be removed by removing the particular parser agent. The data files acquired for this thesis were stored in comma separated files (.CSV format). Figure 6.4 shows an extract from the active power file for the month of April 06.

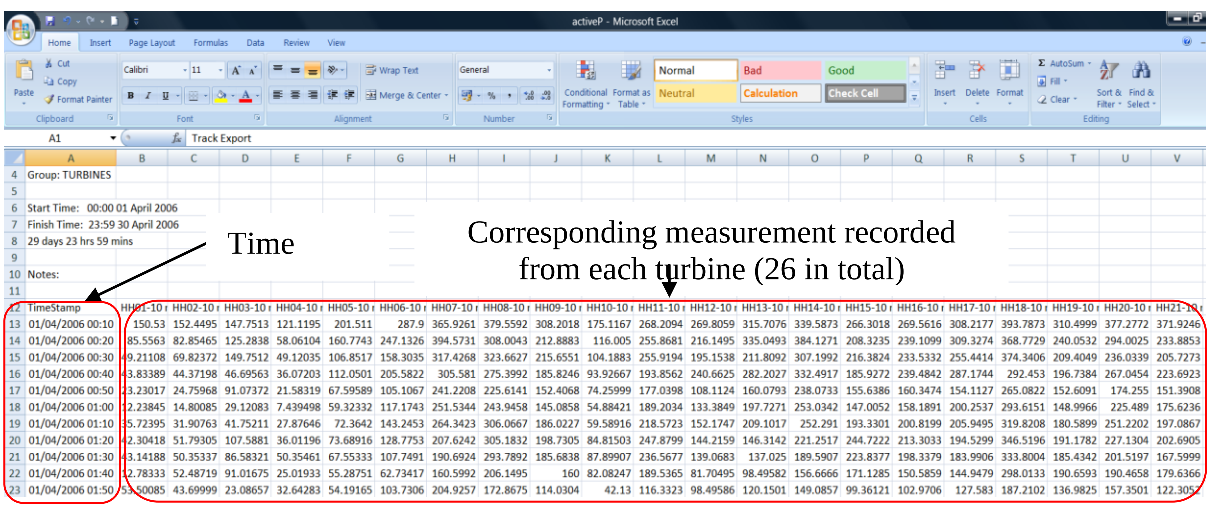

Figure 6.4: Extract from Exported SCADA file from SCADA Database

In order to simulate a real-time environment, the parser agents were developed to operate in a manner which polls the data files periodically to read the next data point. This

method can be applied in the same manner to a real-time environment, the only difference being the data source polled for the newest data would be updated as necessary to collect new data instances as they are generated by the turbine monitoring system.
The functionality of the data parsers is split into the following three main behaviours:

1. Parse data behaviour
2. Parse waker behaviour
3. Subscription responder behaviour

The parse data behaviour is responsible for accessing the data file, finding the required data and storing it temporarily in a format that can be sent to the data management agent. The parse waker behaviour is a timed behaviour that controls the time interval between readings of the data file to parse the next available measured value. The data parser agents are data providers, so upon initiation they register with the DF that they can provide a specific type of data. In order to do so they must exhibit a *subscription manager* (FIPA IP Spec) interface that allows them to handle any number of agents that are interested in receiving this data that they can provide. The subscription *responder* behaviour allows it to do this by handling any subscriptions sent to it. It then takes care of all the subscribed agents (the data management agent that requires this data to prepare the data for processing) by sending out the data to it as soon as it becomes available.

To summarise, the overall functionality of the data parser agents can be described by figure 6.5 below. The raw data is streamed through the appropriate data parser, one data point at a time. This data is converted into the appropriate data concepts so they can be identified by the remainder of the agents within the system and then sent off to the subscribed agents.

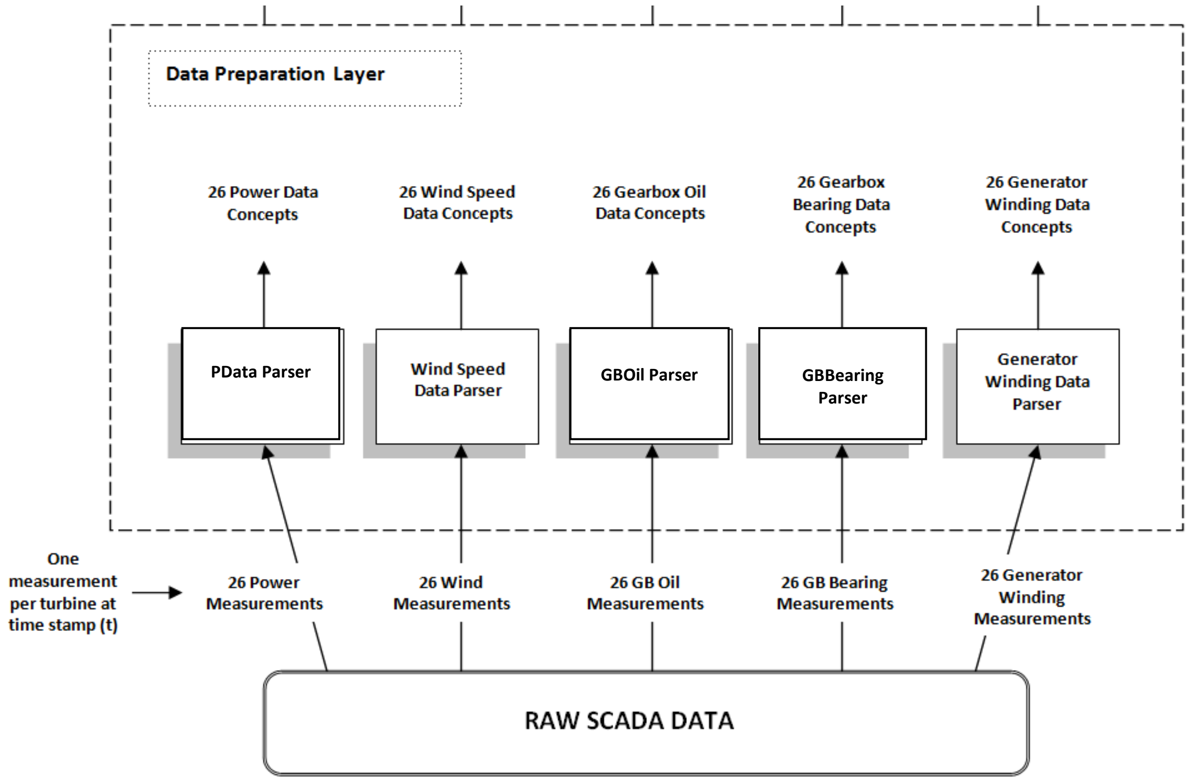

Figure 6.5: Summary of Data Flow through Data parser Agents (the data preparation stage)

The Data Management Agent

The role of the data management agent is to organise the data received by the data parsers into the corresponding input vectors required by the fault detection agents. It therefore subscribes to all of the data concepts prepared by the parsers temporarily storing this data until there are a sufficient number of data points available to create the input vectors. The reason for this is that each input vector contains regressive inputs, with the last of these readings going back to t-3 previous readings for the power input used for the three fault detection models. As the data is read into the system by the parsers one data point at a time at the current time stamp, a moving window is stored for the last four readings (t, t-1, t-2, t-3) for each SCADA parameter type. Upon initiation the data-management agent is supplied with the number of turbines installed in the wind farm (information supplied by the operator). The data-management agent then uses this information to prepare the data storage windows, creating an individual window for each data type for every turbine. With the current system setup and the data available (i.e. 5 data types), there would be 5 data windows created per turbine each holding 4 data instances.

As the data is streamed through the system, the stored data points are propagated through the data storage window allowing new vectors to be created with each data point received by the parsers. In other words each of the 26 data concepts are moved from time point *(t)* in the data storage windows to time point *(t-1)* when a new set of data type concepts is read into the system. This process is repeated as the data is read into the system. These actions are depicted more clearly in figure 6.6 which details the data flow through the data management agent below. It is important to remember that each of the data parsers is an independent agent which can reside on different computing resources (i.e. different machines). Therefore the data concepts sent by them are not necessarily sent and received by the data management agent in a synchronised manner. Therefore, in order to avoid synchronisation errors at the data processing stage by the data processing agents, the data management agent consistently checks if there is sufficient data stored in the data windows to create the input vectors before attempting to send them off for processing. This described functionality is carried out through the following main behaviours:

1. Power data Subscription Initiator
2. GB Oil Subscription Initiator
3. GB Bearing Subscription Initiator
4. Ambient Subscription Initiator
5. Wind Speed Subscription Initiator
6. Store Data
7. Create Vectors
8. GB Oil Vector Subscription Responder

9. GB Bearing Vector Subscription Responder

10. Generator Vector Subscription Responder

11. Power Estimation Vector Subscription Responder

The subscription Initiator behaviours are responsible for subscribing to the appropriate Data-Parser agent in order to receive the required data. The store data behaviour stores the data received from the parsers into the corresponding data-storage window depending on the type of data received by the data-management agent. Once there is sufficient data stored in the data windows, the create-vectors behaviour is initiated which then sorts the data into the appropriate input vectors before sending them on to the subscribed agents via the subscription responder behaviours. Figure 6.6 depicts the data flow through the data-management Agent.

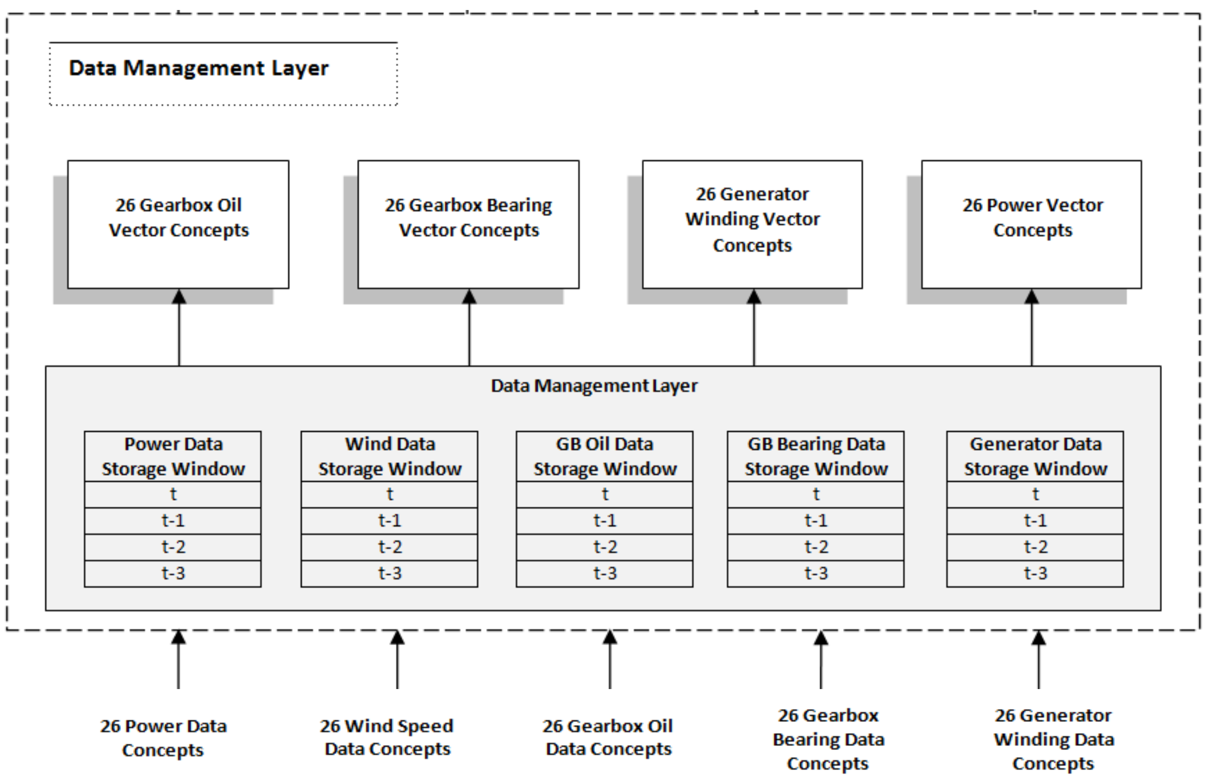

Figure 6.6: Summary of Data Flow through the Data Management Agent

The Data Interpretation Agents

The role of the Interpretation Agents is to identify the abnormal SCADA data instances in the parameters monitored which represent the manifestation of faults within the turbines' components. These agents encapsulate the 3 fault detection models as well as the power estimation model developed and described in chapter 5. The models developed are utilised by the agents to give them the ability to analyse the large volumes of SCADA data prepared and formatted by the data-parsers and the data management agent. These agents automate the process of data analysis to determine which data is of interest to the wind farm operator. By identifying these abnormal instances of data automatically, only a considerably reduced volume of data is required to be passed on to the operator for their attention. This dramatically can save both time and resources from the operation and maintenance perspective.

By highlighting the information of interest to the operator the important and final decisions regarding the maintenance of the machines is still determined by the operator. This ensures that this way the system can offer a form of decision support to the operator without interrupting wind farm operation as was intended.

In order to encapsulate the 3 NN-based fault detection models developed (described in chapter 5), an open source neural network Application Programming Interface (API) named JOONE (Java Object Oriented Neural Engine) was utilised to allow the NN models developed and trained in the MATLAB package to be incorporated into the Interpretation agent's code. The MATLAB NN toolkit conveniently provides the ability to view the trained NN weights and biases. This was useful as these values hold the knowledge attained by the models therefore allowing them to be exported and incorporated into the Agent code. The same NN could therefore be re-constructed using the JOONE API where networks with the same number of layers, neurons and neuron transfer function used by the models to be reconstructed and utilised within the Interpretation agent's behaviours.

The three fault detection based agents are built using the same behavioural architecture as they all operate in the same manner using a NN-based model which provides some numerical output. Their operation is identical in nature with the only discrepancy between them being the NN model loaded in to the each agent's memory. The power estimation agent operates in a different manner due to the differences in the model built and its resulting output.

NN-Based Fault Detection Agents

The functionality of the fault detection agents is split over the following three behaviours:

1. load Model
2. Feature Vector Subscription Initiator (for the appropriate Feature Vector)
3. Process Vector (for the appropriate Feature Vector)

As described in chapter 5, the temperature based fault detection models are trained in an offline fashion where the appropriate data sets are specifically prepared for this purpose. Therefore upon initiation, each fault detection agent loads up the previously trained NN model stored in the numerical form of weights and bias values, and then creates the corresponding NN with the appropriate number of layers and nodes. This model is stored in the agent's memory so that it can be called up for use at any point in time. The process of updating the model can therefore be carried out in a simple straightforward manner as a new model (represented by different weights and biases) can easily be loaded giving the agent different interpretation capabilities depending on the skill learned and captured by the NN used.

The Feature Vector Subscription Initiator behaviour takes care of subscribing to the agent (Data-management agent) which provides the particular vector needed for processing. Finally the process Vector behaviour uses the feature vector received, feeds this as an input to the loaded NN model, which in turn provides an output to the user. The form of the output is the value of the model estimation as well as the actual measured output. In this way the results can be graphed to provide a visual representation of the model estimates compared to the actual temperature trends recorded in the SCADA data.

The important instances of data are highlighted to the user through Alarm concepts which can be generated by each of the processing agents. An Alarm concept is generated depending on whether anomalous behaviour (abnormally high temperatures) is detected for a sustained period of time. This functionality is achieved through a direct difference check between the model estimate and the actual temperature recorded. A difference greater than 3˚C (larger than the 2.21 RMS estimation offset error listed in table 5.10 and the results of the generic models presented in chapter 5 section 3) between the two values triggers a monitoring process in the Process Vector behaviour to determine the severity of the deviation. This monitoring process carries out checks on consequent pairs of model estimates against actual temperatures for a duration of one hour. As data instances are generated every 10 minutes, this represents 6 data pairs which must be checked and compared. If the consequent pairs produce a difference of 3˚C or greater i.e. the deviations consistently become more severe within this duration of one hour, then an alarm concept is generated by the processing agent. This process of monitoring helps to avoid transient errors in both the data recorded by the CM system and the model estimates resulting in a generated alarm concept, thus dramatically reducing the chances of false alarms (or false positives). The sensitivity of these alarms can be adjusted by modifying the duration for which this monitoring process occurs where increasing the duration would result in a less sensitive alarm threshold, and decreasing the duration would create a more sensitive alarm generation system. This level of sensitivity can be left to be determined by the operator as different turbine models are not guaranteed to operate with the same operational characteristics. The results presented in chapter 5 (see sections 5.5 & 5.6) showcase the number of alarms flagged by the system for the faults detected by each model.

Power Performance Agent

The principal difference in operation between the power performance agent and the fault detection based interpretation agents is due to the online training process used to learn the power characteristic of each turbine as opposed to the offline training used for the fault detection models. The functionality of the power performance agent is distributed across four behaviours:

1. PowerPerformance feature Vector Subscription Initiator
2. TrainData
3. LearnCurve
4. MonitorData

Upon initiation, before any of the behaviours are called, the power performance agent prepares the appropriate memory space for creating and storing the learned power characteristic curve for each turbine in the wind farm. A Boolean based variable which states whether a learned curve for each turbine exists is also created and set initially to false since the curves have not yet been learned as the agent is initially started. This Boolean variable controls the mode of operation of the power performance agent calling one of two behaviours, namely the TrainData or MonitorData behaviours. When the turbine exists variable is set to false for a particular turbine, the mode of operation for that particular turbine is set to training mode. Similarly if the Boolean variable is set to true then the corresponding mode of operation is set to Monitoring mode. For each mode of operation the appropriate behaviour is called. In a wind farm employing a CBM policy, each turbine in the wind farm will undergo maintenance operations at different times according to its current health conditions. The functionality provided by the Boolean variable is therefore considered advantageous since it gives the agent the capability of simultaneously operating in different modes for each turbine in the wind farm monitored by the system. This provides the flexibility required by the system to operate in the appropriate mode in order to accommodate every turbine’s specific operating schedule. Since the mode of operation is simply controlled by a Boolean variable, a new power curve for any particular turbine can easily be learned at the request of the operator by simply resetting the value of this variable to false.

The subscription initiator behaviour is responsible for requesting and receiving the power performance feature vector data. The power performance feature vector contains the appropriately time stamped power, power SD, wind speed and wind speed SD pairs necessary for the training and monitoring phases of the developed power performance algorithm detailed in chapter 5 (see section 5.7).

Once a feature vector is received, (when operating in training mode) the TrainData behaviour is called. This behaviour is responsible for storing the data received until there is sufficient data available in the agent memory store to learn the turbines corresponding power characteristic. Once sufficient data has been collected, the LearnCurve behaviour is called calculating the averaged power curve and the associated alarm limits. The turbine exists Boolean variable is then set to true which then triggers the MonitorData behaviour whenever a new feature vector is received switching the agents functionality from training mode for the specific turbine to monitoring mode.

The MonitorData behaviour is then responsible for classing all consequent power and wind speed pairs with the appropriate performance label according to where they lie in relation to the learned power characteristic curve. It stores the labels generated for the complete month, and then using this information creates a performance concept summarising the turbines overall performance over the month. Figure 6.7 below depicts the data flow through the Data-Interpretation layer.

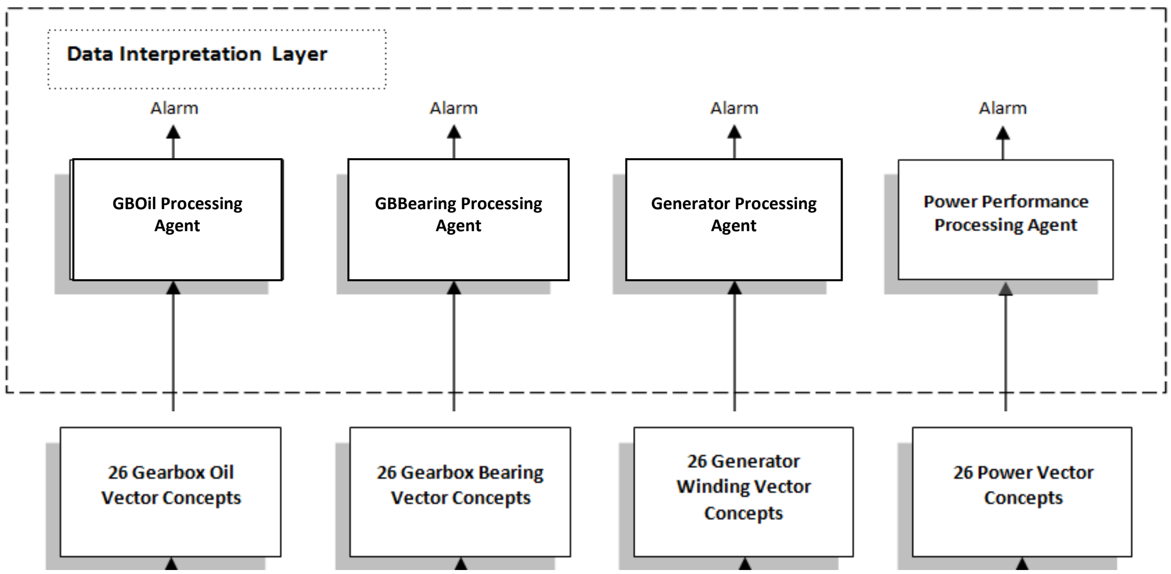

Figure 6.7: Summary of Data flow through the Data-Interpretation Layer

6.1.5 Summary of System Architecture

When figure 6.5, 6.6 and 6.7 are all put together, a complete view of the data flow from the point where it enters the system developed as raw data to the point where it leaves the system as information can be seen. From figure 6.8 below, it can be seen that the data flow is considerably reduced as the data propagates upwards through the layers of the system and out through the data-interpretation layer. This is beneficial from a wind farm CM perspective especially given that the JADE MAS framework allows agents to be distributed and run on a variety of computing resources interconnected through either local or wide area networks or even a combination of both. The current data acquisition setup currently in place at wind farms (at the time of writing) is through the integrated SCADA systems installed at in the wind farms at the time the site is commissioned. These systems collate all of the data recorded from each individual turbine installed at the wind farm and store it at some central repository such as the central PC pictured in figure 6.9.

By distributing the developed agents (described in this chapter) so that all of the processing is carried out on-site at the wind farm, the volume of data for investigating potential problems at a wind farm which would require to be transmitted or downloaded from a remote location would be dramatically reduced. This would result since only the alarm and performance output of the interpretation agents would need to be sent back to the remote location accessing the wind farm information. The alarm information sent by the processing agents would allow the operator the ease of investigating the turbines of interest with sufficient time to decide what maintenance action should be taken from the convenience of a remote location. This is especially beneficial for offshore sites where access is severely limited.

As mentioned previously, an interface agent could be developed and run at the remote offsite location. This is the only part of the system yet to be developed as part of the research presented in this thesis and forms part of the useful future work section in chapter 7. This agent could subscribe to the alarm and performance information produced by the interpretation agents. In this way the alarm information can be presented in a clear and concise way to the operator.

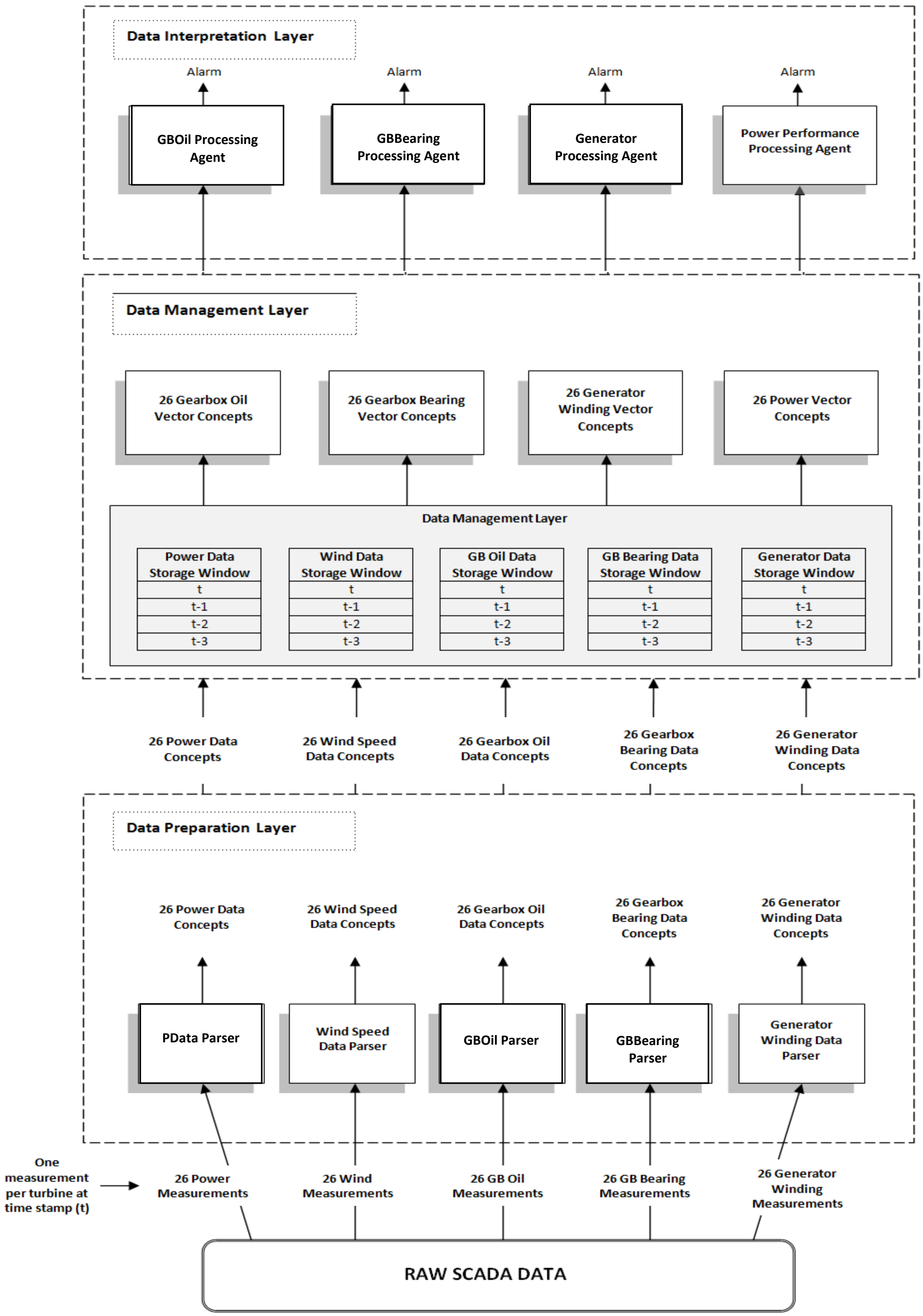

Figure 6.8: The Data Flow through the System's Various Layers

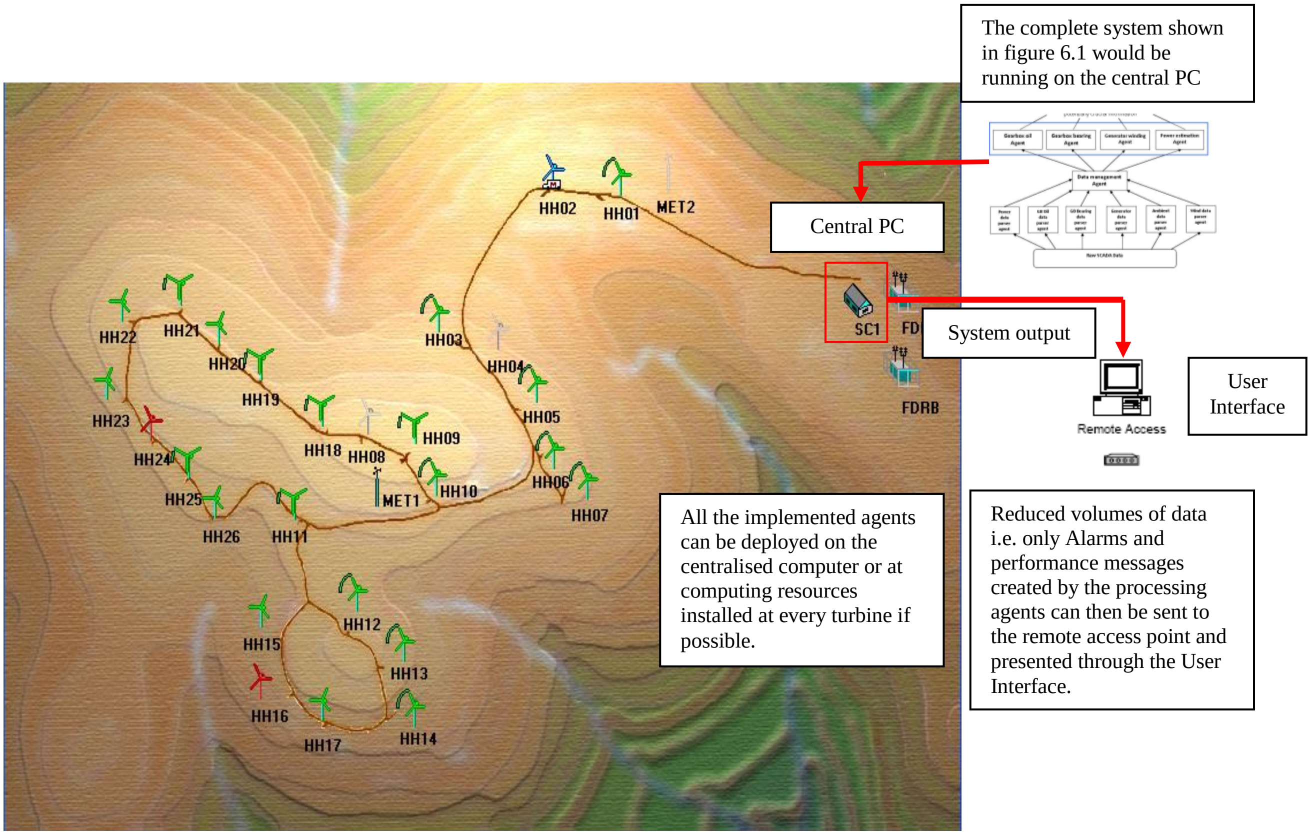

Figure 6.9: SCADA Wind farm network infrastructure

As has been mentioned before the Agents developed within the system can easily handle data from any number of wind turbines. Also the NN based normal behaviour models developed and embedded within the processing agents can be retrained to accommodate new and different turbine models. This would allow the system to be deployed at larger wind farm sites of any size e.g. Whitelee, Europe's largest wind farm making the developed system extremely flexible.

6.2 System Testing through Agent Interaction

Once the individual software agents were developed, a test run of the system was carried out in order to ensure that each agent functioned collectively in conjunction with one another to achieve the overall system objective. When the system is initialised and each of the agents is launched, the interaction between the agents and the DF can be viewed through a JADE graphical user interface (GUI) tool. This tool allows the programmer to supervise the operation of the system to ensure correct operation of the system as a whole by monitoring the messages sent between the agents. While each agent can be launched individually, to simplify this process, an initialisation agent was created to launch the remainder of the agents within the system. For this thesis the agents were all run on the same computing resource. This was sufficient for testing purposes as the necessary agent interactions for system operation still take place between the agents in the same manner regardless of how they are distributed across computing resources. An example of the interactions between the agents on system start up and data being read and passed through the layers of the system is shown in figure 6.10 below:

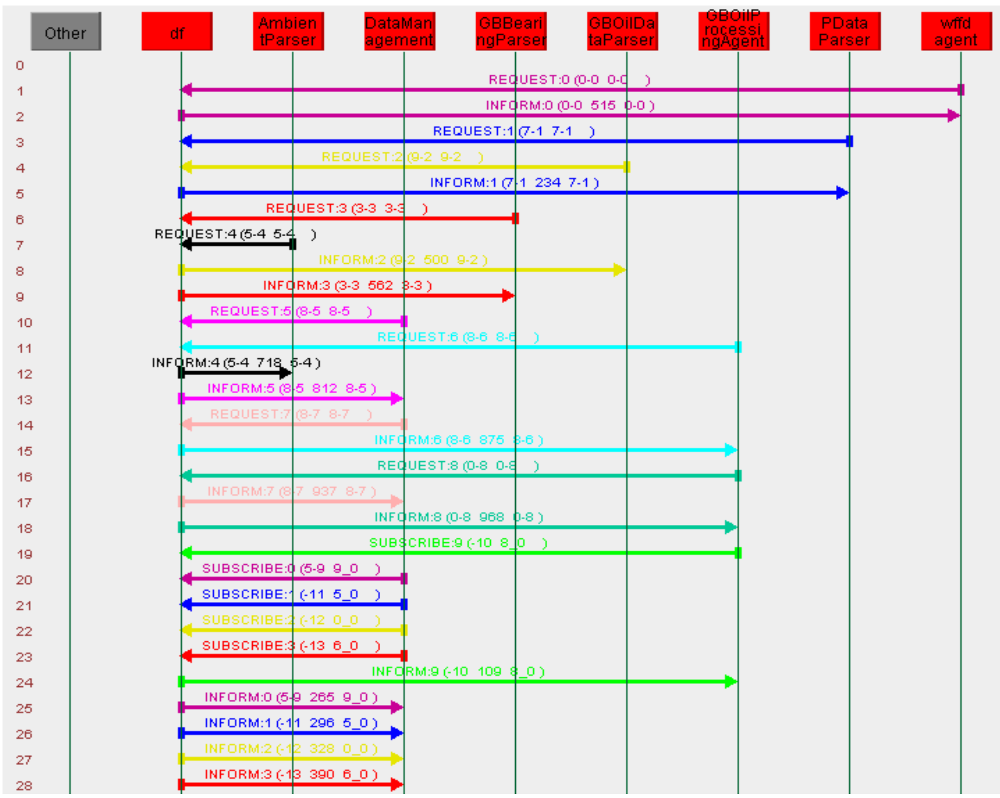

Figure 6.10: Agent interaction as the system is initialised

Figure 6.10 (above) and figure 6.11 (below) depict a scenario of communication where a number of data parsers and the data management agent are preparing data for one processing agent, namely the gearbox oil processing agent. This particular setup was chosen to allow for clarity in the explanation and also to portray agent interaction through all three layers of the system.

Figure 6.10 portrays the communication taking place from message 0 – 28. These messages display the interaction from the point of system initialisation (i.e. as soon as the agents have been launched) up to the point where the agents have registered their services with the DF successfully and are now ready to carry out their processing. The initial request and inform messages sent between each agent and the DF signify the agent checking with the DF that no other agent with the same name already exists on the platform providing the same services. For example messages 3 and 5 represent the PDataParser agent carrying out this process.

In the case where an agent must also subscribe to the DF for a service which it requires to complete its task, further negotiations must take place between the DF and the involved agents in order to set up this link of communication between them. For example the DataManagement agent (which requests all data objects to be sent to it in order to organise and format it into feature vectors), sends its second request message (message 14) to the DF requesting information regarding which agent on the platform provides the particular data it requires. The DF responds with an inform (message 17) supplying the information regarding the agent's which supply this information. For the scenario presented in figure 6.10, only the gearbox oil processing agent was launched along with the appropriate data providers which were required to provide it with the necessary data types. The data management agent therefore automatically sets up subscriptions for the data types available provided by the four parser agents running for this particular scenario sent via the DF (messages 20-23). The DF then responds accordingly to the subscribe messages by informing the DataManagement agent of the details and network addresses of the agents which can supply it with the data it requires.

Figure 6.11: Data preparation and management based messages

Another subscription is also set up between the GBOilProcessing agent and the DF shown by messages 16, 18, 19 and 24. These messages represent the subscription required between the DataManagement agent and the GBOilProcessing agent where the former provides the latter with the feature vectors ready for processing.

Figure 6.11 shows the result of both these subscription procedures. As each parser agent parses the data files and creates the appropriate data concepts, it then sends this as an inform message to the DataMangement agent as requested. This process is continued all the while the subscription is in place where the duration of how long it is established is left at the discretion of the subscribing agent. In the case of this system the process of data analysis is ongoing and so the subscriptions are left intact for the entire duration the system is operational.

After four data concepts from each data parser have been sent to the DataManagement agent (shown by messages 29-40) the data windows contain sufficient data to create the feature vector. Only once the feature vectors have been created does the DataManagement agent begin sending this information to the GBOilProcessing agent regardless of when the subscription was set up. For every four data concepts that are parsed and sent to the DataManagement agent, a new feature vector is created and sent to the GBOilProcessing agent shown by messages 45, 50 and 55.

For this agent interaction test scenario it can be seen that all of the launched agents perform the necessary negotiations which are required to take place in order to facilitate the automated data processing and interpretation objective of the system.

6.3 Research Outcomes & Contributions

This chapter has described the multi-agent system software architecture designed and developed for the purpose of automating the SCADA data analysis procedure in this thesis. It has detailed the different independent modules (agents) within the system and the role they play towards achieving the system's objective. The ability to automate the complete data analysis procedure in real-time as the data is generated by the CM systems simplifies the role of a wind farm operator when it comes to determining maintenance decisions regarding their machinery. Every CM system has the potential to generate data at a faster rate than can be manually analysed by an engineer. The design of the developed system relieves the operator from this problem by highlighting only the interesting information within the generated data automatically without any human intervention. In this way the operator need not consume valuable time monitoring the system as it operates but rather wait until they are notified by the system through the alarms which are generated.

The ontology which defines the concepts that allow inter-agent communication has been explained in detail. The internal behaviours of each agent and how each functions to achieve its desired task as well as the ontology concepts used by each agent have all been detailed. The system test undertaken allowed agent interaction and functionality to be examined. Using the JADE GUI tool the results of this test showed that the desired system operation was realised, evident through the messages sent between the agents which followed the correct course of communication as the data is propagated through the different processing layers of the system.

In summary the novel contributions and conclusions presented within the research presented in this thesis can be summarised as follows:

- An investigation of AI based diagnostic techniques suitable for achieving successful novelty detection models for a variety of parameters found in a real wind farm SCADA data set (chapter 3)

- Working prototypes of AI based diagnostic techniques and an online learning power performance model applied to real wind farm data supplied by Scottish Power. (chapter 5)

- An investigation into the development of an extensible and flexible framework for carrying out data formatting, analysis and interpretation of wind turbine SCADA data for a collection of turbines in a wind farm of any size with any turbine model. (chapter 6)

- Design of an ontology that can be used between agent modules for the application of wind farm SCADA data analysis. (chapter 6)

- An implementation of a novel working prototype of a flexible and extensible wind farm SCADA data analysis multi-agent platform. (chapter 6)

- Presentation of a case study involving a confirmed gearbox failure detected almost 6 months in advance through use of the developed normal behaviour models. (chapter 5)

- Discussion (with a brief example) of how the framework can be used to improve an operator's decision making through corroborating the output of multiple independent data analysis processes. (chapter 5)

6.4 References

Foundation for Intelligent Physical Agents, (FIPA IP Spec), “FIPA Interaction Protocol Specifications”. Collection of Specifications Available from the following address: http://209.85.229.132/search?q=cache:http://fipa.org/repository/ips.html

Noy N. F., McGuiness D. L., “Ontology Development 101: A Guide to Creating Your First Ontology”, Tutorial written at Stanford University, Stanford CA, 94305, Available at: http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

Smith G., (2001), “Development of A Generic Wind Farm SCADA System”, DTI Published Report, Accessed September 2009, Available at: http://www.berr.gov.uk/files/file17854.pdf

7. Discussion, Conclusions and Further Work

Condition Monitoring systems installed on wind turbines have the potential to save a utility company money through the reduction in frequency of unplanned outages. The installation of these CM systems alone is not however sufficient to achieve this and work is required to maximise their potential. Automating the collation and formatting of the data as well as carrying out the difficult task of extracting meaningful information from the raw data streams recorded are all necessary in order to prioritise important information to the operator.

It has been established that the Multi-Agent System developed for the CM of wind farms in this thesis can provide the necessary framework to automate the SCADA data analysis procedure. The normal behaviour models developed can also provide the necessary fault detection mechanism to identify faults within the raw SCADA data. The encapsulation of the models within this automated framework therefore satisfies the objective of relieving the operator from the tedious task of manually analysing and interpreting arduous volumes of data. This was achieved through two main areas of research.

The first area considered for this thesis was the development of the mechanisms used to allow for the interpretation of the raw SCADA data streams generated by the integrated CM systems. Taking into consideration the unlabelled data sets provided for the project a number of well established and applicable techniques were explored in the area of fault detection and diagnosis for the purpose of developing normal behaviour models. Three neural network fault detection models were proposed for the gearbox oil, gearbox bearing and the generator winding temperature parameters as well as a statistical based power model. The development process of all four models each involved data preparation, training and testing phases which have all been described. Both generic and turbine specific models were trained, tested and compared for each of the three temperature based fault detection models. Each time, the accuracy of the generic model was found to outperform its specific model counterpart. A confirmed gearbox fault detected six months in advance by the models presented in this thesis was compared to the most similar model developed in the research literature. The performance of the models developed in this thesis provided a much improved point of detection in advance of the failure over the two day window of the model found in the literature. This earlier point of detection benefits the wind farm operator in that it offers a more practical window of opportunity to establish the most appropriate maintenance decision leading to an overall improved O&M of the wind farm.

Once the fault detection models were developed, the potential for corroborating the output of the models was investigated to determine the benefit of combining the extracted information from each model. Two potential combinations were defined; the output of the two gearbox based models, and, the generator winding model combined with a statistical based power model (replicated from the literature for the purposes of corroboration and providing an overall indication of turbine performance). The results obtained from this investigation showed potential but a limited number of confirmed fault instances in the raw SCADA data provided prohibited a more elaborate and detailed study.

The second area considered for this thesis was the automation of the data interpretation and fault detection process. The Multi-Agent framework was selected to achieve this task for the wind farm CM application. The power of this framework is innate within its ability for flexible communication between independent processing modules (agents) which enables them to form alliances automatically to solve the desired problem. The system designer is also given the flexibility to define the ontology which controls the concepts and objects that agents within the system communicate about. The system and the ontology developed for this thesis utilised these benefits to provide a flexible, extensible and easily distributable framework which can readily incorporate future analysis techniques without requiring modifications to the existing system. The ontology was developed with a generic wind farm monitoring application in mind making the system easily applicable to any wind farm. This was achieved by developing the ontology in a manner which does not constrain future additions for new types of communication which may become necessary between the agents.

The range of agents developed and their ease of distribution on interconnected computing resources provide the necessary processing platform for wind farm SCADA data required to considerably reduce the flow of the raw data as it propagates through the processing layers of the system. The outcome of this process allows the system to provide only the information of interest to the operator at a desired remote location. The advantages of such a system if implemented in a live operating wind farm has the potential to dramatically limit the volume of data which must be transmitted between the onsite wind farm data servers and the more convenient offsite locations which can prove to be beneficial especially for difficult to reach offshore wind farm locations. This would lead to the saving of both time and cost by eliminating the need to implement high capacity communication network infrastructures between the onsite and offsite locations. In conclusion the novel contributions delivered by the research presented in this thesis are:

- An investigation of AI based diagnostic techniques suitable for achieving successful novelty detection models for a variety of parameters found in a real wind farm SCADA data set

- Working prototypes of AI based diagnostic techniques and an online learning power performance model applied to real wind farm data supplied by Scottish Power.

- An investigation into the development of an extensible and flexible framework for carrying out data formatting, analysis and interpretation of wind turbine SCADA data for a collection of turbines in a wind farm of any size with any turbine model.

- Design of an ontology that can be used between agent modules for the application of wind farm SCADA data analysis.

- An implementation of a novel working prototype of a flexible and extensible wind farm SCADA data analysis multi-agent platform.

- Presentation of a case study involving a confirmed gearbox failure detected almost 6 months in advance through use of the developed normal behaviour models.

- Discussion (with a brief example) of how the framework can be used to improve an operator's decision making through corroborating the output of multiple independent data analysis processes.

7.1 Further Work

The overall system developed is of archival value to future researchers working in this field as the platform developed can be easily extended and built upon. While the system has satisfied the objectives which were set out to be achieved in this thesis, it is recognised that there are a number of areas with room for improvement which can be investigated to attain a more robust system. From adding more analysis and interpretation techniques, more data sources containing more recorded parameters and the development of a user interface. These specific areas of future work are investigated and outlined below with descriptions of the benefits they would pose for the system.

7.1.2 More Data and More Parameters Containing Interesting Events

For the research carried out in this thesis only basic SCADA data parameters were available. While the author was fortunate to be supplied with data which contained some interesting events (faults), the frequency of these events was limited. This prohibited extensive testing to obtain an insight into the reliability and accuracy of each fault detection module. By gaining access to more SCADA data along with fault records from both different wind farms which monitor different turbine models, this would allow for a more thorough testing phase to gain an insight into how effective the techniques used for fault detection actually are.

Using data from different wind farm sites would serve the purpose of determining whether the techniques can adjust equally as well to varying environmental conditions and still provide good performance in terms of early fault identification. Also data from different turbine models would serve the purpose of determining how effective the NN models are at capturing different models of normal behaviour.

The use of fault records would prove to be very useful as they could be used to identify if the models accurately detect the correct failures. This would allow an empirical study to be undertaken to determine the percentage of successful faults detected within a sufficient time window thereby determining how robust the methodology is to detecting failures in the various components in their early stages.

7.1.3 The Addition of New Interpretation Agents

One of the aims of gaining access to more data sources is to allow further testing and refinement of the models developed in this thesis. Another aim however is include new data analysis techniques which can classify defects as further research is carried out and incorporate them into new Interpretation Agents. In this way the overall diagnostic result of the system can be improved. The use of fault records for example can also be used to train up classifier techniques reviewed in chapter 3 on diagnosing particular types of defects that occur within the components rather than just identifying the faulty component. This would result in a system which provides more detailed information to the operator. By gaining access to different parameters such as vibration data for example new interpretation techniques can be used to extract meaningful information regarding the health of particular aspects of the turbine for example the overall drive train or even rotor blade imbalances through structural tower oscillations. Therefore by combining new data sources and investigating new data interpretation methods the overall diagnostic capability of the system can be increased. The framework of the system developed in this thesis welcomes the addition of new techniques without requiring any modifications to the existing system.

7.1.4 The Inclusion of a System User Interface

Work is already under way to develop a User Interface for the system. The purpose of a User Interface Agent as has been mentioned in chapter 6 is to provide a clear and concise summary of the information provided by the system through one point of contact for the user. All of the resulting information from the data processing carried out in the data-preparation and data interpretation layers of the system would be depicted by this User Interface. A modification to the Alarm concept can be the inclusion of all of the recorded consequent data instances which are classed as anomalous which trigger one alarm. By doing this a visual comparison can also be graphed on the user interface allowing the user of the system to visually determine the severity of the anomaly. As highlighted in chapter 6 this is an addition to the alarm ontology concept, therefore it can be considered a straight forward upgrade which can be made to the ontology.

Another aspect where a User Interface would prove advantageous is the fact that it can be launched from a remote location where the remainder of the system's processing agents can be located on the onsite data server. The user interface would subscribe to the processing agent's output and therefore receive all of the interesting information from the convenience of a remote location. This would prove extremely useful particularly for offshore wind farm locations since the overall volume of data needing transmitted between onsite and offsite locations would be greatly decreased. In this way the communications network infrastructure installed in wind farms can be built appropriately to suit these lower data transfer rates at a reduced cost.

7.1.5 Testing the System in an Industrial Environment

The results presented by the models developed in this thesis are based entirely on data recorded from a live operating wind farm. Therefore they accurately replicate how the system would operate in a real industrial environment. However the purpose of deploying and testing the system in a real live environment would be to gain an understanding of other user requirements which may only become apparent once the system is running in the proposed environment it was designed for. Other important user features such as being able to request access to specific portions of raw data before the anomaly was detected for example might prove useful from an operator's perspective. The best way to determine such features is through further consultation with the projects industrial partners and the personnel who would typically be using the system once it is in place. By carrying out such a practice, wind farm operators would gain confidence in using the system, therefore helping to transition utility companies away from their typical scheduled maintenance policies to employ a more condition based maintenance approach. The natural evolution of such a process however would be to gain access to more SCADA data from live operating wind farms to further test the system and the models as detailed in the earlier sections of this chapter to ready the system for industrial deployment.

Appendix I

	Aug 05	Sep 05	Oct 05	Nov 05	Dec 05
T8	Good: 3719 Islanding/scheduled: 195 Bad: 382 Unscheduled: 12 Performance figure: 90.8%	Good: 3430 Islanding/scheduled: 115 Bad: 559 Unscheduled: 204 Performance figure: 82.3%	Good: 3688 Islanding/scheduled: 23 Bad: 342 Unscheduled: 0 Performance figure: 91.5%	Good: 3885 Islanding/scheduled: 130 Bad: 259 Unscheduled: 164 Performance figure: 90.2%	Good: 4102 Islanding/scheduled: 14 Bad: 210 Unscheduled: 5 Performance figure: 95%
T15	Good: 3760 Islanding/scheduled: 195 Bad: 545 Unscheduled: 3 Performance figure: 87.3%	Good: 3700 Islanding/scheduled: 53 Bad: 607 Unscheduled: 1 Performance figure: 85.9%	Good: 3701 Islanding/scheduled: 50 Bad: 327 Unscheduled: 2 Performance figure: 91.84%	Good: 4061 Islanding/scheduled: 319 Bad: 242 Unscheduled: 5 Performance figure: 94.3%	Good: 4188 Islanding/scheduled: 245 Bad: 128 Unscheduled: 1 Performance figure: 97%
T16	Good: 3767 Islanding/scheduled: 51 Bad: 508 Unscheduled: 33 Performance figure: 87.4%	Good: 3721 Islanding/scheduled: 0 Bad: 593 Unscheduled: 3 Performance figure: 86.2%	Good: 3619 Islanding/scheduled: 2 Bad: 410 Unscheduled: 1 Performance figure: 89.8%	Good: 3896 Islanding/scheduled: 33 Bad: 389 Unscheduled: 23 Performance figure: 90.4%	Good: 3792 Islanding/scheduled: 0 Bad: 448 Unscheduled: 11 Performance figure: 87.8%
	Jan 06	Feb 06	Mar 06	April 06	May 06
T8	Good: 3761 Islanding/scheduled: 149 Bad: 218 Unscheduled: 338 Performance figure: 87.12%	Good: 3768 Islanding/scheduled: 298 Bad: 206 Unscheduled: 46 Performance figure: 93.73%	Good: 3993 Islanding/scheduled: 364 Bad: 292 Unscheduled: 23 Performance figure: 92.7%	Good: 3957 Islanding/scheduled: 444 Bad: 219 Unscheduled: 132 Performance figure: 91.9%	Good: 3089 Islanding/scheduled: 203 Bad: 218 Unscheduled: 1001 Performance figure: 71.7%
T15	Good: 4185 Islanding/scheduled: 624 Bad: 129 Unscheduled: 3 Performance figure: 96.9%	Good: 3799 Islanding/scheduled: 445 Bad: 221 Unscheduled: 0 Performance figure: 94.5%	Good: 4044 Islanding/scheduled: 444 Bad: 258 Unscheduled: 6 Performance figure: 93.9%	Good: 4034 Islanding/scheduled: 916 Bad: 245 Unscheduled: 29 Performance figure: 93.6%	Good: 3825 Islanding/scheduled: 1287 Bad: 482 Unscheduled: 1 Performance figure: 88.8%
T16	Good: 2432 Islanding/scheduled: 166 Bad: 121 Unscheduled: 1764 Performance figure: 56.3%	Offline for entire duration of month	Good: 3063 Islanding/scheduled: 275 Bad: 402 Unscheduled: 843 Performance figure: 71.1%	Good: 3897 Islanding/scheduled: 270 Bad: 408 Unscheduled: 3 Performance figure: 90.5%	Good: 3632 Islanding/scheduled: 93 Bad: 672 Unscheduled: 4 Performance figure: 84.3%

	June 06	July 06	Aug 06	Sep 06	Oct 06
T8	Good: 3071 Islanding/scheduled: 225 Bad: 187 Unscheduled: 1050 Performance figure: 71.3%	Good: 3944 Islanding/scheduled: 2188 Bad: 231 Unscheduled: 133 Performance figure: 91.55%	Good: 3858 Islanding/scheduled: 758 Bad: 363 Unscheduled: 87 Performance figure: 89.6%	Good: 2795 Islanding/scheduled: 103 Bad: 383 Unscheduled: 1130 Performance figure: 64.9%	Good: 3492 Islanding/scheduled: 221 Bad: 324 Unscheduled: 492 Performance figure: 81.1%
T15	Good: 4015 Islanding/scheduled: 1789 Bad: 290 Unscheduled: 3 Performance figure: 93.2%	Good: 3721 Islanding/scheduled: 516 Bad: 586 Unscheduled: 1 Performance figure: 86.4%	Good: 3255 Islanding/scheduled: 147 Bad: 634 Unscheduled: 419 Performance figure: 75.6%	Good: 3227 Islanding/scheduled: 103 Bad: 1050 Unscheduled: 31 Performance figure: 74.9%	Good: 3489 Islanding/scheduled: 222 Bad: 754 Unscheduled: 65 Performance figure: 81%
T16	Good: 3408 Islanding/scheduled: 142 Bad: 892 Unscheduled: 8 Performance figure: 79.1%	Good: 3433 Islanding/scheduled: 310 Bad: 851 Unscheduled: 24 Performance figure: 79.7%	Good: 3186 Islanding/scheduled: 0 Bad: 1023 Unscheduled: 25 Performance figure: 75.7%	Good: 2956 Islanding/scheduled: 13 Bad: 1234 Unscheduled: 118 Performance figure: 68.6%	Good: 3511 Islanding/scheduled: 142 Bad: 762 Unscheduled: 35 Performance figure: 81.5%

	Nov 06	Dec 06
T8	Good: 2495 Islanding/scheduled: 66 Bad: 368 Unscheduled: 1445 Performance figure: 57.9%	Offline for entire duration of month
T15	Good: 3284 Islanding/scheduled: 19 Bad: 829 Unscheduled: 195 Performance figure: 76.2%	Good: 3363 Islanding/scheduled: 50 Bad: 740 Unscheduled: 740 Performance figure: 78.1%
T16	Good: 3214 Islanding/scheduled: 47 Bad: 513 Unscheduled: 581 Performance figure: 85.5%	Good: 3685 Islanding/scheduled: 40 Bad: 411 Unscheduled: 212 Performance figure: 85.5%