Extracting information from manufacturing data using data mining methods

Giess, Matthew

Award date:
2006

Awarding institution:
University of Bath

Extracting Information from Manufacturing Data using Data Mining Methods

Matthew Giess

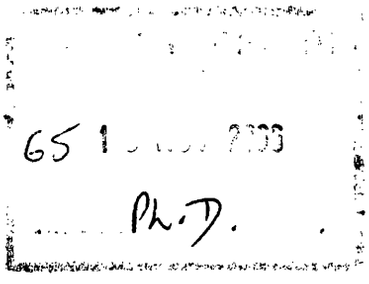
A thesis submitted for the degree of Doctor of Philosophy

University of Bath

Faculty of Engineering and Design

2006

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346


651
Ph.D.

Abstract

The manufacture and test of a product typically results in the generation of quantities of data describing both the characteristics of a product and its performance when tested. Such data contains information of potential use within design, and hence this research seeks to provide a means of both of extracting information from such data and making this information available for the design engineer.

Various methods of data analysis exist and have been utilised for the analysis of manufacturing data. However, many such methods rely upon generating data specifically for analysis, which has cost implications, or simply quantify a previously defined relationship. Others, such as Taguchi's Robust Design, require the ability to control the values of variables, limiting its application when the variance of characteristics within manufacturing tolerance is under investigation. The field of Data Mining (DM) allows for data to be analysed and modelled without the need to specify expected relationships, and without the need to control the value of variables.

In a number of DM implementations within engineering it was noted that little attention was given to the methods by which data were generated, and also how useful information might be extracted from the resultant DM models and subsequently validated. This thesis seeks to address both of these areas.

Information Extraction and Validation

An analytical model was used to generated data mimicking that typically collated during a manufacturing process, and DM modelling methods were used to analyse these data. Various methods of information extraction were used, of which some were novel, and the extracted information was compared against industrially-validated information obtained from the analytical model itself. Encouraging results were obtained.

Data Generation within Manufacturing

Two industrial case studies were used to assist in understanding the nature of manufacturing data. Methods of retrospectively removing erroneous data were tested. Numerous data generation issues were noted, and these examples were used to create a three-tier hierarchy which guides consideration of data generation practices.

Acknowledgements

First and foremost I would like to thank my supervisor Steve Culley, who set this research programme in motion and has offered timely guidance and support throughout its duration.

I would also like to express my gratitude to our industrial collaborators. In particular, I would like to acknowledge the extended and essential contributions of Andrew Shepherd, Rick Glogiewicz and James Cunningham.

I was fortunate to receive significant funding for the duration of my research, for which I would like to acknowledge the contributions made by both the Engineering and Physical Sciences Research Council and our industrial collaborator.

I would also like to thank my colleagues at the University of Bath, both within the Innovative Manufacturing Research Centre and elsewhere, who have provided much appreciated advice, much needed encouragement and much valued friendship throughout my time working with them.

Last, but certainly not least, I would like to acknowledge the support given by both my parents and my partner Fiona. I am aware that the sacrifices I chose to make in seeing this work to competition were shared in great part by them, and I offer my heartfelt thanks for their continual patience and support.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

The issue of communication and transfer of knowledge and expertise has been of interest to designers for many years. It was revealed quite clearly in the development of concurrent engineering, where companies moved from the sequential practice of product development to the more integrated approach commonly found today. The scale of the task is considerable, given the complexity of engineering design. Design has many dimensions, emerging from the understanding of need, the creation of concepts, selection and development of the most suitable concept, through to the detailed specification of the artefact, down to a resolution of defining such elements as fillet radius and surface finish.

Even a simple engineering drawing has a wealth of information elements contained within it. A number of these elements are procedural, such as title, date, issue number and so on, whereas a number will have been entered by the designer indicating the final description of the part. It is not clear, from the drawing in isolation, where values for this description have come from, most significantly in terms of why a particular surface finish was specified or why a tolerance has a certain value. Schulte and Weber (Schulte and Weber, 1993) presented research suggesting that there are 3 methods of specifying the shape or dimension of an artefact, where the value for the dimension is either dictated by analysis, by consideration of appropriate standards or ease of manufacture, or finally by specifying a 'filler' value in the absence of any further information. Whilst the research focused upon dimensions, the rationale also holds for the next level of detail, the tolerances. Formal methods of specifying tolerances include the use of standards (for example BS 1916-1, BSI, 1953), company guidelines (Lowe, 2002) or guidelines published in texts (for example, that given by Groover, 1996). Also exerting an influence alongside these methods are 'filler' factors such as custom and practice, and the simple recycling of values used in similar areas of previous successful designs. Clearly such means of tolerance specification are informed by feedback from manufacture and testing into design, however this area is under-researched and is typified by informal relationships or by relationships with limited structure, where the only firm structure comes in the form of design meetings or formal reviews.

This research aims to investigate and support informal feedback of information from manufacturing into design. The measurement and testing of a product during manufacturing can provide good information as the tests are being carried out upon the actual product, manufactured exactly as the designers intended. The feedback of information from these areas depends upon both an actual interpretation of the tests in the manufacturing process, and upon the form of communication between manufacturer and designer. It is suggested that designers and manufacturers have different agendas, the designer is looking to meet the requirements of or improve the performance of a product via improvements to the underlying structure of a product, whereas manufacturers are seeking to ensure that the product conforms to the manufacturing requirements.

As an alternative to relying upon subjective feedback from manufacturing engineers, it is argued that the data that is generated during a typical manufacturing process, which are recorded during measurement and test procedures, contain useful information that could potentially be of benefit to the designer. It is therefore considered that some form of analysis of manufacturing data could yield useful information. Thus in this research methods of extracting such information from manufacturing data will be considered and a number of approaches will be developed. As a necessary part of this process, an investigation into the nature of manufacturing data is carried out. Such data is the bedrock of the entire analysis, and an appreciation of the methods by which such data is generated and recorded is necessary to successfully implement and evaluate the results of an analysis.

This introduction reflects upon 3 aspects that are critical to this research. Key to this research is an understanding of the nature of manufacturing data, particularly in terms of its suitability for later analysis. Following this it is necessary to understand which methods of data analysis are available, and identify which of these methods is most suitable in this specific application. Finally, it is necessary to consider where within the engineering design process the outcomes of research can be usefully deployed.

These three issues will be briefly introduced, at which point the key research questions will be discussed and the structure of the remainder of this thesis will be stated.

1.1 Data within Engineering

A key requirement for any methodology or guideline developed during the course of this research is that it be generic, with the capacity for easy transfer between enterprises. To achieve this requirement it is necessary to identify the types of data that may reasonably be recorded during a manufacturing and assembly process. The applicability of any analytical method depends not only upon the nature of the required solution, but also upon the form and type of data under analysis. In this respect it should be noted that this research differs from the popular Taguchi and other Design of Experiment (DoE) approaches in that it focuses on historically recorded data, or *ex post facto* data as labelled by Hicks and Turner jr (1999), with the result that there is no attempt to implement engineering operations specifically to generate data for the purpose of analysis.

Many companies subscribe to the ISO 9000 and ISO 9001 series of quality assurance standards (BS EN ISO 9001, BSI, 2000), for which the emphasis is upon monitoring the process via measurement of specific products, rather than monitoring of the specific products themselves. The author is unaware of any other legislation, mandatory or voluntary, which specifies the nature of data measurement and recording within a company. Within the ISO 9000/9001 series, there is little detail given regarding the precise nature of the data that should be recorded, or even how it should be analysed. A review of literature describing research into the analysis of data that could be construed as manufacturing data indicates that there is little attention given to generic data issues, instead the focus of the literature tends to be upon evolving methods of analysis that might be of use in the domain of interest. In this respect, such analyses tend to simply treat each problem on a case-by-case basis, with little consideration of the nature of the data or of what could be considered as typical of data that might be generated within a specific domain.

It is stated at this early stage that the success of this research in finding useful information within manufacturing data will be defined to a great extent by the nature of manufacturing data. Relevant and accurate information can only be extracted from data that accurately depicts the domain or phenomenon of interest. In the absence of any clear understanding of the nature of manufacturing data, it is argued that a significant aspect of this research should aim to provide a description of manufacturing data, and the issues within manufacturing data that would affect accurate analysis.

1.2 Methods of Data Analysis

This research utilises Data Mining (DM) for purposes of data analysis, however it is perhaps necessary to introduce other available methods of analysis, and discuss their associated merits and demerits, in order to provide some foundation to the selection of DM. Greater detail may be found in Chapter 11.

Certain methods of data analysis, such as Taguchi's Robust Design (Taguchi, 1986) require the ability to control various parameters in order to trace the effects of deliberate perturbation of their values. As this research will focus upon analysing data describing the variation of parameters within tolerances such control cannot be guaranteed. More traditional Statistical methods are perhaps most useful in quantifying expected relationships, and hence require some prior understanding of the pattern and structure of data. Statistical methods are therefore unsuited to cases where such prior knowledge is lacking, as may feasibly be the case in this research. Database-oriented methods, such as On-Line Analytical Programming (OLAP), allow users to visualise and manipulate data such that relationships and patterns can be uncovered. This is essentially a tool for assisting in manual analysis of data, and as such requires notable experience and skill on behalf of the operator. DM allows for the utilisation of a variety Machine Learning algorithms in order to seek and identify unsuspected patterns within data. There is no requirement for either prior knowledge and the ability to control the values of certain parameters. DM is therefore considered as the method of analysis most suited to this research.

1.3 Integration of Research within Engineering Design

This research will investigate data generated during manufacturing, but the ambition is to provide information that would be of use to a designer in terms of product behaviour and how aspects of the product's character are related to this behaviour. In order to understand where this information might be employed within the design process, it is necessary to first have an appreciation of the nature of design.

1.3.1 Overview of Engineering Design Process

The term design means many things, each dependant upon context. Used in the context of an engineering process, design may be considered to be the act of prescribing an artefact's physical properties in a manner which will meet a specified goal. This goal is

typically described in terms of what is required of the artefact in question, in terms of the physical functionality that must be provided and the constraints it must meet.

The design process should not be considered as a rigid, algorithmic method of generating a design that will meet the specified goal. It is more a methodology that details the tasks that should be performed and in what manner they should be performed if a suitable design solution is to be reached. In this respect the design process may be considered as a framework which provides the structure necessary to enable a designers intuition to be focused to best effect (Potter, 2000) which acts to harness and direct human creativity (Suh, 1990).

It is suggested that many of the proposed design methodologies have been created in response to particular design scenarios, and although each purports to be generic this variance in background has led to differences in the actual specification of each methodology. In spite of these differences each methodology agrees in general that design proceeds in a stage-wise manner, starting with a consideration of the problem, a search for potential solutions, identification of the most applicable and development of this identified solution through to manufacture (McMahon and Browne, 1998)

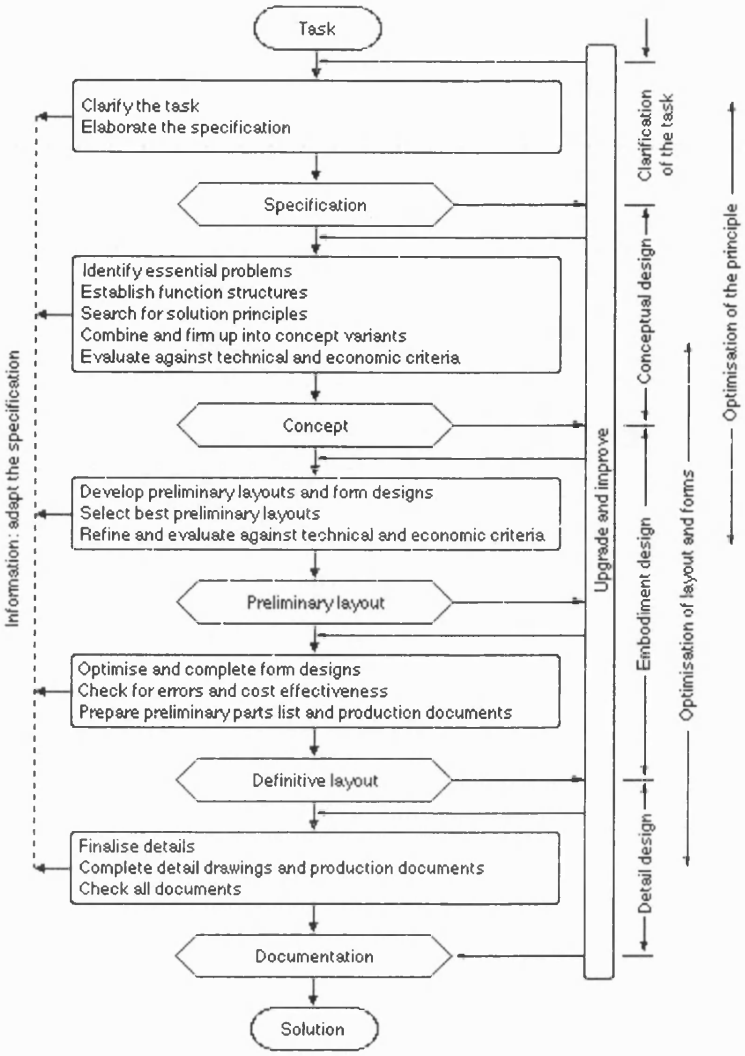


Figure 1 Pahl and Beitz Design Methodology (adapted from Pahl and Beitz, 1996)

The methodology defined by Pahl and Beitz (1996), as shown in Figure 1, is widely referenced in design literature, and introduces the key aspects of design within a systematic framework. There are 4 key stages in this methodology, which are as follows:

1. *Planning and clarifying the task.* This stage involves deducing the requirements and constraints of the artefact, usually from a vaguely worded design brief obtained from a client, and producing a statement (the design specification) which defines these requirements and constraints within terms of reference which the designer has experience of.

2. *Conceptual Design.* The design specification is used to drive the development of a series of working solutions.

3. *Embodiment Design.* The most suitable conceptual design is further developed, and aspects of the design are elaborated upon allowing for evaluation of the design to be undertaken.

4. *Detail Design.* A specific definition of all of the characteristics are developed and stated, including materials, dimensions etc. A more detailed evaluation is undertaken, where both artefact performance and cost are computed. If acceptable, the design can proceed to manufacture, in which case detailed drawings and documentation can be created.

The design process described above could feasibly give the false impression that each of the stages of design are independent and are strictly sequential. As highlighted earlier, concurrent engineering dictates that all factors influencing an artefact, whether they be within design or manufacturing, are considered in parallel. This would suggest that there is some overlap between stages, as aspects of each are carried out simultaneously. The approach proposed by Ohsuga (1989) also specifies stages in which the detail of the design increases, but at each stage there is emphasis upon creating a model which can be evaluated to ensure that all requirements, manufacturing included, are considered and met at each stage. This allows for concurrency, where issues in both design and manufacturing can be addressed at each stage of design and, in the event of the design not meeting requirements, the design can be refined in an iterative loop.

1.3.2 Consideration of Implementation within Design

It is noted that data describing a manufactured artefact, and more importantly the information extracted from that data, are only valid when describing or covering the artefact in question. Such information cannot automatically be transferable to other similar artefacts within the product family, for example information regarding a specific type of motor cannot be assumed to be valid for a different motor, even if they share many similar characteristics or design features. In this respect, any information gained from data analysis will not be of any use in conceptual design but will be of use within detail design. The information extracted from manufacturing data can only be applied to the artefact as it stands, and cannot be applied across all possible conceptual designs. It is suggested that information regarding a design would not appear to be of great use once the design has been completed and manufacturing initiated, however, aside from the idea that products and manufacturing processes are subject to continual improvement, much

design seeks to improve upon other, earlier designs or reuse successful aspects of earlier designs. Pahl and Beitz (1996) refer to three styles or forms of design, *original design*, *adaptive design* and *variant design*. Original design refers to the practice of developing a design with no antecedents, where the means of meeting the specifications are entirely novel. Adaptive and variant design have similarities in that aspects of previous successful design solutions are reused, in adaptive design the principles of previous designs are reused in a different manner, forming fundamentally different designs, whereas in variant design the emphasis is more upon adapting the parameters of existing previous successful designs to produce a new variant. Finger, in her keynote lecture at a conference focusing upon design reuse, suggests that successful design reuse must go further than simply using previous designs for inspiration, and must also consider the rationale involved in design, thus seeking to allow the reuse of logic and knowledge obtained in previous design activities (Finger, 1998).

These areas are the main focus of the research reported in this thesis, where the ambition is to enable information regarding and knowledge of a product to be extracted from manufacturing data in order to feed this design reuse, whether that be in adaptive or variant design. It is argued that the extracted knowledge would be of greater use in variant design, as it would indicate characteristics of similar products that the designer could either seek to replicate or eliminate from subsequent designs, whereas reuse within adaptive design would involve the complication of deducing whether the extracted information is domain-specific or product-specific, in effect deducing the range or coverage of the information. In either case, the extraction of information from manufacturing data can usefully benefit designers within the detailed design stages of variant design and, arguably to a lesser extent, adaptive design.

1.4 Aims, Objectives and Outcomes of Research

There are two separate aspects to this research. The first aspect involves identifying how manufacturing data might be modelled using DM methods, and how information from such methods might be extracted in order to assist designers. The second aspect involves investigating how manufacturing data is generated, focusing upon how accurately such data represents the manufacturing process.

There are 4 key objectives:

1. To propose and evaluate a method of analysing manufacturing data using DM methods.
2. To propose and evaluate a method of extracting information from DM models.
3. To identify how manufacturing data can be retrospectively analysed in order to reduce error within the data and increase their suitability for DM analysis.
4. To investigate the processes of manufacturing data generation such that the suitability for DM analysis of the generated data may be evaluated.

Each of these objectives will be addressed in separate chapters.

The outcomes of the research are threefold:

1. An approach to modelling manufacturing data has been demonstrated, and methods to extract information from models created by both the Artificial Neural Network (ANN) and Decision Tree Induction (DTI) DM algorithm has been tested against an industrially-validated computational case study with encouraging results. The method of information extraction for the DTI algorithm is novel to this research, as is the validation against industrially proven information. Further to this, novel methods of combining information extracted from multiple DM models have been proposed (objectives 1 and 2).
2. A dataset describing a soap powder packaging line was used to test the effectiveness of a range of data cleansing approaches (objective 3).
3. A detailed case study identified issues unduly influencing the accuracy of manufacturing data from a DM perspective, from which a three-tier hierarchy was evolved to provide guidance in both the specification and evaluation of data generation processes within manufacturing (objective 4).

1.5 Structure of Research

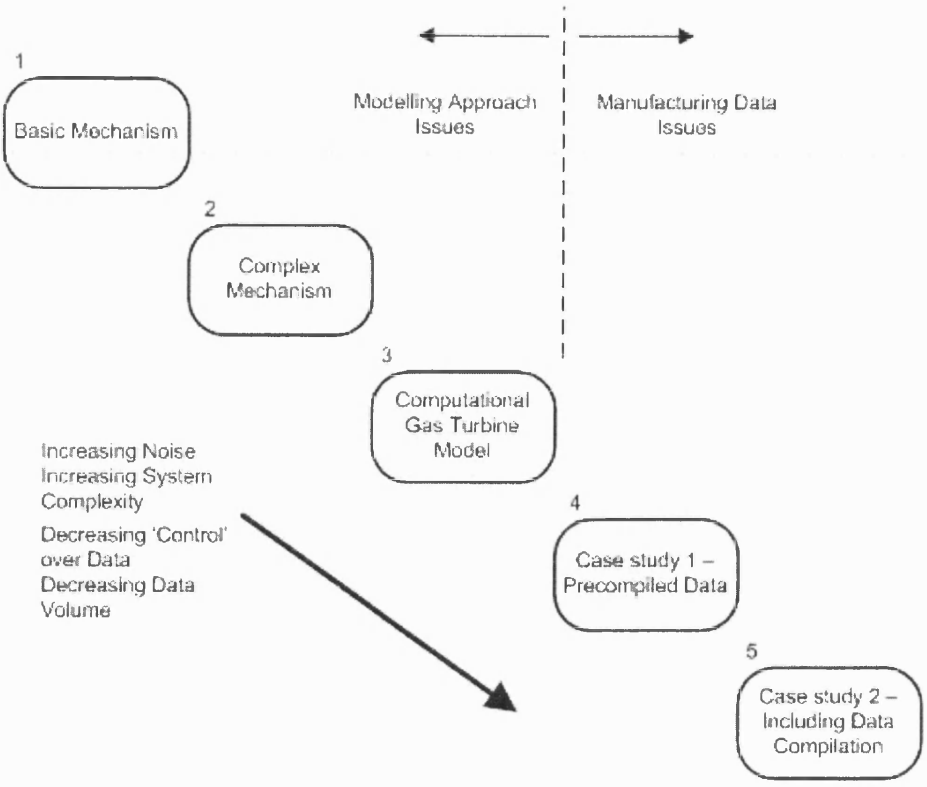


Figure 2 Stages of Research

A map of the 5 stages of research, as initially planned, can be seen in Figure 2, where the progressions from simple, computationally described problems to complex, real-world problems are shown. The third stage, the use of a computational representation of a gas turbine manufacturing process, was initially planned as it would form a useful lead into the 5th stage, a case study focusing upon the manufacture of such gas turbines. This third stage could not be successfully completed as important engineering data was not available, however it remains an interesting avenue for further work.

The different stages address different objectives. The analysis of the basic mechanism is used to evaluate a method of modelling manufacturing data using DM methods. This modelling approach is then applied to a more complex mechanism, and from this methods of extracting information are developed. A comparative analytical analysis of the complex mechanism provides an indication of the accuracy of the extracted information.

The two case studies are used to obtain some insight into the nature of manufacturing data. The first case study considers a precompiled dataset obtained from a soap powder packaging line. It is noted that errors exist in this data, and this chapter focuses upon evaluating methods used to retrospectively reduce these errors. The second case study

goes further, and considers how manufacturing data is generated. A number of issues influencing the accuracy of the generated data are noted. These are used to evolve a generic framework which may be used to specify or evaluate methods of manufacturing data generation.

1.6 Structure of Thesis

This chapter has introduced the research and identified its motivation and scope and how it relates to engineering design. A brief overview of the structure of the remainder of this thesis is given below:

Chapter 2 comprises a review of literature on contemporary Data Mining, where a review of the state of the art of modelling methods will be given.

Chapter 3 reviews literature describing where such methods have been usefully employed to analyse manufacturing data. There is a gap analysis that seeks to identify issues that have yet to be addressed, and the key focus of the research is described.

Chapter 4 identifies the modelling approach to be used throughout this research. This chapter seeks to define a method of analysing manufacturing data using the modelling methods described in chapter 2.

Chapter 5 builds upon the work of chapter 4 to identify how information may be extracted from generated DM models. A comparative study between DM and analytical modelling allows for the accuracy of information extracted from the DM models to be evaluated.

Chapter 6 utilises the logic of existing methods of combining the *predictions* of DM models to develop a method of aggregating the *extracted information* from DM models.

Chapter 7 seeks to identify methods of retrospectively reducing error within manufacturing data. Data obtained from a case study investigating changeover duration of a soap powder packaging line is investigated.

Chapter 8 considers the methods of data generation used within manufacturing. Methods employed at a gas turbine manufacturing facility are critically examined, and the resulting observations are arranged into a framework that is intended to guide the specification and evaluation of methods of data generation

Chapter 9 concludes this research and offers some recommendations for further work

Chapter 2 Data Mining

This chapter introduces the topic of Data Mining (DM), and describes its applicability to the problem in hand. Key to this introduction is the idea that DM is a methodology, and one which there is continual effort to standardise. The tools used for the data analytical aspects of DM are the Machine Learning algorithms, and these will be briefly discussed in following sections. Methods used to combine the outputs of multiple Machine Learning models will also be reviewed

2.1 Principles of DM

The term Data Mining may be considered synonymous with Knowledge Discovery in Databases (KDD), a situation which is complicated by many practitioners considering DM to be the analytical, 'number-crunching' stage of KDD. Along with this slightly European/Continental American difference in terminology, it has been casually observed that in cases where the process has been applied to a particular application it is generally referred to as DM, whereas those who are interested in the theoretical development of the process tend to use the title KDD. This fine point illustrates two things. Firstly, DM is a process with numerous stages of which the numerical analysis is but one part. Secondly, the idea that those interested in practical applications may confuse or oversimplify the process of DM or KDD to only consider the analytical aspects, hence the use of the KDD definition of the analytical aspects (Data Mining) as a title for the entire process. However we choose to define the process, it is important to recognise that DM or KDD is essentially a methodology as opposed to a modelling tool and the analysis forms only one part of the entire process.

Although DM is still a relatively emergent field, it has been successfully used in many domains. Examples of successful DM implementations include analysis of customer energy consumption (Sforna, 2000), investigating civil infrastructure (Buchheit et al., 2000), deducing the most receptive audience for marketing (Forcht and Cochran, 1999), and even includes such unique applications as allowing basketball coaches to analyse opposing teams to deuce strengths and weaknesses (Baltazar, 2000). This widespread

acceptance of DM across divergent fields, allied to its application in areas that have received little analytical attention (such as assisting basketball coaches), suggest that DM is a valid method of investigating data in areas where other, more traditional techniques would be difficult to implement.

As indicated by the differences in terminology across fields, DM has somewhat disparate beginnings. This, combined with its immaturity, have resulted in the formalisation of numerous different methodologies (Lee and Siau, 2001). The majority of these methodologies follow similar lines, containing stages to define problems, collate and analyse data and evaluate the results, although their bespoke nature tends to restrict their range of applicability to certain specific disciplines at the expense of other areas. In order to address this, a European Commission funded consortium has developed a methodology which is intended to be as neutral as possible and allow for standardisation across disciplines, and to allow for tool interoperability (Piatetsky-Shapiro, 1999). This methodology, the Cross-Industry Standard for Data Mining (CRISP-DM) identifies 6 stages as follows (CRISP-DM, 2000).

1. *Business Understanding* - Define what the business wishes to achieve, define problem in DM terms and goals
2. *Data Understanding* - Obtain initial sample, identify problem areas, consider suitable test areas.
3. *Data Preparation* - The 'data warehousing' area, compile, sort, clean and format the data for use in package(s)
4. *Modelling* - Several methods may be applicable, investigate and optimise each.
5. *Evaluation* - Consider if models created reach desired goals
6. *Deployment* - Provide end result of most value to user, whether a formal report, model etc.

These 6 stages form an iterative loop, as shown in Figure 3.

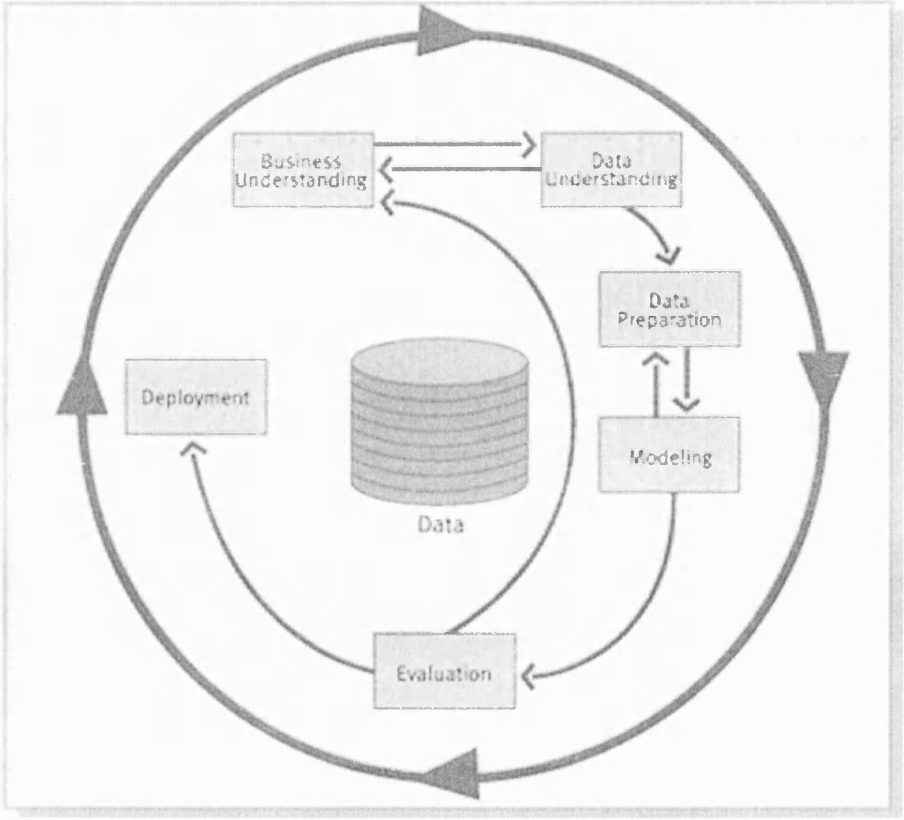


Figure 3 CRISP-DM Methodology (CRISP-DM, 2000)

The first draft of the CRISP-DM model was released in 2000, and at the time of writing this remains the most recent edition. At present it consists primarily of observations and 'tricks of the trade' which have been collated from the experiences of the various consortium members, however as suggested by Clifton and Thuraisingham (2001) it requires further development in order to be considered as an industry standard. In spite of this, it does represent perhaps the most coherent and neutral methodology proposed thus far, and has the support of many in the field, and hence will be used as a framework to steer this research.

2.2 Standardisation of DM

DM has suffered from widely-dispersed development, and that such dispersion has acted to both confuse the terminology and to give rise to numerous methodologies, of which CRISP-DM is simply one of many. The lack of cohesion in the development of DM has also presented problems in the transferability of models created by specific software

packages. This situation has been partly remedied by the construction of a formal arrangement for the description of DM models. This language[1] is called PMML, the Predictive Modelling Mark-up Language (DMG, 2003) and it is an extension of XML that allows DM models to be described and transferred between proprietary software. The XML specification allows for interested parties to create and tailor a language focused upon a domain of interest, and falls outside of the scope of this thesis – the interested reader is directed towards the developers of XML, the World Wide Web Committee, W3C, for more information (Bray et al., 1997). Numerous developers of proprietary DM software have incorporated PMML into their software (Wettschereck et al., 2003) and as such it forms a useful method of model transfer. There remain, however, many developers of software who lack the infrastructure to incorporate this language within their products. A significant section of such software, such as the WEKA program created in Waikato in New Zealand (Witten and Frank, 2004), have been created for purposes of academic research and hence emphasis has been placed upon developing the algorithmic function of the software as opposed to file-handling. It is a measure of the usefulness of the PMML language that a separate filter has been externally developed for the WEKA package to allow models to be exported in PMML format (Li, 2003).

The use of a standard interface language has a further benefit in that models can be created using a proprietary piece of software, with all the attendant licensing costs and hardware considerations, and can then be transferred to remote computers to be viewed on another, simpler piece of software. Such a practice much be considered alongside the need to ensure that the iterative nature of DM is preserved, as problems may be encountered when attempting to further develop the modelling based upon observations whilst viewing or implementing information extracted from the model at a remote site. It is still maintained that the presence of an intermediate language to transfer models between software is a significant benefit of the DM approach.

[1] XML is a language and PMML is, in effect, a vocabulary based upon that language. However, for simplicity PMML will be referred to as a language in its own right.

2.3 Modelling Aspect of DM

The specific implementation of the modelling aspect of a DM application is directed to a large extent by the desired result or focus of the analysis. Grossman *et al* (1999) suggests two types of analysis are typically considered, the first creates a predictor for future use and the second aims to create models that are used simply to provide some description of the domain of interest. In this respect, Grossman *et al* suggest that the second form of analysis has much lower 'burden of proof' in terms of model accuracy, suggesting that more complex models provide better predictive tools whereas information extraction is best achieved using simpler models. This appears to overlook the idea that excessive complexity may in fact cause overfitting, a process of learning both pattern and noise, however the basic premise that there is a tradeoff between accuracy and interpretability is noted. It is further noted that model accuracy forms a prerequisite for any form of analysis, and a lack of accurate evaluation of the accuracy of models created during analysis has been noted and criticised by many researchers, of which Reich and Barai (1999) are an example. This criticism culminated in a direct letter to the editor of a prominent engineering journal (Reich, 1999) which elicited further comment in agreement with this criticism from both a further eminent researcher (Smithers, 2001) and the editors of the journal in question (Tomiyama et al., 2001). It is therefore argued that accurate validation is an essential aspect of any modelling effort, although this is tempered by the suggestion that, in terms of extracting information, model accuracy must be considered alongside other factors such as transparency and ease of interpretation. In cases where the reason *why* a model is making a certain prediction is more important than the value of the prediction it makes, there is little point in having an optimally accurate model if it doesn't meet the requirement of interpretability. In this respect, it is concluded that accurate validation is essential to obtain a clear, representative impression of how well the model characterises the data, and only when this is known can information from these models be used with any degree of confidence.

2.4 Modelling Methods

The idea of the 2 forms of model, the predictive and descriptive, may be further decomposed into 4 modelling methods, which hierarchically add greater levels of predictive resolution. These four main modelling techniques or styles are as follows (Witten and Frank, 2000):

1. *Clustering* - Arranging attributes into groups, such as putting dogs and humans into a group 'mammal' – there is typically some subjective requirement for both forming group boundaries and of designating the label of each group.
2. *Association* - Highlighting links between attributes, such as saying if x is of value a, y is likely to be of value c.
3. *Classification* - Predicting the output class of a new instance from its input attributes, for example predicting if weather will be 'cold', 'warm' or 'hot' based upon information such as wind speed and cloud coverage
4. *Numerical Prediction* - Assigning a numerical value to a new instance based on its input attributes, for example predicting the top speed of an aircraft given characteristics such as wing profile, frontal area and engine thrust.

The 4 methods may be split into 2 camps, where association and clustering fall within the group of *unsupervised* methods and classification and numerical prediction are referred to as *supervised* methods. The simple distinction is that there is no prescribed output for the unsupervised methods, whereas supervised methods attempt to find a pattern that leads to an output prediction for a given input vector. Referring back to the previous notation, supervised methods are the predictive methods and unsupervised are the descriptive. Unsupervised methods are suggested to be of more use in exploratory exercises, where the structure of the data is unclear, whereas supervised techniques are more useful where the structure is known and information regarding the nature of the structure is required. The notion of structure should be clarified, it is not intended to imply that there is any understanding of the nature of interactions between parameters, instead it refers to an understanding of the structure of the process that the data represents. It is suggested that, in an engineering domain, the structure of the data will be well understood as there will be an understanding of where within a manufacturing operation each set of data is generated, but there might not be any understanding whatsoever of the underlying relationships between these data. It is these relationships that supervised methods attempt to deduce, and for this reason predominantly predictive methods will be used in this research.

2.5 Modelling Algorithms – Machine Learning Techniques

The methodology of DM suggests that if data is presented to an appropriate model in an appropriate way, then the model will do the work of 'learning' the structure of the data and elucidate usable knowledge back out. In the work presented in this paper, and adapting the definition of Witten and Frank (2000), learning will be considered as an action which seeks to change behaviour so that performance is improved in future. In the context of supervised learning, previously described as supervised methods of modelling, a series of instances with known outputs can be presented to an algorithm which can then discern a pattern within the data, which can subsequently be used to predict the output of an instance with an unknown output. The modelling stage of DM may therefore be consider as a process of learning the patterns in the data using an algorithm, and when this is performed computationally we may describe such an activity as being a case of Machine Learning.

The methods of modelling have previously been introduced, in terms of both supervised and unsupervised learning, however these are simply methods of specifying what task is required, as opposed to actually specifying how the task should be carried out. Numerous algorithms have been developed which attempt to carry out these tasks in various ways, many of which have been used to good effect.

DM Technique	Characteristics Advantages/Disadvantages	Cluster	Assoc	Class	Num. Pred
Query Tools	• Used for extraction of 'shallow' knowledge • SQL is used to extract Information	Y	Y	N	N
Statistical Techniques	• SPC tools are used to extract deeper knowledge from databases • Limited but can outline basic relations between data	Y	Y	Y	Y
Visualisation	• Used to get rough feeling of the quality of the data being studied	Y	Y	N	N
Online Analytical Programming (OLAP)	• Used for multi-dimensional problems • Cannot acquire new knowledge	Y	Y	N	N
Case-based Learning	• Uses k-nearest neighbour approach • A search technique rather than a learning algorithm • Better suited for small problems	N	N	Y	Y
Decision Trees	• Good for most problems • Gives true insight into nature of the decision process • Hard to create trees from a complex problem	N	N	Y	N[2]
Neural Networks	• Mimics human brain • Algorithm has to be trained during coding phase • Complex methodology	Y	N	Y	Y
Genetic Algorithms	• Uses Darwin's evolution theory • Robust and reliable method for data mining • Requires a lot of computing power to achieve anything of significance	N	N	N	N

Table 1 Data Mining algorithms (Adapted from Gonzalez and Kamrani, 2001)

Table 1 lists the actual algorithms that are practical implementations of the 4 methods of DM, as adapted from Gonzalez and Kimrani (2001), and indicates which of the 4 techniques of modelling each algorithm addresses. Each method will be briefly discussed in terms of applicability to this research.

[2] Unlike either C4.5 or C5.0 The CART algorithm allows for the use of continuous outputs, however owing to the greater popularity of C4.5 and C5.0 in both literature and practical application CART will not be used in this research

2.5.1 Querying, OLAP (On-Line Analytical Processing), Statistics and Visualisation

These four methods are considered distinct to the four methods discussed in the following section as they require a notable amount of human expertise in order to provide useful information.

Visualisation is not considered a DM algorithm in itself, as it merely allows for data to be visualised to highlight erroneous data points or to obtain some general appreciation of the data. There is thus great inferential burden upon the operator, as the Visualisation approach is more of a supporting method than an actual method of analysis.

The methods of querying and OLAP are arguably precursors to Data Mining, and function by permitting a user to interrogate large databases. Querying permits structured questions to be asked of the data, and OLAP allows for data to be transformed in certain manners such that structure within the data becomes more apparent. Much as the case for Visualisation, these two methods do not perform the work of deducing relationships, instead they allow for the manipulation of data such that an operator can uncover interesting patterns. These methods hence also place notable inferential burden upon the operator.

Statistical methods are arguably most useful in quantifying expected relationships, and hence require a degree of prior knowledge. As stated by Hong and Weiss (Hong and Weiss, 2001) when applying traditional statistical approaches 'it is assumed that the correct model is known and the focus is on the parameter estimation'. The practitioner must thus have some preconceived notion of the pattern or structure within the data in order to apply such techniques.

The methods of querying, OLAP and statistics are discussed in greater depth in Chapter 11, a more complete coverage being considered necessary as they are useful and prevalent methods used within the database and data analysis domains, and hence are investigated in detail to further justify the decision to exclude them from this research.

2.5.2 Case-Based Reasoning, Decision Tree Induction, Artificial Neural Networks and Genetic Algorithms

These four approaches are all algorithmic in nature, and operate with a degree of autonomy[3] from the practitioner, essentially shouldering the inferential burden previously assumed by the practitioner.

Case-Based Reasoning (CBR) essentially catalogues previous examples based upon some pre-defined key parameters. When presented with a new, incomplete example, CBR can then extract similar previous cases by comparing the index parameters and infer the missing or incomplete information from these complete examples. The field of CBR has been used to good effect in the analysis of design and manufacturing data, for example setting mould dimensions and parameters (Tong et al., 2004) and in design reuse (Ong and Guo, 2004). The application of CBR to design reuse problems is particularly apt, as design reuse seeks to apply some aspect of a previous design to a new problem, and CBR acts to identify those cases that share some commonality. Both of these example cases are mentioned to illustrate that CBR can usefully be applied to problems within design and manufacturing. In this research, however, greater emphasis is placed upon providing information than upon providing a prediction, and it is suggested that inferring which previous cases are of similar character does not provide a great deal of information regarding relationships between parameters. In this respect, the information provided by this method will be related more to clustering or association as described in the list of 4 modelling methods.

Decision Tree Induction (DTI) seeks to generate a series of logical rules that prescribe the performance of a series of instances or cases given their characteristics[4]. The most

---

[3] It may be necessary for the practitioner to provide some initial parameters that guide the algorithmic development of the model, but such parameters control the manner by which the algorithm executes as opposed to controlling the generated model

[4] The idea of character is used to describe parameters of an entity that are essentially pre-determined or a direct result of design and manufacturing decisions or practice, for example in an engineering context this may include things such as weight, drag coefficient, power etc. The idea of performance, as the title implies, refers to things that are the manifestations of such character, for example the top speed of a car given weight, drag coefficient and power.

seminal of this family of algorithms are C4.5 (Quinlan, 1986) and CART (Breiman et al., 1984), both of which have been developed over two decades. The algorithm divides the set of instances into separate segments, each of similar performance, based upon the values of certain characteristics as seen within the full range of instances. By dividing these segments into smaller and smaller partitions, ultimately to the point where each member of a group has identical performance, a number of logical rules are generated[5] which relate the value of the characteristics to performance. When presented with an instance of unknown performance, the created rule set or tree can be used to provide an estimate of performance based upon the value of relevant characteristics (those characteristics specifically referred to within the logical rulesets). Equally of benefit to this research, the transparent structure of the tree or rule set provides useful insight into *why* a particular prediction has been made.

Artificial Neural Networks mimic the functioning of the brain in order to adapt an interconnected structure such that it gives an appropriate response when presented with a series of inputs. Such networks were proposed in the 1940s, however there was little interest until a number of limitations were addressed in the 1980s by Rumelhart and McClelland (1986). Arguably the most common type of network, the feed-forward back-propagation network, utilises a series of interconnected nodes. An input vector is entered into an a specific input level or layer of nodes, each of which provide a level of output based upon some function of this input. These outputs are factored by an adjustable weighting and these weighted outputs form a part of the input for nodes in a successive level or layer of the network. This 'feed-forward' process continues until the activity reaches a layer of nodes designated as output, where the overall network output is given by the output of these nodes. The network can be adapted to 'learn' a given function by passing input vectors with known or desired outputs, deducing the level of

[5] An example of such a rule would be "if engine power is greater than 100 BHP, and car weight is less than or equal to 800Kg, and the drag coefficient is less than 0.4, and the frontal area is less than 2 square meters, then the car is likely to have a top speed of between 130 and 140 MPH". A Decision Tree is essentially a structure that links rules into a tree-like device, where the individual elements of each rule are branches that are followed until a prediction (a leaf in the tree analogy) is reached. In the example given previously, for example, there would be a separate branch with rules starting with the logical predicate "if engine power is less than or equal to 100 BHP....." and continuing until a leaf is reached.

error seen between actual and desired output, and 'back-propagating' the error to indicate how best to adapt the weights on connections between nodes in order to minimise this error. In such a manner a network 'learns' a function that maps the inputs onto the required outputs, and when presented with an input vector with unknown output the network can be used to generate an estimate of the likely output. Merhods of interrogating the network such as sensitivity analysis (Zeng and Yeung, 2001) or rule extraction (Taha and Ghosh, 1999) can be used to provide important insights inot how a given prediction is reached, an important consideration in this research.

Genetic Algorithms are, in essence, a form of optimisation technique. A population of instances, each with differing characteristics, is created and gradually evolved via a series of operators such as random mutation and crossover ('swapping' or certain characteristics between two instances). The performance of each instance can be evaluated against a given requirement, and a section comprising the most successful instances selected to form a new generation. In this manner the average performance of the population as a whole improves over each generation. Such an approach does not seek to infer any relationship within the data, simply relying upon 'survival of the fittest' to lead towards a selection of instances with the most 'successful' characteristics, and it also requires the ability to predict the performance of each instance. In this research the goal is to provide means to generate a prediction of previously unknown performance, hence Genetic Algorithms are not a suitable method to employ in this research. It is noted, however, that Genetic Algorithms have been successfully deployed in the adaptation of Artificial Neural Networks

2.5.3 Applicability of Algorithms to Research

The techniques of visualisation, statistics, querying and OLAP can be excluded from this research as they require a significant degree of inference from an operator. If the goal of this research is to generate information regarding the relationships within data then it is necessary to have a formal method of extracting such relationships. Reliance upon the operator to perform this task is considered to defeat this purpose.

Of the techniques that interrogate data with a degree of autonomy, Case-Based reasoning is excluded from further use as it does not provide particularly rich information regarding the relationships between parameters, focusing more upon identifying commonalities between cases or instances. Genetic Algorithms do not attempt to learn any pattern,

simply providing a means by which parameters might be blindly adjusted in order to provide a steady improvement in performance.

DTI and ANNs are seen as the most suitable algorithms for further work, as they both seek to identify patterns within data (for the purposes of providing a prediction for future cases) and there are means to understand how and why a given prediction was decided upon. It is this information that this research seeks to provide.

2.6 Machine Learning Algorithms – DTI and ANN

It is necessary to have an understanding of the strengths and limitations of the ANN and DTI algorithms, as these will be extensively used throughout this research. The remainder of this chapter will provide an overview of these algorithms under a broader heading of Machine Learning, along with an introduction to methods used to combine the outputs from multiple models. It is intended that these discussions are brief, much greater coverage is given in Chapter 12 alongside that of Genetic Algorithms and their applicability to the improvement of ANN models.

It is important to recognise what is required of a Machine Learning model for use in a DM application. It is not useful to have a model that simply learns inputs by rote, including errors, and then try to establish some structure or pattern from this learning. It is far more desirable to create a model which learns the underlying structure automatically, as we are interested in the underlying process which the data describes rather than the data itself. A model which describes the underlying process from which data is obtained is said to *generalise* well, whereas a model which can only describe the training data to which it has been exposed is said to *specialise*. This key thought is introduced at this early stage, as it is a key consideration when generating DM models in practice.

2.7 Principals of Artificial Neural Networks (ANNs)

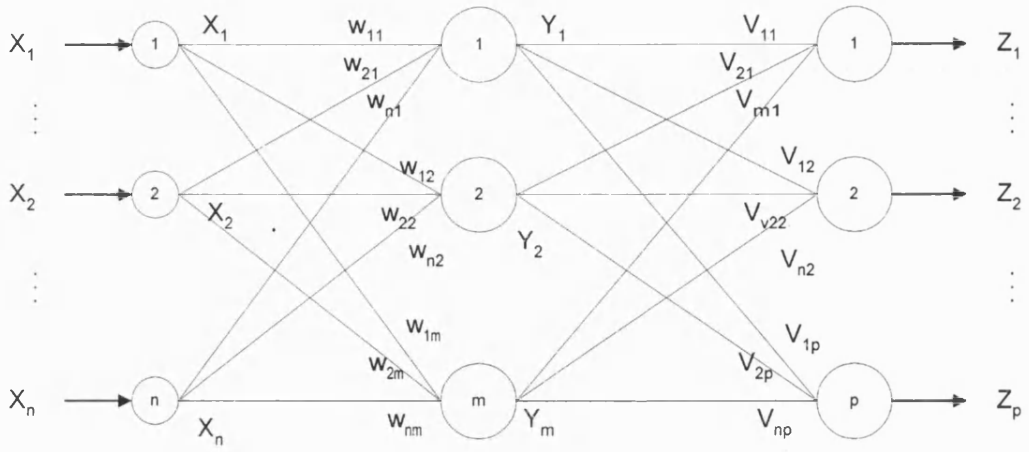


Figure 4 Feed-Forward Neural Network

Figure 4 shows a schematic diagram of a feed-forward network, a common form of ANN. The network provides means by which an input ($X_1 \ldots X_n$) can be translated into an appropriate output ($Z_1 \ldots Z_p$). The activity thus flows from left to right in this example, giving rise to the term feed-forward. The input into each node is translated into an output value using an activation function. This output is multiplied by a weighting (labelled as W and V in the figure), and the factored outputs are summed and used as the input to the next layer of nodes. This proceeds until the output layer is reached.

In order to provide a prediction from an input vector with unknown output, a series of instances with known output are used to adapt or train the network – these instances are known as the training set. The task of the training algorithm (the specific method by which the network is autonomously adjusted) is to adapt the weightings on each connection such that the output given by the network for each instance corresponds with the required output. This training data is passed through the network and the difference between predicted and required output evaluated. This error is passed back to preceding layers of nodes, and the weights adjusted to minimise this error. A range of training algorithms exist, the most popular being the Back-Propagation algorithm (Rumelhart et al., 1986), however other methods of numerical solution to error minimisation may be used (for example Hagan and Menhaj, 1994 use the Levenberg-Marquardt algorithm).

2.7.1 Discussion of the Merits and Demerits of ANNs

ANNs are tools which can model extremely complex domains and domains with noisy or incomplete data. Once created, they are extremely simple to train and validate, merely

requiring the data be presented in a suitable format, and can easily be updated with fresh data. There are numerous varieties of network, however the most ubiquitous is the feed-forward network utilising the back-propagation algorithm for training, and the majority of research in the area focuses on such networks.

One of the most significant problems associated with ANNs are their sensitivity to numerous parameters, such as topology and the form of training data, and the lack of an explanatory mechanism for the decisions they make. There are numerous techniques available to deduce suitable parameter values for ANNs, where the difficulties in defining an optimum architecture could feasibly be addressed using Genetic Algorithms (as discussed in section 12.2.5).

Whilst it is important to be able to create accurate models, such models will be of little use in this research if information cannot be extracted from them. Various rule extraction algorithm shave been proposed, for example by Lu et al (1996), Fu (1999) and later Fu and Shortliffe (2000), Gupta et al (1999) and Taha and Ghosh (1999). These algorithms have yet to be conclusively tested upon practical examples, and are typically limited to specific types of ANN model. It is for these reasons that rule extraction will not be considered in this research.

It is possible to obtain information from a network by perturbing the input vectors and evaluating the influence such perturbation has on output. Sensitivity analysis utilises this idea in order to indicate how much influence an input parameter has upon output. This analysis is typically enacted by varying the value of each input parameter in turn whilst maintaining the value of each remaining input at a pre-determined level, and aggregating and effect seen on network output. In this manner it is possible to rank the input parameters according to the sensitivity of the output to changes in their values.

Rademan et al (1996) and Shelley and Stephenson (2000) both present practical applications of this technique. It should be noted that sensitivity analysis is also used to test the stability of a software-based network prior to it being manufactured as a hard-wired device, where such analysis seeks to ensure that the network is robust in the face of inadvertent input perturbation (noise) in practice (for example, see Zeng and Yeung, 2001).

Sensitivity analysis, whilst first order in nature, indicates those parameters that exert greatest influence upon output. Such an analysis seeks to deduce if perturbation of an

input causes large variation in output, in essence if variation within tolerance of a dimension or measurement of a product causes a large variation in performance. If this variation in performance exceeds permissible levels, or is of a magnitude that is cause for concern, then it can be inferred that attention should be given to those dimensions whose perturbation was the cause of such variance.

This necessarily means that the sensitivity analysis should be carried out using values that would be seen in practice, where it should be ensured that perturbation of a parameter does not, for example, result in the tolerance allocated by the designer being exceeded. It is suggested that deducing where dimensional variance causes excessive variation of performance would be of little use if the dimensional variance would be extremely unlikely to be seen in practice. It is argued that it would be significantly more useful to understand where permissible variations in dimensions acted to cause significant variance in performance, in which case the allocated tolerances could be refined based upon this knowledge. It is further argued that the nature of Machine Learning suggests that algorithms learn from previous cases, and hence can only be used in areas where the data they are presented with is of a similar form to that used in training. This adds weight to the suggestion that perturbation of the parameters should be constrained to physically viable levels to ensure that the output variation is due to a genuine relationship within the data, and that it is not due to use of the network outside of its range of application.

2.8 Principles of Decision Tree Induction (DTI)

This section discusses the function and applicability of Decision Tree Induction to the research at hand. This discussion is based upon the C4.5 algorithm (Quinlan, 1986). Decision Tree Induction and its related technology Rule Induction[6] seek to generate rules by which a prediction for an output can be ascertained by consideration of the value of certain characteristics. This is achieved by partitioning a training set[7] of data into ever-

[6] DTI and Rule Induction operate under the same general principles, with the exception that DTI seeks to generate a complete structure of interconnected rules whereas each rule in Rule Induction is independent of the other rules.

[7] As described in the section on ANNs, a training set is a series of instances with known output, used in the development of a Machine Learning model. The ambition is to provide an explanatory or predictive

smaller segments, with the goal of generating segments whose members all have the same value for output. The characteristics of each instance are analysed, whereby the algorithm specifies a threshold value for a specifically identified characteristic, such that any instances with a value for this characteristic that is greater than the threshold are placed into one segment, and those with a lower value into another segment. In order to carry this out each possible threshold value for each characteristic is considered, and those which most successfully divide the training set in terms of output is the one selected for use. Note that the training set is split via consideration of a given *characteristic* with the ambition of aligning instances if identical *output*. Note also that this method requires the outputs to be discrete variables, where there are a finite number of classes[8]. This process continues on each segment, further and further sub-dividing each segment. This terminates when each instance in each sub-segment has the same output value as each other instance in that sub-segment.

mechanism which gives the output of each instance when it is presented with the characteristics of that instance. In this manner the developed model can be used to estimate output for future cases given only the characteristics of that case.

[8] The CART algorithm allows for continuous variables to be used as output, however the underlying function of the CART algorithm is notably different to the C4.5 algorithm described here. C4.5 was selected above CART as C4.5 is more extensively referenced in Machine Learning literature.

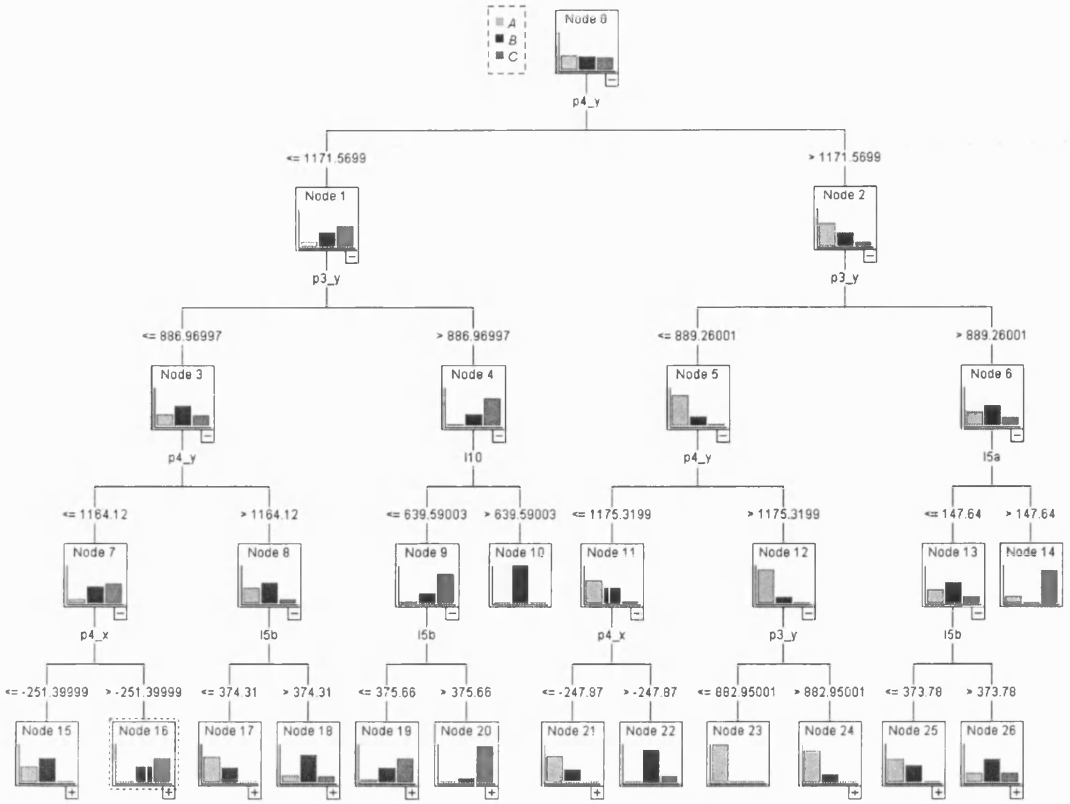


Figure 5 Example Decision Tree

Figure 5 shows an example of a Decision Tree, included here to indicate the completed form such a tree takes. In this example there are three possible outputs, A, B and C. At the very top of the tree Node 0 there is a small histogram indicating the number of instances of each output at that point. At Node 0, the proportion of instances with outputs A, B and C are roughly equal, with any variance being due to unequal numbers of each class in the dataset. Immediately underneath node 0 there is a logical condition, all instances with a value of characteristic "p4_y" less than or equal to 1179.5699 are passed to the left of the tree (towards Node 1) and those with a greater value to the right. It is possible to see that the proportion of instances at Node 1 is biased towards class C, whereas at Node 2 the bias is towards class A. This process can be seen to continue until leaves ('terminal' nodes) are reached or the branch is truncated for reasons of space (as indicated by the small icon at the lower right of Nodes 15 to 21 and 24 to 26. Of the genuine leaves, node 22 contains predominantly instances of class B whereas Node 23 contains those of class A.

DTI has seen extensive utilisation within the Machine Learning community, C4.5 in particular being deployed as a control algorithm in the testing of many new Machine Learning algorithms (as discussed in section 12.3.4).

2.8.1 Discussion of the Merits and Demerits of DTI

As opposed to the ANN algorithm the DTI algorithm requires little work in setting up the network prior to training. There are only a few parameters which may be adjusted to restrict the size of the tree to manageable proportions and prevent the onset of overtraining (the 'specialisation' mentioned at the start of section 2.6).

Of more importance to interpretability and extraction of information, the DTI algorithm has the strength of transparency, where it is immediately clear what particular role a given characteristic has upon the predicted output of a model. This is in direct contrast to ANNs, whose connectionist nature obscure the influence of a given characteristic. Such a strength suggests that the DTI algorithm is suited to the problems identified in this research.

DTI is, however, less robust in the face of noise and is susceptible to the vagaries of data sampling. The identification of a suitable combination for characteristic and threshold by which to divide the data is affected by noise, as errors in the data could result in a sub-optimal combination being selected by either increasing or decreasing the computed accuracy or suitability of a given combination. Such an effect is also possible due to vagaries in data sampling, where one sample might contain data favouring a certain combination whereas an alternative sample might contain data favouring a different combination. As further selections for combinations are influenced by the population of the resultant segments, any change in the selection for the root node (Node 0 in Figure 5) could feasibly lead to massive changes in the structure of the remainder of the tree. Whether this is a problem is open to interpretation, as models of entirely different structure may give identical predictions for a given future case, but should be considered when interpreting the structure of the tree (i.e. when interrogating the model to identify why decisions were made).

2.9 Aggregation of Multiple Models

The iterative nature of DM can give rise to the creation of numerous different models, some of which might be created with different data samples, algorithm types or algorithm

parameters. Assuming that the models have been correctly validated it is possible to use the computed accuracy as a metric by which to select a suitable model for use, however this assumes that an obvious choice is present and can be ascertained[9]. It also ignores the fact that models may only partially cover a domain, in which case selecting a single model might result in the omission of important information.

As a way of addressing these issues, it is also possible to select a range of models, and aggregate or otherwise combine their outputs. A number of formal methods for such an approach have been proposed, of which two have been widely discussed in the literature and will be introduced here (a more comprehensive coverage is given in Chapter 13).

The two techniques both seek to improve prediction accuracy over a single model, but each takes a differing interpretation of the actual cause of error in prediction. Bagging (Bootstrap Aggregating, Breiman, 1996) aggregates multiple models in order to mitigate against problems associated with data sampling. It addresses the instability of certain ML algorithms in the face of variations in training data composition (DTI in particular). A range of models are created using a different random sub-sample of training data, where the random sub-samples are drawn from the main training set via bootstrap sampling. In use, an overall prediction is obtained simply by averaging the prediction from all of the generated models..

Boosting (Freund and Schapire, 1997) essentially attributes prediction error to the inadequate modelling of data by the algorithm. It assigns a weighting to each instance of data, and creates several iterations of model. At each stage of iteration the accuracy of prediction for each instance is evaluated, along with overall model accuracy, and those instances whose outputs are incorrectly predicted are given an increased weighting. During the next stage of modelling the Boosting algorithm forces the ML algorithm to

[9] A range of models might have accuracies that are identical, hence identifying the most appropriate model for further use is nontrivial. It should also be noted that any method of ascertaining model accuracy suffers from some inherent inaccuracy in its own right, and hence even a model with the highest indicated accuracy cannot be guaranteed to give the greatest interpretation of the underlying domain (in mitigation, the use of an appropriate measure of accuracy ensures that such variation in evaluation is minimal). This important point will be returned to in Chapter 4, when methods of evaluating model accuracy will be expanded upon.

focus efforts upon correctly classifying the highly weighted instances. This process continues for a pre-determined number of iterations. In use, a prediction is obtained by factoring the prediction of each sub-model by the pre-determined accuracy of each sub-model, and averaging these predictions. As opposed to Bagging, Boosting therefore takes into account the differences in accuracy of each sub-model.

2.9.1 Applicability of Methods of Model Aggregation to Research

The instability of DTI models to both noise and data sample composition suggests that the application of aggregation will perhaps be necessary. These methods will be applied in due course, and their benefits evaluated.

It should be noted the methods of aggregation focus exclusively upon the combination of *predictions* from ML models. This research seeks to extract *information* from such models, and hence the approaches to aggregating predictions can be used to guide the aggregation of information. This notion forms the basis for the work described in Chapter 6, where a novel method of combining information extracted from ML models is presented.

2.10 Concluding Remarks

This chapter has sought to identify the key concepts of DM, in doing so highlighting where differences in terminology may cause confusion and where efforts to standardise approaches have been developed. The modelling 'engines' of DM, the Machine Learning algorithms, have been discussed and the two key algorithms, DTI and ANN, have been identified as having greatest potential in this research. It has also been noted that the successful validation or establishment of model accuracy is essential.

The following chapter seeks to identify where DM has been applied in engineering, and how comprehensive such applications have been, in order to establish where gaps in knowledge exist and which issues must be addressed in this research if manufacturing data is to be usefully interrogated using DM techniques.

Chapter 3 Applications of Data Mining in Engineering

In common with many other fields, efforts have been made to implement DM within engineering. This chapter will review the scope of these efforts, and indicate where the research described within this thesis fits within this framework. It was noted in Chapter 1 that the specific nature of DM has been misunderstood in many areas, where the term Data Mining has been incorrectly or inappropriately assigned to Machine Learning studies. This has been noted in certain studies reported in the field of engineering and design, and these studies, whilst valid pieces of research, will not be discussed in this chapter[10].

3.1 Nature of Data in Engineering

It is argued that significant differences exist between data generated within areas such as finance (a typical area of implementation for DM projects, as indicated by Hu, 2005, who describes analysis of customer attrition within finance) and engineering. Rudolph and Hertkron (2001) suggest that data generated during engineering suffers from a degree of decoupling from the process it is intended to describe. While data describing financial transactions is typically a reasonably accurate representation of the actual process (in effect, in electronic transactions the data *is* the process), in engineering fields there are attendant problems with obtaining a quantitative value for a physical phenomenon. Aside from the common problems associated with accurate measurement, there are also issues relating to the quantity of data being recorded and the degree to which the recorded data covers the entire process – it is entirely feasible to manufacture a product without once recording a piece of data describing it.

[10] Chapter 12 provides great detail on the functioning of three Machine Learning Algorithms, ANNs, DTI and Genetic Algorithms. Included in this discussion is a coverage of applications of such algorithms within engineering.

There exists significant quantities of data within engineering organisations that result from experimental analysis, which were performed with the exclusive aim of furthering the understanding of the product in question. A survey of these efforts will not be carried out in this research, as it is suggested that any such experiment will, if designed properly, include necessary measures of analysis. It is argued that such experimentation cannot be defined as DM as they intrinsically exclude the difficulties inherent in collating and analysing 'real-world' data, and any method of analysis contained within such approaches cannot be guaranteed as being applicable to such 'real-world' situations. In experimental terms, DM may be considered to reside outside of the scope of such analysis, falling into a group that Hicks and Turner (1999) define as follows:

> 'There are other types of research that are not experimental....values of the variables have been determined by circumstances beyond the control of the experimenter, the variables have already acted, and the research measures only what has occurred....these investigations may use statistical methods to analyse data collected but none is really experimental in nature.'

It is this data that this research seeks to investigate, hence it is worthwhile considering what form such data might take.

Referral to section 8.4 of ISO 9001:2002, the standard describing Quality Assurance that has been taken up by numerous organisations, indicates that a company is obliged to 'determine, collect and analyse appropriate data to demonstrate the suitability and effectiveness of the quality management system' (ISO 9001:2002). It is suggested that, as it forms a requirement for a commonly adhered-to international standard (with over half a million world-wide certifications as of 2001, Helberling, 2002), it is this data that is likely to be encountered on a consistent basis. The ISO 9001:2002 standard is a revised (and arguably more coherent) version of the earlier ISO Quality Assurance standards (Gingele et al., 2003) and is geared towards using '..computers to support decision making by collecting and analysing quality information such as customer requirements, quality goal, product/service design, material inspection, process control, storage, shipment, package and delivery' (Tan et al., 2003). The emphasis is upon improving the overall structure and quality of a process by correct documentation and informed control of the process, a state that can be achieved only by adequate and continual feedback. The standard does not dictate how to assemble or manage a business process, rather it acts to ensure that a company is, in effect, internally auditable by

ensuring that appropriate control, measurement and evaluation procedures are met. It is during this measurement and evaluation that data is generated, however, the form and nature of this data is decided entirely by the company and only has to serve the purposes of enabling this audit – in effect, the only requirement of the data is to prove that the product met all performance requirements and was subjected to all planned stages and testing procedures, and list who authorised its release.

The collection and analysis of data has been suggested to be one of the more awkward tasks specified by ISO 9001, a survey of 227 US firms found that this task was ranked the third most difficult of all those contained within ISO 9001 (Liebesman, 2002) and it is argued that the lack of triviality in performing this stage will result in considerable variation in the manner in which it is approached by different companies. It is therefore suggested that the exact nature of the recorded data cannot be anticipated in those companies accredited to ISO 9000, the only statement that can be made with any degree of certainty is that there should exist data that proves that a given product has both been subjected to all required tests and meets all performance requirements.

3.2 'Pseudo-DM' Approaches

There are a number of analyses within engineering that, despite not being explicitly described as such, follow a methodology that may loosely be classified as DM. Reich has focused upon developing methods of incorporating Machine Learning techniques within engineering, with a particular emphasis upon Design, and has proposed a 7-stage method of implementing a Machine Learning analysis of engineering data. This approach may be seen in Figure 6 (Reich and Barai, 2000). This approach shares many of the features contained within the CRISP-DM methodology, in particular there are specific stages assigned to the processes of data and initial knowledge collection, of post-modelling evaluation and of deployment. There is also a notion that the entire process must be iterative, with feedback loops to previous stages.

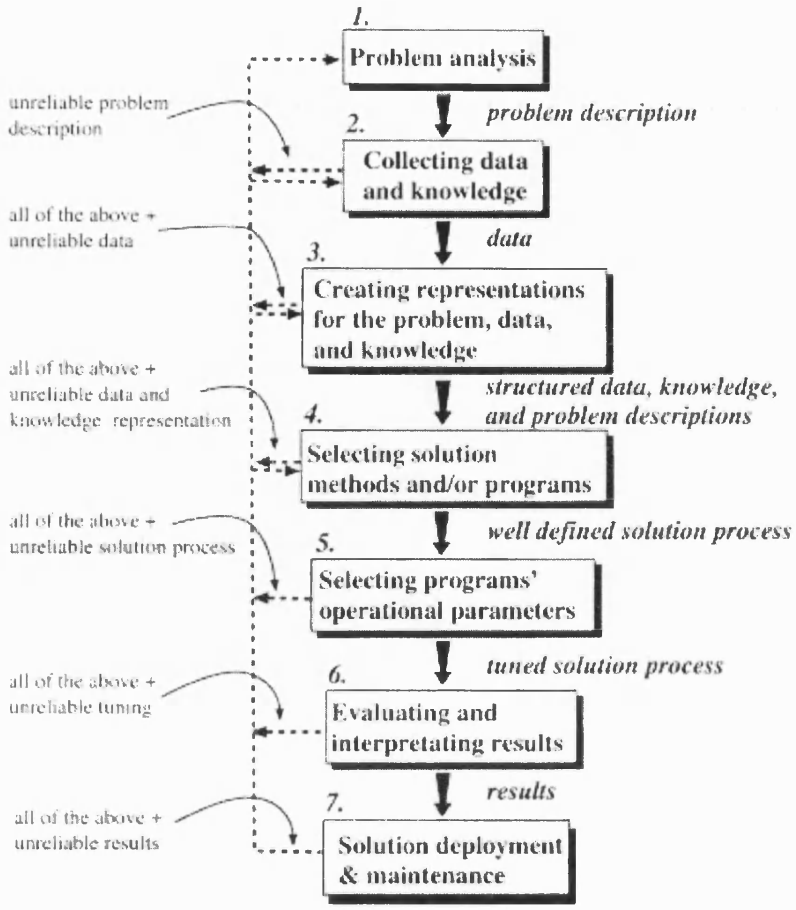


Figure 6 CMLM Methodology (Reich and Barai, 2000)

The actual application of the methodology proposed by Reich and Barai was demonstrated upon data generated by experimental means, and whilst the author does not consider such an analysis as DM in the truest sense, as such analyses may be considered as an intrinsic part of the experimental method, the suggested approach shares many of the features common to DM.

3.3 Other Interpretations of DM in Engineering

Hertkorn and Rudolph (2000), who use the term Knowledge Discovery in Databases (KDD) to represent the entire DM process and Data Mining to denote the analysis stage, present a methodology which relies heavily upon data transformations as a key stage of the DM process. Figure 7 shows the methodology they propose, based upon the more generic work presented by Fayyad *et al* (1996).

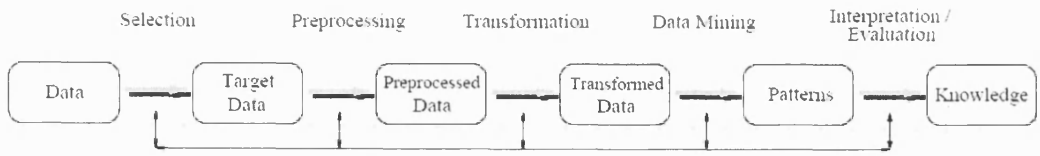


Figure 7 Hertkorn and Rudolph (2000) DM Methodology

There is significant emphasis placed upon sampling and manipulating data, as seen in Figure 8, where a development is proposed which attempts to reduce the dimensionality of the data via assigning dimensionless groupings.

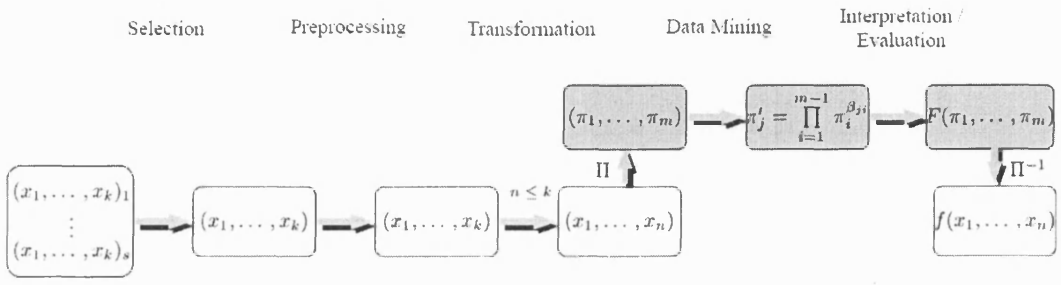


Figure 8 Hertkorn and Rudolph (2000) DM Methodology Incorporating Dimensionless Groups

The use of dimensionless groups was developed by Buckingham (1914) as a means to simplify analysis by creating groupings of parameters that define a specific functionality or behaviour, where the units of measurement cancel out to leave the entire group dimensionless, and where different groups may be compared without a full understanding of the nature of each group. The Reynolds number, used to describe a characteristic of fluid flow, is an example of such a dimensionless group that will be familiar to many engineers. In order to assign such a methodology to the analysis of engineering data involves an assumption of the nature of such data, in that it contains a sufficiently rich set of parameters to construct dimensionless groups. It is argued that such an approach may be useful in the analysis of controlled data, such as that produced by specific experimentation, however it is not considered probable that such complete data will be generated during manufacture. However, excluding the dimensionless groupings, this methodology presents a similar general DM approach to those presented elsewhere.

DM has received considerable attention within the semiconductor industry, representing perhaps the most fertile ground for DM within the engineering domain. As an illustration, in the book edited by Braha (2001) which consists of a series of papers discussing DM within design and manufacture, 5 from 11 of these papers focussing upon DM within manufacturing were based within the semiconductor industry. This

proportion is made even more significant when it is noted that 2 of the remaining chapters deal with such specialised areas as investigation into human interpretation of colour in monitoring equipment and autonomous control of a robotic soccer player, neither of which can reasonably be considered as typical engineering applications. Of the 5 papers dealing exclusively with the semiconductor industry, four of them aim to improve product yield (the number of good chips obtained from each silicon wafer, Maimon and Rokach, 2001) and 3 express concerns over the large volumes of data prevalent in the field. It is argued that these characteristics are notable in differentiating the semiconductor field from other engineering areas. It is suggested that perhaps the most significant difference relates to the huge volume of data available, for example Last and Kandel (2001) describe an example containing 58,076 instances or separate artefacts. It is suggested that, in many engineering fields, the volumes of data will not approach these quantities. The data is also collected for the express purpose of measuring factors affecting yield, as this measure tends to be used as the basic measure of profitability in the semiconductor industry (Last and Kandel, 2001). In this respect there is one over-riding goal that must be fulfilled, and all efforts can be directed towards the meeting of this goal. The actual physical performance of each semiconductor is not measured or even of interest, there is simply a set of rejection limits that must be fulfilled, and hence there is an extremely clear-cut parameter that defines process quality.

It is argued that in many engineering concerns the goal of the manufacturing process is rarely so clearly defined, as more emphasis is put upon quantifying the actual physical performance or characteristics of a product, and as such it is not possible to clearly identify the data that is likely to be necessary to predict or control compliance to this goal. It is also suggested that the volume of data will not be as high in many other engineering domains, the two cases studies in Chapter 7 and Chapter 8 containing less than 500 instances in both cases, as opposed to tens of thousands in the semiconductor industry.

3.3.1 DM in Design

Schwabacher et al (2001) use DM methods to assist in identifying suitable prototypes for a design problem from a repository of previous designs, in effect using DM methods to assist in redesign. As an illustrative example they extract a yacht hull initial design from a series of prototypes based upon the prototype performance under different conditions.

In this way, efforts were made to provide an initial design tailored for a particular set of conditions. In the same area, Grabowski et al (2001) attempt to automatically derive a classification system to enable such redesign using clustering to identify *classification relevant features*, or aspects of different designs that exhibit similar characteristics. A further effort at using DM methods to improve design reuse is proposed by Romanowski and Nagi (2001), where efforts are turned towards using data from product lifecycle support systems

Ishino and Jin (2001) attempt the rather more grand task of capturing designer's intent and 'know-how' by monitoring the modification of features in a CAD system and retrospectively examining the areas of the design that were focused upon and what processes were used to modify the design.

It is argued that many of these efforts differ from the definition of DM given at the start of this thesis. DM is not the process of analysing data, it is the process of interpreting what a business does and how it can improved, collecting and analysing data in the areas that would benefit the business, and then interpreting the results to improve the business. Such applications are not the focus of this research, as they seek to identify patterns across ranges of products rather than focusing upon relationships between different products of identical design.

3.4 Focus of Research

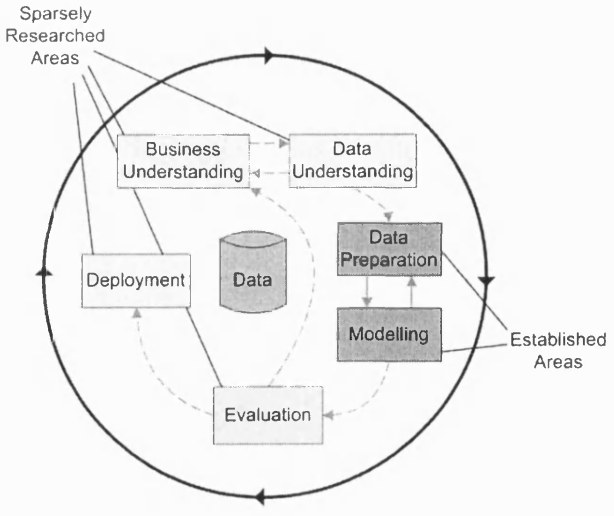


Figure 9 Focus of Research (adapted from CRISP-DM, 2000)

Figure 9 shows the areas that are considered to be under-researched within the CRISP-DM framework, thus indicating where this research will focus. These under-researched

areas can be split into two parts, that of evaluation and deployment and that of business and data understanding. These two parts will be discussed in more detail.

There have been many notable efforts at analysis of engineering data, many of which follow lines similar to this proposed by the CRISP-DM methodology. It is noticeable, however, that there is little mention of how the generated models may be interrogated, save for a consistent reference to DTI methods being more 'interpretable' than ANN models (for example Witten and Frank, 2000, deliberately exclude ANN models from their coverage of DM modelling for this very reason). Much of the work appears to focus upon improvements to the modelling algorithms, in terms of improving the predictive performances, with little attention paid to how to elucidate information from the models. In many cases the construction of a predictive model is the overall goal of the work, as it can be used to indicate likely future performance of a product based upon certain characteristics. The research described in this thesis aims to use the created models in a different manner, and to address the issue of *why* a model makes certain predictions rather than focus upon the actual value of the prediction.

A further issue relates to the nature of data recorded during a typical manufacturing process. DM has been usefully applied to the semiconductor industry, however it is suggested that this area is characterised by good data with coherent means of measurement which have typically been precompiled with analysis already in mind. It is argued that this will not be the case in other engineering concerns, where measurements are not as easily taken and the desired behaviour is not as clearly understood or prescribed – in the case of the semiconductor industry, much of the DM analysis aims to predict or improve yield. Pyle (1999) attempted to describe the issues relating to the preparation of data for DM, however this has focused almost exclusively upon issues within the marketing and finance areas, which is suggested to be of significantly different nature in terms of data and ambition than that of manufacturing.

There is, to the best of the author's knowledge, no literature available to describe the nature of data that is likely to be available for modelling within an engineering organisation. The quality assurance standard ISO9001:2002 contains reference to the recording and analysis of data describing the manufacturing process, but does not explicitly define the specific data.

In cases where engineering data has been analysed, it was noted that the data was reported in summary terms, comprising a simple statement of the list of variables and the number of instances available. In this way the data was treated on a case-by-case basis, in effect the data being treated at face value. Generic observations regarding manufacturing data were not made, and no further understanding of manufacturing data or the errors within it was obtained. This is suggested to be a stumbling block to successful DM implementation within engineering, as the retrospective nature of analysis within DM suggests that problems identified within the data at the time of analysis cannot be rectified. As it currently stands, analysis of the data is the only method of identifying if the data is suitable for analysis, a circular arrangement whose impact is made more severe when it is considered that, by the time analysis is carried out, the manufacturing operations will have been carried out and the data will have already been collected, and hence the data cannot be improved upon.

Given the lacks of means to investigate manufacturing data and indicate its suitability for analysis prior to undertaking such analysis, it is argued that providing an external means of evaluating data collection within a manufacturing operation is a necessary part of this research. In this manner issues within the collection of manufacturing data can be addressed immediately, ensuring the quality and veracity of the data, and hence increasing the likelihood of successful analysis.

Chapter 4 Establishing a Modelling Approach

The purpose of this chapter is to indicate how manufacturing data may be modelled using the DTI and ANN algorithms discussed in Chapter 2. This will be demonstrated via the use of a controlled experiment, where an *analytical* model will be used to generate manufacturing data such that the ANN and DTI algorithms can be used to generate *predictive* models from this data. Comparison between the performance of the analytical model and the predictive models will indicate how accurately such predictive models map manufacturing data.

It is useful at this stage to state precisely the nature of the data that is under consideration. The research seeks to indicate how variations between products of the same type manifest themselves within the performance of the product. This research was undertaken with the goal of deducing how tolerance allocations could be revised by analysing their effects upon system performance during testing. In this respect it is important to capture both the specific measurements relating to the tolerances and the resultant performance of the system. These two types of data are the characteristics and performance values of the system respectively. When passed to the DM models, these data become the input and output values respectively, as the predictive models attempt to estimate the values of performance given the values of the characteristics.

The accuracy of predictive modelling in the presence of noise will be investigated. In an engineering context noise describes the error introduced into measurement data by factors such as inaccuracies in the methods of measurement, operator error and inaccurate data entry. In the experiment described in this chapter, the noise is an artificially introduced random variance within the data used for predictive modelling, where varying amplitudes of noise were progressively introduced.

4.1 Experimental Rationale

The purpose of the experiment is to understand how the modelling aspects of DM might be applied to manufacturing data. To this end an analytical model of a manufacturing system will be used to generate data which approximates the type of data recorded during

a typical manufacturing process. This data will describe the characteristics of the system. The analytical model can further be employed to estimate some measure of the resultant behaviour of the manufacturing system, thus generating the requisite performance values. By performing this process a number of times, and varying the characteristics of the system in a manner equivalent to natural manufacturing variance within tolerance, a set of data can be created that describes a batch of products. This generated dataset, including the estimated performance data for each variation of the system, can then be used to generate predictive models. A comparison can then be made between the estimated performance from the analytical and predictive models, indicating how representative the predictive models are of the manufacturing system.

4.2 Analytical Model - Simple Link Mechanism

The system under investigation in this experiment is a simple link mechanism, where the lengths of the various links and position of the pivots act to influence the kinematic properties of the mechanism, such as the acceleration and velocity of parts of the mechanism.

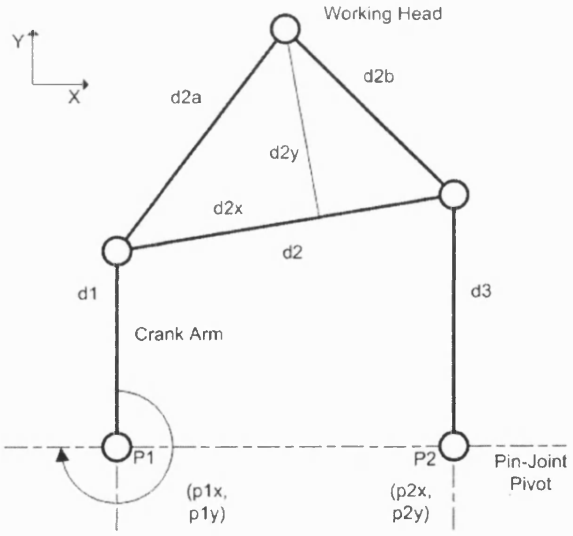


Figure 10 Schematic of Simple Link Mechanism

Figure 10 shows the arrangement of the simple link mechanism used in this experiment. The various parameters of the mechanism are indicated, of which 5 are linkage lengths and 4 are the Cartesian coordinates of the pivot point positions P1 and P2. The mechanism is driven via the crank arm, which pivots around point P1.

In this example the maximum velocity of the working head is selected as the measure of performance to be recorded and used as output within the predictive modelling[11].

4.2.1 Method of Analytical Modelling

The SWORDS constaint modelling software package (Kenney et al., 1997) was used to create the analytical model, allowing for the kinematic behaviour of the constructed model to be evaluated. Initially developed for mechanism design and development, it has proved versatile and has been successfully used for a number of tasks including the modelling of wrist replacements (Leonard et al., 2002). A series of constraints are assigned to the system, for example a certain linkage end point must always remain in contact with a separate linkage end point. SWORDS seeks to satisfy these constraints during motion of the mechanism. It is a simple matter to modify linkage lengths as the nature of the constraint-based approach means that the mechanism is automatically reassembled given any such change.

The creation a range of instances of mechanisms, each with variations in linkage length and pivot point position, was accomplished by repeating the modelling whilst allowing the values for the parameters to randomly vary between limits of 95% and 105% of the original value. In the case of the pivot point positions, which are defined by Cartesian coordinates instead of absolute values as is the case for linkage lengths, the variation of both components of the coordinate were factored by a random number between -5% and +5% of the total height of the complete mechanism. This simple measure ensures that there is a degree of proportionality to the variations when comparing linkage lengths and pivot position.

In total, 100 different mechanisms were created and analytically modelled, representing a small-scale production run. This volume was selected as it is of the same order as the data generated in the second case study as described in Chapter 8.

[11] This decision is purely arbitrary, being selected as it serves the purpose of illustration in this simple example.

4.3 *Predictive (DM) Modelling*

The task of the DM modelling is to generate predictive models that provide an estimate of the maximum velocity of the working head given information regarding linkage lengths and pivot positions. This necessary understanding of the patterns within the data that govern the maximum velocity can be inferred from previous cases.

4.3.1 Structure of Data for Predictive Modelling

Predictive DM algorithms generally require that data be presented to the algorithm as a series of instances, or individual cases or exemplars, divided into appropriate input and output vectors. In this experiment there were 9 input parameters and 1 output parameter, the maximum velocity of the working head.

Input Parameters	**Label**	**Output Parameters**	**Label**
Crank length	d1	Maximum head velocity	vel_max
2nd leg length	d2		
Crossbeam length	d3		
Cartesian component of working head location	d2x, d2y		
Crank pivot co-ordinates	p1x, p1y		
2nd leg pivot co-ordinates	p2x,p2y		

Table 2 Composition of Rationalised Dataset Generated for Predictive Modelling

Table 2 shows the composition of the rationalised dataset, where the inputs are as shown on Figure 10.

4.3.2 Range Definition for DTI

Where ANNs model data with continuous output[12] the C4.5/C5.0 family of DTI algorithms[13] require the output to be divided into consecutive discrete, pre-defined

[12] Continuous data is data whose values are Real Numbers. Textural or symbolic data can be handled by ANNs but require some form of pre-processing.

[13] The C5.0 algorithm is a commercial development of the freely-distributed C4.5 algorithm as discussed in detail in 12.3.

classes or ranges. The maximum velocity of the working head is recorded as continuous data, hence a method of portioning these into ranges is required.

Such consideration of the boundaries used for splitting, or conversely of the composition of each range, is arguably best performed when driven by some external criteria – for example, if the maximum velocity that may be permitted is known a priori, it would be sensible to split the data using this maximum permissible velocity as a boundary. This is one of the strategies proposed by Turney (1995), who attempted to model semiconductor yields using DTI methods. However, Turney neglected this strategy and instead suggested two other methods. The first suggested that a median should be calculated and used as the boundary, thus ensuring each range would have a substantial amount of data within it. This contradicts the position taken by Rademan *et al* (1996) who state that range boundaries must not be placed near the centre of a normal distribution, as this is argued to increase the noise seen in the classification. Turney's second method relied upon manual examination of the data to try to deduce any 'natural' breaks within the data, in effect seeking to identify natural groupings within the data.

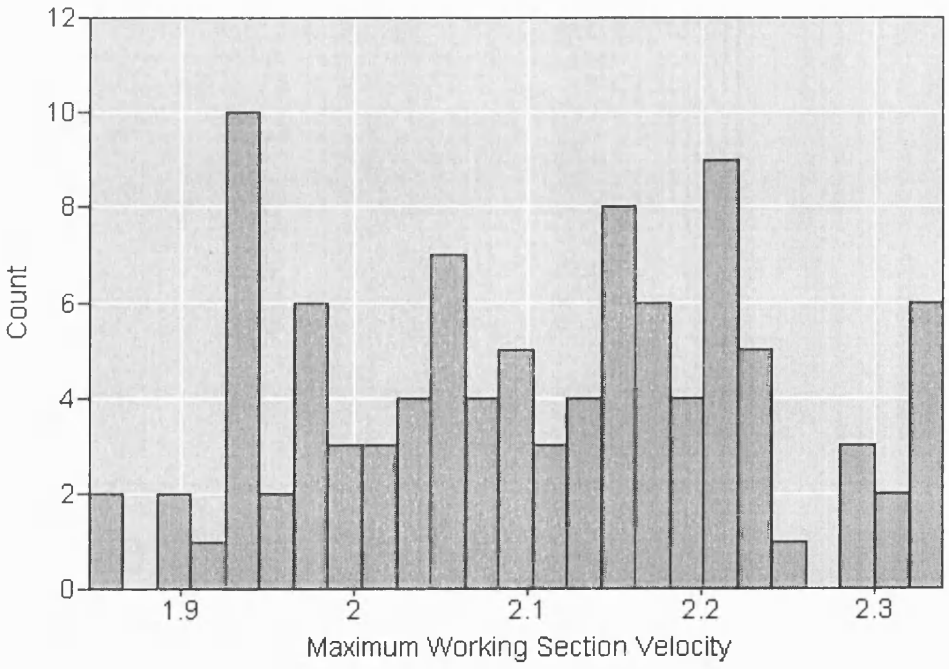


Figure 11 Histogram of Maximum Working Head (Section) Velocity for Simple Link Mechanism

It can be seen in Figure 11 that there does not appear to be any natural grouping of the maximum velocity that would allow for a boundary to be drawn. There are two groups at the extreme ends of the distribution, however these are argued to be too sparsely

populated to use as ranges in their own right. It is also noted here that the identification of ranges by cursory examination of the data requires that the subset of data under examination is descriptive of the population in general. The use of DM methods with limited quantities of data does rely upon some acceptance of the data as being representative of the process under examination, however there are methods of validating the models that are created whilst in this case there is no manual method of deducing if the groupings hold for the general population[14]. It is suggested that the most suitable method of deducing range boundaries, as proposed by Turney, is to create ranges with approximately equal quantities of data within them.

This still leaves the problem of selecting the number of ranges. Many cases are simply binary splits, in effect there is either a positive or a negative range. In cases where the data is transformed from numeric, or continuous, values into ranges, or discrete values, there are considerations relating to the granularity – the greater the number of ranges, the finer the resolution of the classification, whilst the fewer the number of ranges the less complex the analysis (which simplifies the task of the classifier). In the absence of information guiding this choice there is some compromise as to how many ranges to select. It is suggested, based upon observations in the data used in this analysis, that some form of iteration is required to select an appropriate number, where efforts should be made to extend the number of ranges without compromising model accuracy.

In this example 5 ranges were selected with equal quantities of data in each range.

4.3.3 Modelling Accuracy

There are numerous examples of machine learning within engineering applications, however in many of these tasks effort has been expended mainly upon the creation of a model with high accuracy, and attempts to implement or utilise the models, if carried out at all, have only been initiated upon successful creation of the model. This focus upon accuracy may be justified for applications where the model may be used as a predictor, where the only actual implementation of the model relies upon an accurate, informed estimate of the likely value for an output given a vector of inputs, and hence the primary

[14] There are other statistical or machine learning methods available, however this complicates what is a otherwise a relatively straightforward task

metric in judging how suitable a model is for use relies heavily upon how accurately it may produce an estimate of output. Such implementations of machine learning require comprehensive validation, as it is only by post-training evaluation of the models that a representative idea of how well the model will deal with unseen data will be obtained. This is still something of a grey area within engineering applications, where accuracies may be overstated or inappropriate methods of validation used.

This exclusive attempt for accuracy, whilst important, can arguably degrade the level of information available within a model and should be carried out in conjunction with analysis relating to the *usefulness* of the models, a rather subjective term but one that seeks to introduce the idea that a model, no matter how accurate, is (in the context of DM) of little use unless it can be deciphered and information extracted from it. It may be argued that every model is incorrect, as it can only represent the data which is used in its creation and hence cannot fully describe the underlying process, and in the research described in this thesis slightly 'less correct' models are accepted if the latent information they contain is more readily available than a 'more accurate' model. In the words of George Box, Professor Emeritus at the University of Wisconsin-Madison, 'all models are wrong; some models are useful', and it is argued that, as we cannot create the 'perfect' model efforts should be made to create a model that adequately represents the process in question whilst remaining interpretable enough to enable information to be extracted or inferred from it.

This perhaps delineates the differences between the DM approach and traditional uses of Machine Learning within engineering, in which case the model is used 'explicitly'[15] and accuracy is quite legitimately considered to be of paramount importance, whereas Data Miners (who may not use the model explicitly as a predictor) consider the interpretability of the model to be of a similar order of importance as its accuracy (Khabaza, 2002). There is however an important proviso to this argument. It is justifiable to accept some compromise in the overall predictive accuracy of the model, but it is still vital that the accuracy is adequately computed and understood; suggesting that allowing accuracy to be

[15] In this context 'explicit' use of models refers to a model being used for one specific purpose, for example prediction or classification, where no inferred knowledge is extracted alongside this primary application.

reduced at the expense of interpretability is not synonymous with suggesting that accuracy is unimportant or that correct validation is not necessary. Information extracted from a model can only be successfully deployed if the degree of accuracy and hence validity of such knowledge can be evaluated to a sufficient degree, and ignorance of the level of accuracy is a recipe for disaster.

Measures of Model Accuracy

The specific measure used to evaluate accuracy varies depending upon the nature of the data under consideration. The use of discrete data, such as that used in DTI analysis, is evaluated simply by counting the number of correctly classified instances and expressing this as a percentage of the total number of instances. With continuous data a different measure is necessary. Common measures include the sum-squared error, the R-squared error and the Pearson product moment coefficient (Howell, 1989). The Pearson measure indicates the degree of correlation between two series of numbers, and can vary between –1 (for a perfect negative correlation[16]) through zero (no correlation) to 1 (perfect correlation). The coefficient of determination, the R-squared value, is simply the square of the Pearson coefficient, and hence gives a value of between 0 and 1, where 0 indicates no correlation and 1 indicates perfect correlation (both positive and negative). In this research the Pearson coefficient is used as it indicates the *direction* of covariance (in the unlikely event of inverse relationship between predicted and actual output), however the R-squared coefficient can be easily compared against a normalised percentage (a percentage expressed as a ratio between 0 and 1) allowing for the accuracies of both DTI and ANN models to be directly compared.

Methods Available for Establishing Model Accuracy

The measure of accuracy is simply a metric, and consideration must be given to how to obtain and interpret that metric. It is necessary to establish how accurately the model has captured the underlying process that the data describes, instead of 'learning the data by rote.' Simple resubstitution involves passing the training set back though the model to

[16] In effect where one parameter decreases at the same rate as another one increases, or, if plotted on a graph, where the line of best fit is a downwards slope

see how well it is predicted, however this tends to give overly optimistic estimates of accuracy. Many validation methods withhold a portion of the training data (known as the validation set) in order to evaluate the accuracy of model created using the remainder of the data, however as Hand et al (2001) point out this acts to reduce the size and richness of the training data. The specific composition of the training and validation set can also influence the computed accuracy, as the sample in each set might not be representative of the entire dataset. Other methods such as cross-validation mitigate against this (Witten and Frank, 2000). Cross-validation divides the dataset into equal sized segments called folds. A model is created using all but one of the folds, and the accuracy is evaluated suing the remaining fold. This process is repeated, using each fold in turn for validation. The accuracies of each fold are then averaged. The model is typically created using the entire dataset, thus utilising the full richness of the available data.

Ten-fold Cross-validation (where the data is divided into ten folds) is widely use and considered the most dependable metric for assessing the proportion of correct classifications (Witten and Frank, 2000), (Hastie et al., 2001), and is experimentally demonstrated by Reich and Barai (Reich and Barai, 1999) to be preferable to the use of separate training and validation data sets.

Methods Selected for Establishing Accuracy

The C5.0 algorithm was used to create the DTI models, and a feed-forward ANN was trained using the back-propagation algorithm (as discussed in detail in section 12.1.4). The preferential method of establishing model accuracy is cross-validation, and hence ten-fold cross-validation was selected for use in evaluating the DTI models. In a departure from recommended practice a validation set was put aside, simply to indicate the form of error[17]. When using classification models it is possible to create a coincidence matrix, whereby a table is constructed listing actual classification against predicted classification and summing the number of instances falling into each category. This gives some indication of which misclassifications are prevalent.

[17] It is possible that some types of error are more significant than others. For example, the medical domain may consider false positives (incorrect diagnoses of healthy people as ill) as much less costly than false negatives (incorrect diagnoses of ill people as healthy)

The nature of training for an ANN model is by nature computationally expensive, where changes to network weightings are performed incrementally in the face of observed training error. The decision was made to limit validation of such models to the use of a separate validation set, as the more accurate cross-validation requires the construction of a separate model for each fold. It was thought the computational expense was not justified for this simple example.

4.3.4 DTI Modelling

The DTI algorithm has few parameters, simplifying the task of modelling. Unlike the ANN algorithm there is no need to specify a topology.

DTI Algorithm Parameters

Modelling within DM is an iterative process, and hence a range of models were created, each with different parameters. These parameters bias towards a more compact or more expansive tree or ruleset, such options being identified as 'Accuracy' or Generality' within the Clementine software package used in this research. The algorithm attempts to ensure that each instance in a given leaf has the same output, however where there is noise in the data this can act to create leaves that contain a single erroneous instance. To mitigate against this the algorithm prunes the tree or ruleset. The 'Accuracy' setting minimises such pruning, however the 'Generality' setting attempts to eliminate those leaves that contain few instances.

The 'Accuracy' and 'Generality' settings are simply broad settings that control only two parameters. These parameters are the minimum number of objects in a leaf and the pruning severity. The algorithm functions by continually splitting the data until each segment of the data contains instances of the same class The minimum number of objects prevents the algorithm from creating too specialised a class by only allowing a segment to be further divided if at least two instances would end up in each resultant class. Upon completion of training the algorithm then proceeds to prune the created tree by combining adjacent segments if there is minimal effect upon the training accuracy. The pruning severity controls the extent by which model accuracy can be compromised in this manner. In this example the 'Accuracy' settings utilise a small minimum number of objects and a low pruning severity, whereas the 'Generality' settings are biased more towards severe pruning and greater numbers of instances in each leaf. The use of the pruning severity and minimum number of objects allows for greater control when

specifying algorithm parameters, and whilst not necessary in this simple case will be utilised in the research described in Chapter 5.

Results of DTI Modelling

Model No	Boosted?	Validation set size	Options	Train accuracy %	Valid accuracy %	CV accuracy %	No of Rules
1	No	20	Accuracy	90	60	48.7	10
2	No	20	Generality	77.5	65	56.2	8
3	Yes	20	Accuracy	98.75	65	56.2	11
4	Yes	20	Generality	95	70	62.5	11
5	No	0	Accuracy	91	N/A	54	15
6	No	0	Generality	82	N/A	59	12
7	Yes	0	Accuracy	100	N/A	59	12
8	Yes	0	Generality	95	N/A	55	7

Table 3 - Results of 1st Tranche of DTI Modelling for Simple Link Mechanism

Table 3 shows the parameters and results of these models. The use of Boosting as a method to improve classification accuracy has been discussed in section 2.9 and in greater detail in Chapter 13, where its propensity to overtrain models in the presence of noise was noted, and in order to assess its effectiveness models were created both using and neglecting Boosting[18]. In the case of this analysis rule sets were created, as these provide a clear metric of the complexity of the model in terms of the number of rules in each model[19].

[18] To recap, Boosting 'seeds' training instances such that an algorithm will create series of models, each one focusing efforts upon correctly predicting the output of those cases the previous model incorrectly classified. The entire model therefore comprises of a series of 'sub-models'. An overall prediction is given by allowing each sub-model to 'vote' and weighting each vote by the overall training accuracy of the su-model in question.

[19] The C5.0 algorithm allows for both trees and rulesets to be created.

The results shown in Table 3 show the influence of the model parameters upon both model accuracy and complexity. The use of generality options (a low pruning severity) tends to slightly reduce the training accuracy, whilst slight increases are seen in both the validation accuracy and the cross-validation accuracy. As expected, the number of rules created by each model is also reduced when generality settings are used (with the exception of models 3 and 4), a detail which can be of use in situations where trained models are too complex to allow easy information extraction – such a concern is more significant where logical rules are extracted, as each rule acts separately and an excess of rules can cloud their interpretation due to excessive detail. As greater emphasis is put on improving the validation and cross-validation accuracies, and simple models are preferable in terms of extracting information, it is therefore suggested that the use of settings favouring generality will produce models.

The use of Boosting acts to increase the complexity of a model by virtue of creating a series of further antecedent models – the entire model becomes not one single tree or ruleset but a range of such structures. If this increase in complexity is considered in isolation, and when the necessity for information extraction is factored in, it could be argued that Boosting is not useful for the purposes of this research. However, there are further considerations that must be taken into account. In Table 3 it may be seen that the Boosted models typically gave a higher accuracy, both training and validation as well as Cross-Validation.

The number of rules only describes the number of rules in the first fold of the Boosting operation, each antecedent model will have separate rules contained within it, and as there are 10 folds to each Boosted models it is clear that the number of rules is of an order of scale higher than for non-Boosted models. In this respect there is a trade-off, the completed Boosted ruleset offers greater accuracy but at the cost of massively increased ruleset size, which has significant issues relating to the comprehensibility of the model. In order to maintain some clarity whilst describing the methods of modelling, for the remainder of this chapter Boosting will be neglected. Boosting will be revisited in Chapter 6, describing the combination of multiple models, will cover this area in greater detail as it is suggested that Boosting may be a useful tool despite the huge increase in model complexity.

The measures put in place to improve accuracy or generality typically have the greatest effect in the presence of noise, which is absent from the data used in the creation of these

models, hence their effect is slight. This comparison is returned to in later sections of this chapter, where noise is introduced, and the effects become more significant.

Consideration of Error in Trained DTI Models

As mentioned previously, the measure of classification error is not a perfect metric for evaluating the performance of a DTI model, and hence manual inspection of the nature of the error of such a model can provide useful information. Coincidence matrices (also referred to as confusion matrices) list the predicted classification against the actual classification for each class, where correct classifications are listed across the diagonal and incorrect classifications are shown in the remaining subsections of the matrix. An example can be seen in Table 4, where the predicted classifications for 80 instances are shown in a problem with 5 possible classes. In this example, it can be seen that of the 20 instances in range C 18 were correctly classified and 2 were misclassified into range B. Such detail is useful in cases where certain types of misclassification are potentially more serious than others, for example in medical cases where it is more serious to incorrectly classify an ill patient as being healthy than classifying a healthy patient as being ill.

		Predicted Class				
		A	**B**	**C**	**D**	**E**
True Class	**A**	15	0	0	0	0
	B	1	13	1	0	0
	C	0	2	18	0	0
	D	0	0	0	15	0
	E	0	0	0	0	15

Table 4 Training Coincidence Matrix for DTI Model

		Predicted Class				
		A	**B**	**C**	**D**	**E**
True Class	**A**	4	1	0	0	0
	B	0	3	1	0	0
	C	0	0	2	0	0
	D	0	0	1	2	2
	E	0	0	0	1	3

Table 5 Validation Coincidence Matrix for DTI Model

Table 4 and Table 5 show the coincidence matrices for Model 4 from Table 3, which is used as an illustration in preference to Model 8 as it includes a segregated validation stage. It can be seen than that the validation matrix contains more errors than the training matrix, as can be expected from any model training with a non-massive dataset, but each of these misclassifications lies in an adjacent range. It is this nature of error that must be considered when deciding if a model is accurate enough to enable information extracted from it to be used.

The delineation of ranges also plays a part in this adjacent misclassification. In certain situations the delineation of ranges is subject to external requirements, for example the maximum velocity may be limited by legislation in which case the value prescribed by this legislation may be used to define the ranges. In the case described here, the selection of 5 ranges with equal numbers of instances in each range was a purely arbitrary decision, and these may not be the most suitable ranges. Ideally, the instances in each range will share similar features and will be distinct from those in other ranges, however if the limits of these ranges are incorrectly chosen there may be some similarity between certain ranges, and the model will struggle to clearly delineate between these two ranges. It is possible to iterate to deduce the most suitable range limits, using those values which provide the greatest accuracy. There appears to be no simple way of deducing the most suitable range limits prior to modelling, as this requires some understanding of the structure of the data, a prerequisite that is considered undesirable and likely to reduce the range of application of this approach.

4.3.5 ANN Modelling

One of the first tasks to consider in an ANN application, and one of the tasks most influential in assuring good network performance, is selecting an appropriate architecture or topology for the network. Too simple a topology will prevent the desired function from being mapped and too complex a topology creates a situation where overtraining becomes significant. The simple example discussed within this section does not present any significant problems either in terms of system complexity or noise within the data (a contributory factor towards overtraining), hence topology selection is not as critical a procedure as for other more complex situations. In light of this, and of a desire to introduce various complexities and developments in a stage-wise manner, the issues relating to topology selection will be addressed in later sections.

ANN Algorithm Parameters

An ANN model is trained using a specific algorithm which may have a number of parameters. In this research the back-propagation algorithm has been selected for use. This algorithm is a gradient-descent algorithm, in that it follows an error gradient in order to minimise the error (and so improve accuracy) over a number of iterative updates. There is potential for the algorithm to continually overshoot the minimum error, or to settle into a local minimum. The back-propagation algorithm can be adjusted by two measures, the learning rate and a momentum term, in order to mitigate against these problems.

The learning rate is a factor that reduces the *magnitude* of weight changes to prevent instability, much in the same way as reductions in time steps can reduce the chances of instability on a simple Euler approximation. Values for the learning rate are typically large early on, where large errors must be traversed quickly and local minima avoided, but reduced at later stages to allow fine minima to be reached. The momentum term acts to include a function of the previous weight change within the new one, thus alleviating oscillation around a solution by preventing subsequent weight updates from counteracting earlier changes, in effect preventing chances of *direction* of weight updates.

The attempt to drive the training error of the network to zero (essentially the ambition of a gradient-descent solution) means that the network can be prone to overfitting or specialisation, where errors within the data are treated as genuine phenomena and the algorithm tries to cater for them within the model. Training is truncated when the training accuracy has not decreased for a given number of cycles, at which point overfitting may already be in evidence. It is possible to truncate training prior to this event. A test dataset may be used to evaluate the performance of the network after each update iteration, in essence performing validation after each step of training. When the test accuracy is seen to decrease it is assumed that overfitting is occurring and training is terminated. This requires further data to be set aside, reducing the amount available for training, and assumes that any decrease in test accuracy is solely due to overfitting. It is also possible to simply prescribe some stopping criteria other than a minimum error. It is possible to stop training after a certain number of cycles or training duration – such a measure is referred to as *early stopping*. These measures can be effective in reducing overfitting in the presence of noise (as indicated in section 4.4), however as these

methods have no theoretical basis the exact criteria to use can only realistically be found by trial and error.

Results of ANN Modelling

Five different networks were considered, each with a single hidden layer in which different numbers of nodes were used. Each network had 9 inputs, corresponding to the 9 parameters which describe the link mechanism, and a single output, the maximum velocity seen during a cycle of the mechanism. A standard back-propagation algorithm was used, and all input data was normalised in order to ensure all inputs are given equal emphasis in training (Hastie et al., 2001). A value of 0.9 was used for the momentum term along with an initial value of 0.3 for the learning rate, which was allowed to decay to 0.01 over 30 iterations as the training approached convergence. To prevent entry into local minima, the lower learning rate of 0.01 was increased to 0.1 and back over 30 iterations. Training was terminated when the training accuracy had not decreased over 100 iterations.

The accuracy of the networks was deduced by plotting the predicted output against actual output for each instance in both the training set and the validation set, and then computing the Pearson product moment coefficient. This was performed for both the training and validation datasets.

Nodes in hidden layer	Training Accuracy		Validation Accuracy	
	R-Squared	Pearson	R-Squared	Pearson
5	0.9999	0.9999	0.9980	0.9990
10	1	1	0.9818	0.9908
20	1	1	0.9424	0.9708
30	1	1	0.9420	0.9705
50	1	1	0.9346	0.9667

Table 6 Results of 1st Tranche ANN Modelling

Table 6 shows the R-squared correlation coefficient and the Pearson metric for both the training set and validation dataset for 5 networks with various numbers of nodes in the hidden layer.

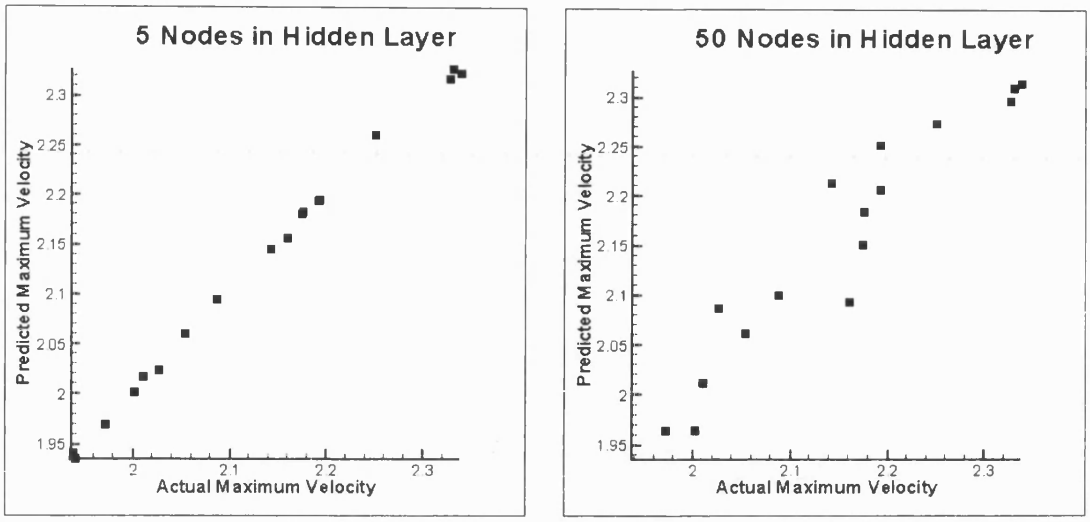


Figure 12 Predicted Against Actual Maximum Speed for 5 and 50 Hidden Node Networks

Figure 12 shows the plots of predicted against actual output for the validation datasets for both the 5 hidden-node network and the 50 hidden-node network. There is evidence of overtraining in the 50 hidden-node network, where the validation accuracy begins to fall as the number of hidden-layer nodes rises, and the system becomes under constrained. This reduction in accuracy is slight, with the Pearson coefficient dropping from 0.999 for a network with 5 hidden nodes to 0.9667 for a network of 50 hidden nodes, suggesting that the level of overtraining in the larger networks is minimal. This is due to the lack of noise in the data, and later sections will discuss the ramifications of excessive network size in the presence of noise.

4.4 Developments – Introduction of Noise

The first tranche of modelling intended to introduce the methods of DM modelling and to describe problems associated with overfitting. This section seeks to identify how these models behave in the presence of noise or error within data, as it is anticipated that such noise will be present in manufacturing data.

In order to evaluate the effectiveness and accuracy of the predictive models in the presence of noise, each parameter, both input and output, was subjected to a pre-

determined amount of random noise. This random noise was generated computationally using the popular Microsoft Excel package(Microsoft, 1994)[20].

4.4.1 Quantification of Random Noise

The anticipated presence of noise in engineering data is argued to be the result of numerous factors, ranging from inaccurate or inappropriate measurement, incorrect data entry through to incorrect post-processing of electronically stored data. The problems of noise or error in data have been addressed in several studies. However these have mostly been in the area of database management where such error is introduced by misinterpretation during the merging of data from dispersed, heterogeneous (individually structured) databases (Hernandez and Stolfo, 1998). At the time of writing there appears to be no metric of data quality or a formal framework to express or quantify noise within data (Kim et al., 2003), and hence there are few means of deducing the amount of noise within engineering data. The two case studies, described in later sections, consider the problems of noise within data and several observations are made regarding the data collated during the more extensive of these two studies.

In this simple example a relatively crude method of adding noise is used, as it is intended merely to demonstrate the phenomena evident in the predictive models in the presence of noise. In the absence of informed methods of simulating noise in engineering data, it was decided to introduce various levels of random noise with increasing severity in an attempt to indicate the degradation in performance of the modelling methods as noise increases. This allows for comparisons between models describing data with different levels of noise, and hence clarifies the effects of such noise.

Case	Random multiplier range	Effective percentage variance
1	0.99 to 1.01	-1% to +1%
2	0.975 to 1.025	-2.5% to +2.5%

Table 7 Extents of Added Noise

[20] All algorithmically-generated random numbers cannot be considered as genuinely random, as there is a deterministic process controlling their value. The bias that this introduces is considered insignificant in the context of this research.

In each case, the complete dataset was multiplied by a matrix of random multipliers. Two different levels of noise were added, as shown in Table 7, where the matrix of random multipliers was populated with individual multipliers strictly limited by the parameters indicated in the random multiplier range column. When the dataset was multiplied by this matrix, the resulting dataset would be subject to a range of random noise as indicated in the effective percentage variance column.

It is argued that random variance is but one possible form of noise, and that errors in the form of bias will also be present in engineering data. For example a measuring device may be poorly calibrated and indicate readings greater than the actual feature in question. It is further argued that the random nature of the noise is perhaps the most problematic of all noise in modelling terms, as there is no pattern to its creation that can be mapped – it is suggested that certain errors may be, to certain extents, predictable, where the causes of the errors are themselves functions of certain parameters. As an example to illustrate this point, consider a tool tip on a lathe. Under high loading forces, this tool tip will deform and the shape of the machined part will therefore have an associated error or deviance from the desired shape, and such a deviance may not be easily measured or quantified. Upon assembly, the completed product may suffer from diminished performance as a result of this manufacturing error. If the tool tip force is known, or factors that may influence tool tip force are recorded, there exists a link, however tenuous, between tip force and performance of the product. This relationship may be difficult or impossible to expressly quantify, however it provides certain levels of information that are not present in truly random noise and may feasibly be used to improve the performance of the model. It is suggested that the inclusion of random noise is the worst-case scenario in terms of error within data, and hence it is argued that this addition of random noise allows for the most robust test possible of the effects upon modelling of noise within data.

4.4.2 DTI Modelling with Noise

Model No.	Boosted?	Options	Validation set size	Training Accuracy %	Validation Accuracy %	Cross-Validation Accuracy %	Number of rules
1	NO	Accuracy	20	90	50	50	14
2	NO	Generality	20	76.25	60	45	8
3	YES	Accuracy	20	100	70	53.7	10
4	YES	Generality	20	88.75	65	55	8
5	NO	Accuracy	0	94	N/A	49	14
6	NO	Generality	0	80	N/A	55	12
7	YES	Accuracy	0	99	N/A	57	15
8	YES	Generality	0	86	N/A	50	8
9	NO	Accuracy	50	92	52	52	11
10	NO	Generality	50	76	50	68	6
11	YES	Accuracy	50	100	52	48	8
12	YES	Generality	50	64	50	42	4

Table 8 Results of DTI Modelling with 1% Noise

Table 8 shows the accuracies of 12 DTI models trained using data with 1% noise, where a series of models were created, using various parameters as for the first tranche of modelling with the exception of models 9 through 12, where 50 instances were extracted for use as validation data.

Model No.	Boosted?	Options	Validation set size	Training Accuracy %	Validation Accuracy %	CV Accuracy %	Number of rules
1	NO	Accuracy	20	90	20	41.2	14
2	NO	Generality	20	76.25	20	40	11
3	YES	Accuracy	20	90	25	45	7
4	YES	Generality	20	80	40	47.5	6
5	NO	Accuracy	0	90	N/A	40	17
6	NO	Generality	0	69	N/A	32	11
7	YES	Accuracy	0	99	N/A	39	11
8	YES	Generality	0	88	N/A	41	10
9	NO	Accuracy	50	82	32	14	9
10	NO	Generality	50	60	48	26	4
11	YES	Accuracy	50	98	42	36	8
12	YES	Generality	50	68	42	30	4

Table 9 Results of DTI Modelling with 2.5% Noise

Table 9 shows the results of DTI modelling with 2.5% noise added. Comparison with Table 8 indicates the reduction inaccuracy with increase in noise, with 1% noise added the cross-validation accuracy varied from 42% to 68%, however this dropped to between 14% and 47.5% with 2.5% random noise.

4.4.3 ANN Modelling with Noise

As overfitting was anticipated, early stopping of training was utilised for a portion of the AN models. In models where early stopping was utilised, the networks were set to truncate training after 5 minutes.

		Training Accuracy		Validation Accuracy	
No of nodes in hidden layer	**Early Stopping**	**R-Squared**	**Pearson**	**R-Squared**	**Pearson**
5	NO	0.9889	0.9945	0.4432	0.6657
10	NO	0.9999	0.9999	0.802	0.8956
20	NO	0.9999	0.9999	0.663	0.8146
30	NO	0.9999	0.9999	0.8768	0.9364
50	NO	0.9999	0.9999	0.7523	0.8674
5	YES	0.8966	0.9469	0.9203	0.9593
10	YES	0.9228	0.9606	0.9689	0.9843
20	YES	0.9589	0.9791	0.9513	0.9753
30	YES	0.9276	0.9631	0.9779	0.9889
50	YES	0.9267	0.9626	0.9765	0.9882

Table 10 Results of ANN Modelling with 1% Noise

Table 10 shows the accuracies of the models created with the addition of 1% random noise. It can be seen that the training accuracies for the networks without early stopping are extremely good, ranging from 0.9945 to 0.9999 for the Pearson coefficient. The training accuracy is lower for those networks where early stopping was used, where the Pearson coefficient ranged from 0.9469 to 0.9791. Taken in isolation, this would indicate that early stopping is detrimental to the training process, however inspection of the validation accuracies lead to quite different results. It can be seen that the accuracies are still maintained at a high level, however those networks that do not utilise early stopping can be seen to have lower validation accuracies and are argued to be exhibiting the onset of overtraining.

The presence of overtraining in the networks created with data subject to 1% random noise has been suggested to be present, where training accuracies exceed validation accuracies in cases where early stopping is not used, however these effects are slight and

the correlation between predicted and actual maximum velocity are still good. In order to demonstrate the effects of overtraining more vividly, the results of networks trained with 2.5% of random noise will be described.

		Training Accuracy		Validation Accuracy	
No of nodes in hidden layer	**Early Stopping**	**R-Squared**	**Pearson**	**R-Squared**	**Pearson**
5	NO	0.9809	0.9904	0.2565	0.5064
10	NO	1.0000	1.0000	0.2837	0.5327
20	NO	1.0000	1.0000	0.2902	0.5387
30	NO	1.0000	1.0000	0.0384	0.1959
50	NO	1.0000	1.0000	0.0663	0.2575
5	YES	0.7477	0.8647	0.5779	0.7602
10	YES	0.7660	0.8752	0.5741	0.7577
20	YES	0.7570	0.8700	0.5372	0.7330
30	YES	0.7166	0.8465	0.5921	0.7695
50	YES	0.7579	0.8705	0.5830	0.7636

Table 11 Results of ANN Modelling with 2.5% Noise

Table 11 shows the results of training with 2.5% random noise. These results clearly show the trade-offs between training and validation accuracy for early stopping. In the case of the first 5 models the training accuracy is approaching 1, whereas the validation accuracy has a Pearson coefficient of between 0.19 and 0.53. In the last 5 models, where early stopping was used, the training accuracy was seen to drop to approximately 0.87 for the Pearson coefficient, although this is tempered by improvements in the validation accuracy which have Pearson coefficient of approximately 0.76. One of the most important features of an ANN is its generality, which describes how well that network will map new data. The validation accuracy is the most accurate method of measuring this, and hence methods that increase the validation accuracy, such as early stopping, are beneficial to accurate training. The Pearson correlation can also be seen to drop for the networks without early stopping as the number of hidden nodes rises, suggesting that increased numbers of nodes are also contributory factors in overtraining. This phenomenon is not replicated in the networks with early stopping, suggesting that increased numbers of nodes are contributory factors only in networks with a predisposition towards overtraining.

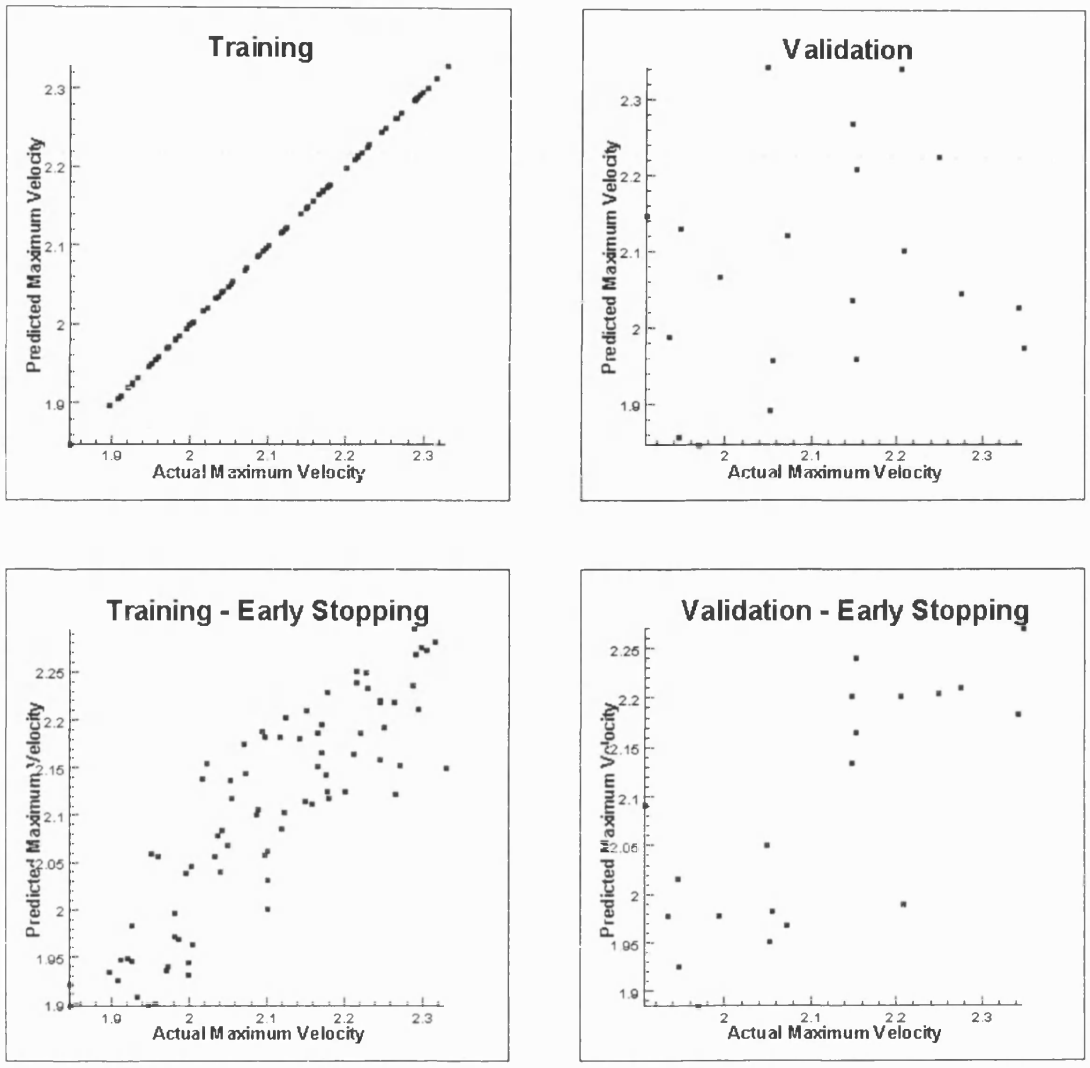


Figure 13 Predicted against Actual Maximum Velocity for 30-hidden node network both with and without Early Stopping, incorporating 2.5% Noise

Figure 13 shows the plots of predicted against actual maximum velocity using both the training and validation data for a 30-hidden node network trained both with and without early stopping. The plots for the network without early stopping indicate overtraining very clearly, as the predictions for the training data are extremely accurate whereas the predictions for the validation data follow little pattern. The plot for the training data for the network with early stopping shows a great deal of scatter, and the scatter is perhaps even greater in the validation data for the same network, although there is a general trend apparent in both. It is unreasonable to expect the fit to be exact as the noise is random in nature and hence impossible to predict, however the plots and the good correlation coefficients indicate that the essential structure of the data has been captured.

4.5 Concluding Remarks

Both DTI and ANN models were seen to successfully map the function relating mechanism geometry to maximum velocity for a simple link mechanism. The data that was used to train these models was generated computationally, and the dataset was limited to 100 instances to ensure that both techniques could be successfully deployed in domains with sparse data of a similar order to that seen in the case study in Chapter 8. The accuracies of the generated models were established via Cross-Validation for the DTI models and via the use of a validation dataset for ANN models. All models were seen to provide high accuracies.

The addition of random noise was seen to reduce the accuracies of both the DTI and ANN models, and the ANN models in particular were seen to suffer from overtraining if efforts were not made to prevent the onset of such a condition. The use of early stopping was seen to be particular effective in this regard, and it is recommended that such an approach be taken where possible.

This chapter sought to identify a method by which DM models could be generated from manufacturing data and their accuracies evaluated. The following chapter builds upon this work, and identifies a method by which the generated models may be used to extract information from such models.

Chapter 5 Extracting and Validating Information from DM Models

The previous chapter established how manufacturing data can be modelled using both the DTI and ANN algorithms. This chapter builds upon such predictive modelling by discussing how useful information can be extracted from the generated predictive models. A novel approach to extracting information from DTI models is introduced, such that the information extracted from DTI and ANN models is of the same form.

The predictive accuracy of DTI and ANN models can be established by careful use of cross-validation, as the predictions of system performance obtained from the models can be directly compared against the known system performance described within the training data. The author is unaware of any previous work that sought to establish the accuracy of *information extracted* from predictive models. It is not possible to evaluate the accuracy of such extracted information in the same manner as for the predictive accuracy, as the training data does not contain any reference information with which to evaluate the extracted information. It is therefore necessary to obtain an evaluative set of information using alternative means.

The results of a separate analytical modelling case study, carried out during the course of a previous research project and whose results were successfully deployed within industry, will be used as a benchmark against which to contrast information extracted from predictive models.

5.1 Structure of Chapter

This chapter seeks to contrast information extracted from predictive models against information obtained from an analytical model. The chapter will begin with a brief coverage of the analytical modelling, as this forms the basis of the comparison. The methods of extracting information from the predictive models will be discussed, where sensitivity analysis is applied to ANN models and the use of a significance metric (which is novel to this research) is proposed for application upon DTI models. Information from both the predictive and analytical models will then be contrasted.

5.2 Analytical Model Analysis

The analytical model was generated during the course of a previous case study, to which the author had no involvement. The approach and results of this case study are briefly covered as there is no publicly available published material which describes this work. The information generated by this analytical case study was applied to good effect within industry, and is considered to be of high accuracy.

The analytical model investigates the kinematic performance of a packaging machine. The mechanism was modelled within the constraint modeller SWORDS as introduced and applied in the previous chapter. The modelling sought to identify how variations in linkage length affected the maximum velocity of the working head, such that those specific linkages whose variation had greatest influence could be identified.

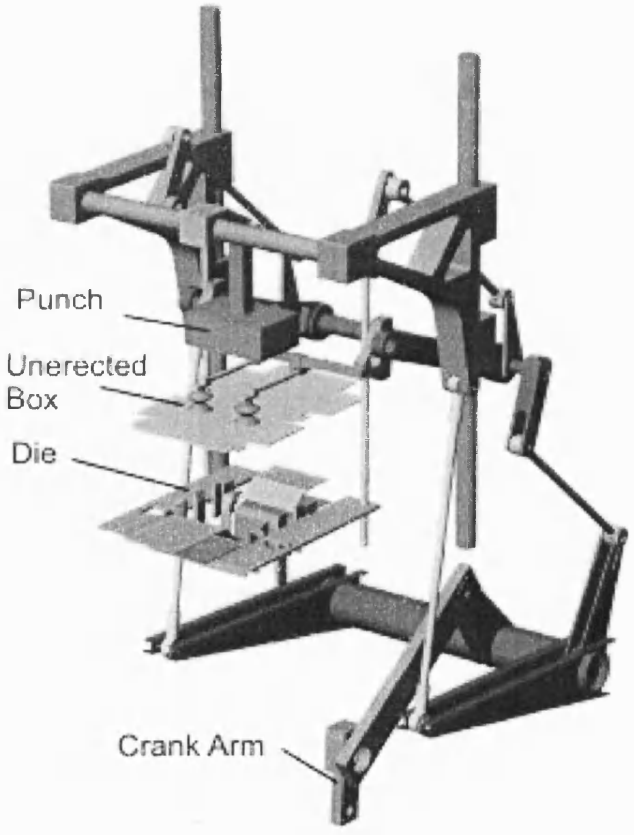


Figure 14 Schematic of Crash Erector

Figure 14 shows a solid model of the mechanism under consideration, where the working head of the mechanism is considered to be the arms that hold the unerected box as opposed to the punch. The mechanism acts by forcing the unerected box through the die

by use of the punch. It was noted in practice that excessive velocity of the working head could cause the arms holding the unerected box to lose grip of the box, resulting in incorrect box expansion.

5.2.1 Sensitivity Analysis of Analytical Model

The geometry of the Crash Erector was entered into SWORDS as a series of linkages of appropriate length, which were then located according to constraints as for the simple linkage in Chapter 4. These constraints defined both the pivot positions and conjunctions of the links.

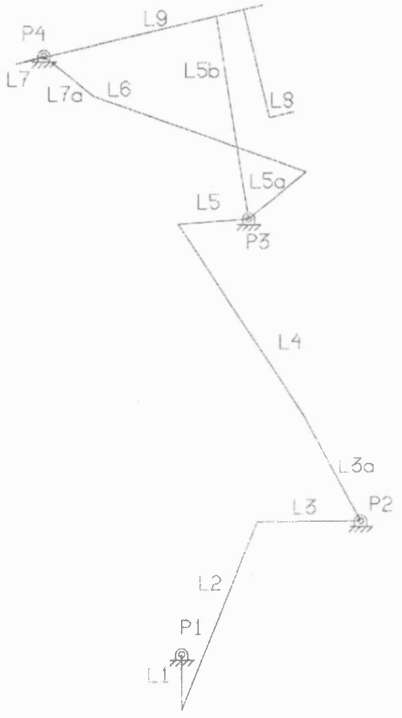


Figure 15 Mechanism Notation (Excluding Punch Subassembly)

Figure 15 shows the annotation used to describe each element within the mechanism, excluding the subassembly mounting and manipulating the punch. This subassembly does not act to link the crank arm to the working head[21], and hence it may be reasoned

[21] The punch subassembly and working head may be considered two separate devices, both powered by the same crank but performing separate activities (the punch forcing the boxes to shape, the working head destacking and locating the boxes ready for punching). In Figure 15 the pivot point P2 is common to both

that it has no influence upon the maximum velocity of the working head. It is included in the analysis, however, to prevent prior knowledge from influencing modelling[22]. The punch subassembly comprises three linkages, identified as L10, L11 and L14.

The mechanism was cycled and the maximum velocity was computed as for the mechanism in Chapter 4. These properties, and the parameters of this base model, were used as a benchmark, representing the ideal arrangement and performance of the mechanism[23]. The sensitivity analysis sought to identify parameters which exert a strong influence upon the velocity of the working head, and this was performed via perturbation of each parameter, whilst maintaining the original settings for the remaining parameters, and recording the variation in velocity of the working head.

Parameter	**Perturbation**
Linkage Lengths	+/- 2% of linkage length
Pivot Points	+/-2% of overall mechanism height

Table 12 Nature of Perturbations for Analytical Model Sensitivity Analysis

Table 12 shows the form of perturbation used for each parameter. These values are arbitrary, but represent a feasible value for variance for each parameter that might be seen in practice. The percentage variation in length for links is simple to implement, however the pivot point positions are relative values as opposed to absolute for the link lengths, and hence a different measure is required for the perturbation of pivot point. This is achieved by perturbing the pivot points by a percentage of the overall mechanism height, giving some proportionality to the perturbation and ensuring that each pivot point is

---

the punch and working head, however all linkages beyond this point (i.e. L3a through to L9) are not connected to the punch.

[22] A justification for using DM in preference to other methods of data analysis is that DM has no prerequisite for prior knowledge. For this reason prior knowledge will not be acted upon in this illustrative example.

[23] As the base parameters represent the mechanism 'as designed', it is reasonable to assume that the performance of such a system is optimum. The analysis seeks to indicate how performance changes as the system deviates from this as designed arrangement.

subjected to equal perturbation. It is highlighted here that the pivot point perturbations comprise 4 separate tests, as the perturbations were carried out for both positive and negative perturbation for both the X- and Y-components of the Cartesian coordinates for each pivot point.

5.2.2 Results of Sensitivity Analysis of Analytical Model

Parameter	Change in Velocity			Rank
	Positive Perturbation	Negative Perturbation	Summed Magnitude	
L5b	30.1	-22.8	**52.9**	**1**
L3	-10.5	-5.5	**16**	**2**
L4	-7.1	-7.6	**14.7**	**3**
L2	2.3	-6.9	**9.2**	**4**
L3a	1.5	-6.5	**8**	**5**
L7a	4	-3.9	**7.9**	**6**
P2_y	-3.6	-3.8	**7.4**	**7**
L1	2.4	-4.8	**7.2**	**8**
L6	0.6	-5.7	**6.3**	**9**
P4_x	2.7	-2.9	**5.6**	**10**
L5	-2.5	2.6	**5.1**	**11**
P3_x	-4	1	**5**	**12**
P4_y	-2.2	2.2	**4.4**	**13**
P3_y	1.6	-2.1	**3.7**	**14**
P1_y	-1.7	-1.8	**3.5**	**15**
L5a	0.9	-1	**1.9**	**16**
L7	-0.8	0.7	**1.5**	**17**
P2_x	-0.9	0.4	**1.3**	**18**
P1_x	0.2	-0.4	**0.6**	**19**
L3b	0	0	**0**	**(20)**
L9	0	0	**0**	**(20)**
L10	0	0	**0**	**(20)**
L11	0	0	**0**	**(20)**
L14	0	0	**0**	**(20)**

Table 13 Results of Sensitivity Analysis of Analytical Model

The magnitudes of the changes in velocity caused by each perturbation were recorded. As each parameter was subjected to perturbation in both a positive and negative

direction, two records were generated for each parameter. The absolute values of these records were summed, and the parameters were ranked according to this summed magnitude.

Table 13 shows the results of the analytical model sensitivity analysis. This ranked list is the most important result of the analytical model analysis, as it indicates clearly which parameter exerts the greatest influence upon maximum acceleration when subjected to perturbation. This information was used to great success within industry, and hence is argued to be an accurate representation of the actual behaviour of the mechanism under analysis. This ranked list will be used as the benchmark for the following DM analysis of the same mechanism, thus providing a means of validating the information extracted from the DM models.

5.3 Creating Data for DM Analysis using Analytical Model

The DM modelling, in essence, looks to replicate the analytical modelling, and provide information (in the form of the ranked list) that agrees with the analytical model. The analytical model was used to provide a dataset for DM analysis that was considered to be similar to that which would be collated during a genuine manufacturing operation. This was achieved in a manner identical to that proposed in Chapter 4, where the parameters were assigned tolerances and allowed to randomly vary within this tolerance, in effect replicating a parameter subject to manufacturing tolerance. The tolerance was set at the same values as used for the analytical modelling in order to attempt to minimise variations in method between the two modelling approaches. These values were as quoted in Table 12, which were +/-2% for the absolute values of linkage length and, in the case of the pivot point positions, +/-2% of the height of the complete mechanism. The variance between these tolerances was random.

A number of different mechanisms were created in this manner, and the acceleration of the mechanism was evaluated for each mechanism. In total 1000 mechanisms were created, resulting in a dataset comprising the 24 parameters (16 link lengths and both Cartesian components of the 4 pivot point positions) and 1 output (the maximum acceleration), for which there were 1000 instances.

5.4 DTI Modelling

As the DTI algorithm is a classification scheme, the output data (in this case the maximum acceleration) must be split into appropriate ranges. This is discussed in the following section. Following this, the algorithm settings are discussed. A range of DTI models is then created using different settings and data compositions, and the results compared. Information is extracted from the most accurate DTI model for comparison with both the results of ANN modelling and from the analytical model later in the chapter.

5.4.1 Appropriate Range Selection/Class Distribution

The previous chapter introduced issues concerning the selection of suitable boundaries to use for defining the separate ranges. It was suggested that such a decision could be made either by consideration of groupings within the data or by ensuring that there is a approximately equal volume of data in each range.

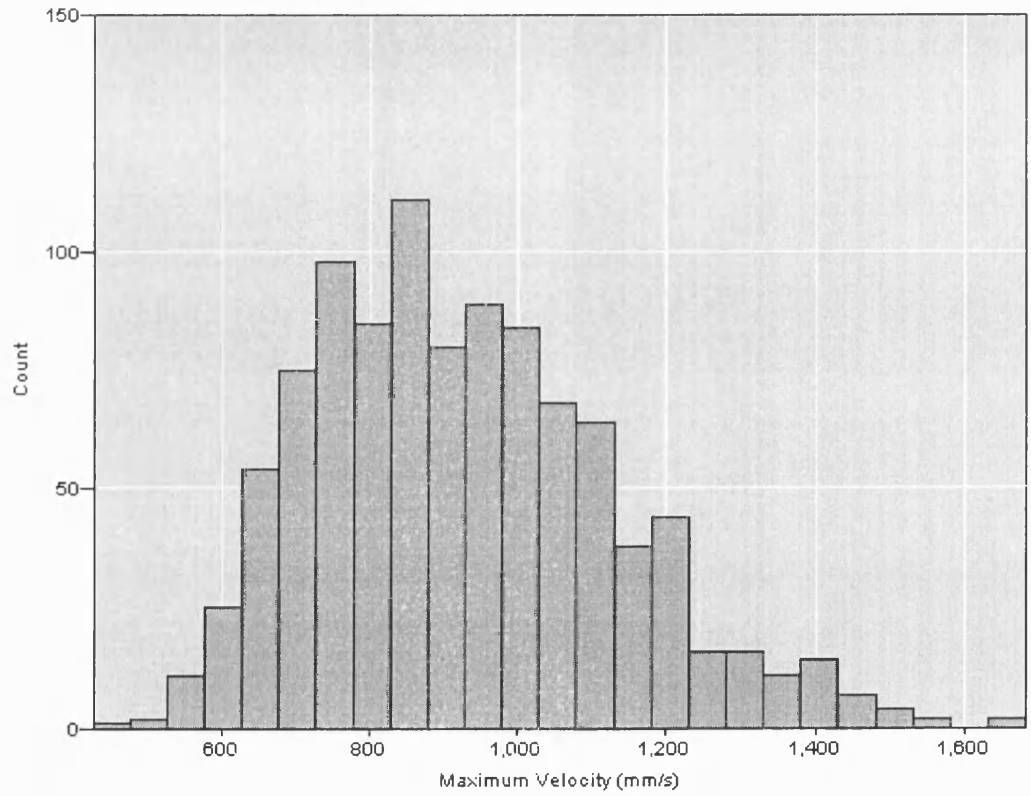


Figure 16 Histogram of Maximum Velocity for Crash Erector

The distribution shown in Figure 16 follows what may be described as a normal distribution, with no clear groupings of any sort. In this respect, the boundaries of the ranges will be selected to ensure an approximately equal distribution of data in each range. Initially the data will be split into 4 ranges with further trials to deduce if 3 ranges will give more accurate models

5.4.2 Algorithm Settings Selection

The DTI algorithm has fewer settings to manipulate than the ANN algorithm, and there are well-established default settings for these settings (Opitz and Maclin, 1999). The two settings of note both regulate how aggressive the algorithm seeks to prune the created ruleset or tree, but do so in different ways. C5.0 is a commercial algorithm with attendant laws preventing reverse engineering, hence the exact implementation of these settings has not been disclosed. However the underlying theory may be understood by examining the freely-available and well-documented precursor to C5.0, the C4.5 algorithm (Quinlan, 1986).

The pruning severity controls the extent to which the created tree or ruleset is pruned. The initial tree will accurately classify all of the training data, in effect overtraining the tree as any errors within the data will be encapsulated within this tree. There will also be a large number of branches, as all permutations within the dataset have to be addressed. In order to reduce this overtraining and remove these superfluous branches, a pruning process seeks to identify branches that can be removed without excessively reducing overall accuracy. The pruning severity dictates the reduction in accuracy that will be allowed for this branch removal, where a greater value will allow for more extreme pruning with a greater reduction in accuracy. The parameter controlling the minimum number of records per child branch seeks to limit the creation of new branches, specifying a minimum number of cases within the training dataset that must be classified by that branch or rule. In practice, the branch limitation controlled by this parameter is retroactive, a 'complete' tree is created and branches with insufficient numbers of rules are removed by a process known as subtree raising (Witten and Frank, 2000).

5.4.3 Results of Modelling

Model No	No. of training instances	Prune Severity	Min records per child branch	No of output classes	Training Accuracy (%)	Validation Accuracy (%)	Cross-validation Accuracy (%)
1	500	75	2	4	95.4	46.71	48.6
2	500	90	2	4	94.8	46.91	50.8
3	100	75	2	3	96	55.27	66
4	100	90	2	3	96	55.27	63
5	200	75	2	3	93.5	57.8	58.5
6	200	90	2	3	90	58.8	51
7	500	75	2	3	95.2	58.28	65
8	500	90	2	3	93.4	60.08	63.2

Table 14 Results of DTI Modelling

Table 14 shows the results of the DTI modelling. Four pairs of models were trained, the first pair attempted to classify the maximum head velocity into 4 ranges, with limited success. An improvement in accuracy was seen in the remaining models where the number of ranges was reduced to 3, although this improvement must be considered against a reduction in the granularity of the prediction. Further discussion of these results will exclude these first two models from consideration, as the further 6 models enjoy significantly better accuracy.

The first of the two models in each pairing were created using the standard[24] algorithm settings, whereas the second model in each pair had a greater pruning severity in an attempt both to ensure a simpler and more comprehensible structure to the model and to reduce overfitting. The second setting, the minimum number of records per child branch, was left unchanged as to a certain extent it replicates the effect of increasing the pruning severity. Of the 3 pairs of models that were created with 3 classification ranges, the

[24] As discussed earlier, the exact implementation of subtree raising and pruning severity is has not been made public. The standard settings comsist of a pruning severiity of 75 and a minimum number of records per child branch of 2, as is default in the Clementine V7.0 Data Mining software package.

number of training instances was varied from 100 to 500. As well as ensuring that a feasible model could be constructed from limited quantities of data, this also takes advantage of the DTI algorithm's instability and sensitivity to data sampling, where perturbing the training data results in changes to the model. It is then possible to investigate whether models of consistent accuracy are created from differing data samples.

It can be seen that the validation accuracy increases as the number of training instances is increased, a result that is perhaps to be expected as the larger data set will suffer from fewer omissions within the data. The DTI algorithm learns by considering which patterns or circumstances had contributed to certain classifications within previous examples, and hence it is possible that smaller datasets will have a skewed content that may not include exemplars of each feasible set of circumstances. As the dataset size increases, this becomes less of a problem, and hence model accuracy (as indicated by validation accuracy in Table 14) rises. It is noted that the validation accuracy dropped from 60% to 55% (models 8 and 4 respectively) when the training data size was reduced from 500 to 100 instances, and it is suggested that this drop in accuracy of 5% for models created with small datasets is acceptable when compared to the accuracy of models created with large data volumes.

When considering each pair of models separately, it is noted that the models created using a less aggressive pruning severity had improved Cross-Validation accuracies with the exception of the first two models, which classified the maximum acceleration into 4 ranges. It is suggested that the data used in this study is not subject to significant noise, and hence measures designed to prevent overtraining (such as increasing the severity of pruning) are not necessary and reduce the accuracy of analysis by means of enforcing an overly simplistic structure to the model. The use of methods of reducing overtraining will be further addressed in section 6.4.9.

5.4.4 Information from DTI Model

In the simple mechanism example of the previous chapter the level of detail of information extracted from the DTI models was minimal, primarily as there was little information that could be verified. In this example, the results of the sensitivity analysis provide a useful comparison that allows the information to be critically evaluated. In

light of this, it is feasible to enter into a much greater level of detail in terms of information extraction.

DTI models have the benefit of transparency, where the structure of the model is presented either in the form of a tree or as a series of rules. Such rules can be extremely useful, forming a useful set of heuristics for designers. However such rulesets are merely an alternative form of output for the DTI algorithm and in their raw form cannot immediately be implemented or considered as useable information. In light of this, a second method is proposed to work alongside rule extraction, which aims to quantify the significance of parameters according to the frequency and position of appearance within the tree. A ranking of the extracted rules will allow a much greater focus to be applied to specific parts or rules of the generated rulesets, removing relatively spurious information and providing only rules that accurately cover a large number of the training data.

The most accurate DTI model seen in Table 14 is model 8, with a training accuracy of 93.4%, a validation accuracy of 60.08% and a cross-validation accuracy of 63.2%. It was created using 500 training instances, classifying into 3 ranges with algorithm parameters set for generality rather than accuracy. The size of the created tree prevents it from being displayed in complete form, rather a reduced version is shown in Figure 17 where lower branches have been rolled up allowing only the higher branches to be seen. This model classifies the maximum velocity of the head into 3 ranges, A, B and C, where A describes the lowest values for maximum acceleration and C the highest.

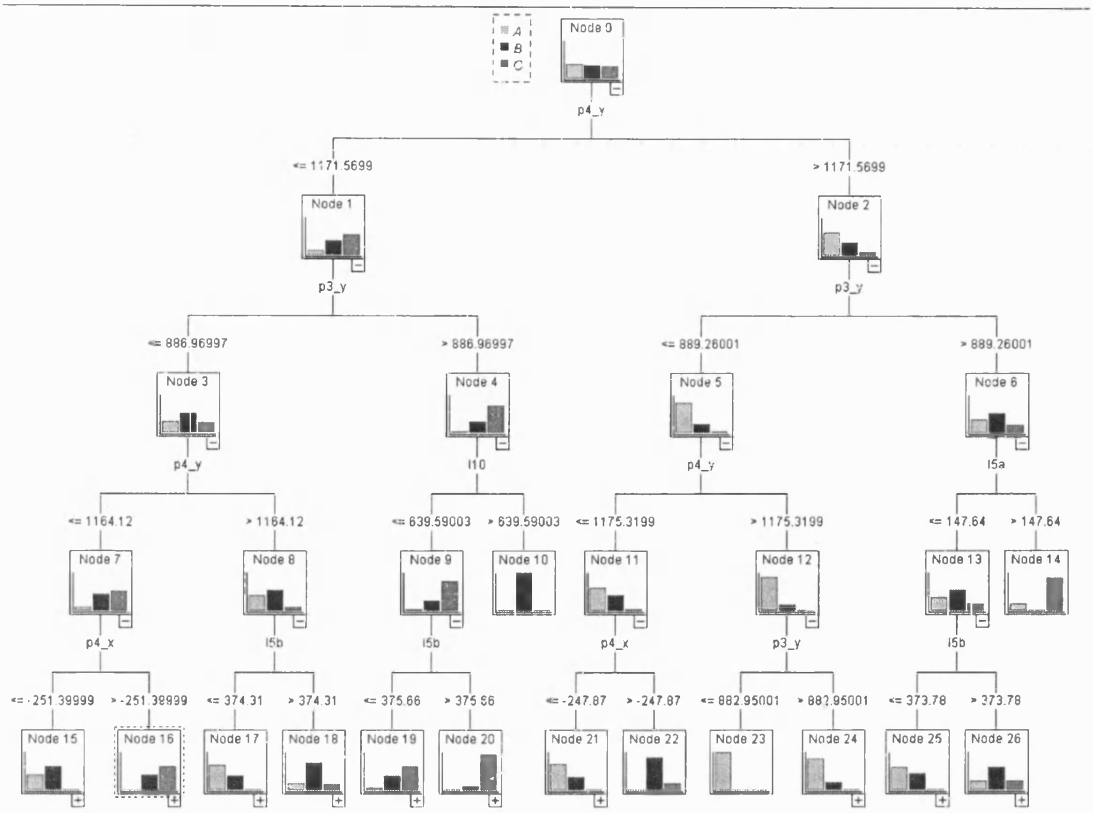


Figure 17 Reduced DTI Model of Model 8

In terms of extracting information we are more concerned with these higher branches, as this is where the major decisions are made regarding partitioning the data – the idea of information gain, used in the identification of parameters to use at the root node, suggests that this parameter and the value used for the split will result in the largest division of data within the dataset. The tree shown in Figure 17 indicates the proportion of each class within the training data at each node, where there are simple histograms showing the distribution of the three classes A, B and C (where A is low maximum velocity and C is high maximum velocity). It can clearly be seen that at the second level nodes (the two nodes immediately succeeding the root node) the node to the left has a distribution biased towards the upper ranges (B and C) whereas the node on the right hand side has a distribution biased towards the lower ranges (A and B). This suggests that the parameter used at the root node is successful at dividing the data into two groups with significant differences, an interesting result in terms of indicating influential parameters. The remaining branches fine-tune this initial coarse division and hence also present useful information, although at a higher level of resolution and a lower level of generality – they are effectively dividing subsets of the original dataset that have similar characteristics. In

this respect, more importance is given to parameters at higher nodes as they are responsible for greater division of the data and act to divide the entire dataset.

5.4.5 DTI Significance Metric

A simple method is proposed to accredit each input parameter with a significance metric indicating the degree of influence that parameter has on the output. For each parameter, add 1 to the significance metric if it appears in the root node, 0.5 (1/2) if it appears in the second-level nodes, 0.25 (1/4) if it appears in the third-level nodes, and so on until each split within the tree has been examined.

Parameter	Additions to Metric	Summed metric
P4_y	1 + ¼ + ¼	1.5
P3_y	½ + ½ + 1/8	1.125
L5b	1/8 + 1/8 + 1/8	0.375
L10	¼	0.25
L5a	¼	0.25
P4_x	1/8 + 1/8	0.25

Table 15 Significance Metrics for DTI Model 8

The example seen in Figure 17 is of a truncated tree, however this serves as a useful example to demonstrate the function of the proposed measure of parameter significance – in actual use, the entire tree would be used in the computation of these metrics. Table 15 shows the metrics obtained for the truncated tree in Figure 17.

Parameter	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8	Level 9	Summed Metric
P4_y	1		2		2		1			1.640625
P3_y		2		1			1	1		1.1484375
L5b				3			1	1		0.3984375
P4_x				2	2			1		0.3828125
L10			1		1	2				0.375
L5			1		1	1	1		1	0.36328125
P3_x						3	2			0.125
L7					1			3		0.0859375
L11					1			1		0.0703125
L9					1					0.0625
L7a					1					0.0625
L3a						1		1		0.0390625
L4						1			1	0.03515625
P2_x						1				0.03125
L3						1				0.03125
L8							2			0.03125
L3b							1			0.015625
P2_y								1		0.0078125

Table 16 Information Content of Decision Tree

Table 16 shows the result for the full Model 8 tree, as expanded from Figure 17. A strong diagonal can be seen, indicating that those parameters that are most significant in the upper nodes of the tree become less significant in the lower branches. This also suggests that it is important to consider the entire tree, as the distribution of parameters within the nodes varies with depth. In this respect the extraction of information could be skewed towards the upper nodes if the tree is truncated.

5.5 ANN Modelling

The difficulty in identifying and specifying suitable parameters for use with ANN modelling suggests that a range of model be created and the most accurate selected from that range. There are 4 factors that require consideration, which are (in no order of importance) the amount of training and validation data, the use of early stopping, the topology of the network and the training algorithm. The amount of training data may be

considered to have an effect similar to that discussed in the DTI modelling in section 5.4.3, where the vagaries of a limited sample can influence the accuracy of the modelling. Early stopping seeks to prevent overtraining by using a small sample of the training data as a test set, and evaluating how accurately the test set is modelled by the network after each iterative stage of training. Where the test accuracy is noted to be reducing, it is concluded that overtraining is evident, and either training is truncated or a network setting (such as learning rate) is adjusted. The topology of the network also plays a part in overtraining, as discussed in section 4.4.3.

The final listed factor, the training algorithm, requires some explanation as the algorithms used in this research are not direct implementations of those discussed in the literature review but are adaptations used with the proprietary DM software Clementine (SPSS, 2002). This software provides facilities to use 6 different algorithms, of which two, the 'quick' algorithm and the 'exhaustive-prune' algorithm, are used here. The quick algorithm is arguably the most basic and uses the standard back-propagation algorithm during training, alongside facilities such as momentum and weight decay that assist in avoiding local minima and in reducing oscillation during learning. The exhaustive prune method seeks to reduce a large network topology to an optimum size by driving certain connection weights towards zero, at which point the connecting node is effectively deleted from the topology. This pruning method is included simply to indicate where performance gains might be found by using a method to automatically deduce parameters and topology. Such methods take advantage of the iterative nature of ANN training to adjust the characteristics of the network whilst the network is undergoing training, allowing for a variety of different parameter combinations to be tested and optimised along with the network weights. It is argued that in the absence of any information that might guide the specification of network parameters and topology, for example previous successful analyses, such automated methods provide an extremely useful tool.

5.5.1 Results of Modelling

Model No	Train Data amount	Algorithm	Test set %	Topology - input, hidden1, (hidden2,) output		Training Time (mins)	Accuracy			
							Train		Valid	
				start	end		R-Squared	Pearson	R-Squared	Pearson
1	100	quick	0	22,20,10,1	22,20,10,1	5	1.0000	1.0000	0.6670	0.8167
2	100	quick	10	22,20,10,1	22,20,10,1	5	0.8494	0.9216	0.7832	0.8850
3	100	quick	0	22,20,1	22,20,1	5	1.0000	1.0000	0.6625	0.8139
4	100	quick	10	22,20,1	22,20,1	5	0.8625	0.9287	0.7883	0.8879
5	200	quick	0	22,20,10,1	22,20,10,1	5	1.0000	1.0000	0.5333	0.7303
6	200	quick	10	22,20,10,1	22,20,10,1	5	0.8124	0.9013	0.7986	0.8936
7	200	quick	0	22,20,1	22,20,1	5	1.0000	1.0000	0.5875	0.7665
8	**200**	**quick**	**10**	**22,20,1**	**22,20,1**	**5**	**0.8624**	**0.9286**	**0.7995**	**0.8941**
9	500	quick	0	22,20,10,1	22,20,10,1	5	0.9972	0.9986	0.5847	0.7646
10	500	quick	10	22,20,10,1	22,20,10,1	5	0.8443	0.9189	0.7962	0.8923
11	500	quick	0	22,20,1	22,20,1	5	0.9959	0.9979	0.4262	0.6529
12	500	quick	10	22,20,1	22,20,1	5	0.8299	0.9110	0.7778	0.8819
13	500	ex-prune	0	22,30,20,1	22,30,20,1	600	1.0000	1.0000	0.5362	0.7323
14	500	ex-prune	10	22,30,20,1	13,30,16,1	23	0.9092	0.9535	0.6965	0.8346

Table 17 Results of ANN Modelling

Table 17 shows the accuracies obtained by the ANN models when using different values for the volume of training data, training algorithm and topology. Different stopping criteria were also used. In cases where a separate test data set was used (the precise quantity of data used for testing indicated as a percentage of training data) the models are effectively arranged in pairs, with an identical model created without the use of a test set. The topology is defined by listing the numbers of nodes present in the input, hidden and output layers respectively. A number of models used only one hidden layer, hence there are three values for topology, however those created with 2 hidden layers have 4 values for topology. When using pruning, the number of nodes is altered by the training algorithm and so the number of nodes at the start and end of training is given.

It can be seen that the validation accuracy is significantly greater for those models that utilised a test dataset, at the expense of training accuracy. This is indicative of

overtraining, where the network attempts to describe the training data by rote at the expense of describing the patterns within the data. Consideration of the training algorithm shows how prone pruning methods are to overtraining, where the validation accuracy of the model without a test dataset had the second-lowest validation accuracy of all models. When a test set was used, however, the accuracy compared favourably with other models. It is noted that the model with no test set (model 13) finished with the same topology as it was initially (and temporarily) specified. The creation of the final model at termination of training uses the topology upon which the greatest accuracy was noted, and in this case the initial topology was noted to give the most accurate model. This is not to say that this is the optimum topology, all that can be concluded is that the act of pruning did not result in an improvement in accuracy.

An interesting pattern is noticeable when considering the models created with different topology. It is noted that, where 100 training instances were used, the models created with only 1 hidden layer had a higher validation accuracy than those created with 2 hidden layers. Where 200 training instances are used, the validation accuracies of the models with the two different topologies were much closer, and when 500 training instances were used the model with the 2 hidden layers was seen to have greater validation accuracy. It is suggested that the models created with 100 instances would have been prone to overtraining, due to a lack of complete coverage of all circumstances in the data as discussed in section 5.4.3, and hence a simpler topology would have acted to curtail overtraining. Where the coverage of the data increases with increased data volume, this pattern reverses and the more complex topology allowed the relationships within the data to be mapped.

It is suggested that prevention of overtraining is crucial for ANN modelling, for which a test set is necessary although topology also plays a significant part. The use of pruning did not lead to significant improvements in accuracy, however of those models created with a static architecture it was noted that those with a simple topology performed well with small datasets and those with a large topology performed well with large datasets. It was not anticipated that dataset size would influence topology, however it is argued that the smaller datasets covered a smaller range of circumstances than the larger datasets, resulting in overtraining when applied to more complex topologies. In this example the lack of noise within the data is argued to have reduced the necessity for rigorous topology

optimisation, however it is suggested that other analyses will require some from of automated or informed topology specification.

5.5.2 Information from ANN Model

The underlying principles of information extraction from an ANN model using sensitivity analysis have been discussed in section 2.7.1, however they will be briefly recapped. An indication of the degree of influence each individual input parameter has on the output of a network can be obtained by tracing the propagation of the effects of each input through to the output. This can be carried out in two ways, the first requires that the individual weights are analysed to compute the 'spread' of an input's influence, whilst the second method manipulates each input parameter in turn and allows the effects of these perturbations to naturally filter through the network to the output, thus indicating directly the influence of each input upon the output. The second method is seen as more practicable, as no investigation into the structure of the network is required, and is most typically used in practical problems. This is the method that will be employed within this research.

The specific algorithm used within this research is that which is contained within the Clementine DM software package (SPSS, 2002), for which greater detail is given by Watkins (1997). The algorithm considers each input parameter in turn, and identifies the maximum and minimum values that are seen for that parameter within the dataset. Three further values are computed at 25%, 50% and 75% of the range between the minimum and maximum. Each instance within the dataset is then passed through the network in turn, and the parameter in question is held at each of the pre-computed 5 values and the output of the network recorded at each of these 5 values. The variation in output seen across these 5 values, in effect the greatest difference between the 5 output values, is recorded. This process is repeated using each instance in the dataset. The variations in output is then averaged across all of the instances. This process is then repeated for each parameter in turn. By comparing these normalised variations for each parameter it is possible to deduce which parameter causes the greatest change in output, equivalent to deducing which input parameter the output is most sensitive to. This form of analysis has the advantage over the more simplistic form seen in the analytical model of ensuring that there is a degree of realism in the perturbations, where the variations seen do not exceed

the values seen in practice, and hence the difficulty seen in selecting a value for perturbation is removed.

P4_y	0.351693
P3_y	0.230426
L5b	0.189777
P4_x	0.0971839
P3_x	0.081283
L3	0.0447659
L2	0.0401943
L1	0.0340102
L6	0.0245493
P2_x	0.0233536
L9	0.0228926
L3a	0.0217417
L5	0.0178251
L7	0.016934
L10	0.0163763
L11	0.0159453
L4	0.0153437
L7a	0.014781
L5a	0.0144411
P2_y	0.0127395
L3b	0.00991063
L14	0.00696565

Table 18 Information from ANN Model Number 8

The results of the ANN sensitivity analysis can be see in Table 18. The parameters defining pivot position have the prefix 'P' whilst those defining link lengths have prefix 'L'. These results will be discussed alongside those from the DTI modelling in the following section.

5.6 Comparison of Extracted Information

Key	
Higher rank in analytical model	Lower Rank in analytical model

Analytical Model Sensitivity Analysis		ANN Sensitivity Analysis		DTI Significance Metric	
L5b		P4_y		P4_y	
L3		P3_y		P3_y	
L4		L5b		L5b	
L2		P4_x		P4_x	
L3a		P3_x		L10	
L7a		L3		L5	

Figure 18 Comparison of Extracted Information from all Modelling Approaches

Figure 18 shows there are significant differences between the results of the analytical model analysis and of both the DTI and ANN analysis. The most significant difference relates to the presence of the pivot points as two of the most significant parameters within the DTI and ANN analysis, whereas referral back to Table 13 indicates that only 1 pivot point parameter appeared in the 10 most significant parameters within the analytical model analysis (P2_y is the 8th most significant parameter).

5.6.1 Removal of Pivot Point Information

Key					
Higher rank in analytical model			Lower Rank in analytical model		

Analytical Model Sensitivity Analysis		ANN Sensitivity Analysis		DTI Significance Metric	
I5b		I5b		I5b	
I3		I3		I10	
I4		I2		I5	
I2		I1		I7	
I3a		I6		I11	
I7a		I9		I9	

Figure 19 Comparison of Extracted Information from all Modelling Approaches, excluding Pivot Points

Figure 19 shows the comparison of extracted information without consideration of the pivot point information. It can be seen that the three approaches each list the same parameter as the most significant. The ANN approach appears to give closer agreement with the analytical model investigation, where three of the four most significant parameters are common to both analyses. In the case of DTI, the agreement does not extend beyond the most significant parameter. It is suggested that this is due to both the simplicity of the method of DTI information extraction and due to the fact that the methods of analysis of both the analytical model and ANN methods shared many similarities, relying upon artificial perturbation of a given parameter in isolation to effect a change in output. The next chapter will propose a method of improving the DTI analysis, as it is suggested that the ANN analysis offers significantly better performance at present.

There is no simple explanation as to why the pivot point information appears within the information extracted from the DM models but not from the analytical model, all that can be suggested is that the artificial manipulation of the pivot points in the generation of data

for DM analysis was of a different form to the perturbations engineered during the analytical model sensitivity analysis. This is argued as the information regarding linkage lengths has good consistency between the two Machine Learning approaches, hence it is argued there are genuine patterns within the data.

5.7 Concluding Remarks

The means of extracting information from a DM model were expanded, where the information is extracted in the form of a ranked list indicating the parameters that exert the greatest influence upon the performance of an artefact.

The information extracted from the DM models was seen to agree with the information from the industrially-validated analytical model with respect to the most significant parameters, with the caveat that the pivot point parameters were assigned greater significance within the DM models than within the analytical model. The consistency between the DM models suggests that both the DTI and ANN models are revealing a genuine trend within the data. It is hence argued that either the method of perturbation within the analytical model sensitivity analysis or measure of tolerance assigned to the pivot points during generation of the DM dataset are unsuitable or suboptimal.

The accuracy of the DTI modelling was seen to increase with a reduction in the number of classes that the model divided predictions into, arguably as the predictions would therefore be more coarse. It was also noted that aggressive pruning led to overtraining, or an improvement in the training accuracy of the model at the expense of the more meaningful and representative validation accuracy.

The DTI algorithm was seen to give information that agreed with the analytical model only for the single most influential parameter, with little correlation in the remaining list. Whilst it is argued that this is useful information, it is suggested that it is necessary to improve on this performance if the approach is to be usefully deployed. It is suggested that the DTI algorithm is unstable, where small changes in the composition of the dataset can cause significant variations in model structure and, by extension, information content, and hence in Chapter 6 a method of combining information from multiple DTI models is proposed to alleviate this instability.

It was noted that the use of simple ANN topologies resulted in accurate models when using small datasets, but when the dataset size was increased a more complex structure

gave the best results. This was argued to be due to the data containing increasing detail, or covering a wider set of circumstances. It was noted that each ANN model was prone to overtraining to a certain extent, and the use of a test set to effectively evaluate validation accuracy throughout training was noted to reduce the onset of this phenomenon.

Chapter 6 Combining Information from Multiple DM Models

DM modelling typically results in the construction of a range of models, which tend to iterate towards a perceived optimum solution as an understanding of the most suitable algorithm settings is developed. The previous chapter indicated how to extract information from one model in isolation, and hence the most suitable model for this task must be selected from the range of models. This was achieved by selection of the model with the greatest validation accuracy.

It is argued that the selection of the most accurate model does not guarantee the model is either the best representation of the underlying process or, by extension, will provide the most accurate information. It is conceded that a model with markedly higher validation accuracy would be more useful in later analysis, however it was noted in the previous chapter that many models had similar levels of accuracy and hence difficulties may arise in justifying the selection of one model at the exclusion of all others.

It is suggested that models with similar levels of accuracy cannot be readily segregated according to accuracy or suitability for further analysis. The data used in validation is a sample of the complete dataset, and hence may be skewed, and the dataset in itself is simply a representation the underlying process, and hence suffers from attendant errors and omissions. When comparing models with similar accuracies it should be noted that the indicated model accuracy is a function of the specific data sample used for validation as well as how accurately the model maps this data. It is thus difficult to judge whether the increase in accuracy seen in one model is the result of better representation of the underlying process, or of the nature of the data used in training and validating that particular model. In this respect the model with the greater accuracy might not actually be a better representation of the underlying process, and hence selection of the most suitable model for subsequent information extraction is thus somewhat subjective.

This subjectivity also occurs where a single prediction is required from a range of models, and methods have been proposed to combine the predictions from a range of models into one unifying prediction (such as Boosting, Freund and Shapire, 1997, and

Bagging, Breiman, 1996), thus removing the necessity of judging which model is likely to give a better prediction. This chapter extends such methods of combining predictions to combining information, and seeks to use the logic of the methods of prediction combination to guide the development of the methods of information combination.

The previous chapter identified the methods of extracting information from DM models, where information was presented as a ranked list indicating which parameters of an artefact exerted greatest influence upon performance of the artefact. The information from the DTI model was seen to have relatively weak agreement with information from an industrially-validated analytical model, and this was suggested to be due to the vagaries of the specific model. A method of combining information from numerous DTI models is proposed, based directly upon the method of Boosting. This is treated in isolation, as the application of Boosting to DTI (specifically the C5.0 algorithm) results in the generation of a range of 'sub-models' or folds within an overall model, where the folds are Decision Trees in their own right, and hence combination of the information contained in these folds could feasibly increase the veracity of the overall information.

Two approaches are developed to combine information from multiple models created using different algorithms. The logic for these methods is taken from Boosting and Bagging respectively. The two proposed approaches seek to replicate the function of these two methods of prediction combination, thus providing a sound basis.

6.1 Structure of Chapter

The methods of combining models will be briefly recapped, and placed into context when considering combing information as opposed to combining predictions from multiple models. The information extracted from the DTI model in Chapter 5 was seen to offer relatively weak correlation with the results of the benchmark analysis (when compared to information extracted from the ANN model), hence the most pressing need is to improve this situation. A method of combining information from multiple DTI models is proposed.

The next stage seeks to use an experiment to outline the two methods of combining information from multiple models created using different algorithms. The two methods will be discussed, and then a candidate set of information will specified using the models created in Chapter 5. This information will then be combined using the two proposed methods, and the results compared and discussed. Some conclusions will then be drawn.

6.2 Consideration of Current Methods of Combining Models

In the review of literature in Chapter 13 it has been established that ensemble methods may be used to aggregate predictions from multiple models, and in doing so improve the overall accuracy of prediction, provided that each model is demonstrably different from the others within the ensemble. The criteria for such demonstrations of difference are wide-ranging, many authors consider that using different parameters, data samples or modelling algorithms result in models that may be aggregated well within an ensemble (Breiman, 1996), (Sharkey, 1996), where changing parameters or data samples may be referred to as *distortion* (Sharkey et al., 2000). There is also an argument that ensembles are most suitable for models that misclassify or otherwise perform poorly on different specific instances within the data, where each model correctly classifies different instances within the data (Freund and Schapire, 1997).

The approach proposed in this chapter builds upon these traditional ensemble methods. The proposed method seeks to combine information within each model into a single body of information, taking both aspects of the Boosting and Bagging methods as a logical basis.

It should be noted that it is not possible to base the method of aggregating models exactly upon either Bagging or Boosting, as the methods of candidate model creation are specifically described. In the case of Bagging, each model is different from the next by virtue of being created from a different bootstrap sub-dataset, and the combination of models takes no account of candidate model accuracies. In Boosting, each derivative model is different from its antecedent as the data is subjected to a different weighting, and the overall model combination considers the accuracy of each model. Neither technique makes a direct reference to any difference between candidate models other than the form of the data – Bagging is aimed mainly at indicating/improving the stability of a model when faced by different data samples, and Boosting takes account of the limitations of a modelling algorithm in terms of encompassing all data instances. In the research described in this thesis the differences between candidate models extend beyond differences in the data, and extend to differences both the algorithms used and in the algorithm parameters specified for each model.

6.3 Boosting within DTI Models

Examination of Figure 19 in Chapter 5 shows that the information extracted from the DTI model has significant differences compared to that extracted from both the ANN and analytical models. Although this may be due to an inadequacy in either the DTI algorithm or in the method of information extraction, it is suggested that the selection of one model in preference to others based solely upon accuracy is a contributory factor.

Boosting was addressed in section 2.9, where the methods of modelling were discussed. It was suggested at that point that Boosting acts to increase model complexity and thus reduce the comprehensibility of the model. That may be true in the case of *rulesets* generated using the C5.0 algorithm, as are discussed in section 12.3, however the method of information extraction proposed in Chapter 5 relies upon scoring parameters located within *trees* generated using the C5.0 algorithm. Comprehensibility is not impaired by the inclusion of multiple trees, as there is no need to consider the structure of the trees *per se*, merely the results of the scoring operation. There is also no significant increase in workload save for the scoring of the additional trees. In the case of rules, it is necessary to investigate each one separately, as they consist of essentially separate, discrete pieces of information. In effect, the scoring operation serves both to extract and condense information into a simple list, and hence Boosting only acts to increase the range of trees that the scoring is computed for.

It was noted in Table 14 (the results of DTI modelling given in Chapter 5) that model number 8 was the most accurate, and it is this model that was used as a base for the Boosted model. The use of identical algorithm parameters between the non-Boosted and Boosted models gives a clearer indication of the effect of Boosting upon information extraction.

Model	No. of training instances	Prune Severity	Min records per child branch	No of output classes	Training Accuracy (%)	Validation Accuracy (%)	Cross-Validation Accuracy (%)
Non-Boosted	500	90	2	3	93.4	60.08	63.2
Boosted	500	90	2	3	100	67.66	66

Table 19 Comparison between Boosted and non-Boosted Model

Table 19 shows the comparison between a Boosted and non-Boosted DTI model. It can be seen that the Boosted model exhibits greater accuracy than the non-Boosted model,

indicting that there are instances within the data that the model cannot adequately map, and hence an adapted sub-model or fold has been created to improve the Boosted model response to these problematic instances. In this case, the structure of the later sub-models or trials is likely to have significant differences to the base sub-model.

6.3.1 Results of Combining Information from Boosting Folds

Key	
Higher rank in analytical model	Lower Rank in analytical model

Original DTI Model	Boosted DTI Model	ANN Model	Analytical Model
15b	15b	15b	15b
110	110	13	13
15	13	12	14
17	15a	11	12
111	19	16	13a
19	16	19	17a
17a	13a	13a	11
13a	15	15	16
14	14	17	15
13	111	110	15a
13b	12	111	17
	17	14	13b
	17a	17a	19
	13b	15a	110
	11	13b	111
		114	114

Figure 20 Comparison of Extracted Information from Crash Erector Including Boosted DTI Model – Pivot Information Excluded

Figure 20 shows the comparison of extracted information from a Boosted DTI model created in Chapter 5. Concerns were raised over the accuracy and veracity of the pivot point information, hence these have been excluded from this comparison. It can be seen that the Boosted DTI model gives improved results when compared to the original model,

where the grey-scale can be seen to agree more closely with that of the analytical model. The number of parameters in the list is also expanded, as each fold or sub-model within the Boosted model has a different structure and will therefore utilise different parameters at each level. It may therefore be argued that Boosting not only can act to improve model accuracy but can also act to improve the veracity of information extracted from DTI models.

6.3.2 Issues with Evaluation of Accuracy for Each Trial

The information was extracted from each tree in an identical manner to the original approach outlined in Chapter 4 and Chapter 5, however the scores for each parameter were summed across all 10 trees or trials – in this respect the information extraction does not follow Boosting exactly, as there is no consideration of the accuracy of each trial within the model. This is due to the manner in which cross-validation accuracy is computed for a Boosted model, for which an understanding of the process is necessary to illustrate why it is not viable to deduce the accuracy of each trial.

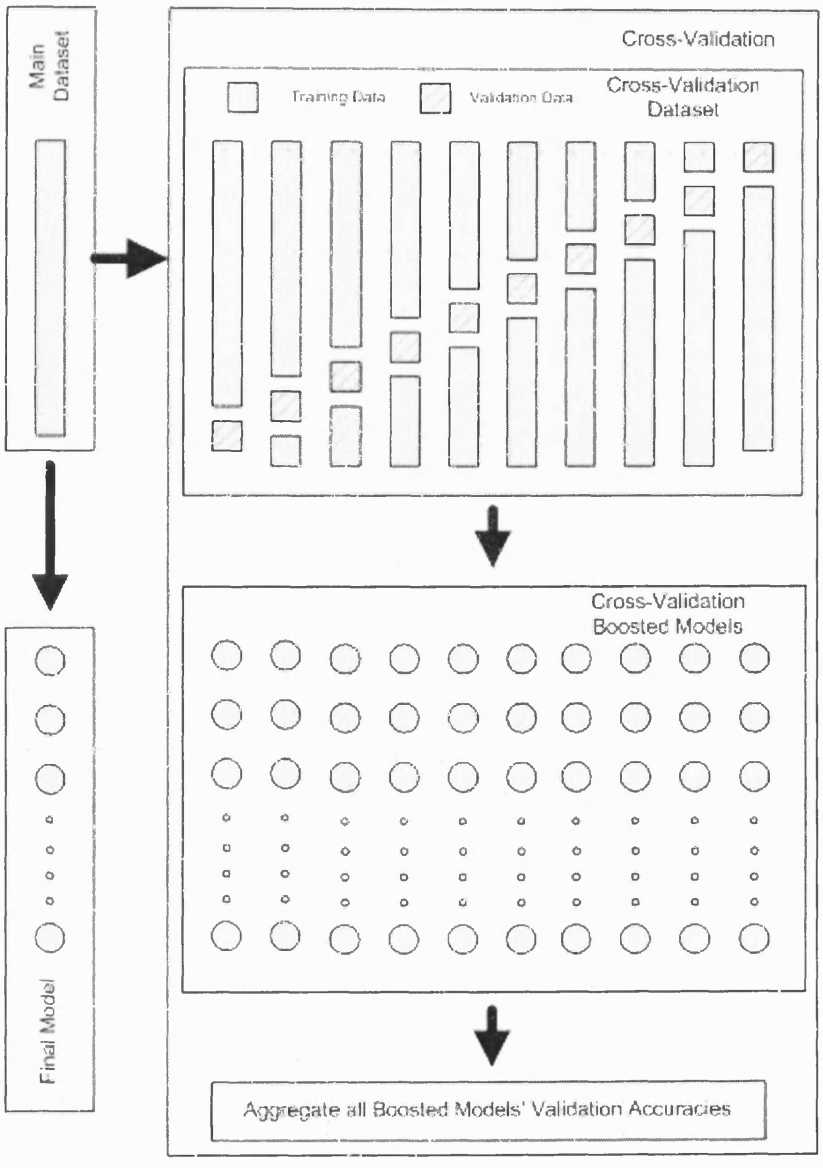


Figure 21 Method of Performing Cross-Validation upon Boosted Models

Figure 21 shows the method which is used to compute the 10-fold cross-validation accuracy of a Boosted model. The main dataset is partitioned into 10 separate folds of identical size, and then these are split into separate 90% training and 10% validation datasets, where 9 folds are appended to create the 90% training dataset. This process is completed 10 times, each time using a different fold for the validation dataset. This results in 10 separate and unique pairs of 90% training and 10% validation datasets. A series of Boosted models is created using the training dataset for each of the 10 pairs of datasets, and the aggregated accuracy over each trial within the Boosted model is calculated using the validation dataset. These accuracies are then aggregated over the 10 folds, giving the value for 10-fold cross-validation accuracy. A Boosted model is then

trained separately using the entire contents of the main dataset, as the cross-validation accuracy is known there is no need to carry out any further validation. It is in this separation between the cross-validation and the actual model that the problems lie. The Boosted models created during the cross-validation are 'lost' as they are only created for the purposes of establishing the accuracy, and once this is finished they are deleted. There is therefore no direct correlation between the cross-validation accuracy and the created model, in particular there is no measure of how accurate each trial in the Boosted model is.

This lack of connection between the cross-validation mechanism and the final model should have little effect, as the Boosting algorithm must use some measure of accuracy for each trial to be able to assign weightings to the prediction from each trial. The C5.0 algorithm and the Clementine implementation of both this algorithm and Boosting are proprietary, and whilst the accuracies of each trial are contained within the model (as they are needed to weight the predictions before aggregation) there is no outward indication of what these values are. It could perhaps be the case that the accuracy of each trial is computed upon implementation, in which case there would be no permanently stored values for accuracy. Whilst it is not possible to extract a value for accuracy for each trial, the sequential sub-models created at each trial can be seen, and information extracted. The only diversion this approach has from Boosting, in terms of weighting and aggregating the extracted information, is the ability to deduce accuracy for each trial. It is conceded that, should such a method become available to achieve this, it would present a useful improvement over this method. It is for this reason that it is suggested that this problem be addressed in future work.

6.4 Combining Models Created Using Different Algorithms

The use of methods akin to Boosting have been argued to be of use only when using the same algorithm, and lend themselves in particular to C5.0 where the measure of information gain allows for weighting of instances to have an effect upon model creation. When efforts move to amalgamating information obtained from models created using different algorithms a more bespoke method of combining information must be developed.

In order to combine the information extracted from DTI and ANN models it is necessary to decide how to accredit the information from each model. If the Bagging model is

followed, it is not necessary to consider the accuracies of the separate models, and simple aggregation of the information will suffice. However, this assumes that both the DTI and ANN algorithm are equally suitable and accurate across all domains, an assumption which is considered to be without adequate basis. It is suggested that ANN models suffer when used in complex domains with sparse quantities of data, as the required data quantities increase vastly with increases in the complexity of the model. It has been suggested that the number of nodes within an ANN is analogous to the order of a polynomial, where the order of the polynomial (and, by analogy, the number of nodes) increases as the 'complexity' of the function under consideration increases. In this respect, more complex domains require more complex ANN architectures, which require more data to adequately train. In certain cases it is suggested that the data quantities will be insufficient for this task, in which case the accuracy will be impaired. In this case, and if the accuracy of the DTI models are greater, then information from the less accurate ANN models can be weighted down to give precedence to the information from the DTI models.

Whilst it might reasonably be concluded that the DTI algorithm will have certain advantages over the ANN algorithm in applications with sparse data, it is not clear which algorithm will be most suitable for any given problem. Lim et al (2000) have tested 33 separate algorithms (including C4.5, the precursor to C5.0, and Radial Basis Functions, a form of ANN, Blanzieri, 2003) on a range of different problem datasets and conclude that different algorithms performed to different levels of accuracy between datasets, although the average level of accuracy across all of the datasets was of a similar level for all algorithms. Whilst there is no clear 'winner' in terms of accuracy for all datasets, there are arguably 'winners' for individual datasets. In this respect, it is expected that for any given problem either the DTI models or the ANN models will be more accurate, which will influence the accuracy of the information extracted from such models. Whilst such differences in accuracy cannot be foreseen, it is necessary to consider how to combine information if the accuracies of the models are not equal. It is perhaps easier to discuss how to achieve this when considering the specific mechanisms that might be employed to combine model information.

The degree of influence of a parameter upon model output can be indicated in one of two ways, the first being the placing in a ranked list and the second the actual value of the significance metric assigned to the parameter. Previous analyses of the individual models

have simply considered the placing or rank of each parameter within a list, where the actual value of the significance metric for each parameter was not considered. This approach will be expanded upon, as it can easily be adapted to aggregate information from multiple models. It is suggested, however, that neglecting the significance metrics perhaps reduces some of the available information, and so a further thread will be pursued that allows these significance metrics to have an input upon information aggregation.

6.4.1 Consideration of Ranking – The 'Medal Table' Approach

It is possible to simply aggregate the rankings of each parameter in the listings generated during information extraction. This is analogous to the idea of a medals table, where the nation of the competitor (the specific parameter) is ranked for each event (each separate model) and then combined in an overall medals table (the aggregated information). Whilst simple, there are certain shortcomings to this approach. In the case of any one parameter being extremely dominant (reflected in a significance metric that are significantly larger than those for all other parameters) then this dominance will be lost as information regarding the metric is not carried across into the combined information. There is also a problem in that a minuscule difference in metric between two notionally identical parameters would be magnified by virtue of one parameter being ranked ahead of the other. This could be alleviated by assigning a tolerance that must be exceeded for parameters with similar metrics to be considered as different, however this would require that the tolerance be suitably specified.

This approach does not easily reconcile with the idea of weighting information from a given model based upon the model accuracy. Extending the medal table analogy, it is difficult to assign 'partial' medals to those events which are considered to be worth less. In this respect it is perhaps more useful to implement this method of combining information along the lines of Bagging, where the accuracy of the models are not considered, and the ranking for each model is simply transferred to an overall medal table. This approach would therefore appear to be most applicable in cases where there is consistent accuracy between models, although there is the side issue of deciding what constitutes an appropriate level of consistency.

6.4.2 Consideration of Significance Metric – The Pseudo-Boosting Approach

If the medal table approach can be considered a Bagging approach, so the second approach can be considered as following the lines of Boosting. This approach considers the significance metric and the model accuracy for each parameter in each model, and uses the model accuracy to factor the significance metrics based upon the model accuracy. A simple summing of the significance metrics across all models will then indicate the overall level of significance for each parameter. Two separate areas must therefore be considered, the method of assigning significance metrics and the method of assigning accuracy for each algorithm

In this method there is a tacit assumption that the significance metrics are assigned in ways that give equal measure to significant parameters between modelling algorithm, essentially assuming that the most significant parameters in each model are rewarded by metrics of similar magnitudes. The two methods of modelling are intrinsically different, developing from two schools of thought, ANN being connectionist and DTI symbolic (Fu, 1994), and it is suggested that their differences are intractable – information is essentially encoded within ANN models via the strength of connections between nodes, whereas information is encapsulated within DTI models by nature of conditions attached to discrete or symbolic objects. The very different characteristics of these two algorithms suggest that there will be differences in the way that the significance of each parameter is evaluated and from this in the way the value of the metric is assigned to that evaluation. It is suggested that the use of two separate algorithms enforces acceptance of these differences, as the same method of information extraction cannot be used for both algorithms. It has been demonstrated earlier in this chapter that the method of information extraction for the DTI algorithm leads to similar results as for the ANN algorithm (this was most apparent with the Boosted DTI model, see Figure 20). It is therefore suggested that there is sufficient agreement in the assignation of metrics between approaches that the differences in the fundamental mechanism may be ignored.

It is important, however, to ensure that the metrics from each model are assigned the same degree of importance in the aggregated information (prior to allowing for model accuracy to be factored in). This can be simply enacted by normalising the metrics in each model, so that the sum of the significance metrics within each model is equal to 1.

Comparing Model Accuracies

The quantification of accuracy is different for both the DTI and ANN algorithms, both in terms of the method used to establish accuracy and in the actual metric used to describe it. Accuracy of the DTI models is measured via cross-validation, whereas computational expense means that the ANN models are evaluated using a separate validation data set[25]. Although the ANN method is suboptimal and will lead to different results compared to cross-validation, both measures can be considered identical for the purposes of describing accuracy in this application. The metric used to describe accuracy is different, although a simple manipulation enables direct comparison. The DTI accuracy is expressed as a percentage of correctly classified instances, whereas the ANN accuracy is described by a coefficient of correlation between the network prediction and actual value for each instance in the validation dataset.

Two separate coefficients were discussed in Chapter 4, the Pearson coefficient and the R-squared coefficient. The R-squared coefficient, as the name suggests, is simply the square of the Pearson coefficient and varies between 0 (no correlation) and 1 (exact correlation). The Pearson varies between -1 (perfect negative correlation) and 1 (perfect positive correlation). The R-squared coefficient is suggested to be most useful in this case, as it provides a metric in the same range as a normalised percentage.

The method of comparing accuracy is therefore quite simple, by normalising the percentage accuracy of the DTI tree (essentially dividing the percentage by 100) the accuracy will be represented on scale of 0 to 1, with 1 being complete agreement. The use of the R-squared coefficient for the ANN models allows accuracy to be presented along an identical scale.

6.4.3 Specification of Candidate Model Set

The first stage of combining the information is to define a candidate set of models, which are those models that have been created as part of the iterative DM process that are considered to be of a high enough accuracy for inclusion. In Chapter 5 the models were

[25] It is suggested that Cross-Validation is a preferable method of ANN accuracy evaluation, hence if computational resources allow this should be used.

predominantly of similar accuracy, arguably due to the lack of noise seen in the deterministically-derived data, and hence the selection of models is perhaps a little arbitrary. The models were created using a range of different data samples and algorithm parameters, and in order to maximise diversity within the candidate model set (both to test the ruggedness of the approach and ensure that the benefit of model combination is maximised - see Sharkey, 1996) a selection of models created from different combinations of data and parameter settings will be included.

Model	ANN Model 8		ANN Model 14		DTI Model 3		DTI Model 8		Boosted DTI Model	
Model Reference	Model A		Model B		Model C		Model D		Model E	
Accuracy*	R-squared 0.7995		R-squared 0.6965		66%		63.2%		66%	
Ranked Extracted Information	L5b	0.1898	L5b	0.1577	L5b	0.5625	L5b	0.3984	L5b	4.8887
	L3	0.0448	L3a	0.0597	L5	0.5625	L10	0.3750	L10	1.7617
	L2	0.0402	L6	0.0576	L4	0.2500	L5	0.3633	L3	1.2627
	L1	0.0340	L3	0.0567	L3a	0.1250	L7	0.0859	L5a	1.2109
	L6	0.0245	L5	0.0563	L10	0.1250	L11	0.0703	L9	1.0078
	L9	0.0229	L11	0.0484	L1	0.1250	L9	0.0625	L6	0.9844
	L3a	0.0217	L5a	0.0433			L7a	0.0625	L3a	0.8931
	L5	0.0178					L3a	0.0391	L5	0.7139
	L7	0.0169					L4	0.0352	L4	0.7002
	L10	0.0164					L3	0.0313	L11	0.6104
	L11	0.0159					L8	0.0313	L2	0.4512
	L4	0.0153					L3b	0.0156	L7	0.3633
	L7a	0.0148							L7a	0.1973
	L5a	0.0144							L3b	0.1484
	L3b	0.0099							L8	0.0938
	L14	0.0070							L1	0.0742

* Accuracy is established via 10-fold Cross-Validation for DTI models and via separate validation set for ANN models

Table 20 Candidate Set Extracted Information

Table 20 shows the models that were selected for use in this experiment, where two DTI models and two ANN models were selected for use along with the Boosted DTI model from section 6.3. The DTI and ANN models were selected from Table 14 and Table 17 respectively.

A cursory examination of the table of extracted information in Table 20 reveals the discrepancies that are evident across models in terms of the extracted information. Whilst all models agree that parameter L5b is the most influential, there is notable variation in the ranking of less significant parameters. It is highlighted that not all parameters will appear in either the ANN or DTI extracted information, as there is no obligation for the DTI algorithm to use all available parameters and the method of

pruning used to generate a suitable ANN architecture necessarily removes input nodes (if the algorithm considers they are not contributing to network response). These absences do not unduly influence the consideration of the significance of each parameter, as those that are absent are those considered to have minimal influence upon model behaviour.

6.4.4 Method for Medal Table Approach

The traditional Medal Table credits participants in individual events via 3 different medals, each with progressively lower levels of prestige, with the most prestigious being credited to the winner. The overall Medal Table is arranged by consideration of the sum of each level of medal awarded to a group or nationality. The ranking is assembled by considering how many of the most prestigious medals each group or nationality has, those receiving more having a higher rank. In the event of two or more groups having equal ranking, the ranking is then further refined by considering how many of the second-most prestigious medals each group has and this is once again repeated in the event of a tie.

It is preferable to be able to expand the range of available 'medals' as the list of significant parameters extends further than the top three. The rationale of considering only the most prestigious level of medals as the key metric, and considering the overall rankings based only upon that metric (except in the case of ties), is argued to effectively ignore parameters that might consistently be ranked second or third whilst favouring those that might fleetingly be ranked first.

To address both of these concerns it is suggested that assigning a score instead of a medal (in effect awarding credit of some description to each parameter within the ranking) allows for consideration of all parameters within the model, and by summing these scores then those parameters which frequently appear high in the ranking will be duly credited.

The scoring is based upon the reciprocal of the listed ranking for each model, where the most significant parameter is given a score of 1, the second-most a score of ½, the third-most a score of 1/3, and so on.

$$R_x = \sum_{i=1,m} r_{x,i}$$

The previous equation indicates how the overall Medal Table should be compiled based upon the scores assigned in each model, where R is the ranking for model x, r is the score assigned to model x within a specific parameter, and m is the total number of models.

6.4.5 Method for Pseudo-Boosting Approach

In the case of the Boosted DTI model, it was not possible to fully mimic Boosting as it was not possible to deduce the accuracy for each trial in the model. In order for Pseudo-Boosting to be usefully employed, it is important that both the measure of accuracy and the measure of significance of each parameter across the entire range of models should be of both a similar form and of the same order. This can be simply achieved for the significance metrics, requiring only that the metrics be normalised for each model.

$$N_x = \frac{s_x}{\sum_{i=1,y} s_i}$$

This normalisation can be computed using the above equation, where N is the Normalised Significance Metric for parameter x, s is the significance parameter for parameter x and y is the total number of parameters contained within a model.

Model	**Model A**		**Model B**		**Model C**		**Model D**		**Model E**	
Accuracy	**0.7995**		**0.6965**		**0.66**		**0.632**		**0.66**	
Normalised	L5b	0.37488	L5b	0.32875	L4	0.14286	L5b	0.25369	L5b	0.31823
Ranked	L3	0.08849	L3a	0.12445	L5b	0.32143	L10	0.23879	L10	0.11468
Extracted	L2	0.0794	L6	0.12008	L5	0.32143	L5	0.23134	L3	0.0822
Information	L1	0.06715	L3	0.1182	L3a	0.07143	L7	0.0547	L5a	0.07882
	L6	0.04839	L5	0.11737	L10	0.07143	L11	0.04477	L9	0.0656
	L9	0.04523	L11	0.1009	L1	0.07143	L9	0.0398	L6	0.06408
	L3a	0.04286	L5a	0.09026			L7a	0.0398	L3a	0.05814
	L5	0.03516					L3a	0.0249	L5	0.04647
	L7	0.03338					L4	0.02241	L4	0.04558
	L10	0.03239					L3	0.01993	L11	0.03973
	L11	0.0314					L8	0.01993	L2	0.02937
	L4	0.03022					L3b	0.00993	L7	0.02365
	L7a	0.02923							L7a	0.01284
	L5a	0.02844							L3b	0.00966
	L3b	0.01955							L8	0.00611
	L14	0.01383							L1	0.00483

Table 21 Candidate Set Extracted Information with Normalised Metrics

The accuracies for each model can be standardised by converting the percentage given by cross-validation of the DTI models into decimal form, in effect dividing by 100. The results of this normalisation and decimalisation can be seen in Table 21. The first stage

of this approach requires that all metrics that will have a bearing on the analysis to be normalised. Once this is done, the significance metrics and accuracies shown in Table 20 can be represented by those seen in Table 21. In order to minimise rounding errors the number of significant figures for each metric have not been further constrained.

Upon normalisation, each significance parameter is then multiplied by the accuracy of the model from which it was computed. These modified significance metrics can then be summed for each parameter across all the models within the candidate model set.

$$S_x = \sum_{n=1,m} s_{x,n} A_n$$

The equation for computing overall parameter significance is given by the above equation, where S is the Combined Significance Metric for parameter x, s is the significance metric for parameter x in model n, A is the accuracy of model n and m is the total number of models within the candidate set.

6.4.6 Results for Medal Table approach

Parameter	Score
L5b	5.0000
L10	1.3000
L5	1.2833
L3	1.1833
L3a	1.1607
L6	0.7000
L4	0.6389
L11	0.5576
L9	0.5333
L1	0.4792
L5a	0.4643
L7	0.4444
L2	0.4242
L7a	0.2967
L3b	0.2214
L8	0.1576
L14	0.0625

Table 22 Results of Model Aggregation for Medal Table Approach

The results of the Medal Table aggregation are shown in Table 22. these results will be examined in more detail in conjunction with the results of the Pseudo-Boosting aggregation.

6.4.7 Results for Pseudo-Boosting Approach

Parameter	Score
L5b	1.111197
L5	0.498875
L10	0.299645
L3a	0.222197
L3	0.219915
L6	0.164613
L4	0.162695
L11	0.149899
L14	0.149588
L5a	0.137633
L9	0.104613
L1	0.10402
L2	0.082865
L7	0.076865
L7a	0.057
L3b	0.028287
L8	0.016626

Table 23 Results of Pseudo-Boosting Aggregation

The results of the Pseudo-Boosting approach can be seen in Table 23, and will now be discussed alongside the results of the Medal Table aggregation.

6.4.8 Comparison of Proposed Methods of Model Combination

Key	
Higher rank in analytical model	Lower Rank in analytical model

Medal Table		Pseudo-Boosting		Analytical Model	
L5b		L5b		L5b	
L10		L5		L3	
L5		L10		L4	
L3		L3a		L2	
L3a		L3		L3a	
L6		L6		L7a	
L4		L4		L1	
L11		L11		L6	
L9		L14		L5	
L1		L5a		L5a	
L5a		L9		L7	
L7		L1		L3b	
L2		L2		L9	
L7a		L7		L10	
L3b		L7a		L11	
L14		L3b		L14	

Figure 22 Results of Medal Table and Pseudo-Boosting Aggregations

The comparison given in Figure 22 indicates that the two methods of information combination give results that are consistent, but these results do not agree with the analytical model analysis. The correlation is notably weaker than for the information extracted from the exemplar DTI model and ANN model in the previous chapter.

It is suggested that those models subjected to pruning or are otherwise adapted to improve generality might be contributing to this situation, as the actual act of pruning intends to sacrifice some portion of accuracy in order to make gains in model generality and interpretability. Such methods also attempt to post-process the model and manipulate the internal structure of the model whilst having minimal impact upon model output. Whilst this is of little significance in situations where only the prediction is

required (indeed, it might assist matters as excessive aspects of the model structure which might result in overfitting are removed) such methods may act to change the internal structure in such a way as to alter the way in which information is stored. The proposed method of information extraction for DTI models can be argued to be significantly affected by this - whereas the method of extraction for ANN models does not consider the model structure, only model response to a series of inputs, the method of extraction for DTI models considers only the model structure, and specifically only where each parameter is used within the model. Excessive pruning may therefore act to remove or reduce the occurrence of certain parameters, and do so in a way which is outside of the original method of constructing the model. In the case of ANN models the removal of input parameters acts to limit the form that the network might take, in forcing the network to reach a final arrangement that would not necessarily be reached in the presence of all input parameters.

It is reinforced here that the structure of a network is simply method of transforming a series of inputs into a series of outputs, and where the nature of the required transformation is inferred from previous cases. The actual transformation is never explicitly known, the network in effect seeks to replicate its effects. There are many ways that the network can achieve the desired effect, and variations within a limited subset of the input data could feasibly provide enough information for only these inputs to be considered – the output might be predicted to a reasonable degree of accuracy using only a small portion of available information. This is essentially what pruning implements, where nodes (including input nodes) that are considered less influential are removed. It is suggested that such a method might act to influence the information contained within a network, as the pruning algorithm attempts to drive connection weights to zero whilst minimising the effect upon network accuracy. Aside from the issue of correctly measuring accuracy, where it can only be established using a specific subset of data that might not be entirely representative of the process, there is the problem that judgement of the degree of influence of a node is carried out during the training process, before it is complete. It is feasible that a node, whilst exerting little influence initially, might go on to exert greater influence towards the end of training as the entire network is adapted to better fit the training data. If, however, this node is removed at an early stage of training then this possibility is lost. It is therefore suggested that pruning on ANN models might act to reduce the veracity of thee extracted

information. The method of pruning is suggested to still be of use, as it allows for some degree of network autonomy in deciding a suitable number of hidden layer nodes, but it is argued that pruning should not be applied to the input layer nodes.

It has been established that Boosting can act to improve DTI model accuracy, and it has further been demonstrated that the accuracy *of information extracted* from a DTI model can be further improved by Boosting. Bearing this (and the previous argument that pruning of a model may act to reduce the accuracy of extracted information) it is suggested that removing both non-Boosted DTI models as well as all pruned models from the candidate set may act to improve the results of model combination.

6.4.9 Removal of Restricted Models from Candidate Set

Key	
Higher rank in analytical model	Lower Rank in analytical model

Medal Table	Pseudo-Boosting	Initial Medal Table	Initial Pseudo-Boosting	Analytical Model
L5b	L5b	L5b	L5b	L5b
L3	L3	L10	L5	L3
L10	L10	L5	L10	L4
L2	L2	L3	L3a	L2
L6	L6	L3a	L3	L3a
L9	L9	L6	L6	L7a
L5a	L5a	L4	L4	L1
L1	L3a	L11	L11	L6
L3a	L5	L9	L14	L5
L5	L1	L1	L5a	L5a
L7	L4	L5a	L9	L7
L4	L11	L7	L1	L3b
L11	L7	L2	L2	L9
L7a	L14	L7a	L7	L10
L3b	L7a	L3b	L7a	L11
L14	L3b	L14	L3b	L14

Figure 23 Results of Medal Table and Pseudo-Boosting Aggregation with Restricted Candidate Models Removed

It can be seen in Figure 23 that the removal of the three pruned models from the candidate set results in a ranking for both the Medal Table and Pseudo-Boosting approach that is closer to the results of the analytical model analysis, where 3 of the top 4 ranked parameters are consistent between the three analyses. It is also noted that the results for the Medal Table and Pseudo-Boosting approaches are extremely similar, where the top 7 ranked parameters are the same for both analyses. This is unexpected as there is a difference of approximately 10% between the models which would act to influence the ranking for the Pseudo-Boosting approach but not for the Medal Table. The list is argued to be skewed by the high rank for parameter L10, which appears 3rd in both of the Pseudo-Boosting and Medal Table approaches, but 14th (or 3rd from bottom) in the Analytical Model list. It can be seen in Table 20 that parameter L10 is highly ranked by the DTI analysis, but not by the ANN analysis. It is suggested that further consideration of how to assign the values of the DTI significance metrics is required, however it is concluded at this point that the methods are useful in identifying significant parameters at the expense of incorrectly inflating the significance of other, less significant parameters.

6.5 Concluding Remarks

The methods of combining models as presented in this chapter are intended to indicate how the accuracy of extracted information might be improved by combining information from different models. The initial modelling and comparison of information extracted from individual models showed good agreement with a separate, industrially-validated case study, however the information from the ANN model was seen to offer closer correlation than that from the DTI model. This situation was improved by combining information from a Boosted DTI model, essentially converging information from the 10 trials within the Boosted model via a simple aggregation. This was seen to offer improvements, however not to the extent as to be of equivalent accuracy to the ANN model. There was a difficulty in evaluating the accuracy of each trial, or separate sub-model, within the DTI model, and hence it was not possible to factor into the aggregation any idea of accuracy or to weight information from each sub-model depending upon accuracy of that sub-model. It is suggested that, if a method of deducing sub-model accuracy were available, such information would present significant opportunity for improvement in the approach.

It was argued that the relatively simple nature of the case study and the high quality of the resultant data meant that differences in the created models were minimal. However, it was anticipated that in practice a range of models would be created each with varying degrees of accuracy, and containing different information depending upon the specific data and parameters used in its construction. In order to cater for this, two methods were proposed to combine information from different models. These two methods were based upon the logic of Boosting and Bagging, two methods of combining the prediction of multiple models. As there is no benchmark or method of evaluating the performance of these two methods, as the initial, single-model results were of good agreement and hence any development will not necessarily yield any performance increase, it was decided to apply the logic of both Boosting and Bagging as faithfully as possible in order to provide a sound foundation to the approaches. In this respect, the function of the two methods of combining information are identical to the methods of combining predictions, but are applied to a different but related attribute of the model.

The initial approaches were seen to produce combined information of significantly inferior quality to that given by the most accurate (in terms of significance ranking) of the individual models. This was argued to be due to deleterious influences from both non-Boosted DTI models, where the vagaries of data and model parameter exert significant influence, and of the pruning of models, where the structure of a model is changed.

It is highlighted here that acceptance of the results of both the initial, single-model approaches and combined approaches requires a degree of acceptance of the veracity of both the original analytical model analysis, the control study, and of the veracity of both Boosting and Bagging. In one case, the results of the analytical model study form the basis of comparison and of validation of this research, and in the second case (and in the absence of any practical method of validation) provide the underlying logic.

The Boosted DTI model showed that combining what are essentially different models can lead to more accurate information. The method of information extraction from a non-Boosted DTI model was seen to offer limited agreement with both information from an ANN model and with the control analytical model results, however when the approach was trialled upon a Boosted DTI model the correlation was seen to have significant improvement. It also appears that efforts to increase the generality (or otherwise reduce model complexity) reduces the veracity of the extracted information, hence it is recommended that non-pruned models are used for information extraction.

Chapter 7 Initial Manufacturing Data Issues – Error within Manufacturing Data

The work described in the thesis thus far has focussed upon modelling data that have been computationally generated to represent manufacturing, test and assembly data. This chapter shifts the focus towards the modelling of genuine manufacturing data and seeks to identify how error within such data might be dealt with prior to modelling. As the presence of noise or error within data has been seen (in section 4.4) to reduce the accuracy of DM modelling, this chapter seeks to identify if such error may be removed or otherwise treated such that the accuracy of DM modelling may be improved.

A case study investigating data generated from a soap powder packaging line is discussed. This investigation aimed to reduce the duration of changeover, the period when a manufacturing line is altered to switch production from one product type to another. The manufacturing process under examination had previously been refined using a industry-standard improvement strategy, however the final stages of this strategy are poorly defined, and it is within these final stages this case study is focused.

The data used in this chapter were generated during an earlier analysis of this manufacturing line. Initial examination of the collected data revealed the presence of significant noise within these data, and different strategies of dealing with such noise were trialled in a series of experiments. These experiments are described and their success in removing erroneous instances is examined.

7.1 Structure of Chapter

The case study investigates the phenomenon of changeover. The phenomenon of changeover will be introduced, and literature describing methods of reducing the changeover duration will be covered. This DM analysis aims to improve upon an earlier analysis, and this previous research will be briefly discussed to provide some context. The method of data collection will be covered, as will the nature of the collated data.

It has been suggested that errors will feasibly have been introduced into the data. Inheritance of a dataset precludes any investigation of the causes of error at the

manufacturing facility, and in light of this a series of experiments are proposed to reduce the error within this inherited data. The degree of success of these experiments will then be discussed.

7.2 Scope of Case Study – Changeover Performance

This case study uses data collected from a soap powder packaging line as a basis for investigation. These data were collected in order to further understanding of factors that affect changeover, the period of time that a production line is inoperative or not at full running speed whilst alterations are made to allow for a different product to be manufactured or assembled. Efforts had previously been made to improve changeover performance, or essentially reduce changeover duration, however an understanding of which parameters or line alterations was considered necessary to steer further efforts at improvement.

7.2.1 Definition of Changeover

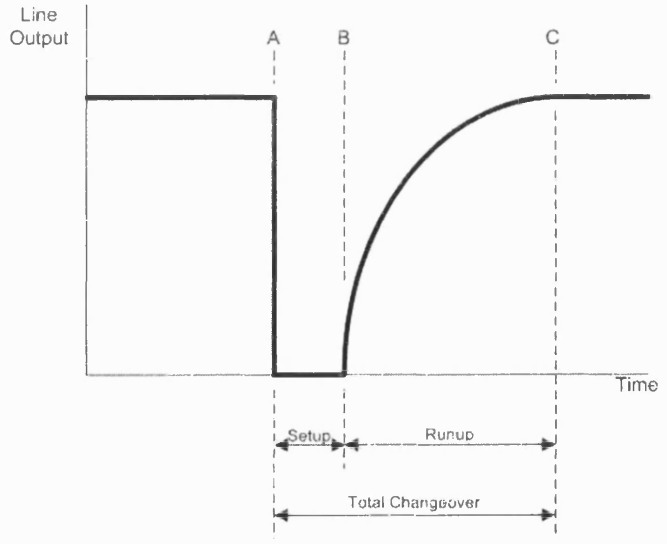


Figure 24 Typical Changeover (McIntosh *et al*, 2001)

Figure 24 shows a plot of manufacturing line output against time for an example changeover, where the individual changeover components of *set-up* and *run-up* are indicated. Line 'A' denotes the time where manufacture of the initial product is terminated. Manufacture of the second product begins immediately after line 'B' and reaches the desired operating output after line 'C'. The time period between 'A' and 'C' may be referred to as the changeover period. The time period between 'A' and 'B' is defined as the set-up period, where the line is stationary and (amongst other operations)

adjustments are made to the manufacturing line to allow for a different product to be manufactured. The section between 'B' and 'C' is the run-up period where further adjustments are made to 'fine-tune' the production line to allow manufacture to proceed at the desired throughput and with the requisite quality. The set-up period involves changing line settings whilst the run-up period focuses upon ensuring that these changes have the desired effect upon the function of the line.

7.2.2 Research into Changeover Improvement

Research into improving changeover performance primarily centres around reducing the changeover time, although there are other areas that may be improved such as reliability or amount of scrap (McIntosh et al., 2001). Reductions in changeover time result in benefits such as increased capacity and improved flexibility (Van Goubergen and Landeghem, 2002). There are a number of practical strategies available to reduce the changeover time, of which the methodology proposed by Shingo, the Single Minute Exchange-of-Die (SMED) system (Shingo, 1985), is by far the most common and has had extensive take-up in many manufacturing organisations. The basic principles of the SMED system tend to feature in most process-based changeover improvement methodologies (Van Goubergen and Landeghem, 2002), largely because SMED was one of the first proven techniques (Szatkowski and Reasor, 1991). This methodology is entirely retrospective in its application, and is based upon two founding principles.

- Identifying, separating and converting 'internal time' tasks into 'external time' tasks
    - External time (the period immediately prior to 'A' in Figure 24) may be considered the period immediately before manufacturing is stopped prior to changeover, where preparations are made to begin changeover.
    - Internal time is the changeover period, which is traditionally refers to the period where the line is static. However, once run-up is considered this definition is no longer realistic, as the line will be moving (perhaps at reduced speeds) during run-up and hence the term internal time will not be used in this case study.
    - By moving tasks from internal time to external, changeover time is reduced as fewer activities take place during this period.

- 'Streamlining' all aspects of the operation
    - General improvements to the entire changeover procedure (Shingo, 1987).
    - This is a somewhat vaguely defined term, referring to all efforts at changeover improvement not directly relating to the 'externalisation' of tasks.

In order to accomplish these 2 goals, 4 steps are specified, of which the first 3 respectively deal with identifying, separating and converting internal time tasks to external time. The fourth step focuses upon the idea of streamlining the entire process, a term that caters for all other improvements that can be made to the changeover process. A diagram of the entire methodology may be seen in Figure 25 where the 4 steps are identified as stages, Stage 3 representing the 4th step of streamlining.

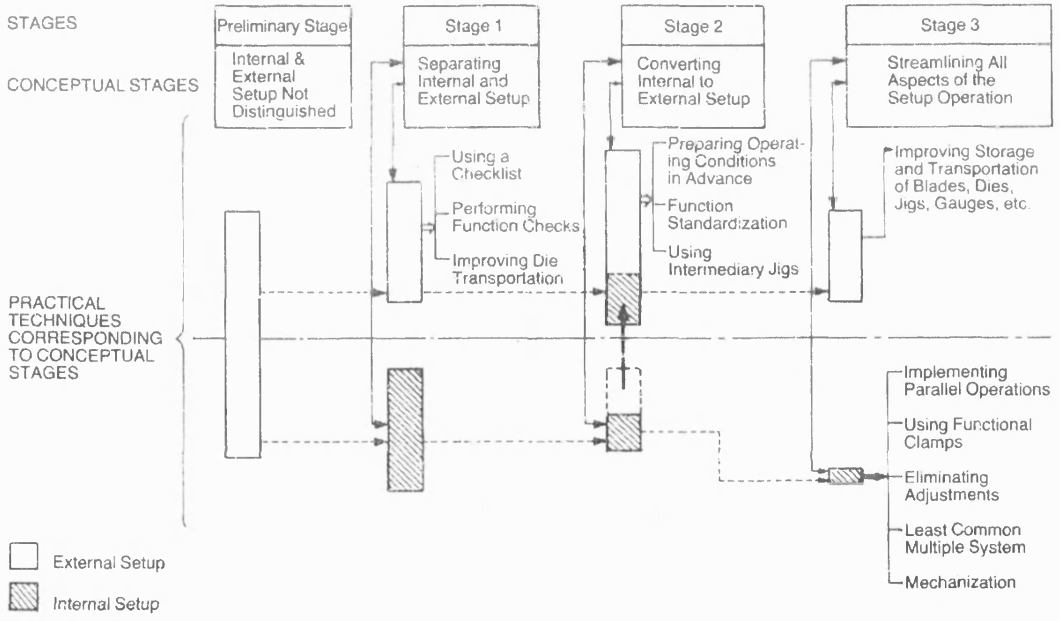


Figure 25 Shingo's SMED Methodology (Shingo, 1985)

The decomposition of the changeover period into separate set-up and run-up periods is not always recognised, indeed the SMED system does not make explicit reference to run-up or provide any specific means to reduce it (McIntosh et al., 2001). Many practitioners appear to interpret implementation of SMED as simply an attempt to reduce a changeover's set-up period (Mileham et al., 2004). However, whilst sparse, there have

been some reference to run-up within literature (Higgins, 2001), (Sladsky, 2001), and hence it is viewed within this research as a genuine phenomenon.

One of the difficulties of implementing the SMED methodology is a lack of structure to the streamlining phase of the changeover improvement process, where the actual practice of streamlining is not explicitly defined and is open to considerable interpretation (McIntosh et al., 2000). There is also the problem of chronology. Streamlining only becomes active within the SMED methodology during the last of the 4 stages, despite forming a significant part of the entire process, and hence it perhaps does not receive the emphasis that earlier stages enjoy. It has been suggested (Shingo, 1985) that only 15% of the overall improvement to changeover given by SMED implementations is obtained within the final SMED methodology's streamlining stage, which agrees well with practical results (Szatkowski and Reasor, 1991). This may perhaps be explained by earlier stages resulting in a level of improvement that is significant enough for further efforts to be truncated. When compared to the stages dealing with transition from internal to external time, streamlining is also arguably more expensive as it is likely to involve modification to the underlying hardware (Leschke, 1997) whereas the alterations performed as a result of the first three stages are likely to be predominantly procedural in nature and hence circumvent the cost implications of physical changes to the line. Inspection of Figure 25 highlights the general improvements that Shingo recommends at each stage, where there is little cost involved (excluding labour) in modifications such as *performing function checks* and *preparing operating conditions in advance*. In contrast the changes involved in stage four, such as *mechanization*, are more expensive to implement and hence would arguably only be carried out if improvements over and above those possible by procedural change were required.

7.2.3 Aims of Changeover Analysis

The manufacturing line investigated in this case study has been the focus of considerable investment and effort in terms of improving the changeover performance. The conversion to external time, which provides a significant portion of the improvements attributable to SMED, has already taken place, and focus has now switched to streamlining operations. As highlighted earlier, such efforts are costly and involve significant (time-consuming) modification to the line, and hence it is desirable to be able to steer streamlining modifications towards areas that would be most beneficial. It is for

this reason that the dataset was originally collated, as it was considered essential to be able to deduce which parameter changes were causing the greatest problem in terms of changeover performance.

This thesis has thus far considered the application of DM to specific artefacts, with the ambition of indicating which of an artefact's characteristics influence some aspect of performance. In this case, the changeover becomes the artefact, and the aspect of performance most of interest is the duration of the changeover. The task of this case study therefore becomes to indicate which characteristic or parameter of the changeover has greatest influence upon the duration of the changeover.

7.2.4 Initial, Non-DM Data Analysis

This initial analysis was performed outside of the research described in this thesis and the author makes no claim for ownership. This work will be discussed briefly in order to provide some background. The initial investigation of the collated data sought to deduce the influence of each parameter via a series of weightings. Each parameter (as listed in Table 24) was weighted according to the judgement of experienced operators, where influential parameters (those parameters that were considered to contribute to lengthy or variable changeover durations) were given high weightings and less influential parameters given low weightings.

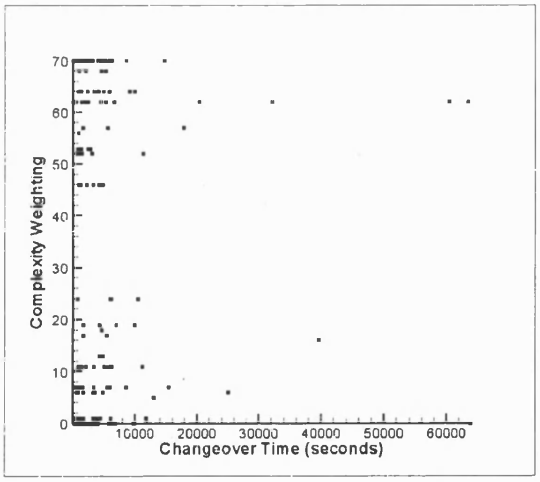


Figure 26 Complexity Rating Versus Changeover Duration for Pre-DM Analysis

Figure 26 shows the summed complexity rating for a series of changeovers against the time taken for each specific changeover. There is no discernable pattern visible, as it is suggested that the method of complexity weighting based upon expert opinion suffers from excessive subjectivity and also relies upon the relationship between parameter and

'complexity' being observable to an operator. It is also suggested that the individual weighting allocations do not allow for couplings within the data, where one parameter might only become influential if another parameter is to be modified within the same changeover.

7.3 Generated Manufacturing Data

The measurement and recording of data from the packaging line was initially intended to give engineers at the plant an understanding of those processes which were the most time consuming, and in doing so allow for both better scheduling (avoiding repeated alteration of troublesome parameters) and focusing improvement efforts upon the longest duration changeovers or specific processes. In order to achieve this it is necessary to evaluate the duration of changeover and to link into each individual changeover record the parameters that were modified in the course of that operation. This was achieved by utilising a data logger to record the output over time (Eldridge et al., 2002). This data logger recorded the output of the line over time, and could be further employed to record specific comments and codes that could be manually entered and aligned at a particular point in time on the output trace.

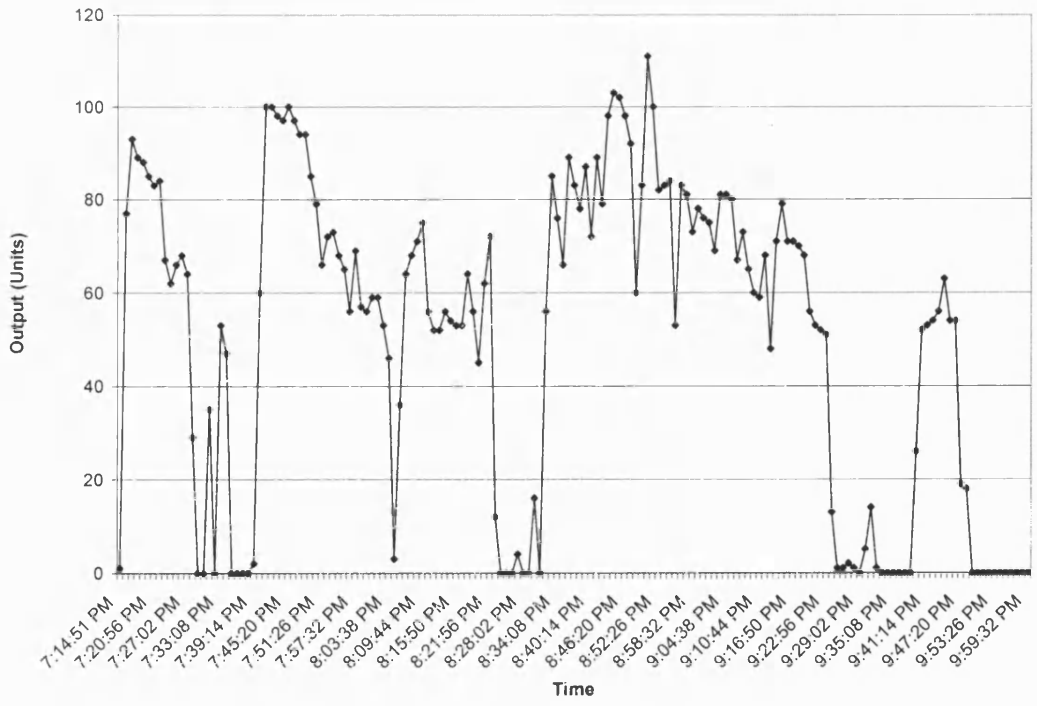


Figure 27 Typical Production Line Output

Figure 27 shows an example of the output of a production line over time[26], where the number of units produced are displayed against time. In this plot the time axis is shown as the time of day, a system which (unlike elapsed time) provides an index with other operations and thus allows for other data to be aligned with out put at any given time.

The packaging line may be stopped for a number of reasons, ranging from routine maintenance and problems with operation through to product changeover (the stoppage of interest in this research). The identification of a changeover is relatively trivial, the time stamping allows records of product code to be aligned with the output plot, and where the product code changes over the duration of a stoppage that stoppage may then be assigned as a changeover. In addition to this, the annotating facilities of the data logger allow for further information to be stored, detailing such occurrences as problems with changeover or extra time for maintenance.

7.3.1 Identification and Quantification of Changeover

Whilst the identification of a changeover is straightforward, an accurate measurement of changeover duration can be more problematic. In the example plot of line output given in Figure 27 it can be seen that the output is unsteady with numerous peaks and troughs, with numerous small periods of activity within these troughs that represent 'false starts' as the line is restarted for only a brief period. These false starts are the basis of the phenomena the author has identified previously as run-up, where efforts are made to ensure that changes in line settings result in acceptable performance (the manufacture of goods of the desired quality).

In most implementations of SMED the practitioner would consider changeover as starting once the last piece of merchantable quality of the initial product was completed, and as being complete once the first piece of merchantable quality was produced immediately after the line was modified, in effect going from 'good piece to good piece' (Trevino et al., 1993). This neglects any stoppages or defects within manufacture immediately after the first piece of the new batch is manufactured, as is argued to be the evident in Figure

[26] This plot is obtained from a related study, included here to illustrate the variable output from production lines. The data used in this case study was received in a pre-compiled state and it was not possible to create a similar type of plot from this data.

27. It may clearly be seen that the line in Figure 27 is subject to numerous short stoppages and limited output immediately following the initial restart at 7:33:08pm, which may arguably be assigned to the effects of the alterations carried out during changeover. In light of this it is difficult to reconcile the idea of changeover being complete once the first piece of the new batch is manufactured, although this may accurately be described as the end of set-up, and hence a different metric to measure changeover is required.

This poses significant problems, as it is unclear how to decide if changeover has been completed – for the sake of argument, a line might run successfully for hours until a problem is noted, and if that problem is decided to be due to the alterations carried out at changeover then the preceding hours of production could conceivably be considered as run-up. This is clearly a poor definition, and hence a more robust method of specifying where run-up and changeover are complete is required.

This choice of metric is still the subject of other changeover research, and falls outside of the scope of this research. As an interim measure, run-up will be considered to have finished once the volume of production has been stable at the desired rate of production for a period of 5 minutes (the five minutes of stability being included in the measure). This is not intended to be, nor should it be construed as, a definitive metric for measuring run-up, however it is argued to be a consistent measurement that indicates when a level of stability has been introduced into the line.

7.3.2 Data Acquisition Process

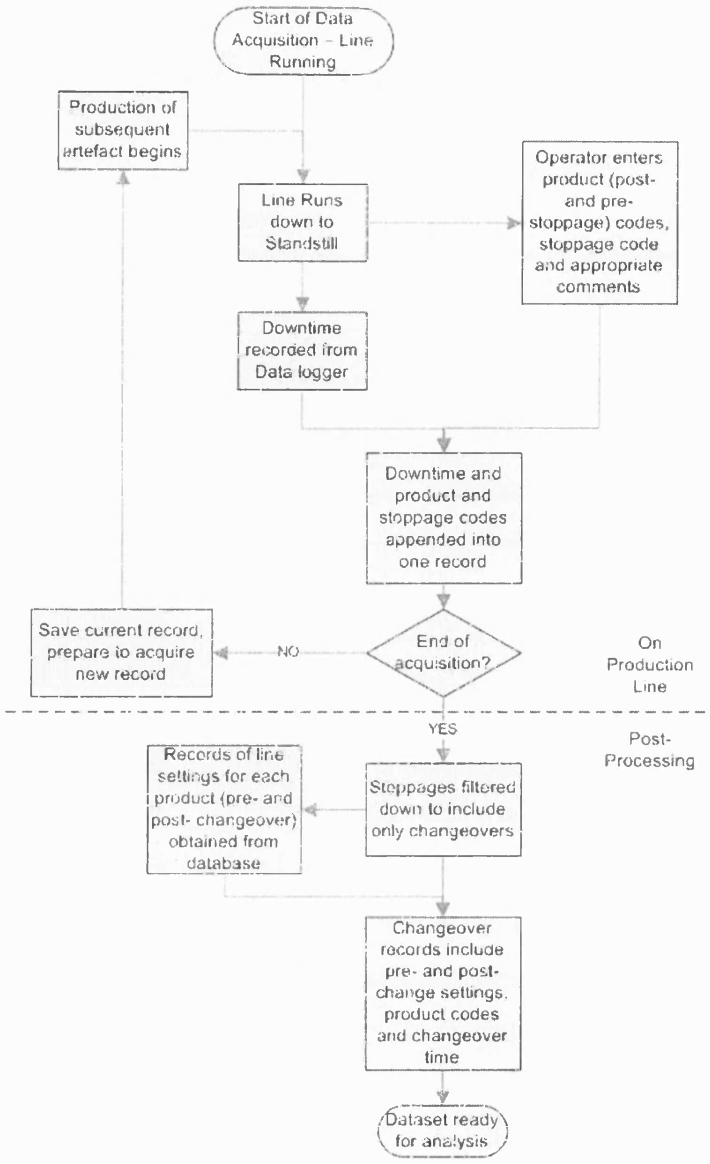


Figure 28 Flowchart of Data Acquisition Process

Figure 28 shows the author's interpretation of the data acquisition process as performed by Eldridge *et al* (2002). This process is carried out in two separate stages. The first stage is the 'real-time' monitoring of the production line output, from which details of each stoppage are automatically recorded. This is the gathering stage of the data acquisition process, being performed at the site of the manufacturing line. In the second stage each stoppage is investigated and genuine changeovers identified by comparing pre- and post-stoppage product codes, and non-changeover stoppages are deleted from the dataset. This is the post-processing or data transformation stage of the acquisition process, and was carried out remotely from the manufacturing line. Cross-referencing

with a database of product settings allows for the pre- and post-changeover product codes to be expanded to include line settings used for each product.

7.3.3 Structure of Acquired Data

The database comprised 470 records of individual changeovers, corresponding to approximately 9 months of manufacture.

Parameter	Nature of Data	Description
Carton*	Discrete	Code (C23, E30) relating to industry standard carton sizes
Flask Size*	Discrete	Size (small, large) of powder storage flask
Scoop Size*	Numerical	Diameter of scoops used for delivering powder
Box Height*	Numerical	
Box Width*	Numerical	
Line Speed*	Numerical	
Product Insert*	Discrete	Presence of promotional free gift
Force-to-Box (FTB)*	Numerical	Force required to expand carton
Riveting*	Discrete	Requirement for riveting during box expansion
Team	Discrete	Group of operators performing changeover
Shift	Discrete	Shift during which changeover occurred

Table 24 List of Parameters for Changeover

The data received from the data logger consisted of the team and shift and the product codes both before and after changeover. A separate data list gave the line settings for each product code, and the product codes were 'expanded' into the full set of line settings, increasing the number of parameters in the main dataset. Table 24 lists these parameters, where those marked with an asterisk indicate that these parameters can be altered during a changeover, whereas the others are fixed (although it is noted that team and shift could feasibly change during a changeover, in practice no such cases were noted). It can be seen that there is little indication of the exact processes that are required to actually alter these settings, for example changing the force-to-box could entail a simple adjustment to a control or it could involve modifying a complex system. It should also be noted that the numerical values have not been labelled 'continuous', as although the means of describing these settings is continuous (such as the diameter of the feed scoop) many of these parameters will have only a few settings, which will result in many Machine Learning Algorithms effectively treating them as discrete.

The duration of both set-up and run-up were extracted from the time plot of production, and appended to the dataset. To recap, set-up was easily measured as being the duration between the end of production of the last piece of merchantable quality and the start of production of the first piece of merchantable quality of the following batch. The run-up is more problematic to measure, where the end of run-up is not always clear, and a decision was made to consider run-up complete when the line had been operating at a desired volume of output for 5 minutes.

7.3.4 'From-to' and 'Delta' Data

The input data describes the line settings before ('from') and after ('to') changeover. There has a limitation in that some parameters might not change during a specific changeover but it is not possible to infer this from the data as they currently stand, as there is no method of indicating that each parameter in the 'to' subset is simply a later state of the same parameter in the 'from' subset. These data do not expressly state information regarding the *change* to a setting, and although such information is contained it is not made explicit. It is possible to manipulate the data to make this information explicit, in doing so generating a further set of input data.

These further parameters, labelled the 'delta' settings, give either a magnitude of change in the case of continuous variables, or a flag to state that something has changed in the case of discrete variables. It is possible to improve this flagging to give some idea of the magnitude of change.[27]

7.3.5 Data Understanding and Preparation

Witten and Frank rather informally state that in terms of preparing data for analysis '[t]here is no substitute for getting to know your data' (Witten and Frank, 2000). This point is made in the CRISP-DM model (CRISP-DM, 2000) where an entire stage of the 6-step process is devoted to what is labelled 'data understanding'. It is perhaps a self-

[27] In cases where there is a progression to the discrete values, for example 'cool', ''warm' to 'hot', it is possible to use a metric for the delta value that indicates the magnitude, where a change from 'cool' to 'hot' would receive a higher score than 'cool' to 'warm'. In the absence of further supporting information this development cannot faithfully be carried out in this case study

evident truth that whatever errors or omissions the training data are subject to will manifest themselves within the results of analysis of that data, although as the actual analysis is a form of conversion these errors within the results may not be easily attributable to the initial errors. It is important to remedy any obvious errors and address any evident shortcomings within the data prior to analysis (even by going as far as deleting any questionable data, Turney, 1995), as although the modelling algorithms can deal with a degree of error within the data these errors will still act to unduly influence the accuracy and stability of these algorithms, and will complicate the task of identifying genuine trends from specious ones.

IN the CRISP-DM methodology, the process of data understanding leads into data preparation, where the dataset is modified to facilitate modelling. In the CRISP-DM process model the data preparation stage comprises a number of tasks ranging from data selection through to deriving new parameters (for example, deriving *mass* from *volume* and *density*). This process tends to be extremely time consuming (Sethi, 2001), Westphal and Blaxton (1998) suggest that up to 80% of the total time for a DM project can be taken up during this stage.

It is important to ensure that data is complete and, as far as possible, free from obvious error. A number of authors refer to the process of outlier removal (Witten and Frank, 2000, Westphal and Blaxton, 1998, Reich and Barai, 1999[28], Turney, 1995[29]) where instances that appear to suffer from excessive error are manually deleted from the dataset. This involves an assumption regarding the 'correct' values for the data, and as such it is suggested that such courses of action should only be taken where there is knowledge and understanding of the data and the anticipated form of the data is well known.

Significant variance was noticed within the data. To illustrate this, the dataset was examined and 6 cases of a changeover between two specific products were extracted.

---

[28] Reich and Barai performed manual removal of outliers retrospectively, where those instances proving problematic during modelling were removed from the dataset

[29] Turney actually removes those instances that fall between two classes, hence they are not true outliers in the traditional sense, but are still instances that act to confuse modelling – they concede that such measures have not been theoretically validated but claim significant practical improvement

These 6 cases are considered to be six examples of ostensibly the same operation, although it is appreciated that, in the absence of any knowledge of the processes in question, this assumption may prove to be unfounded.

Case	Set-up Time (secs)	Run-up Time (secs)
1	475	701
2	474	674
3	1013	426
4	775	705
5	1184	436
6	14610	1853

Table 25 Recorded Durations of 6 identical changeovers

It can be seen from Table 25 that the set-up and run-up times for case 6 is considerably greater than for the other five cases. The set-up time in particular, at over 4 hours, is suggested to be the result of either an error in the recording process or an exceptional event such as a machine breakdown or modifications to the line (which is catered for in other data sections outside of the DM dataset and hence should not be included here). It is noted that cases 1 and 2 are similar in duration, as are cases 3 and 5. It is further noted that cases 1 and 2 have shorter setup but greater run-up than cases 3 and 5, and the total changeover duration is approximately 12,000 seconds for cases 1 and 2 and approximately 15,000 seconds for cases 3 and 5. Excessively large changeover duration can be explained by extraneous events, whilst there is little rationale for reduced duration. It would therefore appear that cases 1 and 2 represent changeover duration accurately, but cases 3 and 5 are artificially increased by means of extraneous processes. It is argued, however, that the differences could be due to different methods of working, where for cases 1 and 2 efforts were made to complete set-up quickly at the expense of run-up, whilst for cases 3 and 5 longer was spent on set-up to improve run-up duration. In this case, neither group of cases can be considered any more 'correct' than the other.

This cursory examination reveals the difficulty in simply collating data without consideration for its nature. It is considered unlikely in practice that examples such as case 6 would be removed from a dataset even if it is found to be the result of an extraordinary event, as engineers who are interested in improving the operation of the line would arguably be more interested in such extreme cases. In such situations there is a greater requirement to understand what causes excessive delay or difficulty within a line, therefore examples which specify the characteristics of such delays are most useful.

In the case of DM analysis it is much more useful to have a dataset that is representative of the typical[30] functioning of the line, and therefore cases which exhibit or describe characteristics that are exceptional are of limited use. It is only possible to analyse the characteristics of each changeover within the confines of those operations that are designed to occur within the changeover period. To qualify that using an example, characteristics such as the size of box or the linespeed are designed to be changed, whereas there is no legislation to deal with processes such as repairing accidental damage. These events may manifest themselves within the other recorded characteristics of the changeover, but are difficult to include as discrete characteristics in their own right. Such extraneous processes, whilst having a genuine influence on changeover time and of interest to those wishing to improve changeover performance, create difficulties within the DM process.

In the absence of any knowledge regarding the processes in use on the manufacturing line the task of identifying representative data is made difficult and imprecise. The only option available is to consider the changeover durations and infer from these if there is a possibility that events other than those designed to occur took place. It is much more preferable to be able to examine the source of each specific instance of data and compare the specifics of that instance to what should typically be seen in practice, or, to rephrase, to be able to distinguish what actually occurred at each changeover and from this to deduce which changeovers varied excessively from the desired process. Pyle (1999) labours the point that efforts should be made to uncover the genuine value or reason for error in preference to estimating the correct value. Alhoniemi (2003) suggests that in practice certain aspects of the data can indeed be identified as invalid based upon known external information. In this way, those cases that were known to have been influenced

[30] The term 'typical' is used with caution, it is used to signify functioning that conforms adequately and consistently to the specified design, and therefore not subject to extraneous, unplanned events. This runs counter to the argument regarding analysis, where the ambition is to deduce which changes and cause divergence from planned operations. A middle ground, where planned changes are overacted or require reparative measures, is in essence the area of interest in this study, which specifically excludes unplanned processes from consideration.

by external factors or other phenomenon that that are beyond the remit of this analysis can be identified and excluded from the dataset.

Level or Granularity of Data

There are different levels or layers of data granularity that can be explored, both in terms of the measure of a specific parameter and in the expansion of data to include more descriptive parameters. To illustrate the first aspect, as an example it might be possible to decide whether to wear a coat based upon a measure of temperature described in terms such as 'warm', 'cool' or 'hot'. However, whilst cooking, it is important to be able to measure the temperature of an oven to a much more specific degree, typically in units of 10°C. In a laboratory experiment, the measure of temperature might be of the order of 0.1°C. This form of data granularity is more likely to be fixed, as it is dependant upon the method of measurement, and cannot easily be modified afterwards.

The second aspect, that of expanding the data to create a more descriptive parameter set, is perhaps less fixed as data can be expanded after being measured and collected assuming there is further information to append. It is also entirely possible to reduce the level of granularity in the event of excessive detail or in an effort to reduce the dimensionality (number of parameters) of the data, in essence simplifying the task of the DM model – the use of Principle Component Analysis has been used to good effect in a number of DM analyses for this purpose (Faba-Perez et al., 2003). However it acts to 'interlink' parameters within the generated components and hence confuses subsequent model analysis, and is therefore excluded from this research. It is also feasible to increase the level of granularity, and it is this case which is of most interest.

The data used in this analysis was expanded from a set of product codes to include the line settings that would be used in the manufacture of a given product. In this way it is possible to compare the settings pre- and post-changeover to deduce which settings have been altered. Whilst this undoubtedly increases the information content of the data, there are still improvements that could be made. The definition of line settings before and after changeover does not indicate the actual individual operations that need to be carried out to achieve changeover, where it is feasible that certain setting changes will require a greater number of operations. The dataset, as it currently stands, does not indicate what is actually performed during changeover, merely what the settings are before and after changeover. It is possible to infer what the changes in settings are, but not what the

required operations are. This is problematic as it is suggested that each setting change will require numerous operations, and these operations cannot be modelled individually. This problem is common in DM applications, Pyle (Pyle, 1999) suggests that even detail data can be a form of summary data, and that the '..level of aggregation is a continuum.' Pyle further suggests, as a rule of thumb, that DM is best carried out using detailed data as opposed to summary or aggregate data, and it has previously been shown that the level of detail of the data used in this case study could be improved upon.

Consideration of a suitable level of data granularity is one that most acutely reveals the problems of inheriting a dataset, where there is no recourse if the granularity of the data is not optimal. It would be a relatively trivial task to expand the dataset still further to include the operations required to change settings if relevant engineering documentation were available.

7.4 Review of Methods of Handling Errors in Data

The review of literature carried out in Chapter 2 and Chapter 3 discussed the field of DM and its application in engineering. It is perhaps useful to further review the means of dealing with missing and erroneous data in some detail, which will be performed in this section.

Data are subject to error and omission as a result of measurement errors as well as handling or recording errors. This section will briefly review methods that have been proposed for dealing with such errors, and seek to identify those which might usefully be applied to this research. It is suggested that both forms of error are similar, where missing data can arguably be considered as a subset of erroneous data, with the exception that the form of error is immediately noticeable. However, the problems of missing data and of erroneous data will be addressed separately as these two forms of error are handled in notably different manners.

7.4.1 Missing Data

Many instances within a dataset may suffer from missing data, where the value for one or more parameters is unknown. There are generally 3 methods of dealing with missing values (Batista and Monard, 2002):

1. *Ignoring and Discarding Data.* This can be done instance-wise, where all instances suffering from missing values are removed, or attribute (parameter)-

wise, where parameters which can be demonstrated to be insignificant to modelling and that suffer from large amounts of missing data can be removed.

2. *Parameter Estimation.* Procedures such as the Expectation-Maximisation (Dempster et al., 1977) allow the complete dataset to be used to estimate missing values

3. *Imputation.* Missing values can be obtained by consideration of patterns and relationships within data.

It is highlighted that an understanding of the pattern of missing data is required in order to select an appropriate means of dealing with missing values. It is entirely possible that bias may be introduced into analysis if inappropriate methods are chosen (Schafer, 1999). For example, it is possible that an electronic measurement device will cease providing a measurement in extreme environmental conditions, for example if the temperature exceeds a certain threshold. In such a case it is reasonable to infer that where the measurement is absent the temperature was greater than the threshold value. By choosing to exclude all cases where there is no such measurement, the practitioner risks excluding all instances occurring at high temperature. This acts to bias the dataset towards instances occurring at low temperature.

Little and Rubin (2002) suggest 3 classes of randomness for missing data, indicating the extent to which the probability of an instance having a missing value is linked to either the remaining known parameters or upon the actual value that is missing[31].

1. *Missing Completely at Random (MCAR).* This describes the situation where the probability of a missing value does not rely upon the known values or upon the missing data.

2. *Missing at Random (MAR).* In this case the probability of a missing value may depend upon the known values, but not the missing value itself. For example, a pressure sensor may stop transmitting a signal when temperature exceeds a

[31] This refers to the probability of an instance being missing, not to the actual value of the missing parameter itself. Knowing this probability, a decision can be made regarding to deal with the missing value, for example by discarding the instance or by imputation.

threshold value, hence the temperature measurement provides some indication of the likelihood or probability of a missing pressure measurement.

3. *Not Missing at Random (NMAR).* The probability of an instance having a missing value may depend upon the missing value itself. For example, a temperature sensor may stop transmitting a signal when the temperature exceeds a threshold value.

In cases where the probability of noting a missing value is truly random (MCAR), any method of missing data treatment method may be used (Batista and Monard, 2002). In cases of MAR and NMAR, the use of methods such as ignoring or discarding data might introduce bias (Conklin and Scherer, 2003), as only instances describing a certain condition are included.

Many machine learning algorithms have methods of dealing with missing attributes. In the case of the DTI algorithms C4.5 and CN2, as discussed in section 12.3, two different methods are proposed. CN2 (Clark and Niblett, 1989) uses a simple method of replacing missing values with the mean (or mode, in the case of discrete values) of the population. C4.5 uses a more sophisticated probabilistic method, which may be considered to reside somewhere within parameter estimation and imputation.

It is suggested that efforts to address missing data using imputation will require some understanding regarding the randomness of the missing data, in terms of the 3 groupings proposed by Little and Rubin. There is the further consideration that Imputation is, in its simplest sense, an estimate of the likely measure for the missing value and hence will add more uncertainty to the data. This must be aligned with the potential improvement presented by increased data volumes, as those instances with missing data can be 'completed' and used within analysis.

Conklin and Scherer (2003) concede that the most common form of handling missing data is by deletion of instances that are affected by missing values, and this can be improved by the use of case-wise deletion, where only those parameters that will be used in the analysis will be checked for missing values. In this manner, greater amounts of data can be used as an instance will not be excluded if the missing values occur in parameters that are not of immediate interest. It is this method that was used in the analysis described in the following chapter, with the understanding that the potential of

imputation was noted near the conclusion of this research and thus after this modelling work was completed.

It is anticipated that engineering data may be of low volume, as evidenced in the case study described in the following chapter, and instance- or case-wise deletion of data with missing values is argued to be detrimental to the success of the DM analysis. In light of this, it is suggested that imputation will allow for greater volumes of data to be usefully employed in the analysis of such data. The field of imputation has been subject to considerable research interest in recent years, notable examples including the EUREDIT project (Chambers, 2000) which focused upon data editing and imputation and the project funded by the US department of transport (VDOT) which sought to impute road traffic information (Conklin and Scherer, 2003). It is argued that imputation within DM is still evolving, Batista and Monard (2002) suggesting that many Machine Learning algorithms are still naïve in their treatment of missing values, and hence imputation is suggested to be a technology with considerable potential in future DM research.

It is maintained that the optimal method of addressing missing values is via further examination of data stores, and efforts to uncover the actual value in question, as in this manner there is both an opportunity to obtain the actual measurement of interest or, if this proves impractical, to be able to infer the likely nature of the error and thus inform the estimation of a suitable value.

7.4.2 Erroneous Data

Errors in data can be identified either by means of a case-by-case examination of the data, based upon criteria that might be subjective or based upon empirical knowledge, or by some form of algorithmic analysis.

Case-by-Case Examination

The case-by-case examinations are perhaps most useful in identifying those instances whose parameter values that are outside of a permissible range and are thus clearly outliers, and are commonly used in practical applications (examples within manufacturing that incorporate such error handling include Turney, 1995, and Alhoeneimi, 2002). Alhoeneimi states that "..in practice, it is usual that part of the values are missing or some existing values are known to be invalid based on available external information.' Witten and Frank (2000) state that, when performing linear regression,

statisticians frequently remove outliers by visually identifying them, although they highlight that it is '..never completely clear whether an outlier is an error or just a surprising, but correct, value.' These two views indicate that it may be possible to remove an instance either by knowing that its value is infeasible or by observing that it lies some distance from other instances, and that the second case is somewhat subjective and runs the risk of eliminating genuine data.

Analytical Methods

The analytical methods are perhaps most useful where visual inspection does not reveal which data are subject to error, or where there is a risk of removing potentially useful data alongside those considered to be extraneous.

The analytical methods may be further subdivided according to their area of application. Many DM texts are aimed specifically at certain domains, and as such it is possible to 'second-guess' the manner of error and provide a method of automatically correcting any error. As an example, North American participants in a study might list their dates of birth in month, day and year (mm/dd/yy) form, whereas their UK counterparts might list theirs in day, month, year (dd/mm/yy). If this error can be anticipated then it can be dealt with automatically. This method does, however, require some knowledge of the form of error which is not always available.

A second division of analytical techniques relies upon modelling of the data to deduce incorrect instances, in terms of automatically filtering incorrect instances from the dataset, or by estimating likely values for parameters given some knowledge regarding the noise that the data is subjected to.

The first branch is simple both to visualise and to implement, and is best served by an example. In certain cases, it might be necessary to include a person's age as a parameter, and it is feasible that information relating to this might be in the form of either a date of birth or and age in years. If the requirement is for date of birth, it is a relatively simple task to create a script that will automatically convert an age in years into the year of birth. A further complication is introduced in cases of deliberately incorrect entries, where a respondent might refuse to enter an age and include an obviously false value. Whilst being obvious under cursory inspection, such errors can be easily overlooked and are typically dealt with by limiting a particular parameter to a specific range or form of value, and when this range is exceeded the instance can be flagged for further analysis.

The aggregation of data seen in this case study, where product parameters are obtained and linked to individual changeovers via the product code, it is important to ensure that the specific format of each parameter is consistent across the entire dataset. As an example, one of the parameters describes the length of the box to be filled, and it is feasible that data from different sources may use different units (perhaps centimetres and millimetres), hence a measure for checking for consistency is required.

The second aspect of analytical methods use Machine Learning algorithms to effectively pre-cleanse the data for use in a later analysis. Witten and Frank (2000) suggest the creation of a 'pre-model' to identify and remove misclassified instances, thus cleansing the data for use in a second tier of modelling. The idea of analytically cleaning data has also been implemented in the process monitoring field. The measurements obtained during process monitoring can be subject to significant error which take the form of both random and gross error, where the random error is the normally distributed error and the gross error is effectively an offset caused by phenomena such as leaks, electrical spikes and miscalibration (Kim et al., 1997). Process control is an on-line activity, and hence it is not feasible for operators to continually manually intervene to cleanse data. Data reconciliation is an automated method of performing this task (Kim et al., 1997) (Mingfang et al., 2000), (Himmelblau and Karjala, 1996). These methods aim to use knowledge of the expected measurement to remove noise from the actual measurement. The gross and random errors are typically handled separately, although by similar means (Mingfang et al., 2000). A limitation of these methods is that they require some knowledge about the pattern of noise or error within the data, typically assuming that the mean of the noise is zero and the distribution is Gaussian. Himmelblau *et al* (1996) describe the use of recurrent ANN, in the form of an Elman network as discussed in Section 12.1.6, as a means of data reconciliation which intrinsically caters for both gross and random error. Although the artificial noise added to an example case study is Gaussian, the use of an ANN does not intrinsically require that the noise be Gaussian, increasing its applicability to the research described in this thesis. Meert (1998) integrates the pre-cleansing model directly into the monitoring model by using two recurrent ANNs in series, although the actual pre-processing network is trained separately and can be used in isolation. In both cases, where noise and error are introduced artificially, improvements in the quality of the reconciled data are noted. As mentioned in the literature review recurrent networks have some concept of time,

explaining why they are widely used in process monitoring. In this research the analysis is carried out entirely off-line, and hence other Machine Learning algorithms may be used in place of recurrent ANNs.

7.4.3 Summary and Applicability of Methods of Handling Errors in Data

It has previously been suggested that, in an ideal situation, where errors are noted efforts should be made to find the correct value instead of attempting to correct or complete the data retrospectively. It is suggested that this will not always be possible, as the case can be foreseen where the erroneous data is the sole historical record of a particular event, and retrospective attempts to correct it cannot be undertaken in the absence of any supporting information. In the case of missing data it is possible that the value of the missing parameter might be obtainable from a separate disparate database or record, however it is suggested that some missing values are missing precisely because the value was not measured or has been deleted. In such situations an alternative is needed.

The method of imputation was suggested to represent a useful method of avoiding the deletion of instances with missing values, unfortunately this technique was noted only as the conclusions of this research were made. In this respect, the practice of case-wise deletion of instances with missing values was used throughout this research. Reconciliation was seen to provide a means of autonomously correcting erroneous data, however it relies heavily upon knowledge of the likely form of error, and as this knowledge is considered to be an unreasonable prerequisite for engineering data, reconciliation will not be used in this research. However, an adaptation of reconciliation will be tested, where engineering data will be analysed by a pre-model to remove instances that are considered to be outliers, thus cleansing the dataset for subsequent analysis.

It is argued that DTI methods will be more useful than ANNs in identifying outliers as such methods perform classification as opposed to numerical prediction[32]. In this respect, it is immediately clear which instances have been incorrectly handled, and these

[32] It is noted that ANNs can perform classification via the use of output nodes with step activation functions, however these have not been selected for use in this research.

can be excluded from subsequent analysis. It is equally possible to use methods of numerical prediction for this task, however a threshold criterion must be specified to indicate at which point to exclude poorly predicted instances. It is suggested that this will require some optimisation, significantly increasing complexity as the entire process of cleansing the data, creating the predictive models and evaluating performance must be carried out at each iteration within the optimisation.

7.4.4 Proposed Method of Handling Errors within Manufacturing Data

In the examples given in Table 25 it may be argued that case 6 is the result of extraneous, and hence unaccountable, events and is therefore unsuitable for inclusion within the DM dataset. The absence of any further information tempers this argument, as there is no indication of what constitutes a 'standard' changeover of this type – this argument merely uses as a base the logic that the remaining cases are 'correct' and the sheer magnitude of the differences in changeover duration therefore implies that it is in some way different or 'incorrect'. Whilst there is insufficient evidence to fully substantiate this claim, it is still maintained that the presence of outliers or of cases where unaccountable processes took place will act to detrimentally influence the results of the analysis, and hence further experimental work will be carried out to deduce which instances should be excluded from the DM dataset.

Two separate methods have been suggested as applicable for this task

- Removal of extreme values - instances where set-up or run-up exceeds a prescribed value (As performed when investigating manufacturing data in Reich and Barai, 1999)
    - There is an implicit assumption that excessive set-up or run-up duration is the result of extraneous influences and is hence 'incorrect'
    - There is a problem of defining the criteria for exclusion
        - Can result in loss of 'correct' data
- Use of DTI techniques to identify misclassified instances (based upon Meert, 1998, Witten and Frank, 2000)
    - Treat as outliers and remove from dataset

- Assumes that misclassification is due to error within data, not limitation or inability of DTI algorithm to classify correctly.

It is appreciated that removing cases with extreme values for run-up or set-up might result in genuine, but problematic, cases being excluded from modelling. However the primary goal is to reduce the occurrence of outliers or cases describing extraneous events within the DM dataset and hence such a compromise is noted and accepted. The second method is perhaps the more reasoned, as there is a genuine criterion for data omission. However it is unclear if an instance is to be excluded based upon genuinely spurious characteristics or because the DTI algorithm cannot adequately deal with the range of instances presented to it.

There are certain similarities between the second, data cleansing method and the technique of Boosting. Both of these methods rely upon identifying misclassified instances, and utilise these instances to present a modified training set for creating a second-generation model (and, in the case of Boosting, further generations of models). The major difference in the approaches is, in effect, that Boosting considers misclassified instances to simply be 'correct' data that could not be adequately encapsulated within the model, whereas the data cleansing method considers misclassified instances to be 'incorrect'. It has been shown experimentally that Boosting does not perform well on noisy data (Opitz and Maclin, 1999), arguably because it assumes that misclassifications are the result of limitations of the modelling and not of undue influence of noise. It has been shown that the data used within this case study is subject to considerable variation, and hence may be described as noisy, and it is therefore suggested that the majority of misclassifications will be due to this noise and not to limitations within modelling. In this respect, Boosting is considered to be of restricted suitability for use within this case study whilst data cleansing is argued to be of considerable use in excluding outliers.

7.5 Data Cleansing

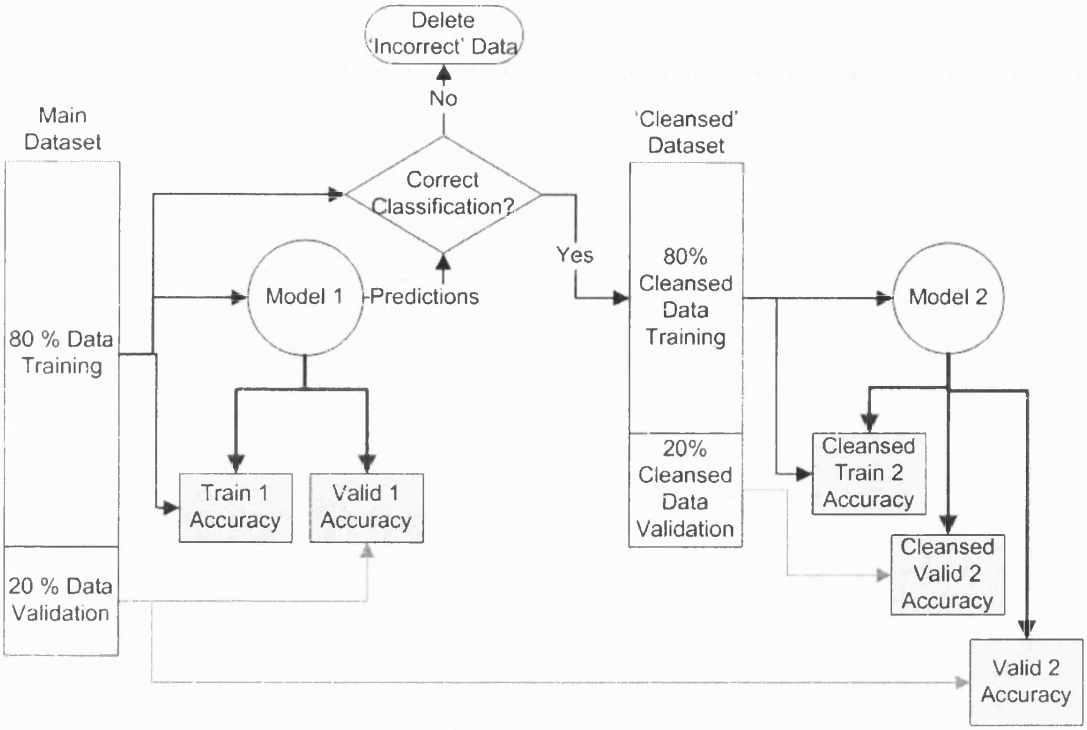


Figure 29 Schematic of Data Cleansing Analysis

Figure 29 shows the methodology that will be followed in the data cleansing stage. The initial dataset will be split into a separate training set, comprising 80% of the data, whereas the remaining data will form a validation dataset. An initial model (labelled Model 1) is then created and the training and validation accuracies established by propagating the separate datasets through the model and recording the percentage of correctly classified instances. The training data is then passed through the model once more, and only those instances that are correctly classified are passed on to form a second, cleansed dataset. This is split into separate training and validation datasets as before, and a second model (Model 2) created. The training and validation accuracies are then computed and a further test of accuracy is performed using the original, pre-cleansed validation dataset.

The raft of possible permutations of modelling algorithm, algorithm parameters and datasets could act to confuse the analysis, and hence only the data describing set-up was investigated using the standard algorithms settings for DTI (a pruning severity of 75 and a minimum number of records per child branch of 2). If successful, this approach could be expanded to encompass run-up data and ANN analysis and allow for fine-tuning of the modelling algorithm parameters.

7.5.1 Results of Data Cleansing

Model	Training Data Quantity (no of Instances)	Accuracy (%)			
		Train	Valid	Valid 2	Cross Validation
1a	358	70.67	57.3	n/a	63.5
2a	239	97.54	92	59.55	95
1b	358	67.88	62.92	n/a	58.7
2b	239	98.46	91.67	60.67	95.9
1c	358	64.25	65.17	n/a	61.2
2c	235	100	95.65	64.04	97.3
1d	358	64.25	48.31	n/a	59.5
2d	217	97.83	93.48	50.56	94.5
1e	358	66.48	66.29	n/a	60.3
2e	234	97.91	93.62	65.17	94.8

Figure 30 Accuracies of Level 1 and Level 2 Models

The results of the data cleansing analysis may be seen in Figure 30. The numbered models are arranged in pairs, where 1*x* refers to the *x*th example of a level 1 model (Model 1 in Figure 29) and 2*x* to the *x*th instance of a level 2 model (Model 2 in Figure 29). In total, 5 pairs of models were created from randomly sampled datasets in order to ensure that the specific composition of both the training and validation datasets did not result in large variations in terms of accuracy of the models.

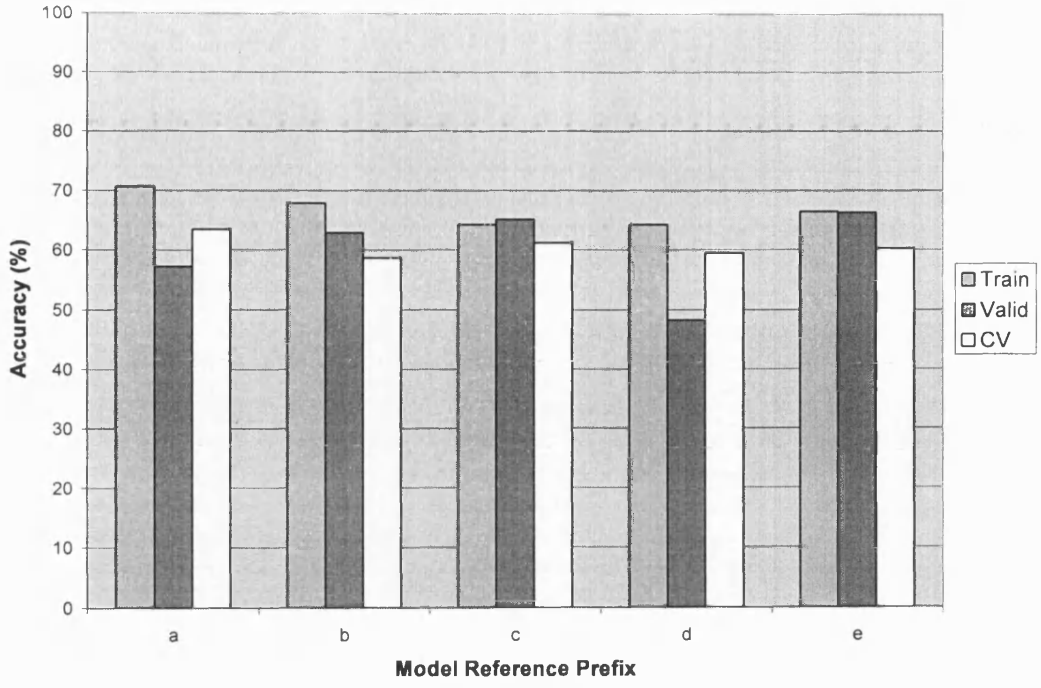


Figure 31 Comparison of Accuracy in Level 1 Model

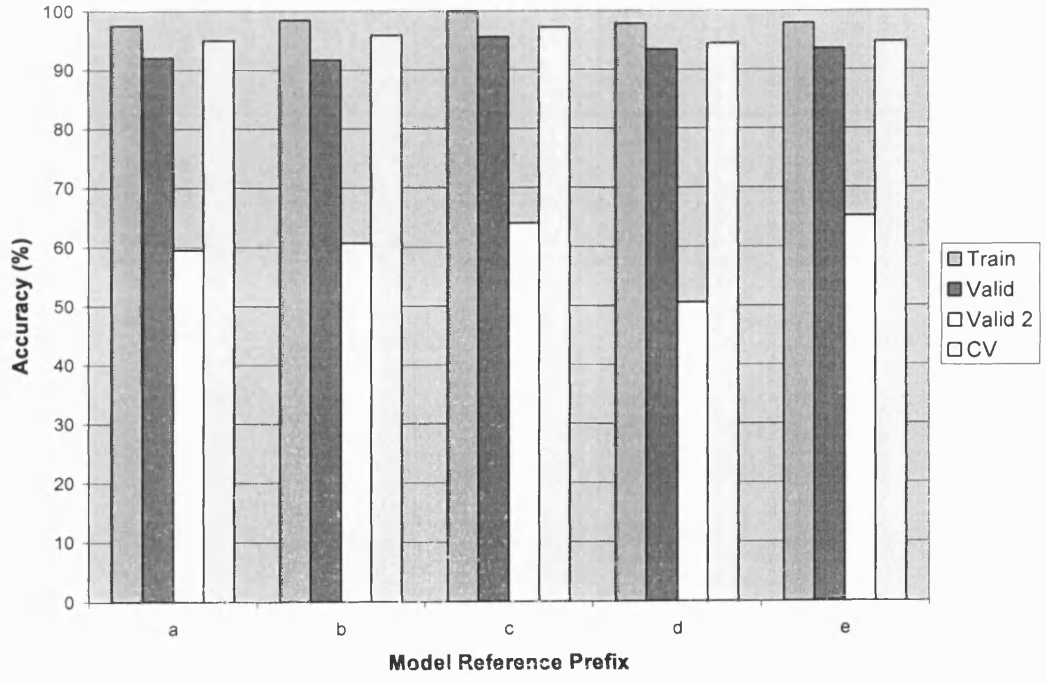


Figure 32 Comparison of Accuracy for Level 2 Model

The comparisons made in Figure 31 and Figure 32 show the variations in accuracy seen for the 5 sets of models for the level 1 and level 2 models respectively. It can be seen that the validation accuracies for the level 1 models display the most significant variance

across the 5 model sets, ranging from 48 to 66%. In comparison, the validation accuracies for the level 2 models are much more stable at around 95%, but when the original validation dataset (as created for the level 1 model) is passed through the level 2 model the variance in accuracy returns, going from 50 to 66%.

The results so far have indicated that the use of a data cleansing level 1 model acts to improve the accuracy of level 2 models, however it is still unclear whether such cleansing actually removes outliers or simply presents a reduced dataset that the level 2 model can more easily model. In this respect, it is not obvious whether the use of a data cleansing model actually acts to improve the veracity of the modelling or simply reduce the information that is imparted by it.

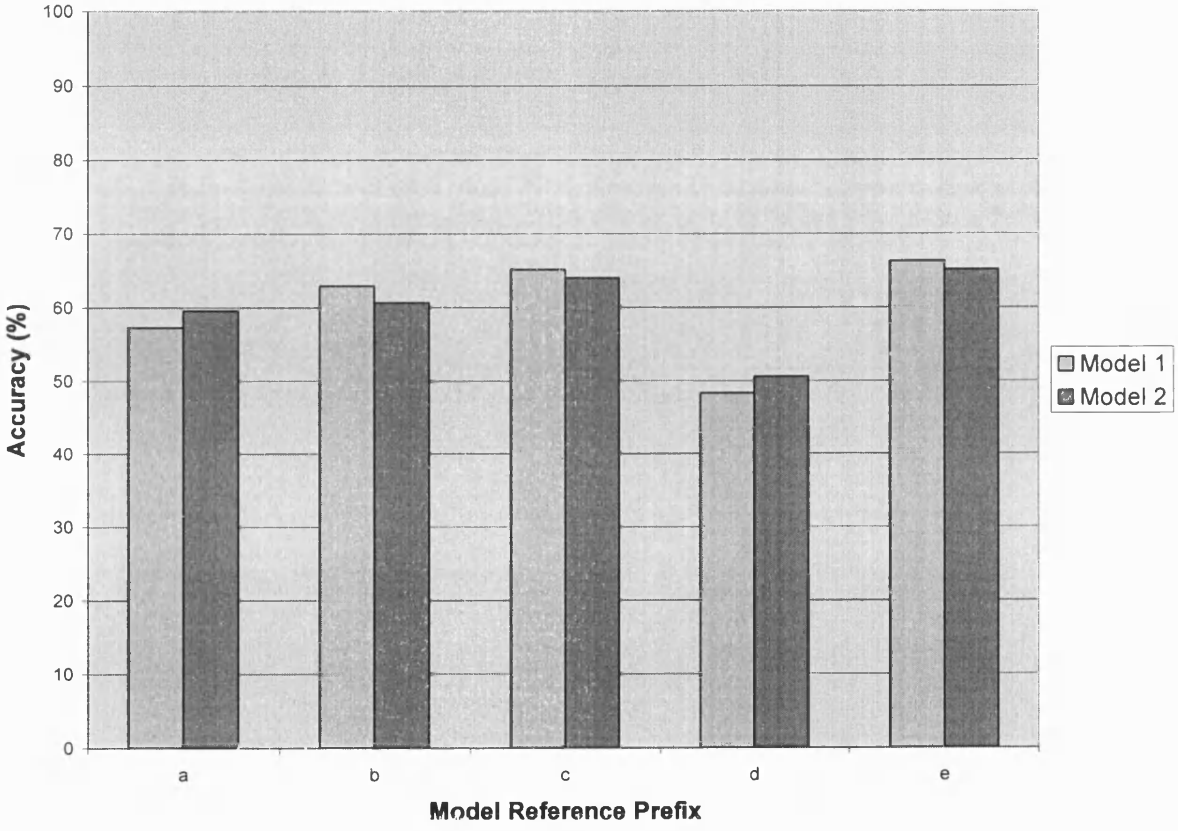


Figure 33 Accuracies of Level 1 and Level 2 Models using Uncleansed Validation Data

Figure 33 shows the validation accuracies of the 5 sets of models for both the level 1 and level 2 models using the validation dataset compiled from a random 20% sample of the original uncleansed dataset. It is immediately apparent that there is little difference between the level 1 and level 2 models, in 3 of the 5 cases the level 1 model has a higher

validation accuracy and the difference in accuracy between the two levels does not exceed 2.5%. It is suggested that, although the uncleansed dataset will necessarily contain noise and error, there will be data within it that is representative of the manufacturing process and hence it is reasonable to expect that the accuracy of the level 2 models should be greater as they have been created using more representative data. This statement hinges around the argument that the erroneous instances will be misclassified regardless of the quality of the model and that the representative instances will be better handled due to improvements in the model. Judging by the resultant accuracy levels this has not occurred. The lack of clear improvement suggests that the cleansing model is merely removing data that the algorithm cannot deal with effectively, and is not actually removing data that does not represent the process. It is for this reason that data cleansing via the use of a pre-model has not been adequately proven to be effective in improving classification performance.

7.6 Removal of Extreme Values

The manual identification and removal of outliers is preferably guided by some prior understanding of the acceptable range of values a parameter might take (Turney, 1995) (Alhoniemi, 2003). In cases where the acceptable range is not known it may only be inferred that the acceptable range is that which the majority of cases reside within, and that those with markedly different values fall outside of this range. This section describes attempts to identify and remove outliers by considering the distribution of output parameter values, and treating those instances that are outside of this distribution as outliers.

DTI modelling was once again used to investigate the modelling of set-up with both 'from-to' and 'delta' data.

7.6.1 DTI Modelling of Set-up

The first round of modelling sought to generate a control dataset where no instances were removed. The set-up was split into three equally populated ranges, A, B and C. The boundary between ranges A and B was 780 seconds and the boundary between ranges B and C was 2000 seconds.

Model number	Boosted?	Train	Prune severity	Min no of objects	Accuracy %		CV %
					Train	Valid	
1	N	80%	75	2	70.69	57.3	64.5
2	N	80%	90	2	68.44	59.55	65.7
3	Y	80%	75	2	63.97	55.06	61.2
4	Y	80%	90	2	66.2	59.55	63.9
5	N	90%	75	2	68.73	54.55	62.7
6	N	90%	90	2	66.25	54.55	61.3
7	Y	90%	75	2	65.26	54.55	61.3
8	Y	90%	90	2	67.25	52.27	63
9	N	100%	75	2	67.56	n/a	63.6
10	N	100%	90	2	64.66	n/a	64.6
11	Y	100%	75	2	65.55	n/a	61.8
12	Y	100%	90	2	64.88	n/a	62.8

Figure 34 Results of Initial Modelling for Set-up using 'From-to' Data

Figure 34 shows the results of the initial modelling. The cross-validation accuracies for each model are good, each above 60%. The models may be logically arranged in 3 groups, where the 1st group (models 1 to 4) were constructed using 80% of the available data with the remaining 20% used for validation purposes, the 2nd group (models 5 to 8) were constructed using 90% of the available data and the final group (models 9 to 12) were constructed using all of the available data and hence have no corresponding validation accuracies. In only 1 group did Boosting act to increase accuracy, where the accuracy for model 8 is 63% and the accuracy for models 5 and 6 are 62.7% and 61.3% respectively. The highest accuracies in the remaining groups are from non-Boosted models.

Identification of Outliers

The identification of outliers was performed manually, where histograms of the changeover period were plotted and those that were judged to fall outside of the main distribution were considered extraneous and hence deleted. It was noted in section 7.4.2 that such an approach is by nature subjective.

The changeovers could be generally broken down in two types, size changes and brand changes, each with different durations of changeover, and hence these were split into two separate datasets and examined separately.

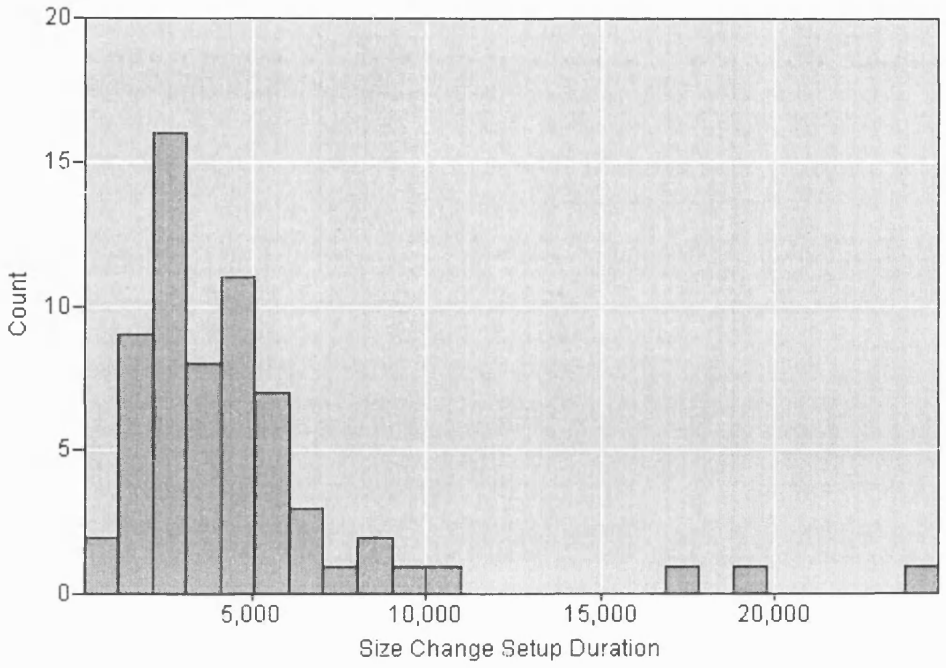


Figure 35 Histogram of Setup Duration for Size Changeover

Figure 35 shows the histogram of setup duration for a size changeover. It can be seen that there is a distribution centred around approximately 2,500 seconds, and whilst there are obvious extreme values beyond 15,000 seconds there is a skew to the distribution.

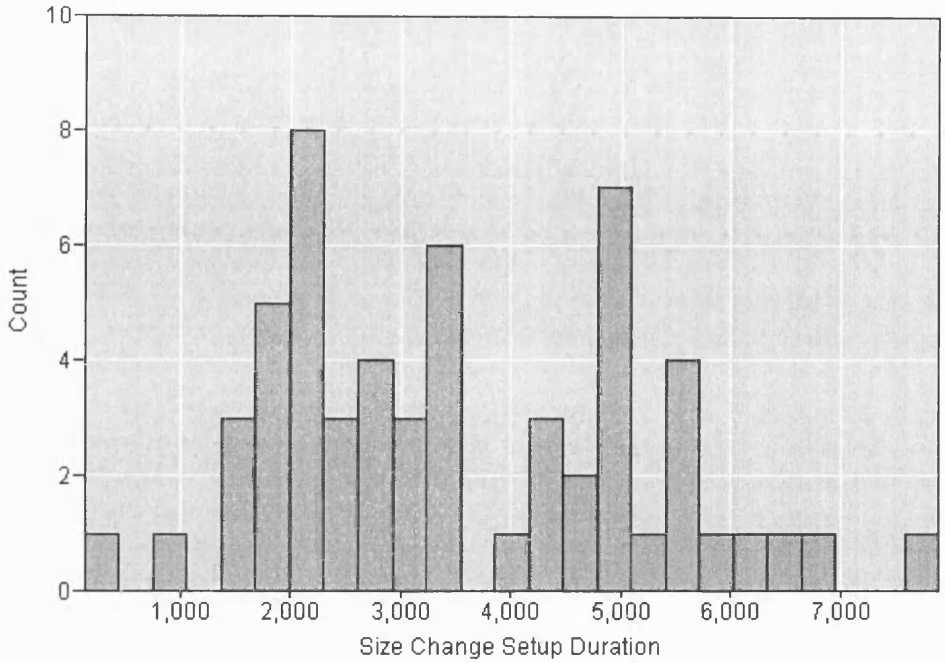


Figure 36 Histogram of Setup Duration for Size Changeover truncated at 8,000 Seconds

Figure 36 shows the distribution for setup duration for size changeover where the data is truncated at 8,000 seconds, and any instances with durations beyond that limit are removed from the dataset. This limit was deduced visually, consideration of Figure 36 suggests that a value of 7,000 seconds may also have been used. What is clear, however, is that the histogram represents a more coherent distribution than the histogram seen in Figure 35. It is suggested that this distribution is multimodal, but there is less evidence of skew and the extreme values have been removed.

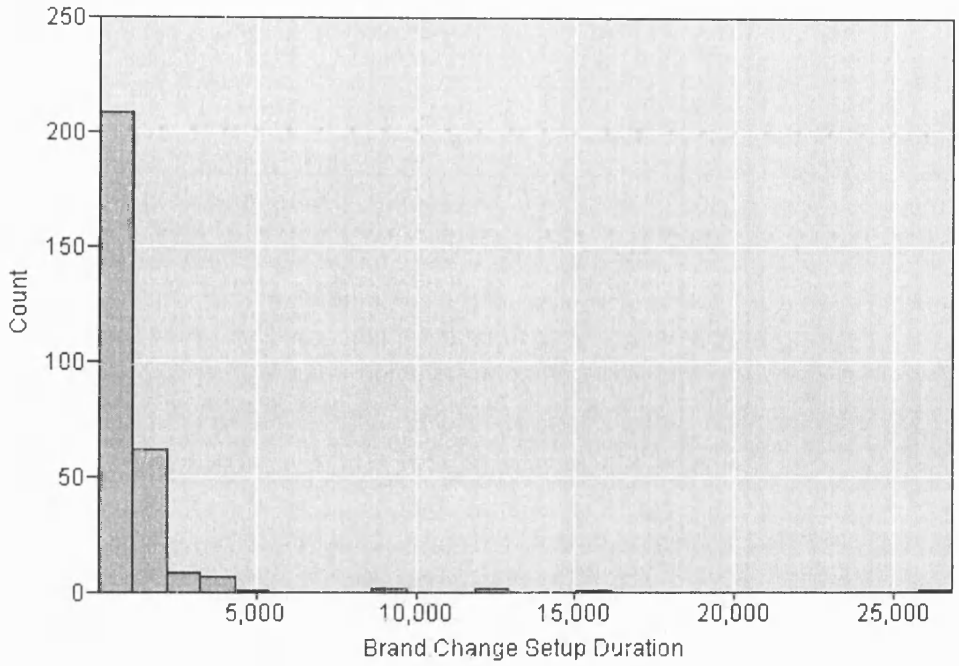


Figure 37 Histogram of Setup Duration for Brand Changeover

Figure 37 shows the histogram for brand changeover, where extreme values skew the distribution hugely.

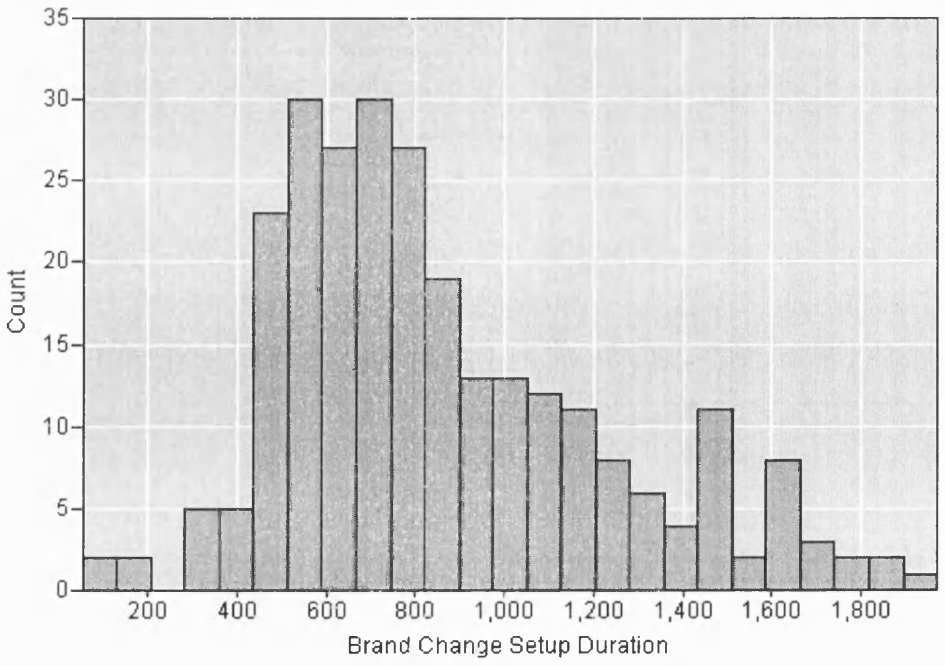


Figure 38 Histogram of Setup Duration for Brand Changeover truncated at 2,000 seconds

Figure 38 shows the histogram for brand changeover where those instances with duration greater than 2,000 seconds have been removed from the dataset. It can be seen that the skew to the distribution has been greatly reduced by this measure.

Combination of Truncated Brand Change and Truncated Size Change

Model	Boosted?	Training data amount	Prune severity	Min no. of objects	Accuracy		CV %
					Train	Valid	
1	N	80%	75	2	64.48	53.12	55.6
2	N	80%	90	2	61.39	53.12	53
3	Y	80%	75	2	64.48	53.12	56.8
4	Y	80%	90	2	61.39	53.12	52.9
5	N	90%	75	2	62.59	43.75	51.2
6	N	90%	90	2	62.2	43.75	48.4
7	Y	90%	75	2	63.23	37.5	51.2
8	Y	90%	90	2	61.51	40.62	51.9
9	N	100%	75	2	65	n/a	52
10	N	100%	90	2	61.3	n/a	52
11	Y	100%	75	2	64.09	n/a	52.3
12	Y	100%	90	2	61.3	n/a	53.8

Figure 39 Results of Initial Modelling for Set-up using 'From-to' Data with removed outliers

The distributions shown in Figure 35 to Figure 38 describe either brand changes or size changes. Once the extreme values were removed the data for both brand change and size change were merged together to form one dataset giving a total of 323 instances. The range boundaries had been previously decided to equally distribute the instances between the 3 ranges, and since the majority of the longer duration instances have been removed it was necessary to reconfigure these boundaries. The values decided upon were 715 seconds and 1160 seconds for the boundaries between ranges A & B and Ranges B & C respectively.

Figure 39 shows the results in terms of model accuracy of the DTI analysis of the data describing set-up duration with removed outliers. It can be seen that the CV stays relatively constant across all 12 models, ranging from 48.4% to 56.8%. These values are not as high as those seen prior to the removal of extreme values, which ran from 61.2% to 65.7%. If the removal of extreme values is of benefit to modelling it is necessary that such an approach acts to improve accuracy. The reduction in accuracy seen in this

example suggests that removing instances that fall outside of the main distribution does not increase modelling accuracy.

This lends further weight to the argument that retrospective elimination of data that is considered extraneous is not always the best course of action, as it neglects the idea that instances with extreme values might have those extreme values for a legitimate reason. In many cases the extreme values might fall outside of what might reasonably be expected, or there are established criteria which define what range of values is acceptable, and armed with such information it is acceptable to remove data. It is argued, however, that the lack of knowledge of what constitutes an acceptable boundary has hampered this analysis.

A further observation, and one that was also expected, is that Boosting has minimal influence, and the influence it does exert acts to reduce model accuracy. It has been stated earlier that the use of Boosting can act to decrease overall accuracy in the presence of noisy data, and as highlighted in the comparison of the time taken for 6 identical changeovers (as shown in Table 25), the data used in this case study is subject to significant levels of noise. It is for this reason that Boosting will be neglected as a viable method of improving model accuracy for the remainder of this case study. It has been noted in Chapter 6 that the information extracted from a Boosted DTI model is superior to an un-Boosted model, and I cases where the information is to be deployed it is recommended that Boosting be used in at least an exploratory role within DTI modelling.

Confusion Matrices

The accuracies of the DTI models were not as high as those created within the analytical model analysis described in Chapter 5. This is perhaps to be expected as the analytical model analysis investigated data with minimal noise, whereas this case study uses data obtained from an industrial process with resultant errors and noise.

		Predicted Value		
		A	B	C
True Value	A	12	7	2
	B	9	10	2
	C	3	7	12

Figure 40 Validation Data Confusion Matrix for Model 3

Figure 40 shows the confusion matrix for the validation data classifications using model 3, the model shown in Figure 39 to have the greatest cross-validation accuracy. It can be seen that, of the 30 misclassified instances, 25 were misclassified into adjacent ranges, suggesting that there is a genuine pattern to the data but that either the pattern is 'weak' or the range selections are suboptimal.

'Delta' Values

Model number	Train	Prune severity	Min no of objects	Accuracy		CV %
				Train	Valid	
1	80%	75	2	60.34	61.8	57
2	80%	90	2	59.22	60.67	55.3
3	90%	75	2	60.55	59.09	54.1
4	90%	90	2	59.55	59.09	55.9
5	100%	75	2	63.76	n/a	54.6
6	100%	90	2	61.52	n/a	53.7

Table 26 Results of Initial Modelling for Setup using Delta Values

Table 26 shows the results of modelling set-up using Delta values. The cross-validation accuracies are good, ranging from 53% to 57%, but these are not as good as for the equivalent models created using 'from-to' data, where the cross-validation accuracies ranged from 61% to 66%. It is suggested that these results are comparatively poorer as the flags used to indicate a change in discrete values do not indicate the magnitude of change, instead merely highlighting a change has occurred, and is therefore a reduction in the information content of the data.

Model number	Train	Prune severity	Min no Of objects	Accuracy		CV %
				Train	Valid	
1	80%	75	2	58.69	46.88	44
2	80%	90	2	52.51	50	49.8
3	90%	75	2	61.59	50	52.9
4	90%	90	2	55.33	46.88	48.5
5	100%	75	2	60.37	n/a	52.3
6	100%	90	2	54.49	n/a	49.2

Table 27 Results of Initial Modelling for Setup using Delta Values with removed outliers

Table 27 shows the results of a repeat of the setup modelling using delta values with the outliers truncated. The outliers were removed in exactly the same manner as for the 'from-to' analysis, and the results are included here to enable a comparison between

models made both with and without the extreme values removed from the data. It is intended to use these comparisons as verification of the observations made previously.

It can be seen that the Cross-Validation accuracies of the models range between 44% and 53%, compared to 53% to 57% for the models created using the complete dataset. It is therefore concluded that the removal of extreme valued data does not improve model accuracy. This conclusion is consistent with observations made of the models created using 'from-to' data.

7.6.2 DTI Modelling of Run-up

Run-up was also included in this analysis in order to extend the range of the experiment. An identical round of analysis to set-up was performed, once again modelling both with and without the removal of extreme values. The run-up durations were different to the set-up durations, hence a new set of boundary values were required. These were placed at 555 seconds and 1200 seconds respectively, giving 3 equally populated ranges.

Model number	Train	Prune severity	Min no of objects	Accuracy		CV	
				Train	Valid	Mean	SE
1	80%	75	2	61.41	43.96	45.9	2.4
2	80%	90	2	55.16	34.07	44.1	3.3
3	90%	75	2	59.42	48.89	44.2	1.4
4	90%	90	2	56.28	55.56	42.3	1.5
5	100%	75	2	57.08	n/a	44	1.6
6	100%	90	2	55.12	n/a	44	1.8

Table 28 Results of Modelling for Run-up using 'From-to' Data

Table 28 shows the results of the DTI modelling for run-up using 'from-to' values. The cross-validation accuracies are consistent at between 445 and 46%, however these values are considerably lower than those seen in the modelling of set-up reflecting the inherent noise in the measurement of run-up.

Model number	Train	Prune severity	Min no Of objects	Accuracy		CV	
				Train	Valid	Mean	SE
1	80%	75	2	48.43	41.27	41	2.5
2	80%	90	2	51.18	38.1	35.8	2.7
3	90%	75	2	55.94	45.16	41.2	3.4
4	90%	90	2	51.4	45.16	35.7	2.7
5	100%	75	2	53.37	n/a	40.4	2.2
6	100%	90	2	55.21	n/a	41.3	3.1

Table 29 Results of Modelling for Run-up using 'From-to' Data with outliers removed

Table 29 shows the results of modelling of run-up where the extreme values within the dataset were removed. The boundaries for the ranges were adjusted to 500 seconds and 900 seconds to account for this manipulation of the data. The method of removal was identical to that used in the modelling of set-up. Comparisons to Table 28 reveal that the removal of data with extreme values reduces the accuracy of analysis, the range of cross-validation accuracies being between 35.7% and 41.3% compared to between 42.3% and 45.9% for modelling involving the entire dataset. This further verifies the observations made in the analysis of set-up.

7.7 Concluding Remarks

This case study sought to identify if errors within manufacturing data could be removed or treated prior to DM analysis. Significant error was noted within the data, noise which could be attributed to the heuristic nature of changeover and graphically illustrated by the significant variance in the duration of ostensibly identical changeovers. It was argued that many of the more extreme durations of changeover could be attributed to external operations or events, however it is maintained that it is not ideal to retrospectively deduce which instances within the data are subject to error unless there are guidelines or a prior understanding of the nature of the data. Such an observation was made by Bucheit *et al* (2000) who concluded that instances describing large temperature variations in a cooling system, which were outside of expected values, could not be discounted as it could not be ascertained '...what is truly noise, and what is variation that is essential to developing a complete model of the system.' Pyle (1999) suggests that efforts should be made to

positively identify outliers from further analysis of the source of the data, and in light of the results presented in this chapter it is suggested that, whilst effort-intensive and not always practical, such an approach should be considered as most favourable.

Efforts to algorithmically remove erroneous data were argued to be inconclusive. A 'pre-model' was constructed which modelled the complete dataset and removed instances that it failed to correctly predict or otherwise map. The reduced dataset could then be analysed, where the assumption was that the pre-model had removed the erroneous data. It was suggested that this pre-model in fact simply rejected instances that it was incapable of encompassing within a model, in effect the reason for rejection of an instance was not an error within the instance but a limitation of the pre-modelling algorithm.

A further method of identifying erroneous instances was to examine a histogram of the changeover duration and remove those instances that were seen to be located away from the main distribution of data. This was seen to reduce the accuracy of DTI modelling, suggesting that genuine instances that were subject to extraordinary processes were mistakenly being considered as erroneous.

It is suggested that the most reliable method of removing error from a dataset would be approach it from the opposite direction, in effect proactively applying the suggestion of Pyle, by actually recording or otherwise indicating where external processes have influenced a changeover and deciding whether to omit the instance based upon that knowledge. Failing this, it is argued that a greater understanding of the nature of the data would allow for those instances that fall outside of what typically might be expected to be identified and a decision then made as to their veracity. The data was obtained in a pre-compiled state, and hence the learning process that would accompany a data collection exercise did not take place. The CRISP-DM methodology (CRISP-DM, 2000) suggest that a DM implementation should begin with efforts to understand both the business practices of an organisation and the nature of the data that describes those processes. It is suggested that the problem of changeover improvement has been understood in some depth, but the exact mechanisms and processes in place at the manufacturing line have not been understood in any great detail. As an example, there are references to changing the height of a box but there is little indication either of what is meant by this or by what mechanism could be used to achieve this, and an understanding of what is meant can be obtained only from a generic understanding that soap powder boxes are of different heights and that assembling and sealing the box must require different settings for boxes

of different height. If more was understood, it is argued that problems in the data would become more apparent and the process of eliminating or otherwise dealing with erroneous data could be carried out using this knowledge.

The error within modelling was argued to be due to two features of the data. It has been highlighted that the data contains noise, where the actual measurements of set-up and run-up are both variable and, particularly in the case of run-up, difficult to measure accurately. A further issue is the 'summary' nature of the data, where the data only describes the settings of the line for each product and, by considering the change in product, the settings that are altered during changeover. This gives little information regarding the actual operations that take place. It is argued that a given parameter might have multiple settings and changing between one subset of these settings might involve trivial operations, such as moving levers or pressing buttons, whereas changing to another subset of settings might involve complex operations, such as the replacement of part of the manufacturing line. It is also possible that certain settings use identical operations, for example changing the height and width of the box might require the manipulation of the same lever. The use of 'summary' data does not include information on either of these cases. It is suggested that the granularity or aggregation of the data is, in effect, continuously variable, and that it would be a relatively trivial matter to increase the level of detail in the data if the appropriate information were known. In this case study such information would consist of details regarding the precise operations necessary to go from an initial given line setting to a further given setting, and it is further suggested that such information could be readily derived given suitable access to either operators or procedures.

Chapter 8 The Suitability of Manufacturing Data for DM Analysis

The early chapters of this thesis described the development of a DM modelling approach that could be used to model manufacturing data. The previous chapter sought to identify if error within manufacturing data could be dealt with retrospectively, such that the accuracy of DM modelling could be improved. It was concluded that retrospective consideration of manufacturing data is not ideal, and argued that an understanding of the methods of data generation (and of the manufacturing processes such data describe) are necessary if erroneous instances are to be identified. This chapter takes this idea as its central focus, and seeks to investigate the methods of manufacturing data generation. This investigation seeks to identify how error is introduced into data, and to provide a framework by which manufacturing data generation procedures may be evaluated (or indeed generated) in order to establish how suitable the generated data is for DM analysis.

An investigation of manufacturing operations at an industrial power generation gas turbine manufacturing plant was undertaken, with focus primarily upon understanding how data is generated and what it describes. Allied to this were efforts to deduce how to steer analysis towards areas that would be of benefit to engineers within the organisation. Some early modelling results are presented, as this modelling used a different method of information extraction to that proposed in Chapter 5. The DM methods used in this case study represent early development of the methods proposed in the previous chapters, indeed much of the work described at the start of this thesis was in fact carried out towards the end of the period of research.

Of particular interest in this case study is the nature of the data itself, and in the methods used to generate and record such data. The problems that might be seen in manufacturing data are addressed and a method of investigating the data to ensure that it is representative of the process is discussed. Lessons learnt regarding the generation of manufacturing data are presented within a hierarchical framework, with illustrative examples given of issues seen within each of the three tiers in the hierarchy. This

hierarchy proved useful in deducing the extent of the engineers' understanding of the methods of data generation, highlighting where further effort will typically be required in ensuring that data of sufficient quality for DM analysis is generated.

8.1 Structure of Chapter

This chapter is divided into two parts. The first part introduces the nature of the case study, defining the nature of the manufacturing operation under examination and highlighting the manufacturing process and areas of data generation. A brief summary of the initial modelling will be given, where an alternative method of evaluating the information within DTI models is proposed.

The second part investigates the nature of the manufacturing data, identifying where errors are introduced (and hence how they may be eliminated) and how it can be ensured that such data is suitable for DM analysis The manufacturing data is evaluated on three tiers or level of detail, and issues within each of these three tiers are identified. Examples of problems or issues within the data at each tier are given to clarify the nature of these tiers. The prevalence and severity of data issues within each tier is then discussed, and this is contrasted against the understanding an engineer has of the data within each tier. In this way, it can be identified where an engineer might not be aware of significant problems within the data, and therefore where data that is not representative of the manufacturing process might be introduced into analysis.

8.2 Scope of Case Study

The case study presented here is concerned with the manufacturing process of an industrial power generation gas turbine, and whilst the function of the turbine is not subject to scrutiny it is perhaps beneficial to briefly cover the basic operation and construction. The critical performance parameters that are of importance during the manufacturing process are not always the performance parameters that would be of interest in actual usage (to illustrate, when manufacturing a car, the top speed is not a manufacturing concern, whereas meeting a suitable voltage over the spark plug gap might be), hence emphasis will be placed more upon the performance and characteristics of the turbine during manufacture than upon the turbine in use.

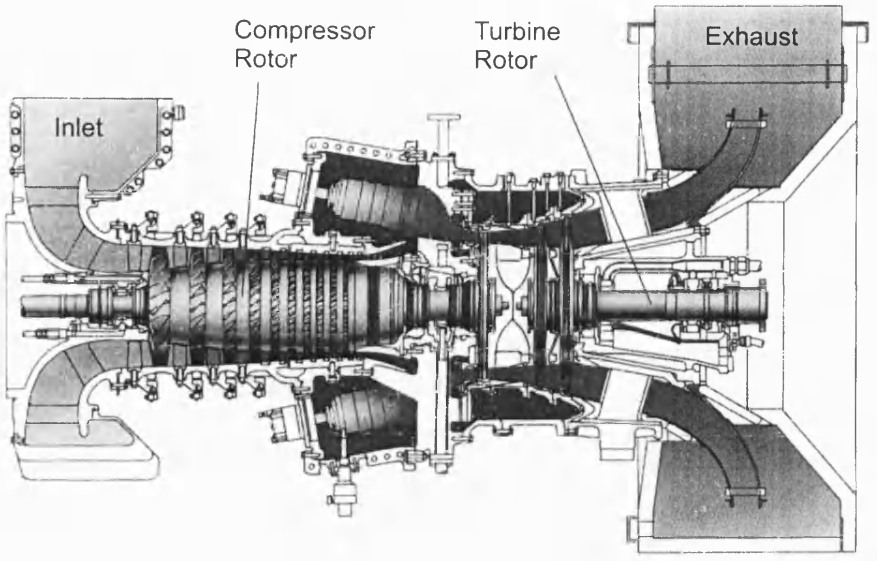


Figure 41 Cross-Section of Gas Turbine (Alstom, 2000)

Figure 41 shows a cross-section of a Gas Turbine of the type investigated in this case study. The Gas Turbine functions by compressing gas using a series of rotating blades located at stages along a rotor, adding fuel to this compressed air and igniting it within a combustion chamber, and then harnessing the gas expansion via a series of turbine blades. The torque generated by the turbine is then used to drive the compressor and an external gearbox.

The compressor and turbine are typically located on a single rotor, Figure 41 shows an exception to this where the power turbine is located on a separate shaft (identified as the Turbine Rotor) which allows the main shaft to run at constant speed whilst the output shaft is free to run at variable speed. These rotors are fabricated from solid metal discs, and rings of blades are clamped between each set of discs. Control of the concentricity and swash of the discs (degree of flatness between the mating surfaces of each disc) is essential in ensuring that the overall balance of the rotor is good, and to this end manufacturers expend significant effort to ensure that rotors do not suffer excessive unbalance[33] (Rolls-Royce, 1986). The mass of the rotor might be of the order of 1250kg

[33] The term unbalance is a noun typically used within the Gas Turbine industry to refer to the state of the rotor in terms of a measure of out-of-balance forces.

and the rotational speed around 16,000 revolutions per minute, and as the unbalance increases as a square of speed the potential for huge forces due to this unbalance is great.

It is this area that the case study will focus upon, in effect the vibration of the rotor will be considered as the critical performance parameter. Efforts will be made to model the vibration seen in the complete engine under final testing, based upon a series of input parameters taken from earlier measurement and testing procedures.

8.2.1 Manufacturing Process and Areas of Data Generation

In keeping with the CRISP-DM methodology, it is essential to generate an understanding of the business and data processes as a matter of priority. It is intended to focus this chapter upon issues relating to the nature of manufacturing data, hence the manufacturing process will be covered in some depth.

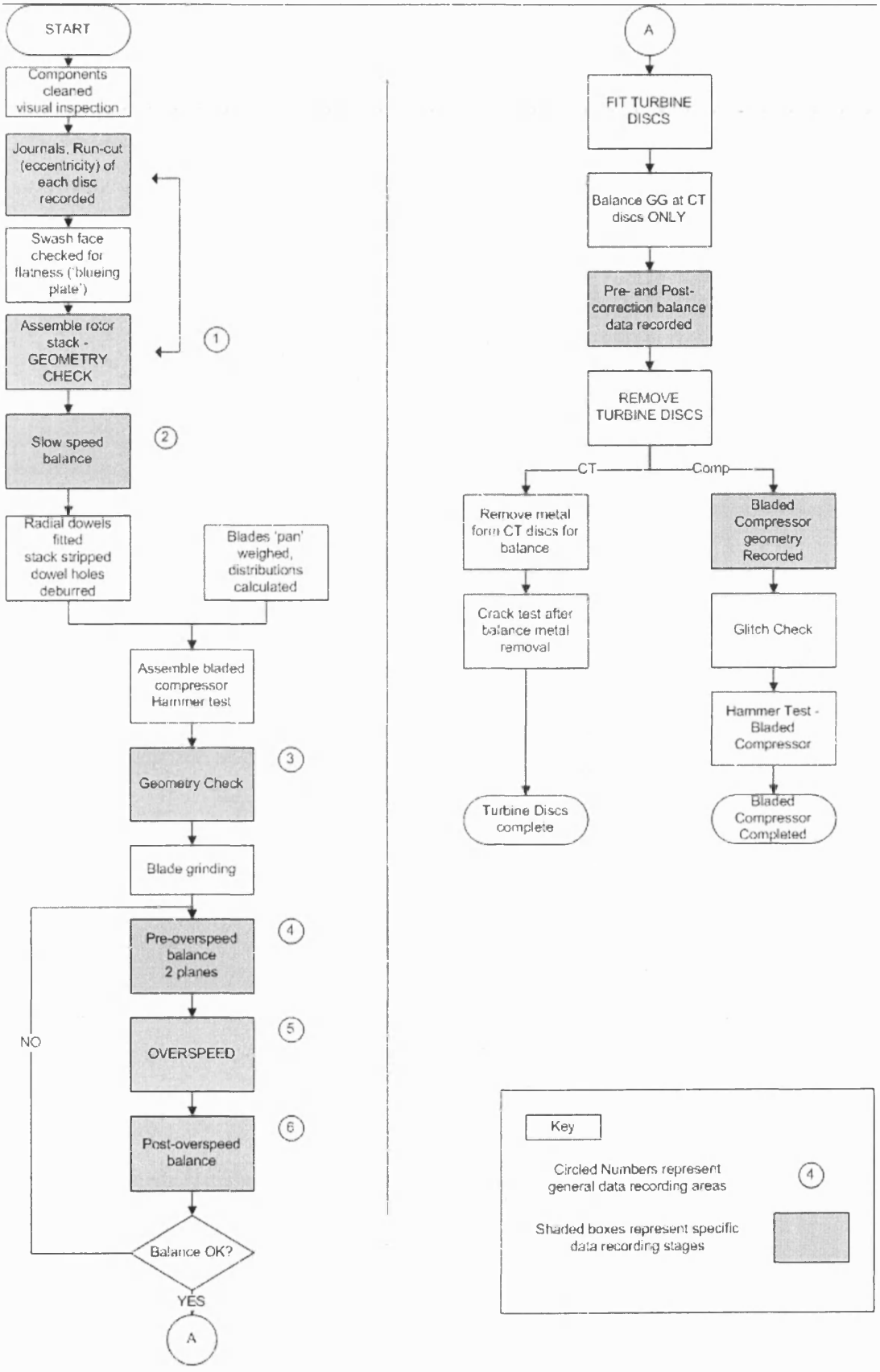


Figure 42 Simplified Gas Turbine Manufacturing Process

Figure 42 shows a simplified gas turbine rotor assembly process, which is described in detail as this is where the dynamic performance of each rotor is decided. The full flowchart is not included for reasons of confidentiality. Other factors or processes that decide rotor vibration in practice that are not included in this flowchart are the properties of the journal bearings and the aerodynamic properties of the interior of the engine casing. There is little data to describe these factors, and it considered by the engineers managing the rotor final assembly that the majority of problems with a gas turbine are introduced by the manufacture and assembly of the rotor. It is for this reason that attention will be focused upon this area.

8.2.2 Data Collection stages

The stages of data production can be broken down into 7 main areas[34], of which 6 are illustrated in Figure 42 and the 7th is the Final Engine Test vibration level which is carried out upon the complete rotor. The seven areas are as follows:

- Unbladed Geometry
  - The measure of the journal and individual disc run-outs and disc eccentricities of the assembled unbladed compressor rotor
- Unbladed Balance
  - The measurement of the unbalance seen at the two bearing positions for the unbladed compressor rotor
- Bladed Geometry
  - The eccentricity of the compressor rotor discs when the blades have been added (due to physical restrictions, not all discs can be measured)
- Bladed Balance
  - The measurement of the unbalance seen at the two bearing positions for the bladed compressor rotor immediately before entering overspeed
- Overspeed Vibration Level
  - The vibration characteristics of the bladed compressor rotor when rotated at 125% of designed operating speed within a vacuum chamber
- Post-Overspeed Check Balance

[34] Other stages of data production exist, for example pre- and post-correction balance, however many of these have sparse data or simply record whether deviation from previous measurements is evident after certain operations.

- The measure of the unbalance seen at the two bearing positions for both the bladed compressor immediately after overspeed and the trim-balanced complete rotor

- Final Engine Test Vibration Level
  - The complete rotor is fitted into a test engine core and the engine is run. Engine vibration level is monitored throughout a test cycle.

8.3 Initial Modelling

This chapter focuses to a large extent upon the methods of data generation, however collaboration with the manufacturing company meant that modelling was an important aspect of the research as successful modelling would have significant commercial benefits.

An initial exploratory stage of modelling was undertaken, assisting both in understanding how the modelling algorithms functioned and of the nature of data generated during manufacturing. The modelling was carried out using the DTI algorithm as this was considered to be the most useful for an exploratory exercise, due to simple and rapid training and the minimal number of algorithm parameters that would require optimisation.

8.3.1 Consideration of Suitable Areas of Investigation

The decision regarding which areas to investigate was made by considering three factors

- The availability of data in the input and target areas
- The accuracy of the data
- The usefulness of the test in terms of improving the manufacturing process

Each of these three factors required evaluation across the 7 areas of data collection described previously. The two data factors, accuracy and availability, can only be evaluated upon successful compilation of a dataset, which (in an organisation of such size) is a significant undertaking. It is suggested to be more practical to evaluate which areas will yield the best results beforehand, or perhaps more accurately eliminate those areas that are of little use, and then to focus data collation efforts in those areas that are considered to be the most likely to offer the best results and the most useful information.

These three factors were analysed for each pairing of the 7 data collection areas described previously. Scores for the three factors were deduced by consultation with a group of 10

engineers, comprising the rotor manufacturing group manager, rotor assembly manager and a range of engineers from the rotor manufacturing group. The subjective opinion of this committee was translated into a score of between 1 and 10, where a higher score indicates greater accuracy, availability or usefulness. The committee for this scoring process comprised the managers overseeing the rotor assembly and the engineers responsible for each specific area.

	Accuracy	**Availability**
Data Source	Score 1 - 10 (10 is Best)	Score 1 - 10 (10 is Best)
Unbladed Geometry	5	1
Unbladed Balance	9	5
Bladed Geometry	3	1
Bladed Balance	8	5
Overspeed vibration level	6	4
Post Overspeed Check Balance	8	5
Engine Test Vibration	8	5

Table 30 Consideration of Availability and Quality of Data

Table 30 shows the scores for the accuracy and availability of data in the 7 data collection stages. It is clear that the areas where the data accuracy is considered to be good are also those where availability is considered good. These areas are typically characterised by data obtained from an automated process, such as the balance measurements, where the actual data to be recorded is indicated on a display and does not need to be inferred from a series of measurements (as is the case for the geometry measurements). The bladed balance and post-overspeed check balance tests are considered marginally less accurate than the unbladed balance (with scores of 8 compared to 9 for unbladed balance) as there are issues with the blades moving slightly under rotation or load, and hence the attachment of these blades adds some variation to the characteristics of the rotor.

	How useful would it be? Score 1 – 10 (10 is Best)			
Data Source	**Unbladed Balance**	**Bladed Balance**	**Overspeed vibration level**	**Engine Test Vibration Level**
Unbladed Geometry	1	2	6	10
Unbladed Balance	N/A	2	6	10
Bladed Geometry	N/A	1	4	7
Bladed Balance	N/A	N/A	4	7
Overspeed vibration level	N/A	N/A	N/A	6
Post Overspeed Check Balance	N/A	N/A	N/A	6
Engine Test Vibration	N/A	N/A	N/A	N/A

Table 31 Consideration of Usefulness of Comparisons/Relationships

Table 31 shows the consideration of usefulness of comparison for each combination of data collection areas. It can be seen that it was considered most useful to be able to compare Final Engine Test performance to other, earlier tests, as the Final Engine Test is perhaps the most rigorous evaluation of the complete engine and also is the most costly test to run. The scores for usefulness could now be combined with the scores for data accuracy and availability in order to establish the most suitable areas for investigation.

$$S = \left(U_{ab}\left(A_a + A_b + V_a + V_b\right)\right)$$

The overall suitability of each area for investigation was computed using the above equation, where S is the suitability score for each input-output pair, a and b denote the input and output data collection areas respectively, A is the accuracy score, V is the availability of data, and U is the measure of Usefulness of comparison between a and b.

Total Score	Usefulness, Accuracy and Availability			
	Unbladed Balance	Bladed Balance	Overspeed Vibration Level	Engine Test Vibration Level
Unbladed Geometry	20	38	96	190
Unbladed Balance	N/A	54	144	270
Bladed Geometry	N/A	17	56	119
Bladed Balance	N/A	N/A	92	182
Overspeed vibration level	N/A	N/A	N/A	138
Post O/S Check Balance	N/A	N/A	N/A	156
Engine Test Vibration	N/A	N/A	N/A	N/A

Table 32 Combined Score for Areas of Investigation

It can be seen in Table 32 that the four areas of investigation considered most suitable for investigation all concern the Final Engine Test. It is suggested that this is due to the significant benefits that could be obtained if a greater understanding of factors affecting engine test vibration levels could be established. This is reflected in the scores for usefulness of pattern seen in Table 31, where those relationships that act to assist in describing the Final Engine Test vibration levels receive scores of 6 out of 10 or greater, whereas the remaining relationships are scored at 6 out of 10 or lower.

8.3.2 Nature of Modelling

In light of the results obtained Table 32, modelling began by comparing the unbladed balance against Final Engine Test vibration level. Discussion with engineers indicated that vibration levels of the exit bearing (the bearing located between the compressor and turbine, immediately next the combustion chamber) was the cause of the majority of Final Engine Test failures. All modelling described here therefore focuses upon exit bearing vibration levels and the vibration of the inlet bearing is not considered.

There are numerous parameters within these stages that must be analysed and considered on an engineering basis to be able to select those which will be suitable for use in the modelling.

Data	Measurements	Converted Data	Number of Parameters	Nature of Parameters
Balance	Mass-Distance and angle	Cartesian components	4	Two Cartesian components at both bearings
Final Engine Test	Amplitude of vibration	(unchanged)	2	Measurements along Cartesian axes at exit bearing

Table 33 Composition of Data

Table 33 shows the composition of both the balance and Final Engine Test data. The balance measurements comprise a polar measurement of the unbalance seen at both bearing positions. This represents the unbalance as a given mass at a given distance at a given radial angle. This data was converted to Cartesian Coordinates[35] for this analysis in order to directly link the angle and magnitude of the unbalance. The Final Engine Test data comprised a maximum level of vibration seen along the Cartesian axes within the engine test rig. The maximum amplitudes of vibration seen on both axes were recorded.

For the first stages of modelling, the C4.5 algorithm was used to model the data, and was implemented using the Weka software package (WEKA, 2002). Decision trees were created, from which rules could be directly extracted. Such rules were extracted to allow engineers to directly infer information, this approach being a precursor to the use of the significance metric (as applied to trees) proposed in section 5.4.5.

8.3.3 Results of Modelling of Identified Areas

The compiled data consisted of instances for 132 engines, of which 27 had incomplete records for the areas in question. As discussed in section 7.4.1, missing data may be dealt with using methods such as imputation, however in this case study all instances with missing parameters were removed from the dataset. The initial modelling was therefore performed with a dataset comprising of 105 instances. The data was converted

[35] All references to Cartesian co-ordinates assume only two axes, both of which point directly outwards radially from the centre of rotation of the rotor. The third axis would represent the axial length of the rotor, however no measurements exist of the centre of mass of each disc along this axis, and the Final Engine Test suite is not equipped to measure any movement along this axis.

in the manner shown in Table 33, giving 4 input variables and 2 separate output variables.

For the purposes of C4.5 algorithm the continuous Final Engine Test vibration level must be divided into ranges, and in this case four approximately equally populated ranges were used. The Final Engine Test vibration level was described by two variables, the X- and Y-components, hence two separate sets of models were created for each, using each variable in turn as the single output or classified[36] variable.

Model Number	Classified Variable	Prune	Min num	No. of Learning Instances	Training Accuracy %	CV Accuracy %
1	X-Component	75	2	105	84.76	43.3
2	X-Component	90	2	105	83.81	39.3
3	Y-Component	75	2	105	80.95	40
4	Y-Component	90	2	105	70.48	43.3

Table 34 Results of DTI Modelling of Unbladed Balance against Final Engine Test Vibration Level

Table 34 shows the results of the initial modelling using the C4.5 algorithm. Two models were created for both the X-component and Y-component of engine vibration, using more aggressive pruning for the second of each pair of models. The models show good training accuracy but poor cross-validation accuracy, reaching only a maximum of 43%. This is similar to the accuracies seen in the presence of noise as described in section 4.4.2, where accuracy reached a maximum of 47.5% (as was the case when 2.5% noise was added). This compares to a maximum accuracy seen of 62.5% with no noise present. It is desirable to deduce where such noise can be reduced, and hence a great part of the further work seeks to identify where errors are introduced to manufacturing data.

[36] The term 'classified variable' is used to avoid confusion between the 'output' of 'inlet bearing vibration' and the 'classified variable' of 'X-component of inlet bearing vibration'..

Breakdown of Models

The models were analysed to see if there were any specific rules that would be of use, or if a unifying pattern could be seen. The creation of a ruleset allows for specific rules to be extracted and considered in isolation. The C4.5 algorithm gives a coverage and an accuracy for each rule, indicating how many instances from the original dataset were covered by this rule (in effect how many instances complied with this rule) and the ratio of correctly to incorrectly classified instances for this rule.

Rule	Coverage	Accuracy	Inlet bearing x-Component		Inlet Bearing Y-Component		Exit Bearing X-Component		Exit Bearing Y-Component		Prediction
			Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper	
1	7	1		-539	224		-1563	-409			A
2	7	0.857			-1930	-933		1615		-1223	A
3	2	1	1379					155			A
4	2	1			224		2296			1186	A
5	12	0.583			-933		727	2296			A
6	14	0.643				-1930					B
7	97	0.34								2797	B
8	7	0.857		1481	-1930	-370	727				C
9	4	0.75		-487					2797		C
10	3	0.667			-1930	224	-1563	-409			C
11	15	0.467			-1930	-933					C
12	2	1	1481		-933	-370	2225				D
13	4	0.75		712	-933		155	727			D
14	7	0.571			-933			-1563		311	D
15	7	0.571							2797		D

Figure 43 Rules Describing X Component of Exit Bearing Vibration

Figure 43 shows a breakdown of the individual rules extracted from model 1 as seen in Table 34. These rules comprise logical conditions and an ultimate prediction, and are given in their constituent parts in Figure 43. The prediction gives the range that an instance meeting the logical conditions of the rule is estimated to belong to, where the ranges are denoted by A to D, with A being the lowest. In this case, these predictions estimate the likely X-component of engine test vibration level. The upper and lower bounds describe the logical conditions, where a given parameter must be above the lower bound and below the upper bound to fulfil the conditions of that rule. The coverage indicates how many instances (engines) within the dataset are covered by that rule, essentially how many instances conform to the logical conditions stated, and the accuracy

indicates the ratio of correctly covered rules against the total number of covered rules (both correct and incorrect)

It can be seen that the coverage and accuracy for the rules are subject to significant trade-off, where the rule with the greatest coverage classifies 97 instances, 34% of them correctly, whereas the rules with greatest accuracy classify between 2 and 7 instances with 100% accuracy.

Rule	Coverage	Accuracy	Inlet Bearing X-Component		Inlet Bearing Y-Component		Exit Bearing X-Component		Exit Bearing Y-Component		Prediction
			Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper	
1	3	1		-539	1628					2797	A
2	2	1				-1762		-1800			A
3	2	1		2946			3579				A
4	8	0.75			1216	1628					A
5	97	0.34								2797	A
6	77	0.416					-1075			2797	B
7	4	1					-1800	-1563			C
8	2	1								-3854	C
9	2	1						-3200			C
10	2	1	2946				3579				C
11	4	0.75		-487					2797		C
12	7	0.571							2797		D
13	15	0.467						-1800			D

Figure 44 Rules Describing Y Component of Exit Bearing Vibration

Figure 44 shows the individual rules extracted from model 3, as listed in Table 34. There is a pattern that indicates that large unbalance in the Y-component of exit bearing balance will result in greater vibration in the Y-axis during Final Engine Test, where cases where the Y-component of exit bearing balance is less than 2797 the vibration level is predicted as being low (either range A or B, as shown in rules 1, 5 and 6), whereas in cases where this balance is greater than 2797 the vibration level is predicted as being high (range C and D, as for rules 11 and 12). There is also an indication that large Y-component unbalance in the inlet bearing will actually act to reduce vibration in the Y-component of Final Engine Test vibration, as shown by rules 1,2 and 4.

Consideration of Rules

A key benefit of producing rules, as opposed to decision trees, is that each rule may be taken in isolation if that rule is considered to reveal important information in its own right. A difficulty arises in that it is not always clear how to decide which rules are the most significant, as there is a trade-off between coverage and accuracy, or how many instances in the dataset conform to a given rule, and how many are incorrectly covered by the rule. This is analogous to the problem seen in information retrieval where there is a trade-off between the number of returns of a search and the accuracy of each return. These factors, respectively entitled recall and precision, have been the focus of attention in information retrieval research and aspects of such research will be used to guide the identification of key rules.

The C4.5 algorithm gives the total number of instances covered by a rule followed by a ratio indicating the proportion of correctly classified instances. It was noted that those rules with higher accuracy tended to have smaller coverage, whereas the rules with high coverage tended to have lower accuracy. It is perhaps a statement of the obvious to suggest that more accurate rules would be of greater use, but coverage is important as a rule which classifies only a small portion of the data might simply be describing a spurious pattern, and even if genuine the rule will be of little use in practice as it covers a relatively uncommon set of circumstances. In order to extract the most significant rules, it is therefore necessary to address the trade-off, as deducing which rules have the better combination of coverage and accuracy.

Methods of Comparing Coverage and Accuracy

The problem of coverage and accuracy mirrors to some extent the issues of recall and precision in the Information Retrieval (IR) community. IR algorithms seek to return a set of documents from a corpus that is considered relevant to a specific query. When evaluating the performance of such a system, recall and precision are used as measures. Recall describes the proportion of relevant documents returned against relevant documents in the corpus (in essence comparing the number of relevant documents the system found against those it missed), whereas precision describes the proportion of relevant returned documents against irrelevant returned documents (of the returned documents, how many are correctly returned). There is a degree of compromise between these measures. It is possible to obtain only those documents that are certain to be

relevant and omitting whose relevance is considered marginal, in which case risking missing a number of relevant documents, thus achieving a high precision but poor recall. It is also possible to retrieve all documents that are considered even marginally relevant, in which case the risk is of returning irrelevant documents along with relevant ones. As it is less likely that relevant documents will be missed, this approach tends towards high recall but low precision. The compromise therefore involves balancing between these extremes, achieving an acceptable level of both precision and recall[37].

The methods proposed to address recall and precision in the information retrieval field will be briefly summarised, and then adapted for use in identifying important rules.

$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}}$$

The Harmonic Mean (as described by Baeza-Yates and Ribeiro-Neto, 1999) is defined above, where F is the value of the Harmonic Mean for the jth document and r and P and the recall and precision for the jth document respectively. Assuming that values for recall and precision will be given in the range of 0 to1, then the value of F will range from 0, where there are no relevant documents classified, to 1 where all the relevant documents have been classified. All incorrect classifications will reduce the value of F, and hence those with good precision and recall will receive the highest value of F. There is a limitation in this approach, in that it is not possible to factor in which property is more important to the user, recall or precision. In some cases it might be necessary to retrieve all relevant documents, and hence a compromise in precision would be tolerated, whereas in other situations it is more important to only select accurate documents and hence a compromise in recall would be accepted.

[37] It is possible that certain situations will require different levels of compromise. For example, when seeking a basic introduction to a new area of research it may be better to receive only a few documents that are highly relevant, even if other relevant documents are missed (high precision and low recall). In areas such as patent searching it is essential that all relevant documents are returned, even if the returned document set includes some irrelevant documents (high recall and low precision)

$$E(j) = \frac{1+b^2}{\frac{b^2}{r(j)} + \frac{1}{P(j)}}$$

The E Measure as proposed by van Rijsbergen (1979) introduces a factor *b* that essentially allows for the Harmonic Mean to be biased towards recall or precision. If a value if 1 is used for this factor, the E measure simply replicates the Harmonic Mean, whereas if the factor is greater than 1 it biases the E measure towards recall and a value of less than 1 introduces bias towards precision.

Application of Methods to Rules

The notion of recall cannot be directly applied in the evaluation of rules, as a score of 1 implies that every instance is classified by a given rule. This is suggested to be unlikely to occur in practice, as the C5.0 algorithm actually attempts to divide the data as much as possible, in doing so preventing each instance from being classified by the same rule. It is therefore suggested that an improved measure of recall would be to deduce the number of instances classified by the rule with the greatest coverage, and normalise the value of recall based on that count. If, for example, the rule with greatest coverage classified 30 instances, then the value for recall for each rule would be the number of classified instances divided by 30.

Consideration of Figure 43 indicates that there are 5 rules that could be considered to be useful based upon either a high accuracy or a wide coverage, however there is a trade-off between these two characteristics.

Rule	Coverage	Accuracy	Normalised Coverage	Harmonic Mean	E-Measure		
					b=0.01	b=0.5	b=2
1	7	1	0.072	0.134615	0.9987	0.2795	0.0884
5	12	0.583	0.124	0.204111	0.5828	0.3350	0.1472
6	14	0.643	0.144	0.235744	0.6428	0.3798	0.1705
7	97	0.34	1	0.507463	0.3400	0.3917	0.7203
11	15	0.467	0.155	0.232342	0.4669	0.3330	0.1789

Figure 45 Measures of Significance of Rules

Figure 45 shows the coverage and accuracy of the 2 rules with greatest accuracy and coverage respectively, and of 3 rules which perhaps present a more useful balance. Also

included are the values of E-Measure and Harmonic Mean for each rule. It can be seen that rule 7 has the highest value for Harmonic Mean, arguably because it has an overwhelming advantage in terms of coverage. The E-Measure is given using 3 values of b, each biasing the measure towards accuracy or coverage. In this case it can be seen that rule 1 has the greatest value of E-Measure when a value of 0.01 is used for the factor b, but when this factor is given a value of 2 rule 7 has the highest E-Measure. There is little to suggest which value of b is most suitable for ranking rules, however it is suggested that in cases where there is one rule that has poor accuracy but a vast coverage that the factor should be weighted towards accuracy (greater than 1).

8.3.4 Remarks on Modelling

The methods of analysis proposed in this section have sought to provide individual rules as a means of extracting information from a DTI model. This was proposed prior to the adoption of the ranking list, as discussed in Chapter 5. However, a rule is a direct statement that an engineer can use to deduce what series of events will be most likely to result in either disadvantageous or desirable performance. Methods of identifying the most significant rules have been discussed, however utilisation of a subset of rules necessarily involves rejecting the remaining rules, and hence a portion of the information contained within the model is lost. The use of a ranked list was intended to rectify this situation, by incorporating all information contained within the model, and was also intended to allow direct comparison with information extracted from ANN models as well as combining information from multiple models. These ranked lists do not provide any information regarding the interactions between parameters, as the sequence of logical conditions in rules can provide, and hence there is still a role for rule extraction to play in providing information to manufacturing and design engineers.

It was originally intended to test the created models by assembling a rotor based upon the results of modelling, in the form of a practical experiment. This would assist in deducing whether the patterns contained within the models were spurious or did indeed contain useful information. The accuracies of the models were similar to those generated in the presence of noise in section 4.4.2, and hence attention turned to identifying where such noise could be eliminated prior to model creation, such that the models would be of the greatest accuracy possible. During the course of this investigation several issues were identified that gave some cause for concern, indicating that notable errors were present

within the data. Attention will now turn to the methods used to generate this data, identifying where errors are introduced and indicating how such error might be eliminated at source.

8.4 Data Measurement and Recording Issues

The initial understanding of the manufacturing and data collection process focused upon obtaining a broad understanding of the process, in doing so deducing a flow of operations describing the entire fabrication process from receiving the basic disc billets through to signing off the complete engine. A comprehensive flowchart of the manufacturing process was created, however for reasons of commercial sensitivity this cannot be included here. This flowchart included feedback loops and indicated where data was recorded. There was little consideration at this point as to how precisely the data would be recorded or what form the data would take, although the data records were checked to ensure that data was consistently recorded for all rotors that were manufactured – if the data recording was only carried out on specific rotors, for example those with characteristics that were cause for concern, the stage was not categorised as being one of data recording.

8.4.1 Intangible Feedback

The retrospective construction of a flowchart describing the manufacturing process is arguably subject to inaccuracies, as it can only be deduced from the processes actually in place, rather than being used as template for the process. A flowchart constructed in such a manner can only illustrate those processes that are described by documentation, or are either witnessed during the creation of the flowchart or are highlighted during consultation with engineers. There is scope for certain processes to remain excluded from the flowchart in the event that their presence is not revealed by any of these measures. It is also possible that gradual changes to the process might not be reflected in documentation or be understood or noted by all engineers, and as these means of process evaluation are used to guide the construction of the flowchart any errors or misunderstandings will have implications on the degree of representation of the flowchart.

In the case study it was noted that there was a degree of subjectivity to the treatment of rotors that had poor test performance or whose measurements were outside of permissible

limits. The financial and scheduling implications of stripping a rotor down and restarting assembly mean that, where possible, the minimum amount of remedial work is carried out, where the rotor will be passed back and re-entered into the manufacturing process at a point as close to the required point of repair as possible. There is some assessment of the problem the rotor is experiencing, typically heavily reliant upon the expertise of the engineer, and from this a decision is made as to which point in the manufacturing process the rotor should be returned to in order to make good the problem. There are few hard-and-fast rules governing or guiding this decision, it is mainly down to the judgement and experience of the engineer on whose shoulders the decision rests. There are certain small-scale experiments that can be used as a guide, such as the use of a hammer-test[38] to indicate the degree of damping within the rotor, a key indicator of good coupling between discs. The results of these impromptu tests are not recorded, as there is no easily definable output to measure, and they are merely used as an indicator to guide the decision of the engineer.

The problem that this subjective feedback causes is twofold. The first difficulty is in tracing the path of a given rotor through the manufacturing process. There are numerous permutations for feedback, and the decision of which one to follow is dependant upon the diagnosis from the engineer. There is a need to ensure that the data recorded at each stage is obtained from the most recent iteration, as failure to ensure this can lead to discontinuities in the data where the data from a previous iteration is used to describe the final state of the rotor.

In a perfect manufacturing process, any reworking would merely revert the rotor back into a previous form, in effect simply 'undoing' the changes made during the incorrectly completed manufacturing operations. In such a situation it is not necessary to know if the rotor has been through feedback, as there is no fundamental change in character due to this feedback. It is unclear if this is the case, whilst there is little evidence to suggest that

[38] An accelerometer is attached to one end of a rotor whilst the opposite end of the rotor is struck by a rubber-faced hammer. The reduction in amplitude of vibration (the degree of damping) indicates the quality of location between components in the rotor. An experienced engineer can infer the longitudinal location of a poor joint by visual examination of the trace of the decay of vibration.

a reworked rotor will have different characteristics upon being stripped to a previous state, there is also no evidence to suggest it will not.

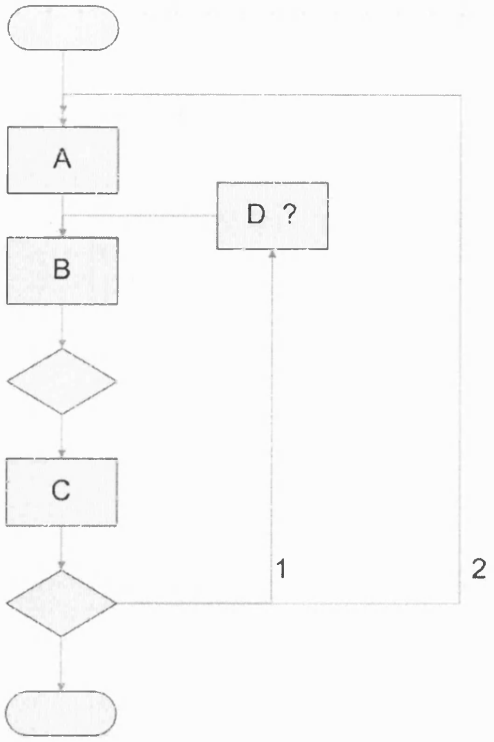


Figure 46 Schematic of Intangible Feedback Processes

The most critical problem, however, is the presence of intangible operations in the feedback loop. Figure 46 shows a schematic that illustrates what is meant by an intangible process within a feedback loop. The processes A, B and C are all clearly defined and understood, and, in data collection terms, are easy to account for. The operations seen on the feedback loops are much more difficult to account for both operationally and in terms of data collation, as there is not always a definitive description of the operation or of the data generated during the operation. As described previously, there are occasions where the actual feedback path is not known, where a problem after process C could result in a rotor being fed back through either path 1 or 2, however the problems created by a lack of understanding of process D is significantly more pressing.

The presence of intangible feedback operations renders the sequential linking of operations via flowchart meaningless, as it is difficult to maintain a direct link between the state of a rotor as it leaves one process and the state of the rotor as it enters the following process. In Figure 46, if process D comprises nothing more than a pass-through stage it is reasonable to assume that a rotor entering process B would have the

same characteristics as one leaving process A, regardless of whether it had been stripped down after feedback or whether it had passed directly from process A. If process D acts to change the characteristics of the rotor in any way then these assumptions no longer hold, and a rotor that is subject to feedback will no longer have identical characteristics to one that has passed directly from the preceding process.

Examples of Intangible Feedback

To illustrate these intangible processes two examples are given. It has been reported that, for reasons of expediency, a problematic disc was replaced in an unscheduled operation during one feedback loop during the initial balancing. It had previously been noted that the eccentricity of the disc in question could not be improved using the standard measures, hence it was suggested that there was an intrinsic problem with the disc which could only adequately be resolved by direct replacement. A second series of example cases were witnessed under Final Engine Test, where the turbines of problematic rotors were re-phased (reconnected to the compressor stack at a different angular position) on the test bed in order to fine-tune vibration performance under full test. The set-up times and associated costs for Final Engine Tests are huge, costs running into tens of thousands of pounds, hence re-phasing of the turbine in situ represents a huge financial saving over disassembly and repetition of previous processes.

These two example cases indicate the reasons why such feedback is necessary. In both cases the unscheduled operations carried out during feedback allowed for a rapid and economically beneficial resolution to a problem that would otherwise require extensive remedial work which might not ultimately result in the required improvement in performance. In both cases, the nature of the rotor was intrinsically modified by such actions, and comparisons cannot be drawn between measurements taken either side of these modifications as they no longer describe the same artefact. It is noted that the nature of production described in this case study is one of low volume and high complexity, where each product can be considered on a case-by-case basis, however it is argued that the problems highlighted in the feedback of problematic rotors can be extended into other higher-volume fields.

8.4.2 Ambiguous Measurement

The characteristics of vibration of a rotor are, to a large extent, defined by the out-of-balance forces created by the centre of rotation of the rotor being offset from the centre of mass. This offset is caused by two factors, the first is an inherent imbalance in each disc in the rotor and the second is eccentricity between discs in the rotor. The end result of both of these factors is an unbalance profile, where the degree of unbalance in the complete rotor varies with longitudinal length. In general, it is not possible to correct for each individual disc unbalance, instead two correction planes are assigned at opposite ends of the rotor, and by measuring the unbalance at these two points it is always possible to find a combination of balance weights that will be equivalent to the unbalance seen in the entire rotor (Rolls Royce, 1986). The unbalance is typically indicated as a specific out-of-balance mass at a specific distance at a specific angle, and by adding or removing the equivalent mass at the appropriate distance at the appropriate angle, it is possible to correct the unbalance at these two planes.

The difficulty in this case arises from the fact that the unbalance measurements are representations of the unbalance in the rotor. The unbalance profile has not been corrected, but the overall unbalance of the rotor *as seen at the bearing positions* has been reduced. This is acceptable for meeting performance requirements, as the correction acts to improve the dynamic performance of the rotor, but in terms of understanding the rotor unbalance it does not meet requirements as there are an infinite number of possibilities as to the unbalance profile throughout the rotor.

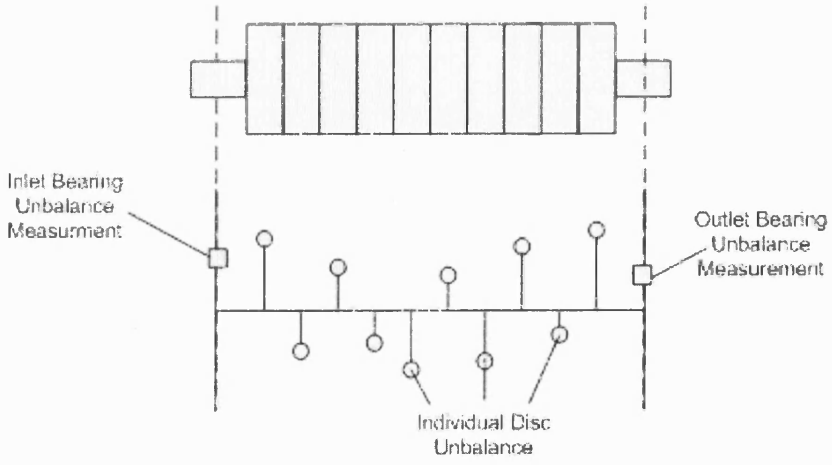


Figure 47 Schematic of Rotor Balance

Figure 47 (not to scale) illustrates this problem, where the individual unbalance of each disc within the rotor stack is denoted by a series of circles and the overall rotor unbalance as seen at the two bearing positions are indicated by two squares. It can be seen that the unbalance seen at the bearing positions (the only indication given during measurement) does not give an impression of the unbalance along the rotor.

An overall impression of the unbalance profile can be obtained by comparing the eccentricities of each disc to the unbalance measurements, where it assumed that larger eccentricities act to cause greater local unbalance. The two balance measurements at the bearing positions give the overall magnitude of the unbalance, whereas the individual disc eccentricities indicate how this unbalance is distributed along the length of the rotor. It is therefore feasible to infer a profile of the individual disc unbalances, although the accuracy of this inference will be impaired by errors in the measurement of disc eccentricity (discussed in the following section). This individual measure of unbalance is used to deduce the optimum blade distributions, the centre of rotation of which is deliberately offset to counter these estimated individual disc unbalances. Whilst a specific measure of individual disc unbalance would be preferable, this measurement of unbalance is driven by pragmatism as it is simple to measure unbalance in this manner and such measures give some indication of individual disc unbalance.

8.4.3 Error in Measurement

A significant proportion of the unbalance seen in Figure 47 is due to eccentricity between the discs in the rotor, where the centre of mass of each disc is offset from the centre of rotation, and for this reason the eccentricity of the rotor is measured.

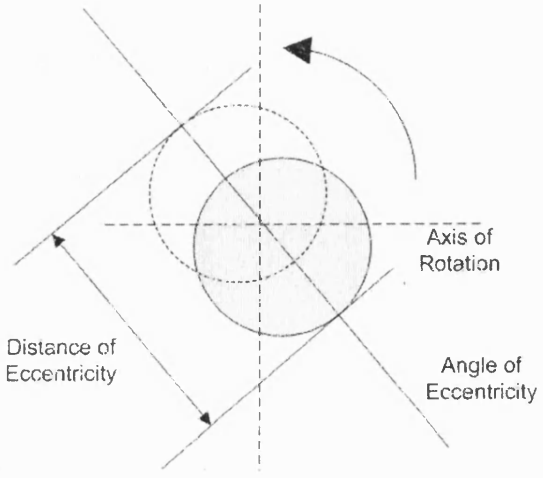


Figure 48 Rotor Disc Eccentricity

Figure 48 indicates the distance and angle of eccentricity which is measured for each disc. These values are measured by fixing a depth gauge to a datum surface and rotating the rotor through 360°. The maximum and minimum readings are recorded, along with the angular position of the rotor where the maximum and minimum were seen.

The difficulty in this method of measurement lies in the fact that a perfect disc is assumed, where any runout is attributed solely to eccentricity. It was noted, upon detailed examination of the data, that in many cases the angular distances between the maximum and minimum could be as little as 10°, which cannot rationally be assigned to eccentricity and can only logically be explained by either debris underneath the runout probe or by a non-round disc. The magnitudes of eccentricity are minute, as little as a few microns, and hence it is suggested that such undue influences could have a significant effect. The means by which the measurement is corrupted is not of importance, as the end result is that the measurement for eccentricity for each disc does not accurately represent the actual eccentricity.

8.5 Generic Principles of Data Recording Issues

The data generation issues raised in the previous section, and the examples used to describe them, will be to a certain extent unique to the case study. It is anticipated that the issue of intangible feedback will be traceable in many organisations as reworking is arguably common, and thus there is potential for cutting costs by tailoring reworking. It is foreseeable, however, that many manufacturing processes will have more defined means of feedback.

As an example, the author worked for a period within a semiconductor fabrication facility, being involved in adapting fabrication equipment from use as research and development equipment into a full-scale manufacturing facility. This adaptation meant that many batches of semiconductors were subject to considerable rework as the processes were fine-tuned. The feedback paths for rework were carefully defined, and a computer logging facility tracked each batch through the manufacturing process and also allowed approved rework paths to be presented to the operator when a fault was logged. A digital record of each batch was created and stored, indicating precisely which remedial measures had been undertaken. Of course, the large volume of production meant that deducing bespoke feedback for each batch would have been impractical, however the careful definition of feedback paths means that intangible feedback would

not become an issue (assuming such planned feedback paths were rigidly followed). This example illustrates that intangible feedback processes might not be prevalent in every manufacturing process.

It is also foreseeable that both ambiguous and erroneous measurements will be introduced into many manufacturing operations; however the mechanisms by which these errors could be introduced will depend heavily upon the specifics of the manufacturing process under investigation. It is suggested to be almost inevitable that there will be some ambiguity and error to the data, as no manufacturing process can be considered perfect, however it is unclear how prevalent such issues will be in other manufacturing processes.

As the specific issues and examples listed cannot be guaranteed to exist in all companies, it is important that the *principles* underpinning these issues be identified. These principles can then be used as a guide in ensuring that generated data is suitable for DM analysis.

8.5.1 The Three-Tier Hierarchy of Data Generation Issues

The three different data generation issues discussed previously can be argued to operate at different levels of granularity or specificity within the manufacturing process. The issue of intangible feedback functions on a process-wide level, whereas the issues of erroneous data are much more specific, functioning on an operation-wide level. This observation has led to the establishment of a hierarchy, containing three tiers, which is proposed to facilitate some understanding of where undue influence may be introduced into data generation, and where efforts must be made to eliminate such influences.

Tier	Scope	Example
Process	How areas of data generation relate to each other	Intangible Feedback
Implementation	Which phenomena uniquely identify each specific area of data generation	Ambiguous Measurement
Measurement	The means by which each specific phenomenon is to be recorded	Erroneous Measurement

Table 35 Identification of Hierarchy within Data Generation

Table 35 defines the three tiers within the hierarchy, where the level of granularity increases from the Process tier towards the Measurement tier. At the Process tier, the

issues of interest define how the individual areas of data generation (for example Final Engine Test or unbladed balance testing) relate to each other. In the Implementation tier the focus is within each specific area, where attention turns to considering which phenomena are recorded, and how these phenomena define the state of the artefact within that area. The Measurement tier focuses upon identifying how the specific phenomena are actually measured and recorded.

Issue	Description	Tier
No 'failed' data	Data describing 'failed' tests is overwritten by rework – there is only data describing 'good' pass and 'bad' pass, reducing extent of data	Process
Sparse Data	For example, no engine core (casing, combustion chamber etc.) data.	Process
Introduction of Electronic equipment	Records different phenomena to old equipment – lack of continuity within historical records	Implementation
Vibration measured along Cartesian axes	The orbit of vibration is elliptical, and if the centre-line of the ellipse does not align with either of the two measurement axes then the maximum displacement will not be measured	Measurement

Table 36 Further Data Generation Issues

The examples of data generation issues described previously were those that were considered to effectively preclude modelling. There were other issues noted, and whilst these further issues did not directly impinge upon the modelling that was undertaken, it is feasible that they could have undue influence on other data modelling activities. Table 36 shows these identified issues, and places them within the appropriate tier. These are included both to provide clearer indication of the composition of each tier and to provide some further substantiation of the suitability of considering individual issues within the framework of this hierarchy.

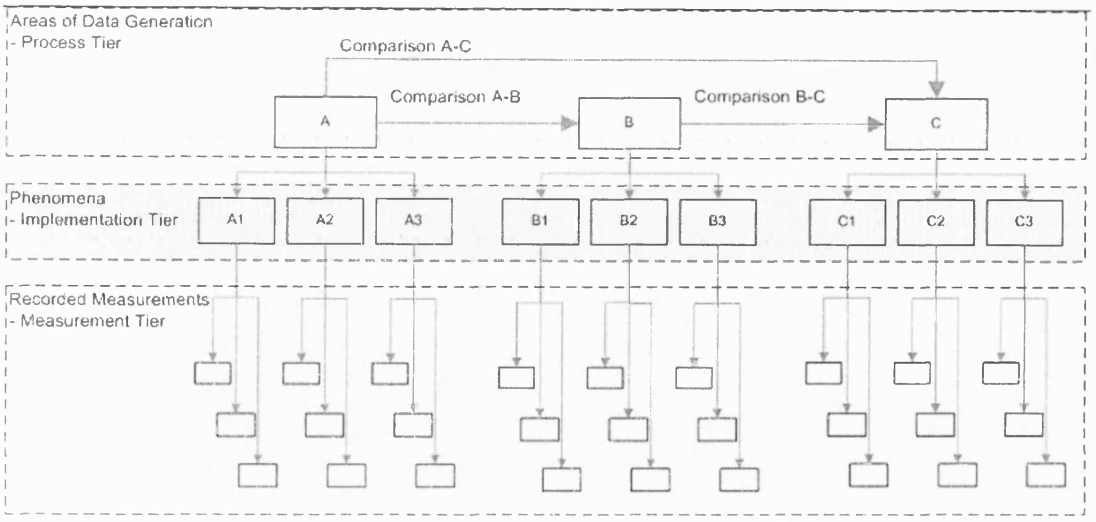


Figure 49 Graphical Representation of Data Generation Hierarchy

Figure 49 graphically depicts the hierarchy, indicating the areas of data collection at the upper tier and filtering down into the phenomena and specific measurements. It can be seen that each area of data collation may comprise multiple phenomena, and that each phenomenon may comprise multiple measurements. In this illustration multiple inheritance[39] is not allowed, as although it is feasible that a measurement could legitimately cover multiple phenomena it is argued that this leads to problems in distilling how much 'credit' to assign to a given phenomenon. The problem of multiple inheritance is essentially the same as that of erroneous measurement, as described in section 8.4.3, where a single measurement was seen to be applied to multiple phenomenon and it was not clear how to distil an accurate value for each phenomenon from the single measurement.

In reflecting upon the illustration in Figure 49, it is useful to simultaneously take into account the level at which analysis will be considered and acted upon. The modelling efforts undertaken in this research seek to draw comparisons between areas of data generation. It was at this Process level that engineers at the gas turbine manufacturing facility considered the overall extent of analysis, seeking to identify which areas to

[39] Multiple Inheritance is a term used in Object-Oriented Programming to depict a situation where a given class inherits characteristics from more than one superclass. This form of inheritance can lead to ambiguity, as in the event of more than one superclass incorporating independent implementations of the same characteristic, it is not clear which superclass takes priority in inheritance.

compare in order to obtain greatest benefit (section 8.3.1 discusses the decision making rationale in more detail). In contrast to this, the DM practitioner typically considers the analysis on the level of phenomena, where a decision is made upon which specific phenomena to compare. In the modelling described previously in this chapter, the engineers indicated the areas they considered suitable for comparison, and did so independently from the author. The author, liaising heavily with engineers, then indicated which phenomena to analyse within these identified areas. Taking the author as an independent DM practitioner, it may be argued that those involved in manufacture consider modelling on a Process level, whereas a DM practitioner is involved (and thus is interested) more heavily at the Implementation level. Along with this variation between engineer and DM practitioner is the level at which modelling acts. Ultimately, the modelling is enacted by applying a Machine Learning algorithm to specific measurements, thus utilising data generated within the Measurement tier.

The overall effect of these disjoints between engineer, DM practitioner and Machine Learning algorithm is that separations can occur between measurements and phenomena. In cases where two areas are to be compared, for Machine Learning algorithms to be successful it is essential that 'input' measurements from the first area can be linked to 'output' measurements in the second area. This is achieved by considering how accurately the measurements depict the specific phenomena of interest, and subsequently how the phenomena depict the state of the artefact at the given areas of interest. If both of these considerations are adequately addressed, then attention may be turned to ensuring that a direct, clearly understood link exists between the two areas under consideration. In the case study described in this chapter it became clear that no one individual was responsible for considering linkages at each of the three tiers, and hence many of the issues discussed in section 8.4 were not immediately identified. This led to considerable wasted effort until a clearer understanding of all three tiers had been developed. The hierarchy is intended to guide the DM practitioner in addressing all relevant tiers prior to modelling, thus avoiding many of the problems faced in this case study.

Tier	What criteria must be met?
Process	• Areas for comparison must be directly linked in terms of manufacturing process • Each artefact must be subjected to the same range of intermediate processes between each area of comparison (no 'intangible feedback' stages)
Implementation	• Phenomena must adequately depict the state of the artefact at that point o Phenomena must be unambiguous representations of the state of the rotor o Sufficient coverage of phenomena – each aspect/characteristic of artefact should ideally be described by complete range of phenomena
Measurement	• Each measurement should exclusively apply to only one phenomenon • Measurement should adequately quantify phenomena

Table 37 Three-Tier Hierarchy Guidelines

Table 37 shows the guidelines that the hierarchy encompasses, as obtained from the issues previously identified in section 8.4 and discussed in this section. It is emphasised that these guidelines evolved from those issues identified in the two case studies described in this thesis. Different applications will have different issues, however it is argued that the basic idea that there are three levels at which data generation should be considered is a useful thought to carry into a DM analysis.

8.5.2 Hierarchy Relating to Soap Powder Packaging Case Study

The case study described in Chapter 7 utilised data that was precompiled, and described a manufacturing process to which the author had no direct access. Nevertheless, it is still possible to make some general observations regarding the identified data issues and place within the structure of the hierarchy.

It is highlighted at this juncture that the data was generated from (what is essentially) a single complex operation on a single machine, and as such there are unlikely to be any issues relating to the process tier – this tier considers issues relating to the *sequence* of manufacturing operations and data generation areas, and in the case of the soap powder packaging line the data is effectively concurrently generated.

The two identified issues relating to the generated data were the aggregation of the data, where the data was expressed in a form of summary, and the difficulty in accurately distilling a value for the changeover duration. These two issues can be expressed in the implementation and detail tiers of the hierarchy respectively.

Summary Data – Implementation Tier

The key consideration at the Implementation tier level is whether each phenomenon contained in a given area is adequate to fully and reliably capture the state of the artefact at that point. In the soap powder packaging example, the artefact is actually the *process* of changeover, and hence the recorded phenomena should accurately and adequately define the process of changeover. It is difficult to verify this with any authority given the limited involvement the author had within data generation and collation, however a number of points may be inferred from consideration of the inherited data.

The methods of data generation, collation and preparation were discussed in section 7.3.2, however it will be briefly recapped here. A data logger captured manufacturing line output over time, and periods of inactivity were cross-referenced to product codes which indicated the product under manufacture at any given time. This cross-referencing allowed for changeovers to be identified, as indicated by periods of line inactivity in combination with a change in product code across this period. The product codes were expanded into a more comprehensive dataset by considering the settings used during the manufacture of a given product. The changes made to the line were found by deducing which specific settings were altered when switching from one product to the following one.

It is suggested that the scope of the recorded phenomena was adequate to capture the nature of the line before and after changeover, as the expanded list of line settings specified each parameter that required adjusting or setting prior to manufacture. A problem was identified in the granularity of the data, where the expanded dataset did not indicate each specific operation necessary to change each parameter. The data is ambiguous, as simply listing the settings for each manufacturing line parameter before and after changeover does not reveal what specific operations were necessary to enact those changes.

This problem with the data was argued to limit to some extent the range of possible modelling, and of interpreting the results. It was intended that the DM analysis would

allow for streamlining[40] efforts to be focused upon areas most problematic in changeover, for which the focus would be upon the operations carried out. As it stands, the analysis would not focus upon these operations but upon the line settings that they result in. As a change to a line setting involves a number of operations, identification of problematic setting changes would indicate the group of operations causing the greatest difficulties. Whilst this is a useful piece of information, it would be preferable to be able to identify the specific operation causing greatest difficulties.

This problem of granularity could be addressed by consideration of the guidelines in Table 37, which states that efforts should be made to avoid ambiguity in the phenomena used to capture the state of the artefact.

Quantification of Changeover – Measurement Tier

The periods of inactivity recorded by the data logger suffered from some subjectivity in that certain operations required the line to be restarted for a period during changeover. In such a case it is difficult to accurately delineate the duration of both Set-up and Run-up[41] as there is no precise boundary between them. This situation was more noticeable when attempting to identify the end of Run-up, where the line might run for a period of a few minutes before being stopped. It is difficult to state with any certainty whether the subsequent stoppage was related to the previous changeover, in which case the short period of manufacture should be included under the measure of Run-up.

The identification and attempted quantification of these two measures is a demonstration of the necessity of considering issues within the Measurement tier. A number of authors simply refer to changeover duration as the time taken to go form 'good piece to good piece, however further investigation of the nature of changeover suggests that this would be a poor measure to take.

---

[40] Streamlining is the final stage of the SMED changeover improvement methodology, which is argued to be subjective and difficult to enact. For a more complete description of both SMED and Streamlining please see section 7.2

[41] Set-up is defined by the author as the period taken to carry out all operations necessary to switch manufacture between two different products, whereas run-up is the period occuring after set-up to ensure that the line is operating successfully. For a more detailed description please refer to section 7.2.1

8.5.3 Severity and Prevalence of Issues

The construction of the three-tier hierarchy is useful in defining the nature of issues within data generation, in which case a DM practitioner can more readily identify where data might not be suitable for analysis. Such identification is reactive, where analysis can be steered away from unsuitable data, however this does not actually improve analysis, instead it helps to prevent inherently flawed analyses from being followed through. For the hierarchy to be of *active* use it is important that it assists in improving the methods of data generation, thus pre-empting issues that might arise. It is suggested that this might be achieved by considering how the issues within each tier influence the veracity of the recorded data, how prevalent these issues are within the manufacturing process, and how great the engineers' understanding of these issues are.

Data issues in each of the three tiers influence the task of Data Mining in different ways, some of which are retrievable or rectifiable at varying degrees of effort or cost. The notions of the severity and of the prevalence of these issues are, in this context, interlinked as a problem or issue related to data collection could be made more severe by the nature of increased prevalence. Consideration of the severity and the prevalence allows for some idea to be gained of how 'terminal' issues in each tier might be to the DM process, and from this it is possible to direct attention within the area of data generation to improve the quality of the data and hence of the DM analysis. It is noted that all understanding of the prevalence and severity of issues in each of the three tiers is obtained solely from consideration of the case study, and as such this analysis cannot be considered as a complete authoritative examination, although it serves as a useful indication of the nature of manufacturing data collection.

Process Tier

It is suggested that data will not be recorded that will completely define a given product, and it is anticipated that there will be gaps or omissions within the data where certain operations are not recorded. It is further suggested, however, that those operations that are considered either problematic or of greatest significance will have greater volumes of recorded data. This was noted in the case study, where computational methods of recording the overspeed test data were introduced as this test was seen as key in defining how the specific gas turbine rotor would behave in the Final Engine Test (the ultimate test of performance). This cannot be guaranteed to be the case in all organisations,

however it appeals to the notion that the recording and analysis of data is most likely to be carried out in situations where problems are noted or anticipated as there will be greater checking of compliance.

The issues of intangible feedback indicate that the process of reworking can prevent the establishment of a continuous linkage between the data generated during manufacture. Data describing the state of a given product at adjacent stages of production might not actually refer to the same rotor, as unrecorded operations may have taken place between the stages. This is not irretrievable, providing that some administrative or descriptive documentation is maintained alongside the data that describes the path of the product through the entire process. If there is any doubt as to which operations have been carried out on a given product, data obtained from the product in question may be excluded from analysis. It is also possible to use the administrative data to deduce if any intangible operation has been performed, as even if the actual process is not defined or measured, it is reasonable to expect there to be some written or recorded indication of the fact that a process has been carried out. In the example given previously a problematic rotor disc was replaced outside of the typical manufacturing process, however a note was made of the change. Hence, a decision could be made as to whether the documented change had an adverse effect upon the data sufficient to require the removal of this record from the dataset. It would only be necessary to reject the rotor in question from analysis if the areas of interest straddled the point at which the disc was replaced, in such a situation the rotor entering the second stage could not be considered as the same rotor that left the previous stage.

The prevalence of intangible feedback is difficult to ascertain, as it relies upon identifying operations that might not be documented, or where such documentation is poorly defined. It is suggested, however, that careful examination of each record will allow for changes to a product during feedback to be identified, and once identified a decision can be made as to whether a product re-entering a process after feedback has characteristics identical to those recorded during the first iteration of the manufacturing process. If this is not the case, and the product has been subjected to an intangible process, then data collected before the point of re-entry into the manufacturing process cannot be compared against data collected after the point of re-entry. In this respect, issues relating to the process tier can act to limit the range of investigation, or to require

the removal of specific instances from the dataset, and hence do not act to prevent successful analysis of the data but can act to limit the extent of analysis.

Implementation Tier

The issues that are contained within the implementation tier all consider the nature of which phenomena are to be measured at each stage, and by what means they should be measured and recorded. The problems surface when phenomena are not readily measurable, in which case a method may be proposed that will obtain an impression of the phenomena but will not completely define them. This can be seen in the method of measuring rotor balance, it would be preferable to measure the unbalance of the rotor along its length but practical concerns limit the measurement to the two bearing positions.

It is suggested that many of the tests or measurements carried out during the course of a manufacturing process will not measure the phenomenon of interest, but will simply be used to ascertain whether the individual product is conforming to a required metric. This is an acceptable compromise, as useful information can still be gained by a further understanding of how these measures are related, and from these measures it is possible to infer how the underlying phenomena are related. The issue then becomes inferring knowledge of the product by relationships found between the measured data.

It is suggested that issues relating to the implementation of measurement and recording may be prevalent, but the problems and issues raised at this level act only to diffuse the information obtained from modelling.

Measurement or Detail Tier

The detail tier covers the actual practice of extracting a piece of data that describes a given phenomenon. All measuring devices are subject to error, where the indicated measurement does not exactly capture the actual value of the phenomenon in question. This problem is well understood (for example Holman, 2001 discusses error in measurement during experimentation) and measures to address it would be better enacted by those designing the measurement devices. The focus of the problems in the detail stage, from a DM perspective, is the problem that the data does not always describe exactly what it is intended to describe. In effect, the measuring device performs its task correctly, but the actual measurement test has not been arranged in a manner that

accurately indicates the value for the phenomenon of interest. This is immensely damaging to the DM process, as data should be considered as merely a representation of an underlying phenomenon, if the data no longer describes the phenomenon in a consistent manner then there is little possibility of being able to either map the data or to deduce how the mapped relationship transfers from the data to the phenomenon.

There should be no ambiguity over the nature of the measurement, and it should be entirely repeatable. In the case of the gas turbine, the measure for eccentricity measured not only the eccentricity but also the roundness of each disc, and it is not possible to separate the two phenomena simply by examining the data. In the case of the soap powder packaging, the actual measurement of both the setup and runup times were argued to contain both the actual time and time taken by extraneous factors (which could feasibly include things such as misplaced tools and accidental damage). Such problems are intractable, in both of the examples given there is no way of retrospectively investigating the data to deduce precisely which phenomena the data describes.

Rademan *et al* (1996) analysed an industrial leaching process, and as part of their preamble argued that noise is endemic in manufacturing data as each operator will have an individual method of working and all those involved with the recording of data will make errors. The errors in recording are difficult to prevent, as unless an operation is entirely automated there is potential for human error. The difficulties arise in the idea that each operator performs a task in a slightly different manner, and hence this variability lends credence to the idea that not every operator will be recording the phenomena of interest.

8.5.4 Summary of Severity and Prevalence of Issues

It has been suggested that issues relating to the process or methodology tier have perhaps the least influence upon successful analysis, as data collection efforts will arguably tend to be based around areas of greatest interest, and any issues in this area will act to reduce the range of analysis rather than preclude analysis. Where errors, such as intangible feedback, are introduced it is possible to configure the scope of the DM analysis *to only analyse data from known good sources of data that are sequentially linked.* Issues relating to the implementation tier do not act to prevent analysis, instead they act to perhaps remove the results of the analysis from the underlying process, where knowledge of the relationships between parameters must be translated into relationships of the

phenomena that the data describes. In this respect it is necessary to consider how best to relate the results of DM analysis to the process in question, as it might not be a linear application as would be the case if the phenomena under DM analysis were the phenomena of direct physical interest. Issues relating to the detail or measurement tier are argued to have the greatest impact upon analysis, as the data used to describe the phenomena might be detrimentally influenced by phenomena that are not under analysis. If more than one phenomenon is recorded by a single measure then it is generally not possible to distil measures for each separate phenomenon. In cases where the value for the phenomenon that is not of interest cannot be predicted or otherwise evaluated, then the overall measure will be a function of both variations in the phenomenon of interest and of the impinging phenomenon. It is suggested that this is intractable, as there is no method of extracting the measurement of interest and the overall measurement would be subject to variance caused by an unknown process. It is therefore concluded that problems with data in the process and implementation tier can be handled at some expense, whereas problems within the detail tier present less easily rectified obstacles.

8.6 Observations Regarding Variations in Data Collation Practices

As previously highlighted, to be of greatest use the hierarchy must allow for future data acquisition and collation practices to be improved, and not simply form a structure for retrospectively considering how a given dataset came to be generated. As part of this, it is important to consider how detrimental variations in data generation practices are introduced. During the case study, observations were made regarding how each member of the manufacturing team perceived the nature of both the data generation process and the data itself, and these observations will be used to suggest how such variations impinge upon data generation.

It was noted in the case study that many engineers had different perceptions of the manufacturing process, in that those who were more removed from the process, such as those engineers in more senior management roles, had a good understanding of the general process but often had an incomplete or out-of-date impression of some of the more detailed work. This limited coverage was also seen to be the case for the parallel data generation practices. Once again, the arguments raised in this section are based

upon observations made during the case study, and cannot be considered to be entirely representative of manufacturing in entirety.

8.6.1 Levels of Understanding

The levels of understanding of the manufacturing process were, perhaps unsurprisingly, defined strongly by the breadth of involvement of an engineer within the manufacturing process. Those with involvement within relatively small areas, such as a specific operator for an individual machine, had greater understanding of that isolated area but had a weaker understanding both of other specific processes and of the process as a whole. Those in management roles had a good understanding of the complete process but a relatively weaker understanding of the fine detail. This is not to say that either operator or manager was ignorant of the nature of processes outside of their immediate attention, rather that they lacked specific knowledge and detailed understanding of these other areas.

To illustrate this point, in a meeting with the manager of the rotor manufacturing group and the line manager for rotor assembly, the more senior rotor manufacturing group manager was unaware that the rotors of the gas turbine were occasionally rephased (the turbine repositioned relative to the compressor) in Final Engine Test in the event that a given rotor had undesirable characteristics that could feasibly be remedied by this action. The line manager, in contrast, was aware of this practice. The senior manager's lack of knowledge of this practice is understandable, as the practice itself is not listed or discussed in any procedural documentation, instead it has been adopted as a 'quick fix' in cases where the performance of the rotor is marginally outside of tolerances and can thus be quickly (and, most importantly, cheaply) rectified in this manner. As the line manager was more closely involved in the processes within the assembly operations his level of understanding of the detail of these processes was greater.

Any DM analysis will arguably be instigated at management level, where issues relating to the process tier and implementation tier are likely to be well understood. When considered alongside the manufacturing process both the areas of data collection and problematic areas, or areas where an increase in knowledge would be most beneficial, would be known. It was noted during the case study that the senior engineers had a good understanding of the complete process and could identify where problems were occurring, be that in situations where there was a history of test failure or where

considerable effort was required to meet conformance requirements. There was also a good understanding of the quantity of data, although it was noted that data quality was generally optimistically estimated. This is argued to be due to the contrasting needs of data collection, where it is important to note that not all data is recorded for the purposes of analysis (Liu and Motoda, 2002).

Purposes of Data Generation

Within an organisation data might be recorded for the purposes of product tracking and ensuring that a product has been fully tested, for which even the most cursory data collection is suitable. The emphasis of the ISO 9000 series of quality assurance standards is upon process control, and hence data recorded to meet such standards describe process performance, as opposed to artefact performance. This was noted in both discussions with rotor assembly engineers and meetings with engineers responsible for monitoring Final Engine Test performance levels. In both cases, engineers were able to pinpoint periods of time during which there were increased numbers of test failures, both in intermediate tests and in Final Engine Test. As an example, a protracted period of increased Final Engine Test failure due to excessive vibration were revealed to be caused by changes in anti-corrosion coating processes, where variable coating thickness altered the centre of mass (and hence vibration characteristic) of each disc. The identification of the nature of the problem assisted in both locating the cause of the problem and guiding the necessary remedial action, highlighting the value of recording such data.

In DM analyses, the collected data has to be a highly accurate representation of the processes under examination, and hence data that would feasibly be perfectly capable of recording a product's progress through manufacture or of indicating process performance may not always be suitable for such DM analysis. The engineers would arguably have previously examined data, and found it met the purposes that it was created for, and hence considered the data accurate. That might be true for product tracking or process control, but assessing its accuracy and suitability for a separate and relatively weakly understood task is suggested to be difficult to judge.

It is perhaps useful to consider where there were misunderstandings regarding the data. As previously stated, it was observed that the senior engineers optimistically judged the quality of the data, as they tended to understand the method by which the measurement should be taken, but had little idea of the specific procedure followed to accomplish this

task. It is argued that data quality is strongly influenced by issues within the detail tier, and hence this misunderstanding could have significant ramifications upon the accuracy of modelling.

'Shortcuts' in Methods of Measurement

The operators tended to have a very good idea of the *procedure* specified for each measurement, but a comparatively weaker understanding of the actual *requirement* of the measurement, in essence understanding what task should be performed but not fully understanding the reason for carrying the task out.

It is possible that such a procedure can be inappropriately specified, where it is possible either for the operator to introduce 'shortcuts' or for a number of different phenomena to be measured whilst following procedure. Further to this, and as mentioned previously, the operator's extent of understanding was restricted to the tasks at hand simply by virtue of range of responsibility. In this respect there was no requirement to understand precisely what phenomenon was under examination whilst performing a measurement procedure, simply to follow the procedure and record the required results of measurement. In the case of the eccentricity measurement, it is clear that the actual readings (as given by the measuring device) were recorded, and not the phenomena of eccentricity.

It is suggested, however, that many problems in the detail tier are the result of insufficient precision in the specification of the method of measurement. In the example of eccentricity measurement, it is entirely possible to specify a new procedure that ensures eccentricity is measured exclusively of other phenomena. In the case of measuring the true maximum level of vibration of the rotor in the Final Engine Test (as highlighted in Table 36) it is possible to develop the method of measurement so that the true maximum is recorded. This could, for argument's sake, be achieved by requiring the *profile* of the rotation of the rotor to be measured, as opposed to simply recording maximum and minimum deflection of the outer edge of the rotor and the angles at which these deflections are seen. By recording the offset at a number of points during rotation it is possible to build up an image of the outer edge of the rotor during rotation, such that eccentricity can be disassociated from out-of-roundness.

It is noted that such improved methods come at a cost, and unless analysis of such measurement is required then such a cost could be argued to be unnecessary. However,

in both of the case studies examined in this research it is suggested that the factors most limiting to accurate modelling were problems within the detail tier, as these problems are much more difficult to deal with on a retrospective basis than problems within both the process and implementation tier.

It is further noted that the external DM practitioner will suffer from similar difficulties to the senior engineers, as an understanding of the detail stages of analysis are perhaps the last to be obtained. In this case study, a comprehensive flowchart of the rotor assembly and test process was constructed and improved via consultation with engineers on site. This flowchart was intended to map individual manufacturing operations or stages to the data collection stages, and as such is a more comprehensive form of the flowchart shown in Figure 42. This flowchart was subject to several major changes over the period of the research, culminating in the identification of many of the issues discussed in section 8.4. These issues were noted at the conclusion of a lengthy period of modelling, where interpretation of the results of modelling was underway for the purposes of process improvement. The identified issues suggested that refinement of the methods of data generation would be necessary in order to improve modelling. Such refinement could prove a lengthy process, and could not be accomplished within the bounds of this research project.

The external practitioner will invariably begin with an investigation of the manufacturing process on a broad level, understanding what is performed and where data is generated. This ensures that the subsequent analysis can be carried out in areas where data of suitable coverage is generated and results can be usefully employed within the manufacturing and design process. The understanding of the detail tier is dependant upon a gradual refining of knowledge, which may be developed by consultation with engineers who themselves might have weak understanding of the issues within the detail tier. It is therefore suggested that the DM practitioner takes a somewhat cynical view of the perceived situation within the detail tier, as there are risks that arguably counterproductive information will be obtained. Discussion with operators will arguably lead to better results on this front, as there is likely to be more immediate understanding of the exact methods used within measurement. At all times the practitioner should be conscious of the fact that there may be a degree of misunderstanding by those involved in the manufacturing process, as many within such a process utilise the data for entirely

different purposes to the DM practitioner, and hence an effort should be made to obtain an independent understanding of the issues within the detail tier.

8.7 Concluding Remarks

The case study described in this chapter was originally intended to be used as a proof of concept, validating the developed DM for manufacturing methodology. The initial modelling gave cause for optimism, as some interesting patterns were seen, although low cross-validation accuracies suggested that further work would be required. This further modelling was precluded, as it became clear that significant portions of the data did not describe the intended phenomena, and in the case of the eccentricity readings it was not possible to retrospectively manipulate the data to correct for this error.

The greatest benefit obtained from the case study was a greater understanding of the issues relating to data collection within a manufacturing process, and hence this has been the focus of this chapter. Numerous examples of issues influencing data generation were identified, leading to the development of a three-tier hierarchy as a method of describing these issues within a generic framework. The three tiers had progressively increasing granularity, starting at the process level, the overall consideration of what needed to be recorded, through the implementation tier, which addresses the phenomena that will be recorded, through to the detail tier, which considers the method used to record the phenomena. It was noted that issues within the process and implementation tiers could be resolved with some additional workload and with some restriction upon modelling, but issues within the detail tier could not be effectively rectified, and thus errors in this tier had the greatest detrimental effect upon modelling.

It was observed that senior engineers had a greater understanding of the overall process and methods of data collection, but tended to overestimate the quality of the recorded data. This lead to a situation where it was possible to identify the areas where analysis of the data would yield the most useful results from an engineering perspective, and where the data would be the most useful, but where there was little accurate understanding of the exact mechanics of measurement and recording. In effect, the senior engineers had a good understanding of the process and implementation tiers of data collection, but had limited understanding of the detail tier.

This situation is made more serious by the misconception of the operators, where it was suggested that they did not always understand precisely what phenomena they were required to record, and instead simply followed procedure and recorded the value indicated by the measuring device. In the case of the eccentricity measurement the procedure is clear, the operators are to measure the maximum and minimum runout and the angle at which they are seen. The problem arises in that even if each operator fulfils this criteria, and hence have faithfully carried out their allotted task, the data would still fail to accurately depict the intended phenomena. Operators are unlikely to have as broad an understanding of the process as senior engineers, preventing an analytical interpretation of the reason for recording a piece of data, and hence cannot be expected to adjust or alter the measurement process in order to more accurately measure this phenomenon. There is hence a degree of misunderstanding between operators and senior engineers, where the senior engineers understand what theoretically is being measured, but the operators are trained to manufacture and measure a metric using a pre-defined operation that, in terms of data collection, will not always meet with the engineers' expectations.

This discrepancy had not been identified as a potential problem prior to the initiation of the DM analysis, and it is for this reason that an understanding of issues relating to data generation is seen as particularly valuable. The three-tier hierarchy can be used to steer consideration of the methods employed in data generation such that problems similar to those noted in this case study can be avoided.

Chapter 9 Conclusions and Further Work

This research is based around the notion that manufacturing data contains useful information which describes how the parameters of a given artefact influence its later performance. These data are typically generated for quality assurance purposes, but it is possible to analyse them in order to identify interesting relationships between the parameters and later performance. The research described in this thesis sought to identify how information useful to the engineering designer could be extracted from manufacturing data. A number of different approaches to the analysis of such data were investigated, of which Data Mining (DM) was seen to be the most suitable.

DM is a methodology encompassing identification of the requirements of analysis, the form and coverage of available data, analysis of data and subsequent evaluation and deployment of the results of analysis. A review of literature describing the application of DM within engineering indicated that the majority of work focused upon the analysis of data, of how predictive DM models could be generated from manufacturing data. There was little formal discussion of the nature of manufacturing data or how the results of analysis could be employed in terms of extracting useful information from the generated DM models. This research sought to address these two neglected areas.

9.1 Extraction of Information from DM models

The analytical 'engine' of DM is, collectively, the set of Machine Learning algorithms. Two such algorithms, Decision Tree Induction (DTI) and Artificial Neural Networks (ANNs) were used to model 'artificial' manufacturing data generated by a computational analytical model. This analytical model described a linkage mechanism, and the DM models were used to provide a prediction for the performance of this mechanism (the maximum velocity) given the mechanism parameters (linkage lengths). The DM models were seen to provide accurate estimates of mechanism performance given these parameters. It was noted that the presence of noise led to overtraining, where the algorithms began 'learning' the profile of the noise within the data, and methods of mitigating against this were proposed.

This analytical model was previously used to generate information indicating *how* the parameters of a linkage mechanism influenced the performance of the mechanism, and this information was successfully utilised in industry. Methods of extracting equivalent information from both the DTI and ANN models were introduced (the method for DTI models being novel to this research), and the extracted information contrasted against the industrially-validated information from the analytical model. In this manner the information extracted from the DM models could be evaluated.

It was noted that information from the ANN models showed good agreement with information from the analytical model. The information extracted from the DTI model showed less clear agreement. However this agreement was improved by merging the information from multiple different DTI models using an adaptation of the popular Boosting algorithm.

It was noted that methods intended to reduce the complexity of DM models (either by reducing the number of parameters used within the model or by compacting the structure of the generated models) acted to impair the accuracy of information extracted from these models.

9.2 The Nature of Manufacturing Data

Two separate case studies were used to provide some understanding of the nature of manufacturing data. Data collected from a soap powder packaging line during the course of an earlier study were seen to contain error, and methods of retrospectively removing erroneous cases or instances from the database were tested. These data were precompiled and the author had no involvement with their collation. It was anticipated that the removal of erroneous instances would act to improve modelling accuracy.

The use of a 'pre-model', a DM model trained and used to filter out cases or instances that it could not correctly classify, was inconclusive in that it could not be established if the pre-model removed erroneous instances or those it could not describe. Manual identification and elimination of instances falling outside of the main distribution was also tested, however modelling accuracy was seen to drop suggesting that such an approach was ineffective. It was argued that erroneous instances could only be identified as such (and hence removed from the dataset) if knowledge of the manufacturing process and hence data was sufficient to identify what constitutes an 'acceptable' instance and therefore what constitutes an erroneous instance. In the absence of supporting

information it could not be established if instances with extraordinary values were erroneous or were the result of an extraordinary but still valid process.

The second case study investigated the manufacture and test of power generation gas turbine rotors. This case study sought to build upon the previous case study by identifying how error may be introduced into manufacturing data, and how data generation processes may be specified such that the resultant datasets are suitable for DM analysis. A further method of information extraction from DTI models was discussed, where a series of logical rules could be ranked according to either their coverage or their accuracy.

A number of issues relating to data generation processes were noted. These issues were observed to operate at different levels of detail which were described within the structure of a three-tier hierarchy.

The first tier of this hierarchy is related to the process or methodological level issues, or issues that effect data collection on a manufacturing process-wide scale, and where the areas suitable for data collection are identified. The second tier addresses the implementation issues, where the phenomena to be recorded are identified. The third tier, the detail tier, addresses issues of actual measurement, where the precise mechanics of measuring the required phenomena are considered. The essence of the hierarchy is that comparison of two areas of data generation involves considering which phenomena should be recorded at each stage and which specific measurements are required to accurately depict these phenomena.

Discussions with senior engineers and operators indicated that senior engineers had a good understanding of the methodological and implementation tiers, and were able to identify those areas where modelling would be useful. This was mitigated against an optimistic opinion of the detail tier, where it was assumed that the methods of measurement acted to record the required phenomena. The operators had a comparatively poor understanding of the broader manufacturing operations, and had a good understanding of the detail tier. This led to various problems, as any ambiguity within the measurement process could not readily be reconciled by the operators as it was rare that they knew precisely what the phenomenon of interest was. In many cases the measurements from the instruments were entered directly into record sheets, despite that fact that they could not physically represent the phenomenon of interest.

9.3 Further Work

The most pressing piece of further work is to be able to 'close the loop' in terms of the analysis, and to demonstrate physically that the information from DM analysis can act to improve manufacturing and design processes. Data Mining is a cyclical process, where information obtained from modelling should be fed back into the business and used to improve it further. The accuracy of information extracted from DM models has been demonstrated via computational study, however a practical application of such DM methods would allow a further series of study to be undertaken that could refine the approach.

It is also important to understand how an engineer, manufacturing or design, would use the extracted information. The information has been presented as a ranked list, indicating which parameters of a given artefact most influence its performance. It is proposed that further work could beneficially be carried out to attempt to tie in the methods of information extraction (and hence nature of extracted information) to the engineers requirements for information. This depends upon a successful analysis of manufacturing data, which in itself depends upon collecting accurate, representative data. It is anticipated that the hierarchy developed during the second case study could assist in ensuring the generated data is of sufficient accuracy for this purpose.

In terms of modelling, it is suggested that alternative modelling and information extraction methods should be investigated. The use of an ANN sensitivity analysis negates many of the inherent problems of lack of transparency in the approach, but it does not provide particularly rich information. Two methods of extracting information from DTI models were proposed, the first sought to provide information in a similar form to the ANN sensitivity analysis to allow direct comparison and aggregation of information between the algorithms, and the second method considers individual logical rules from a generated ruleset and identifies which ones are most pertinent. It is suggested that further work to improve the method of information extraction and of rule identification would improve the veracity of the extracted information. This work ties in the need to understand how engineers would the extracted information, and it is suggested that knowledge in this area should guide the development of alterative modelling techniques.

Chapter 10 References

Ackley, D. H., Hinton, G. E. and Sejnowski, T. J. (1985) A Learning Algorithm for Boltzmann Machines, *Cognitive Science,* **9** 147-169.

Agarwal, S., Agrawal, R., Deshpande, P. M., Gupta, A., Naughton, J. F., Ramakrishnan, R. and Sarawagi, S. (1996) On the Computation of Multidimensional Aggregates. *22nd International Conference on Very Large Databases*, Mumbai, India,

Aleksander, I. (1991) Introduction to Neural Nets. *In* Warwick, K., ed. *Applied Artificial Intelligence.* Peter Peregrinus Ltd, Stevenage. 85-94

Alhoniemi, E. (2003) Simplified time series representations for efficient analysis of industrial process data, *Artifical Intelligence for Engineering Design, Analysis and Manufacturing,* **17** 103-114.

Andrews, R., Diederich, J. and Tickle, A. B. (1995) Survey and critique of techniques for extracting rules from trained artificial neural networks, *Knowledge-Based Systems,* **8**(6), 373-389.

Angeline, P. J., Saunders, G. M. and Pollack, J. B. (1994) An Evolutionary Algorithm that Constructs Recurrent Neural Networks, *IEEE Transactions on Neural Networks,* **5**(1), 54-65.

Ball, N. R., Sargent, P. M. and Ige, D. O. (1993) Genetic algorithm representations for laminate layups, *Artificial Intelligence in Engineering,* **8** 99-108.

Baltazar, H. (2000) NBA Coaches' Latest Weapon: Data Mining In *PC Week*, Vol. 17, pp. 69.

Batista, G. E. A. P. A. and Monard, M. C. (2002) An Analysis of Four Missing Data Treatment Methods for Supervised Learning. *1st International Workshop on Data Cleansing and Preprocessing*, Maebashi, Japan, 9 December 2002.

Beale, R. and Jackson, T. (1990) *Neural Computing, an Introduction,* IOP Publishing, Bristol, UK.

Bertels, K., Neuberg, L., Vassiliadis, S. and Pechanek, D. G. (2001) On Chaos and Neural Networks: The Backpropogation Paradigm, *Artificial Intelligence Review,* **15** 165-187.

Bishop, C. M. (1993) Curvature-Driven Smoothing: A Learning Algorithm for Feed-Forward Networks, *IEEE Transactions on Neural Networks,* **4**(5), 882-884.

Bishop, C. M. (1995) Regularisation and complexity control in feed-forward networks. *International Conference on Artificial Neural Networks*, Paris, October 9-13, 1995.

Blanco, A., Delgado, M. and Pegalajar, M. C. (2001) A real-coded genetic algorithm for training recurrent neural networks, *Neural Networks,* **14** 93-105.

Blanzieri, E. (2003). *Theoretical Interpretations And Applications of Radial Basis Function Networks.* Trento, Italy:Department of Information and Communication Technology, University of Trento, (DIT-02-023)

Braha, D. (2001) *Data Mining for Design and Manufacturing: Methods and Applications,* Kluwer Academic Publishers, Dordrecht, The Netherlands.

Bray, T., Paoli, J., Sperberg-McQueen, C. M., Yergeau, F. and Cowan, J. (1997) Extensible Mark-up Language (XML) 1.1 W3C.

Breiman, L. (1996) Stacked Regression, *Machine Learning,* **24**(1), 49-64.

Breiman, L. (2001) Using Iterated Bagging to Debias Regressions, *Machine Learning,* **45** 123-140.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984) *Classification and Regression Trees,* Wadsworth, Belmont, CA.

BS 1916-1: 1953. *Limits and Fits for Engineering - Part 1: Limits and tolerances.* BSI

BS EN ISO 9001: 2000. *Quality Management Systems.* BSI

Buchheit, R. B., Garrett jr, J. H., Lee, S. R. and Brahme, R. (2000) A Knowledge Discovery Framework for Civil Infrastructure: A Case Study of the Intelligent Workspace, *Engineering with Computers,* **16** 264-274.

Buckingham, E. (1914) On Physically Similar Systems; illustrations on the use of dimensional equations, *Phys. Rev.,* **4** 345-376.

Carse, B., Fogarty, T. C. and Munro, A. (1996) Evolving fuzzy rule based controllers using genetic algorithms, *Fuzzy Sets and Systems,* **80** 273-293.

Chambers, R. (2000). *Evaluation Criteria for Statistical Editing and Imputation.* London, UK:Office for National Statistics, (NSMS28)

Chen, M. S., Han, J. and Yu, P. S. (1996) Data Mining: An Overview from a Database Perspective, *IEEE Transactions on Knowledge and Data Engineering,* **8**(6), 866-883.

Clark, P. and Niblett, T. (1989) The CN2 Induction Algorithm, *Machine Learning,* **3**(3), 261-283.

Clifton, C. and Thuraisingham, B. (2001) Emerging Standards for Data Mining, *Computer Standards and Interfaces,* **23** 187-193.

Conklin, J. H. and Scherer, W. T. (2003). *Data Imputation Strategies for Transportation Management Systems.* Virginia, USA:Centre for Transportation Studies, University of Virginia, (UVACTS-13-0-80)

CRISP-DM (2000) *Cross-Industry Standard Process for Data Mining* [Online]. CRISP-DM Consortium. Available from: http://www.crisp-dm.org [Accessed 1 Jan 2005]

Davalo, D. and Naim, P. (1991) *Neural Networks,* Editions Eyrolles, Paris.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society, Series B (Methodological),* **39**(1), 1-38.

Deutsch, D. (2002) *Speeding Up the Standards Development Process* [Online]. ANSI. Available from: http://public.ansi.org/ansionline/Documents/Standards%20Activities/Information%20Systems%20Conference%20Committee/2002%20Conference%20Presentations/DDeutsch.ppt [Accessed 11 Feb]

DMG (2003) *PMML Version 2.1* [Online]. The Data Mining Group. Available from: [Accessed

Domingos, P. (1997) *Why Does Bagging Work? A Bayesian Account and its Implications* [Online]. Available from: http://www.ics.uci.edu/~pedrod/kdd97.ps.gz [Accessed 1 Jan 2005]

Efron, B. and Tibshirani, R. J. (1993) *Monographs on Statistics and Applied Probability Vol 57: An Introduction to the Bootstrap,* Chapman and Hall, New York.

Eldridge, C., Mileham, A., McIntosh, R., Culley, S., Owen, G. and Newnes, L. (2002) Rapid Changeovers - the Run-Up Problem. *CARS & FOF International Conference*, Oporto, Portugal, 263-270

Engelbrecht, A. P. (2001) Sensitivity Analysis for selective Learning by Feedforward Neural Networks, *Fundamenta Informaticae,* **XXI** 1001-1028.

Engelbrecht, A. P. and Cleote, I. (1998) Feature Extraction from feedforward Neural networks using Sensitivity Analysis, *Int Conf on advances in Systems, Signals, Control and Computers,* **2** 221-225.

Eriksson, L., Johansson, E., Kettaneh-Wold, N., Wilkstrom, C. and Wold, S. (2000) *Design of Experiments: Principles and Applications,* Umetri AB, Stockholm.

Faba-Perez, C., Guerrero-Bote, V. P. and Moya-Anegon, f. D. (2003) Data mining in a closed web environment, *Scientometrics,* **58**(3), 623-640.

Fang, J. and Xi, Y. (1997) Neural network design based on evolutionary programming, *Artifical Intelligence in Engineering,* **11** 155-161.

Fausett, L. V. (1994) *Fundamentals of neural networks: architectures, algorithms and applications,* Prentice-Hall, New Jersey.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996) The KDD process for extracting useful knowledge from volumes of data, *Communications of the ACM,* **39**(11), 27-34.

Fayyad, U. and Stolorz, P. (1997) Data Mining and KDD: Promise and Challenges, *Future Generation Computer Systems,* **13** 99-115.

Finger, S. (1998) Design reuse and design research - Keynote paper. *Engineering Design Conference '98 - Design Reuse*, Brunel University, UK, 23-25 June 1998. Professional Engineering Publishing. 3-9

Fogel, D. B. (1994) An Introduction to Simulated Evolutionary Optimisation, *IEEE Transactions on Neural Networks,* **5**(1), 3-14.

Forcht, K. A. and Cochran, K. (1999) Using data mining and data warehousing techniques, *Industrial Management and Data Systems,* **99**(5), 189-196.

Freund, Y. and Schapire, R. E. (1997) A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences,* **55**(1), 119-139.

Freund, Y. and Schapire, R. E. (1999a) A Brief Introduction to Boosting. *16th International Joint Conference on Artificial Intelligence (IJCIA'99),* Stockholm, Sweden, 31 July - 6 Aug.

Freund, Y. and Schapire, R. E. (1999b) A Short Introduction to Boosting (In Japanese, Translated by N. Abe). *Journal of Japanese Society for Artificial Intelligence,* **14**(5), 771-780.

Fu, L. (1994) *Neural Networks in Computer Intelligence,* MacGraw-Hill, Singapore.

Fu, L. M. (1999) Knowledge Discovery by Inductive Neural Networks, *IEEE Trans on Knowledge and Data Engineering,* **11**(6), 992-998.

Fu, L. M. and Shortliffe, E. H. (2000) The Application of Certainty Factors to Neural Computing for Rule Discovery, *IEEE Transactions on Neural Networks,* **11**(3), 647-657.

Gingele, J., Childe, S. J. and Miles, M. E. (2003) Incorporating links to ISO 9001 into manufacturing process models using IDEF9000, *International Journal of Production Research,* **41**(13), 3091-3118.

Goh, A. T. C. (1995) Back-propagation neural networks for modelling complex systems, *Artificial Intelligence in Engineering,* **9** 143-151.

Gonzalez, R. and Kamrani, A. (2001) A Survey of Methodologies and Techniques for Data Mining and Intelligent Data Discovery. *In* Braha, D., ed. *Data Mining for design and Manufacturing: Methods and Applications.* Kluwer Academic Publishers, Dordrecht.

Grabowski, H., Lossack, r.-S. and Weisskopf, J. (2001) Automatic Classification and Creation of Classification Systems Using Methodologies of 'Knowledge Discovery in Databases (KDD)'. *In* Braha, D., ed. *Data Mining for design and Manufacturing: Methods and Applications.* Kluwer Academic Publishers, Dordrecht. 127-144

Groover, M. P. (1996) *Fundamentals of Modern Manufacturing: Materials, Processes and Systems,* Prentice-Hall, New Jersey.

Grossman, R., Bailey, S., Ramu, A., Malhi, B., Hallstrom, P., Pulleyn, I. and Qin, X. (1999) The Management and Mining of Multiple Predictive Models Using the Predictive Modeling Markup Language, *Information and Software Technology,* **41** 589-595.

Gupta, A., Park, S. and Lam, S. M. (1999) Generalised Analytic Rule Extraction for Feedforward Neural networks, *IEEE Trans on Knowledge and Data Engineering,* **11**(6), 985-991.

Hagan, M. T. and Menhaj, M. B. (1994) Training Feedforward networks with the Marquardt algorithm, *IEEE Transactions on Neural Networks,* **5**(6), 989-993.

Hagiwara, M. (1993) Removal of Hidden Units and Weights for Back Propagation Networks. *1993 Joint Conference on Neural Networks*, Nagoya, 25-29 Oct 1993. 351-354

Hammerstrom, D. (1993) Working with Neural Networks, *IEEE Spectrum,* **July 1993** 46-53.

Hand, D., Mannila, H. and Smyth, P. (2001) *Principles of Data Mining,* MIT Press, Cambridge, Mass, USA.

Hastie, T., Tibshirani, R. and Friedman, J. (2001) *The Elements of Statistical Learning - Data Mining, Inference, and Prediction,* Springer-Verlag, New York.

Helberling, G. (2002) ISO 9000 and ISO 14000 certifications reach record levels in 2001, *ISO Management Systems,* **Sept/Oct 2002** 11-15.

Hernandez, M. A. and Stolfo, S. J. (1998) Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem, *Data Mining and Knowledge Discovery,* **2** 9-37.

Hertkorn, P. and Rudolph, S. (2000) A Systematic Method to Identify Patterns in Engineering Data. *SPIE Vol. 4057, Data Mining and Knowledge Discovery: Theory, Tools, and Technology II,* Orlando, Florida, 24-25 April. 273-280

Hicks, C. R. and Turner jr, K. V. (1999) *Fundamental Concepts in the Design of Experiments,* Oxford University Press, New York.

Higgins, K. T. (2001) Faster Better Changeovers, *Food Engineering,* **73**(7), 46-51.

Himmelblau, D. M. and Karjala, T. W. (1996) Rectification of Data in a Dynamic Process using Artificial Neural Networks, *Computers in Chemical Engineering,* **20**(6/7), 805-812.

Holland, J. H. (1975) *Adaptation in natural and artificial systems,* The University of Michigan Press, Michigan.

Holman, J. P. (2001) *Experimental Methods for Engineers,* McGraw-Hill, Boston, USA.

Hong, S. J. and Weiss, S. M. (2001) Advances in predictive models for data mining, *Pattern Recognition Letters,* **22**(1), 55-61.

Hornik, K., Stinchcombe, M. and White, H. (1989) Multilayer Feedforward Networks are Universal Approximators, *Neural Networks,* **2** 359-366.

Howell, D. C. (1989) *Fundamental Statistics for the Behavioural Sciences,* PWS-Kent, Boston, USA.

Hu, Z. (2005) A Data Mining Approach for Retailing Bank Customer Attrition Analysis, *Applied Intelligence,* **22**(1), 47-60.

Huang, S. C. and Huang, Y. F. (1991) Bounds on the Number of Hidden Neurons in Multilayer Perceptrons, *IEEE Transactions on Neural Networks,* **2**(1), 47-55.

Huang, Y. and Wanstedt, S. (1998) The introduction of neural network system and its applications in rock engineering, *Engineering Geology,* **49** 253-260.

Hudomalj, E. and Vidmar, G. (2003) OLAP and Bibliographic Databases, *Scientometrics,* **58**(3), 609-622.

IPVR (1995). *SNNS User Manual, Version 4.1.*Stuttgart:IPVR, Stuttgart, (6/95)

Ishino, Y. and Jin, Y. (2001) Data Mining for Knowledge Acquisition in Engineering Design. *In* Braha, D., ed. *Data Mining for Design and Manufacturing:*

Methods and Applications. Kluwer Academic Publishers, Dordrecht, The Netherlands. 145-160

Jagielska, I., Matthews, C. and Whitfort, T. (1999) An investigation into the application of neural networks, fuzzy logic, genetic algorithms, and rough sets to automated knowledge acquisition for classification problems, *Neurocomputing,* **24** 37-54.

Jiang, W. (2004) Boosting with Noisy Data: Some Views from Statistical Theory, *Neural Computation,* **16** 789-810.

Joerding, W. H. and Meador, J. L. (1991) Encoding A Priori Information in Feed-forward Neural Networks, *Neural Networks,* **4** 847-856.

Kamrani, A., Rong, W. and Gonzalez, R. (2001) A Genetic Algorithm Methodology for Data Mining and Intelligent Knowledge Acquisition, *Computers and Industrial Engineering,* **40** 361-377.

Kenney, L. P. J., Rentoul, A. H., Twyman, B. R., Kerr, D. R. and Mullineux, G. (1997) A Software Environment for Conceptual Mechanism Design, *Proceedings of the IMechE Part C - Journal of Mech. Eng. Science,* **211** 617-625.

Khabaza, T. (2002) Hard Hats for Data Miners: Myths and Pitfalls of Data Mining. *In* Zanasi, A., Brebbia, C. A., Ebecken, N. F. F. and Melli, P., ed. *Data Mining III.* WIT Press, Southampton, UK. 1032

Kim, I.-W., Kang, M. S., Park, S. and Edgar, T. F. (1997) Robust Data Reconciliation and Gross Error Detection: The Modified MIMT Using NLP, *Computers and Chemical Engineering,* **21**(7), 775-782.

Kim, W., Choi, B.-J., Hong, E.-K., Kim, S.-K. and Lee, D. (2003) A Taxonomy of Dirty Data, *Data Mining and Knowledge Discovery,* **7** 81-99.

Kitano, H. (1990) Empirical studies on the speed of convergence of neural network training using genetic algorithms. *Eighth National Conference on AI,* Boston, MA, July 29 -Aug 3. 789-795

Kohonen, T. (1988) *Self-Organisation and Associative Memory,* Springer-Verlag, Berlin.

Kohonen, T. (1995) *Self-Organising Maps,* Springer, Heidelberg.

Last, M. and Kandel, A. (2001) Data Mining for Process and Quality Control in the Semiconductor Industry. *In* Braha, D., ed. *Data Mining for Design and Manufacturing: Methods and Applications.* Kluwer Academic Publishers, Dordrecht, The Netherlands.

Lazzerini, B. and Marcelloni, F. (2000) a genetic algorithm for generating optimal assembly plans, *Artifical Intelligence in Engineering,* **14** 319-329.

Lee, S. J. and Siau, K. (2001) A Review of Data Mining Techniques, *Industrial Management and Data Systems,* **101**(1), 41-46.

Leonard, L., Sirkitt, D. M., Langdon, I. J., Mullineux, G., Tilley, D. G., Keogh, P. S., Cunningham, J. L., Cole, M. O. T., Prest, P. H., Giddins, G. E. B. and Miles, A. W. (2002) Engineering a new wrist joint replacement prosthesis - a multidisciplinary approach, *Proceedings of the IMechE Part B - Journal of Engineering Manufacture,* **216**(9), 1297-1302.

Leschke, J. P. (1997) The Setup Reduction Process: Part 1, *Production and Inventory Management Journal,* **38**(1), 32-37.

Li, J. (2003) *PMML Output and Visualisation for WEKA*, Thesis (MSc), University of Bristol

Liebesman, S. (2002) Implementing ISO 9001:2000 - US survey of user experiences, *ISO Management Systems,* **Nov/Dec 2002** 39-47.

Lim, T.-S., Loh, W.-Y. and Shih, Y.-S. (2000) A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms, *Machine Learning,* **40**(3), 203-228.

Ling, C. X., Yang, Q., Wang, J. and Zhang, S. (2004) Decision Trees with Minimal Cost. *21st International Conference on Machine Learning*, Banff, Alberta, Canada, July 4-8, 2004.

Little, R. J. and Rubin, D. B. (2002) *Statistical Analysis with Missing Data,* John Wiley and Sons, New York.

Liu, H. and Motoda, H. (2002) On Issues of Instance Selection, *Data Mining and Knowledge Discovery,* **6** 115-130.

Looney, C. G. (1996) Advances in Feedforward Neural Networks: Demystifying Knowledge Acquiring Black Boxes, *IEEE Transactions on Knowledge and Data Engineering,* **8**(2), 211-226.

Lou, K.-N. and Perez, R. A. (1995) A novel application of artificial neural networks to structural analysis, *Artificial Intelligence in Engineering,* **9** 211-219.

Lowe, A. T. (2002) *Studies of Information Use by Engineering Designers and the Development of Strategies to Aid in its Classification and Retrieval*, Thesis (PhD), University of Bristol

Lu, H., Setiono, R. and Liu, H. (1996) Effective Data Mining Using Neural Networks, *IEEE Trans on Knowledge and Data Engineering,* **8**(6), 957-961.

MacLeod, C. and Maxwell, G. M. (2000) Incremental Evolution in ANNs: Neural Networks which Grow, *Artifical Intelligence Review,* **3**(16), 201-224.

Maimon, O. and Rokach, L. S. (2001) Data Mining by Attribute Decomposition with Semiconductor Manufacturing Case Study. *In* Braha, D., ed. *Data Mining for Design and Manufacturing: Methods and Applications.* Kluwer Academic Publishers, Dordrecht, The Netherlands. pp 311-336

Malinov, S., Sha, W. and McKeown, J. J. (2001) Modelling the correlation between processing parameters and properties in titanium alloys using artificial neural networks, *Computational Materials Science,* **21** 375-394.

Maniezzo, V. (1994) Genetic evolution of the topology and weight distribution of neural networks, *IEEE Transactions on Neural Networks,* **5**(1), 39-53.

McCulloch, W. S. and Pitts, W. (1943) A logical calculus of ideas imminent in nervous activity, *Bulletin of Mathematical Biophysics,* **5** 115-133.

McIntosh, R. I., Culley, S. J., Mileham, A. R. and Owen, G. W. (2000) a critical evaluation of Shingo's SMED (Single Minute Exchange of Die) Methodology, *International Journal of Production Research,* **28** 2377-2395.

McIntosh, R. I., Culley, S. J., Mileham, A. R. and Owen, G. W. (2001) *Improving Changeover Performance,* Butterworth Heinemann.

McMahon, C. and Browne, J. (1998) *CAD/CAM - Principles, Practice and Manufacturing Management,* Addison Wesley, Harlow, UK.

Meert, K. (1998) A real-time recurrent learning network structure for data reconciliation, *Artificial Intelligence in Engineering,* **12** 213-218.

Mehrotra, K. G., Mohan, C. K. and Ranka, S. (1991) Bounds on the Number of Samples Needed for Neural Learning, *IEEE Transactions on Neural Networks,* **2**(6), 548-558.

Microsoft (1994) *Microsoft Excel user's guide version 5.0,* Microsoft Corporation, Redmond, USA.

Mileham, A. R., Culley, S. J., Owen, G. W., Newnes, L. B., Giess, M. D. and Bramley, A. N. (2004) The impact of run-up in ensuring rapid changeover, *Annals of CIRP,* **53/1/2004** 407-410.

Mingfang, K., Bingzhen, C. and Bo, L. (2000) An Integral approach to dynamic data rectification, *Computers and Chemical Engineering,* **24** 749-753.

Nearchou, A. C. (1999) Adaptive navigation of autonomous vehicles using evolutionary algorithms, *Artificial Intelligence in Engineering,* **13** 159-173.

Ng, W. W. Y. and Yeung, D. S. (2002) Input Dimensionality reduction for Radial Basis neural network Classification Problems using Sensitivity Measure, *Proc. of 1st Int. conf. on Machine Learning and Cybernetics,* 2214-2219.

Oh, S.-H. and Lee, Y. (1995) Sensitivity Analysis of Single Hidden-Layer Neural Networks with Threshold Functions, *IEEE Transactions on Neural Networks,* **6**(4), 1005-1007.

Ohsuga, S. (1989) Towards Intelligent CAD Systems, *Computer-aided Design,* **21**(5), 315-317.

Ong, S. K. and Guo, D. O. (2004) Online design reuse tool for the support, embodiment and detailed design of products, *Int J Prod Res,* **42**(16), 3301-3331.

Opitz, D. and Maclin, R. (1999) Popular Ensemble Methods: An Empirical Study, *Journal of Artificial Intelligence Research,* **11** 169-198.

Pahl, G. and Beitz, W. (1996) *Engineering Design: A Systematic Approach,* Springer-Verlag, London.

Phadke, M. S. (1989) *Quality Engineering Using Robust Design,* Prentice-Hall, USA.

Pham, D. T. and Karaboga, D. (1999a) Self-Tuning fuzzy controller design using genetic optimisation and neural network modelling, *Artificial Intelligence in Engineering,* **13** 119-130.

Pham, D. T. and Karaboga, D. (1999b) Training Elman and Jordan networks for system identification using genetic algorithms, *Artificial Intelligence in Engineering,* **13** 107-117.

Piatetsky-Shapiro, G. (1999) The Data Mining Industry Comes of Age, *IEEE Intelligent Systems,* **14**(6), 32-34.

Pohlheim, H. (1994) *Geatbx Documentation* [Online]. Geatbx. Available from: http://www.geatbx.com/docu/algindex.html [Accessed Jan 2, 2005]

Potter, S. (1995). *A Survey of Neural Network Algorithms and their Applications.* Bath:Mechanical Engineering, University of Bath, (Internal Report Number 048/95)

Potter, S. (2000) *Artificial Intelligence and Conceptual Design Synthesis*, Thesis (PhD), University of Bath

Pyle, D. (1999) *Data Preparation for Data Mining,* Morgan Kaufmann, San Francisco, USA.

Quinlan, J. R. (1986) Induction of Decision Trees, *Machine Learning,* **1** 81-106.

Quinlan, J. R. (1996a) Bagging, Boosting and C4.5. *Thirteenth National Conference on Artificial Intelligence*, Portland, Oregon, 4-8 August 1996. 725-730

Quinlan, J. R. (1996b) Boosting First-Order Learning. *7th International Workshop on Algorithmic Learning Theory (ALT'96)*, Sydney, Australia, 23-25 October. 143-155

Quinlan, J. R. (1996c) Improved Use of Continuous Attributes in C4.5, *Journal of Artificial Intelligence Research,* **4** 77-90.

Quinlan, J. R. (1999) Simplifying decision trees, *International Journal of Human-Computer Studies,* **51** 497-510.

Rademan, J. A. M., Moolman, D. W., Lorenzen, L., Deventer, J. S. J. v. and Aldrich, C. (1996) Neural Net Based Knowledge Extraction from the Historical Data of an Industrial Leaching Process, *Hydrometallurgy,* **43** 95-116.

Ray, T., Gokarn, R. P. and Sha, O. P. (1996) Neural network applications in naval architecture and marine engineering, *Artificial Intelligence in Engineering,* **1** 213-226.

Reese, G., Yarger, R. J., King, T. and Williams, H. E. (2002) *Managing and Using MySQL,* O'Reilly, Sebastopol, USA.

Reich, Y. (1999) Letter to the editors, *Artificial Intelligence in Engineering,* **14**(2), 199.

Reich, Y. and Barai, S. V. (1999) Evaluating Machine Learning Models for Engineering Problems, *Artificial Intelligence in Engineering,* **13** 257-272.

Reich, Y. and Barai, S. V. (2000) A methodology for building neural networks models from empirical engineering data, *Engineering Applications of Artificial Intelligence,* **13** 685-694.

Romanowski, C. J. and Nagi, R. (2001) Analysing Maintenance Data Using Data Mining Methods. *In* Braha, D., ed. *Data Mining for Design and Manufacturing: Methods and Applications.* Kluwer Academic Publishers, Dordrecht, The Netherlands.

Rudolph, S. and Hertkorn, P. (2001) Data Mining in Scientific Data. *In* Braha, D., ed. *Data Mining for Design and Manufacturing.* Kluwer Academic Publishers, Dordrecht, The Netherlands. 61-85

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986) Learning Internal Representations by Error Propagation. *In* Rumelhart, D. E. and McClelland, J.

L., ed. *Parallel Distributed Processing: Explorations in the microstructure of cognition, Volume 1: Foundations.* MIT Press, Cambridge, MA.

Rumelhart, D. E. and McClelland, J. L. (1986) *Parallel distributed processing: explorations in the microstructure of cognition, Volume 1: foundations,* MIT Press, Cambridge, MA.

Sarawagi, S., Agrawal, R. and Megiddo, N. (1998) Discovery-driven exploration of OLAP data cubes. *Sixth International Conference on Extending Database Technology (EDBT)*, Valencia, Spain,

Schafer, J. L. (1999) Multiple imputation: a primer, *Statistical Methods in Medical Research,* **8** 3-15.

Schapire, R. E. and Singer, Y. (1999) Improved Boosting Algorithms using Confidence-rated Predictions, *Machine Learning,* **37** 297-336.

Schulte, M. and Weber, C. (1993) The relationship between function and shape. *9th International Conference on Engineering Design*, The Hague, WDK. 9-20

Schwabacher, M., Ellman, T. and Hirsh, H. (2001) Learning to Set Up Numerical Optimisations of Engineering Designs. *In* Braha, D., ed. *Data Mining for Design and Manufacturing: Methods and Applications.* Kluwer Academic Publishers, Dordrecht, The Netherlands.

Sethi, I. K. (2001) Data Mining: An Introduction. *In* Braha, D., ed. *Data Mining for Design and Manufacturing: Methods and Applications.* Kluwer Academic Publishers, Dordrecht, The Netherlands.

Sforna, M. (2000) Data mining in a power company customer database, *Electric Power Systems Research,* **55** 201-209.

Shannon, C. E. (1948) A mathematical theory of communication, *Bell System Technical Journal,* **27** 2 parts, pp 379-423 and 623-656.

Sharkey, A. J. C. (1996) On Combining Artificial Neural Networks, *Connection Science,* **8**(3 & 4), 299-313.

Sharkey, A. J. C., Sharkey, N. E., Gerecke, U. and Chandroth, G. O. (2000) The 'test and select' approach to ensemble combination. *1st International Workshop on Multiple Classifier Systems*, Cagliari, Italy, 21-23 June. 30-44

Shelley, B. and Stephenson, S. (2000) The use of artificial neural networks in completion stimulation and design, *Computers and Geosciences,* **26** 941-951.

Shingo, S. (1985) *A Revolution in Manufacturing: The SMED System,* Productivity Press, Portland, USA.

Sladsky, R. (2001) Achieving faster, more efficient tube mill changeovers, *The Tube and Pipe Journal,* **9**(4), 28-31.

Smith, K., Palaniswami, M. and Krishnamoorthy, M. (1996) traditional heuristic versus Hopfield neural network approaches to a car sequencing problem, *European Journal of Operational Research,* **93** 300-316.

Smith, K. A. and Gupta, J. N. D. (2000) Neural networks in Business: Techniques and Applications for the Operations Researcher, *Computers and Operations Research,* **27** 1023-1044.

Smithers, T. (2001) Letter to the editor, *Artificial Intelligence in Engineering,* **15** 3.

Smyth, P. (2000) Data Mining: data analysis on a grand scale?, *Statistical Methods in Medical Research,* **9** 309-327.

SPSS (2002) *Clementine 7.0 User's Guide,* Integral Solutions Limited, Chicago.

Suh, N. P. (1990) *The Principles of Design,* Oxford University Press, New York.

Szatkowski, P. M. and Reasor, R. J. (1991) The SMED System for Setup Reduction - A Case Study. *Proceedings of 1991 International Industrial Engineering Conference*, Detroit, USA, Institute of Industrial Engineers. 123-129

Taguchi, G. (1986) *Introduction to Quality Engineering: Designing Quality into Products and Processes,* Productivity Pr., Tokyo.

Taha, I. A. and Ghosh, J. (1999) Symbolic Interpretation of Artificial Neural Networks, *IEEE Trans on Knowledge and Data Engineering,* **11**(3), 448-463.

Tan, B., Lin, C. and Huang, H.-c. (2003) An ISO 9001:2000 quality information system in e-commerce environment, *Industrial Management and Data Systems,* **103**(9), 666-676.

Ting, K. M. and Zheng, Z. (1998) Boosting Trees for Cost-Sensitive Classifications. *10th European Conference on Machine Learning (LNAI-1398)*, Berlin, 190-195

Tomiyama, T., Smith, I. and Kunz, J. (2001) Editorial comment, *Artificial Intelligence in Engineering,* **15** 1.

Tong, K. W., Kwong, C. K. and Yu, K. M. (2004) Intelligent process design system for the transfer moulding of electronic packages, *Int J Prod Res,* **42**(10), 1911-1931.

Trevino, J., Hurley, B. J. and Friedrich, W. (1993) A mathematical model for the economic justification of set-up time reduction, *International Journal of Production Research,* **31**(1), 191-202.

Tsoukalas, L. H. and Uhrig, R. E. (1997) *Fuzzy and Neural Approaches in Engineering,* John Riley and Sons, New York.

Turney, P. (1995) Data Engineering for the Analysis of Semiconductor Manufacturing Data. *IJCAI-95 Workshop on Data Engineering for Inductive Learning*, Montreal, Canada, 50-59

Turney, P. (2000) Types of Cost in Inductive Concept Learning. *Workshop on Cost-Sensitive Learning at the 17th International Conference on Machine Learning*, Stanford University, USA, 29 June - 2 July. 15-21

Van Goubergen, D. and Landeghem, H. V. (2002) Reducing Set-up Times of Manufacturing Lines. *FAIM 2002*, Dresden, Germany, July 15-17 2002.

Veelenturf, L. P. J. (1995) *Analysis and applications of artificial neural networks,* Prentice Hall, London.

Wang, X. Z. (1999) *Data Mining and Knowledge Discovery for Process Monitoring and Control,* Springer-Verlag, London.

Wasserman, P. D. (1993) *Advanced Methods in Neural Computing,* Van Nostrand Reinhold, New York.

Watkins, D. (1997). *Clementine's Neural Networks Technical Overview.* Woking, UK:SPSS,

Weiss, S. M. and Kulikowski, C. A. (1991) *Computer Systems That Learn : Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems,* Morgan Kaufmann, San Franciso, USA.

Werbos, P. (1988) Backpropogation: Past and future. *IEEE International Conference on Neural Networks*, San Diego, 24-27 July 1988. 343-353

Westphal, C. and Blaxton, T. (1998) *Data Mining Solutions: Methods and Tools for Solving Real-World Problems,* John Wiley, USA.

Wettschereck, D., Jorge, A. and Moyle, S. (2003) Visualisation and Evaluation support of Knowledge Discovery through the Predictive Model Markup Language. *7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, Oxford, UK, 493-501

Witten, I. H. and Frank, E. (2000) *Data Mining: Practical machine learning tools with Java implementations,* Morgan Kaufmann.

Witten, I. H. and Frank, E. (2004) WEKA v3.4.2 Waikato, NZ.

Wong, B. K., Lai, V. S. and Lam, J. (2000) A bibliography of neural network business applications research, *Computers and Operations Research,* **27** 1045-1076.

Wong, B. K. and Selvi, Y. (1998) Neural Network Applications in Finance: A Review and Analysis of Literature, *Information and Management,* **34** 129-139.

Zeng, X. and Yeung, D. S. (2001) Sensitivity Analysis of Multilayer Perceptron to Input and Weight Perturbations, *IEEE Transactions on Neural Networks,* **12**(6), 1358-1366.

Zhang, Y. F. and Fuh, J. Y. H. (1998) A Neural Network Approach for Early Cost Estimation of Packaging Products, *Computers in Industrial Engineering,* **34**(2), 433-450.

Zheng, Z. and Webb, G. I. (1998). *Multiple Boosting: A Combination of Boosting and Bagging.* Geelong, Australia:Deakin University, (Technical report TR C98/01)

Chapter 11 Appendix A – Methods of Data Analysis

Any attempt to make use of recorded data requires some form of analysis, methods for which range from simple manual interpretation to statistical inference and machine learning techniques. Appropriate selection of the most suitable method of analysis for any given problem can only be carried out if both the scope of analysis and nature of the method of analysis are known. This appendix describes the methods suitable for use within the context of applications within engineering design using manufacturing and assembly data. The techniques will be briefly described and their applicability to the problem at hand will be considered.

In order to avoid precluding suitable technologies from consideration, investigation of the various methods of data analysis has been extended beyond those used within engineering to other fields where technologies of this type are prevalent. In particular, research into databases has led to the development of numerous technologies which allow large datasets to be examined and information extracted. The following sections will describe, in turn, the methods of data analysis which have been identified in both the engineering and database fields.

11.1 Engineering Applications

When examining traditional methods of data analysis within engineering, the various approaches may generally be placed into two separate groups. A significant proportion follow a methodology that may broadly be classified as a Design of Experiments (DoE) approach, which encompasses the implementation of experimentation and data collation alongside the analysis of data. A second class of approaches may be described as being statistical in nature, ranging from basic 'curve fitting' through to complex Machine Learning techniques. There is significant overlap between these groups, for example the statistical approaches may include aspects of DoE in order to obtain the data for analysis. Furthermore, it should be noted that the actual analytical stage of the DoE approach relies upon statistical analysis to quantify relationships. These two groups are separately classified by the author however, as each has distinct features which influence their range

of application, and in the case of DoE represent such a significant aspect of engineering analysis that separate coverage is warranted.

11.1.1 DoE Analysis, incorporating Taguchi's Robust Design

Taguchi's Robust Design (Taguchi, 1986) provides a useful application of a DoE approach towards problems within the field of engineering design. Robust Design may be familiar to many engineers, but it should be noted that the field of DoE encompasses numerous techniques of which the method popularised by Taguchi is but one example (Hicks and Turner jr, 1999). These methods formalise a structure to establish a sequence of experimentation and data measurement followed by analysis from which information relevant to the particular domain can be established.

The development of Robust Design began in post-war Japan, which suffered an endemic shortage of quality materials and skilled labour. These constraints motivated development of a method of ensuring successful manufacture and product performance from an early stage in product development, thus removing the need for expensive remedial action. Taguchi approached this by considering the quality of a product as '....the loss a product causes to society after being shipped, other than any losses caused by its intrinsic functions' (Taguchi, 1986) and sought methods of ensuring the maximum possible quality of a product throughout each stage of design and manufacture.

The rationale of the approach may perhaps best be illustrated using an example as given by Phadke (1989). A tile company in Japan in the 1950's suffered from large dimensional variability in the tiles they manufactured, resulting in large numbers of reject tiles. The cause of the variance was diagnosed as uneven heating within the kiln, which could only be remedied at huge expense, and so Robust Design was used to identify other, cheaper solutions. The results of the analysis indicated that increases in the lime content of each tile would reduce the dimensional variability to an acceptable level, and hence this modification was introduced in place of the kiln refurbishment. In this way, Taguchi's desire to improve quality at each stage, rather than attempt to resolve inherent problems through expensive remedial work, resulted in significant improvements at minimal cost.

The classical method of Robust Design is based around the idea of segregating *design* and *noise* factors, which may be considered as being the 'selected', easy-to-control parameters (for example the lime content in the example given previously) and the

'uncontrolled' parameters (the temperature in the kiln) respectively. It should be noted that design factors are typically considered as parameters which affect the mean of any measured output, whilst the noise factors may or may not influence both the deviation and the mean of the output. This consideration is important, as the underlying goal of Robust Design is essentially to deduce which factors influence the mean and which influence the deviation, and quantify their effects (Eriksson et al., 2000). These factors are arranged in *arrays*, where the design factors form *inner* arrays and the noise factors form *outer* arrays. During experimentation, each factor in the inner array is perturbed whilst the remaining design factors are held at a constant value, and for each of these combinations of values for the inner array the noise factors are perturbed between pre-described limits (typically corresponding to the expected or deduced maximum and minimum that will be seen in practice)[42]. The mean and standard deviation for each inner array combination can thus be computed over the recorded outputs for each of the outer array perturbations.

The resultant data is analysed using a series of regression analyses, which indicate those parameters that influence the deviation in the presence of essentially uncontrollable noise and also which modifications are necessary to reduce the influence of such a parameter and hence improve the robustness of the design. There is little need to expand upon these regression analyses, as consideration of the data generation stage of Robust Design reveals its inapplicability to the problem in hand. The subsequent analysis is based upon receiving data that has been generated by evaluating the output at specific input parameter states, a situation which is arguably highly unlikely to naturally occur within manufacturing data, and even if such input states were assumed or could be proved to be present in specific cases, this would considerably reduce the generality of any approach developed within this work. Robust Design requires the ability to maintain the value of a parameter to a specific value and also to deliberately perturb certain parameters by prescribed amounts, which arguably cannot be done if the item under examination has been produced within required tolerances using a representative manufacturing process, and the measured perturbations are simply variations within tolerance. It is noted that

[42] It is noted that it is not always possible, in practice, to completely control these noise factors even under experimental conditions

experimentation external to the manufacturing process may be possible, but this loses the benefit of examining the performance of a product that has been manufactured in exactly the same way as a product in service.

11.2 Statistical Analysis

When analysing data using traditional statistical techniques it is necessary to have some degree of understanding of the nature of both the expected pattern of the result and of the intrinsic nature of the data under examination. When applied to problems within manufacturing, Oh and Lee (1995) suggest that 'The statistical approaches make impractical assumptions, such as uncorrelated inputs, small-scale data and only one input variable.' Perhaps more damning in terms of this research is the statement by Hong and Weiss (2001), that 'In classical statistics approach' (sic) 'it is assumed that the correct model is known and the focus is on the parameter estimation.' It is suggested that statistics are mainly of use in confirming or disproving hypotheses, or of confirming the fit of a specified relationship. The expected relationship or hypothesis must be stated in advance, and either the nature of the relationship is defined or the veracity of the hypothesis is considered. There is little novelty in either of these results, merely a refining or quantification of previously expected relationships. It is argued that such methods do not present the most suitable method for analysing manufacturing data, as essentially the investigator must understand the nature of the relationships within the data before modelling begins. It is suggested that this could act to hamper analysis in situations where little is known or understood about the data, and there is no guidance as to the relationships within the data.

11.3 Database-oriented Methods

Recent increases in the amounts of electronic data collection and storage have been so vast that new methods are required to both collate and analyse this data. To emphasise the volume of data under consideration, Piatetsky-Shapiro (1999) documents the 11 terabyte database of customer transactions installed by Wal-Mart in 1998. The development of tools to assist in the interrogation of such databases started with Structured Query Language (SQL), which is primarily a system that allows data to be browsed and investigated in terms of data extraction. Online Analytical Processing (OLAP) builds on this and presents a series of tools that allow for data to be visualised

and presented in different ways. Data Mining (DM) represents a significant leap from OLAP, and utilises various methods developed in Machine Learning and Artificial Intelligence to investigate and model these databases.

Structured Query Language (SQL) and Online Analytical Processing (OLAP)

SQL may be considered to be a language that facilitates the creation, management and investigation of databases. It is perhaps the most common of all methods of interpreting databases, its ubiquity and transferability leading to its definition as the '..Lingua Franca for database access' (Deutsch, 2002). SQL consists of a set of standard operators which permit simple queries to be made regarding a database, and it is the differences in the nature and definition of these operators that give rise to the various forms, commercial or otherwise, of SQL implementation (Reese et al., 2002). These operators range in function from commands to delete records to functions that allow records containing specifically requested content to be displayed. In this way SQL provides a method to extract and present information contained in a database based upon known quantities within that data, but it does not intrinsically present a method of deducing information not already known or anticipated.

OLAP goes some way to addressing these shortcomings by providing extended functionality with which to interrogate and view data. OLAP automatically aggregates data into hierarchies, which provide different resolutions within the data and allow examination on a more general scale. There is also the facility to display and compare data in different projections, assisting in identifying patterns which may not be immediately visible in a flat file. This change in projection is typically achieved by browsing using *dimensions* and *measures* (Sarawagi et al., 1998), terms for which Hudomalj and Vidmar (2003)give the following simplified definition: '..dimensions are the data usually treated as independent variables in tables or charts, while measures are the characteristics tabulated or depicted there.' Various operations allow for data representations using these dimensions and measures to be varied, facilitating the discovery of patterns within the data.

Key to the success of OLAP is the skill of the operator in selecting appropriate dimensions, measures and aggregates to investigate, and in carrying out the analysis in such a way that patterns in the data are clearly identified. Agarwal *et al* (1996) suggest that areas indicated as anomalous during investigation may be considered to represent

interesting areas for further analysis, and hence present opportunities to discover patterns that are previously unknown, but it is suggested here that such results depend too heavily upon the operator. It is also argued that the nature of the analysis, which relies heavily upon essentially visual comparisons between different representations of the data, may prevent patterns in highly dimensional data from being clearly identified or evaluated.

11.4 Data Mining (DM)

DM addresses some of the shortcomings of OLAP, allowing queries to be presented at a much more abstract level than those possible using OLAP (Fayyad and Stolorz, 1997). DM may be defined as the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Chen et al., 1996). Smyth (2000) adds the further caveat that DM is a '...search for previously unsuspected structure and patterns in data', alluding to the freedom that DM has from requirements for prior assumptions or knowledge, such as is present in more traditional statistical analysis. A detailed review of DM may be found in Chapter 2.

Chapter 12 Appendix B – Machine Learning

This appendix discusses the Artificial Neural Network (ANN), Decision Tree Induction (DTI) and Simulated Evolution algorithms in detail. The ANN and DTI algorithms have been usefully applied in this research, and the greater detail given here is intended to further ground the modelling. The Simulated Evolution family of algorithms, whilst not applied in this research, are described here as they offer useful utility when used alongside the ANN algorithm and may usefully be considered for further work.

12.1 Artificial Neural Networks

Despite the huge improvements in computing power, driven by advances in both hardware and software, computers still struggle to perform tasks taken for granted by humans. The diversity of tasks for which human intelligence outshines AI is remarkable, from the recognition of a face from a split-second glance to the balancing of a busy social and work life.

After early Greek theories suggesting the heart was involved in the process of thought, it was not until the 1800s that the brain was accepted as being the central source of intelligence (Davalo and Naim, 1991), focussing subsequent research. Many eminent mathematicians, such as Alan Turing, looked to the function of the brain for inspiration for developing computing machines (Aleksander, 1991).

The first model created specifically to mimic the function of the brain was McCulloch and Pitt's perceptron (McCulloch and Pitts, 1943) which combined series of neurons into systems which could be set to perform simple logical tasks. The inability of the perceptron to model non-linearly separable problems, such as an Exclusive Or (XOR) logical function, was pointed out by Minsky and Papert in 1969, as described by Beale and Jackson (1990), Fu (1994)and Veelenturf (1995). This drawback was attributed to the lapse in research into Artificial Neural Networks for the next 20 years. In Rumelhart and McClelland's (1986) seminal work 20 years later the non-linear problem was effectively solved by producing a network with multiple layers of nodes, illustrated by their example which effectively modelled the XOR function. Alongside Rumelhart and

McClelland, an eminent physicist, John Hopfield, is also credited with renewing interest in the field for his work on content-addressable networks (Davalo and Naim, 1991).

12.1.1 Biological Basis

The brain and central nervous system are composed of massive numbers of basic units which are highly connected. Numbers of these units, called neurons, are estimated at around 5 billion (Davalo and Naim, 1991) with an estimated 10,000 connections each (Beale and Jackson, 1990). These neurons perform 5 distinct tasks:

- Receiving input from connected neighbours
- Sum input signals
- Initiate pulses in the body of the neuron (*activate* the neuron)
- Channel the pulse throughout neuron
- Conduct pulse to connected neighbours

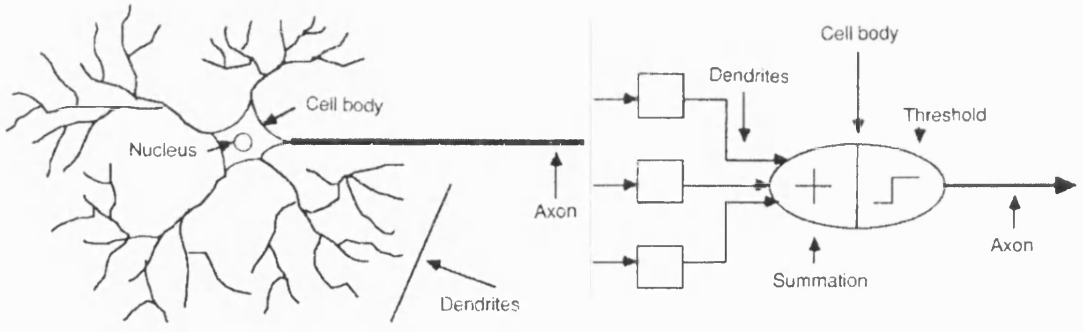


Figure 50 Components and Model of Neuron (Davalo and Naim, 1991)

Figure 50 shows the arrangement of a neuron and, more usefully, an associated simplified model. The input to the neuron is influenced both by the number of inputs and by a function acting on each individual input. There is a threshold, indicated by a hard limit in this case, which will activate the neuron once the input exceeds a certain magnitude. Once activated, a pulse of constant, pre-determined amplitude is produced and propagated by the neuron. In this way, 'information' may be represented by the frequency of these pulses, analogous to the pulsing of a laser in a fibre-optic system.

In order to optimise the network, to allow for learning, there must be some adjusting of both the amount and nature of the connectivity between neurons, and some internal

adaptation of the neuron as well. This is facilitated by various means which may be summarised as follows:

- Creation or removal of links between neurons
- Amplification or diminution of input signal (adjusting input 'weights')
- Modification of summing and activation function

12.1.2 Artificial Network

As in the brain, there are numerous arrangements and types of network, however, arguably the most common artificial network is the feed-forward as shown in Figure 51

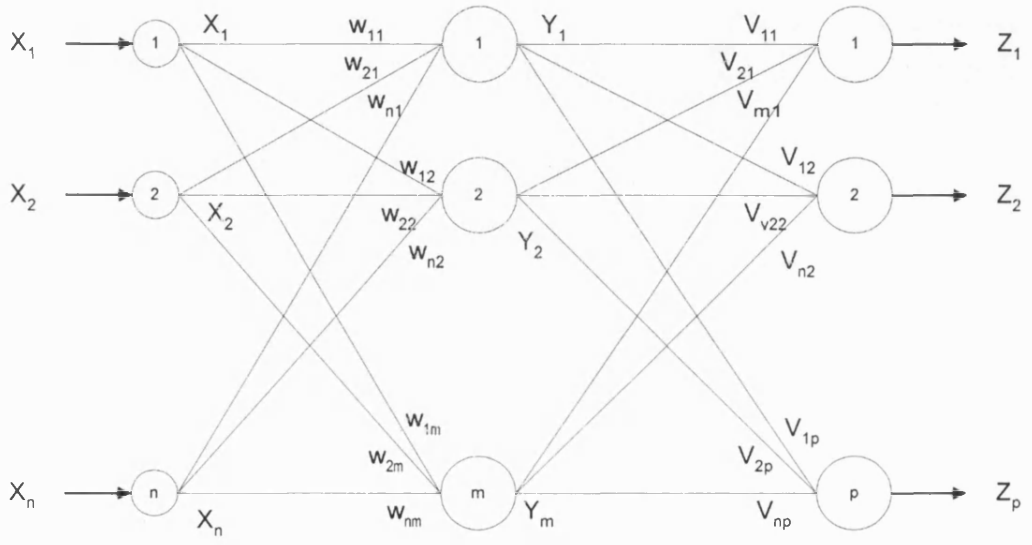


Figure 51 Feed-Forward Neural Network

The network seeks to map the input vector $X = (X_1, X_2, \ldots, X_n)$ to the output vector $Z = (Z_1, Z_2, \ldots, Z_n)$. Nodes are connected to each node in the following layer, and the output from each node is multiplied by a weight, here described by the vectors $W = (W_{11}, W_{12}, \ldots, W_{nm})$ and $V = (V_{11}, V_{12}, \ldots, V_{mp})$. The inputs to each node of the subsequent layer consist of the sum of the weighted outputs from the nodes of the preceding layer. From this input, an activation function translates an appropriate output, and this in turn forms part of the input for the next layer. In order to obtain a desired output vector from a given input, it is necessary to select appropriate activation functions for the nodes and then to adjust the values of the weights. This can be done using the back-propagation algorithm proposed by Rumelhart *et al* (1986), and will be described in later sections.

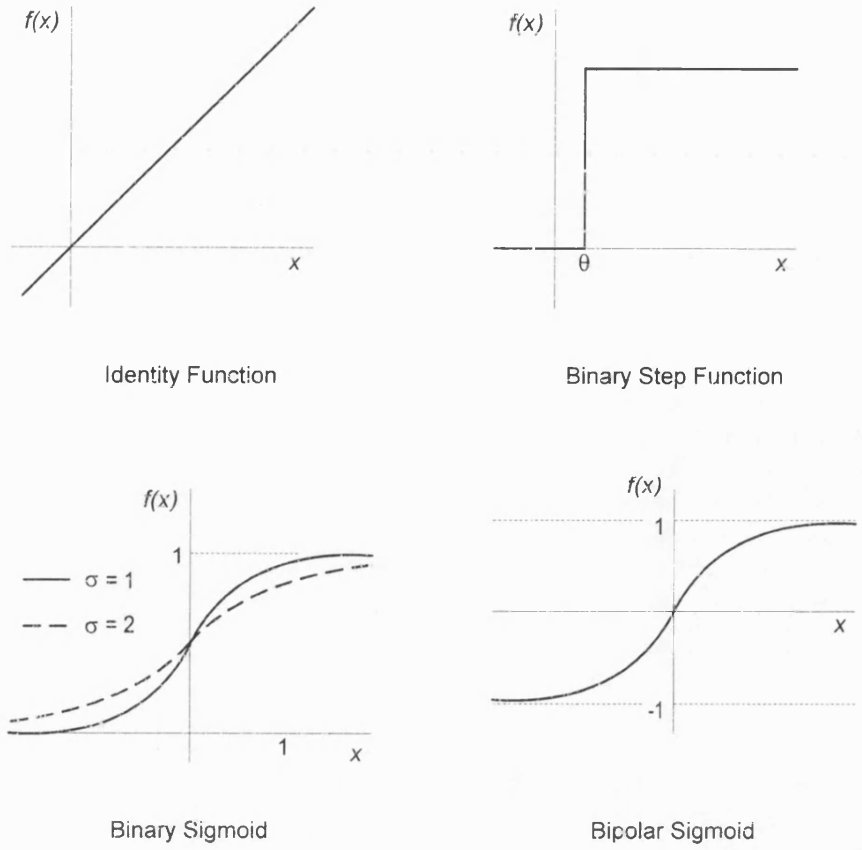


Figure 52 Activation Functions for Nodes of ANNs

The activation function plays a significant role in the function and successful training of a network. This function seeks to translate the sum of inputs into an appropriate output. Four of the most common are illustrated in Figure 52, and are typically used for different purposes. The identity function merely passes the summed inputs through, and is typically used as the activation function for the input nodes (although there is no requirement for this). The identity function is rarely used in the hidden or output layers, as without the use of a non-linear function the benefits of multiple layers are lost – two or more layers with linear activation functions will provide results no different to a single layer (Fausett, 1994). The binary step function has a specified threshold θ, once the summed input exceeds this amount the activation (output) is 1, otherwise it is zero. The sigmoid functions are perhaps the most useful, with two commonly used types, the binary (logistic) and bipolar (hyperbolic tangent). The binary sigmoid utilises a steepness factor σ, for which Figure 52 shows the curves for $\sigma = 1$ and $\sigma = 2$. As these functions translate notionally quite large inputs to values between –1 and 1 they are often referred to as squashing functions. Most networks are arranged with identity functions for nodes in the

input layer and sigmoid functions for the hidden and output layers, either binary or bipolar depending on the desired output range.

12.1.3 Training the Network

Upon creating a network, the adjustable parameters are typically initialised to random values between two limits (IPVR, 1995). In order to arrange the network to describe the process in question, it is necessary to adjust the various parameters to suitable values. This problem was another contributing to factor in the decline in the popularity of Neural Networks during the 1960's and 70's, due to limitations in the training method in common use. This method, Hebbian learning, stated that a weight should be increased if the two neurons being connected were activated at the same time. This rather simple method is impractical for multi-layered networks, and only a certain range of problems could be modelled. There was also no method of diminishing a weight, thus introducing a possible cause of instability (Fu, 1994).

A viable solution was presented by Rumelhart *et al* (1986) with the back-propagation algorithm, although it should be noted that others should also credited with similar discoveries, notably Parker in 1982 and Werbos in 1974 (Tsoukalas and Uhrig, 1997). The application of this rule involves passing a pattern with a known, desired output through the untrained network and computing an error function. This error function indicates the difference between the given output and the desired output. The back-propagation algorithm then precedes back through the network scaling the weights of connections leading to high-error nodes and leaving 'correct' connections unchanged.

12.1.4 Principles of Back-Propagation Algorithm

It is perhaps useful to describe the computations involved in back-propagation, both to understand how the algorithm functions and also to provide a base from which developments to this algorithm may be presented.

The weights within the network are initially randomly set. When an input vector is passed to the input nodes, the various activations of these nodes are passed through the network, factored by the weightings, and serve as inputs into nodes in the following later. This process continues until the output layer is reached, at which point the activations of these nodes is taken as the network output. The initial randomisation of the weight

vectors tends to result in large errors between the given network output and the desired output.

The observed error is used to inform the updating of the values for weights. It is necessary to deduce how much error to assign to each weight. In the case of weights leading to the output nodes the error is relatively easily deduced as the overall error at the node is explicitly stated. The weight change is therefore informed by this error value, however a learning rate is used to control the severity of the weight adjustment. Such a measure ensures that the network remains stable when training.

It is not possible to deduce the error of nodes in the hidden layers (the credit assignment problem, Davalo and Naim, 1991). The error is computed by considering how the activation or output of the hidden layer node contributes to the error at each node in the following layer, or in other words how much of the error is propagated back to the hidden layer node.

This process is repeated for all hidden layers; when the error for each node in a given layer is computed and the resultant weight changes deduced and implemented, the process can be repeated for the proceeding layer. When all the weights are adjusted the next input vector is passed through and the updating repeated. The training vector set is cycled to create a continuous flow of input vectors, and training is considered complete when some stopping criteria is reached. This criteria is usually a pre-defined level for the sum-squared error, which by its very nature means that training cannot be exact (Malinov et al., 2001) but this is typically to the benefit of network generalisation.

12.1.5 Considerations and Developments of Back-Propagation

There are various pre-set factors which influence the performance of feed-forward networks trained using back-propagation, such as learning rate (the rate at which weights are adjusted), range for initial weight randomisation etc. Bertels *et al* (2001) note the sensitivity of the network to changes in such parameters, and describe research into the onset of chaos, and its detrimental affect on the speed of learning. As the back-propagation algorithm is a gradient-descent method it is possible that it may fall into false minima (Potter, 1995), and where the minimum error has a very small surface (in effect resides in a narrow trough within the error curve) problems with oscillation become common. These two problems can be addressed by the inclusion of a momentum term, which adds a specified fraction of the previous weight change to the new weight change.

This reduces the magnitude of weight changes where the direction is opposite to the previous change (where the error is assumed to have overshot the minimum), helping to reduce oscillation. As the addition of the previous weight change will act to increase he subsequent weight change, flat spots on the error surface, where the error value is still some way from the minimum, will be traversed much more rapidly.

Alternatives to the backpropagation algorithm exist, such as Curvature-Driven Smoothing (Bishop, 1993). This method makes use of prior knowledge, something which Joerding and Meador (1991) suggest is important in areas with low data volume.

12.1.6 Variations of Neural Networks

The standard feed-forward network is widely accepted as being the most common form of ANN, however several other networks have been developed either to address certain limitations with feed-forward networks or to fulfil a role to which feed-forward networks are not best suited.

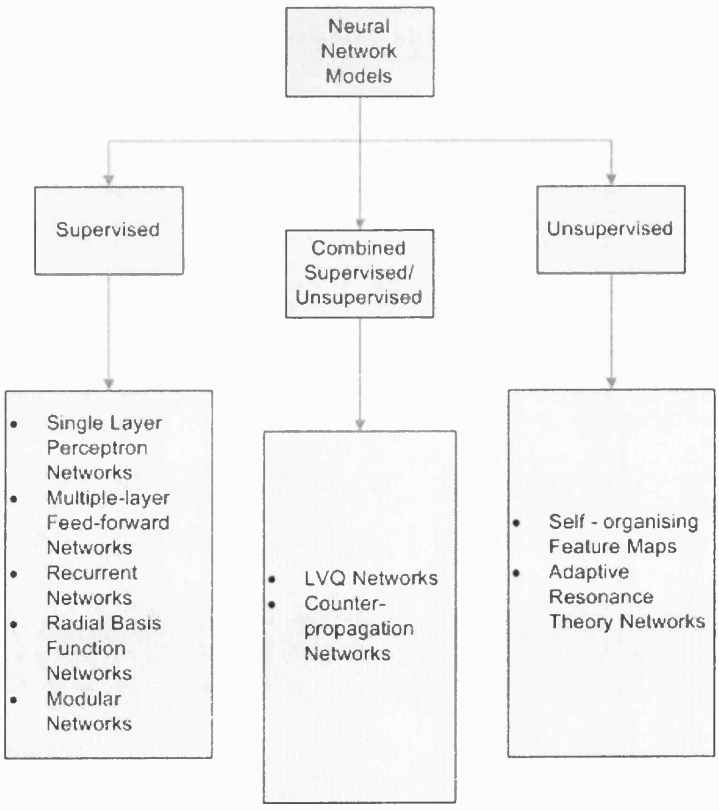


Figure 53 A Classification of ANN Models (Sethi, 2001)

Figure 53 shows a list of variations of ANN models as given by Sethi (2001). A distinction is made here between supervised and unsupervised networks, which may

simply be described as the presence or absence of a target output respectively. In this research focus will be upon supervised learning, where attempts are made to create a model that offers a prediction as to the value of an 'output' parameter based upon the values of a series of 'input' parameters. A few key variations from this list will be briefly discussed, extending the scope of the study.

Recurrent Networks

Recurrent Networks are in essence developments of the feed-forward type of network, where nodes are connected to nodes in previous layers in a form of feedback loop, and information is passed both backwards and forward through the network. Figure 54 shows the layout of such a network, the feedback nodes are referred to as context nodes, although they are also referred to as memory nodes as they store a component of the output from the hidden layer node. The weights for the context node connections is set at a pre-determined value between 0 and 1, allowing the network to be trained using the back-propagation algorithm with no significant penalty over non-recurrent networks (Fausett, 1994). The network shown in Figure 54 is typically referred to as an Elman Network, where the context units have an input from the hidden layer nodes. The network can be rearranged with the output layer nodes forming the input into the feedback loop, whilst still maintaining the context output connection to the hidden layer, and this is termed a Jordan Network.

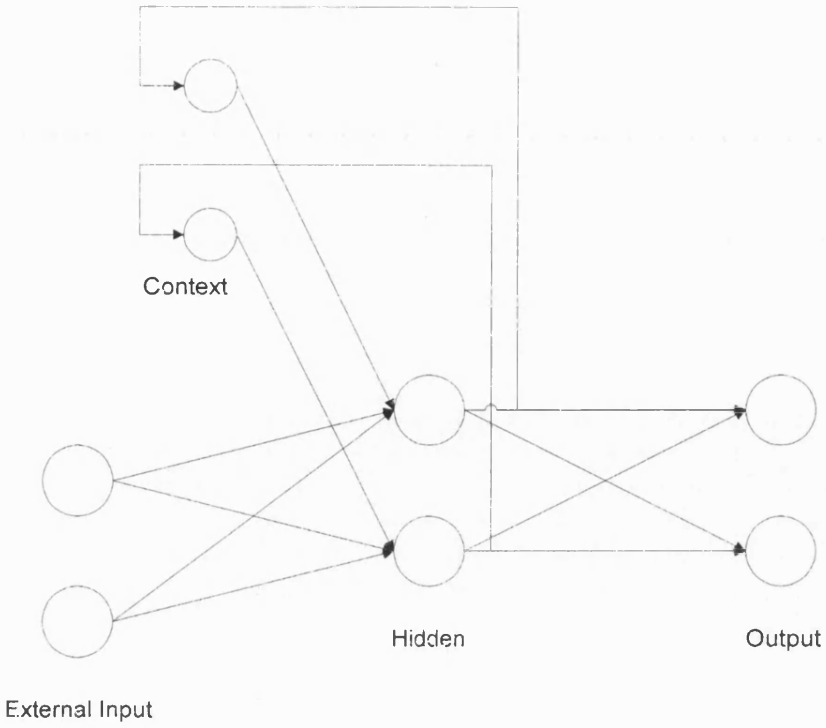


Figure 54 Recurrent (Elman) Network

The introduction of the context memory units gives the network a concept of time, which although present in a feed-forward network is not intrinsic to its function. This temporal property makes recurrent networks ideal for dynamic problems, such as the use of a network in a control system (Pham and Karaboga, 1999a). A major drawback of such networks is the method of training; although back-propagation works well for adjusting the forward weights, the feedback weights require manual adjustment and are typically selected using an iterative process. This is both time-consuming and lacks precision.

Hopfield Network

The Hopfield Neural Network, as described by Fu (1994), is a 'content-addressable' or associative network, in that it attempts to relate an input pattern to a pattern stored in memory. This network has a different architecture to feed-forward, as shown in Figure 55, with complete interconnectivity between a single layer of nodes. This single layer serves as both the input and output layer, and has one node for each feature of the input data.

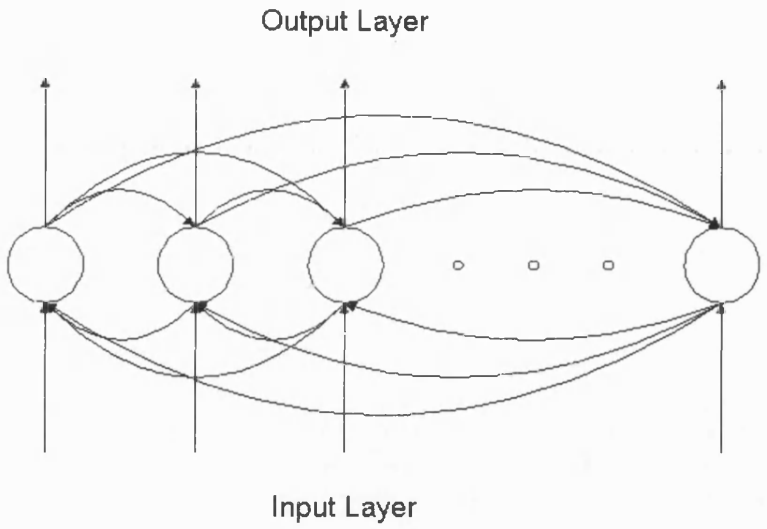


Figure 55 Hopfield Network

This type of network differs from a traditional network in that the weights are not adjusted during training, rather the activations of the nodes are adjusted over a period of steps to deduce a stable activation pattern which serves as a best representation of the input pattern. These networks are typically used in optimisation and constraint satisfaction problems (Potter, 1995). By using individual nodes to represent hypotheses and the weights between each node to represent constraints (i.e. by assigning a –1 to weights where two hypotheses cannot both be true and a +1 where both are true) it is possible to use the activations of each node to indicate whether such statements are true or false.

There are numerous issues surrounding Hopfield networks, such as the limitations on the number of memories that may be stored, the likelihood of settling into local minima or not settling at all, and the vast increase in number of connections when the number of nodes is increased (the number of connections being proportional to the square of the number of nodes).

Simulated Annealing can reduce the possibility of settling into local minima, by mimicking the continually reducing energy state of a cooling of a metal. An analogy given by Davalo and Naim (1991) presents this phenomenon in simple terms. By imagining a ball rolling on an energy 'surface' consisting of numerous peaks and valleys, the likelihood of settling into a small valley, a local minima, is large. By heating the surface, thus increasing the overall energy, the peaks and valleys all rise to a similar level. During cooling, the valleys gradually extend, and by only cooling in small

increments and waiting for the ball to settle in the minimum at each stage, the ball will remain in the overall greatest minimum and thus avoid local minima. In the Hopfield Network, energy states can be represented by degrees of randomness within the activation calculations, and as the 'temperature' decreases the degree of randomness also reduces. Networks designed to incorporate these ideas are termed Boltzmann Machines, in recognition of the discoverer of the relationship between temperature and probability of energy state within metal. For further details the reader is directed towards the work of Ackley *et al* (1985), who are credited with the inception of such a network.

Self-Organising Maps (SOMs)

In a case where no desired output data exists, it is not possible to train a standard feed-forward network using the back-propagation algorithm. In such cases it is useful to cluster the patterns, thus deducing which parameters are closely related, by the use of SOMs of the type pioneered by Kohonen (Smith and Gupta, 2000). These SOMs were inspired by the discovery that the brain is organised into discrete areas, labelled 'receptive fields', which are activated by various sensory features (Kohonen, 1988).

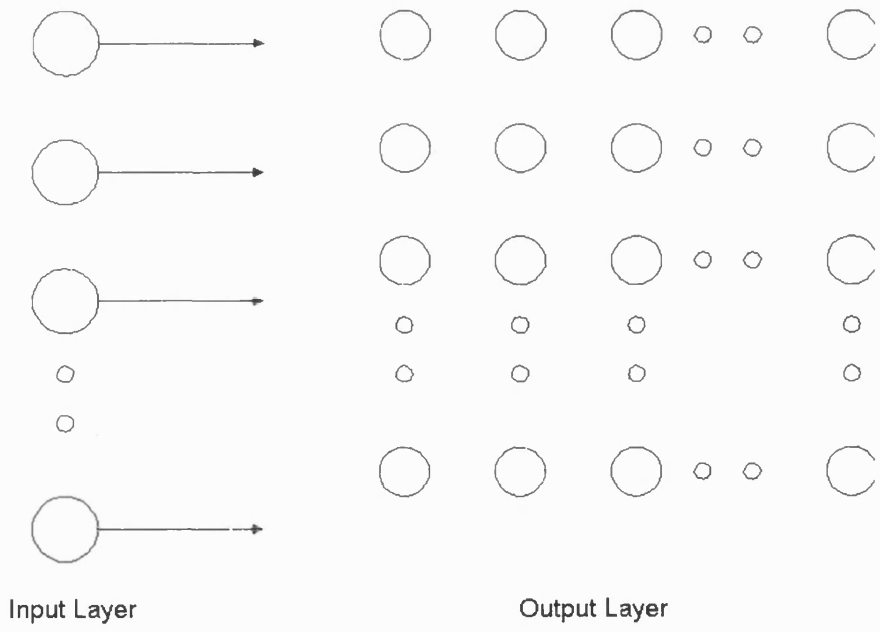


Figure 56 Self-Organising Network (Kohonen Network)

Figure 56 shows such a network with a 2-dimensional output space (it is possible to use higher dimensional output spaces if the situation requires). When an input is presented to the network, each node is activated to a degree determined by the strength of the weights between itself and each input node. The node with the highest magnitude of activation is

selected as the 'winning' node, and the strengths of the weights for this node (and to a lesser extent those of its immediate neighbours) are adjusted so as to provide an activation of even greater magnitude upon presentation of a similar input. No other changes take place to any other part of the network. As training progresses, the network will begin to organise itself into numerous regions, indicated by their response to inputs with certain characteristics, similar inputs will cause activity within the same region. The physical location of a node or group of nodes carries semantic information (Potter, 1995), and the locations of specific regions within the network will depend on the initial weights assigned to each connection.

This network has the strengths of self-organisation and of unsupervised learning, where data with no designated output may be examined for clustering of similar instances. Besides being useful in areas where the outputs are unknown, this has benefits in areas where the output is not easily defined or there is a degree of subjectivity to what is considered a good output.

The problems of network design are as prevalent within SOMs as with feed-forward networks. The initialisation of weights influences which nodes will initially become winners, and there is a danger that certain nodes will become dominant. This will remove the generalisation of the network, where inputs with quite disparate characteristics will cause greatest activations within the same region of the network. The use of a 'conscience' term, which penalises repetitive winners, can help to alleviate this. The topology design is also significant, with too small an output layer (i.e. number of nodes in the grid or array) it becomes difficult to differentiate between overlapping regions, and too large a layer is unlikely to be divisible into easily recognisable regions.

The knowledge obtained from a SOM consists of information regarding the clustering of patterns, indicating which have similar characteristics. The usefulness of this knowledge is tempered by the inability to select which parameters to attempt to cluster inputs by; as an example it would not be particularly useful to cluster engine data according to the date tests were performed on. It is much more useful to classify instances, defining a particular variable by which to segregate all inputs, and a form of SOM, Learning Vector Quantization (LVQ), can be used for this purpose. For a description of LVQs, the interested reader is directed towards the work of Kohonen (Kohonen, 1995) whose work is acknowledged as the foundation of research into SOMs.

12.1.7 Development of Neural Networks

The majority of the developments listed here relate to feed-forward networks, primarily as these are the most prevalent networks in use. Many of the developments can be applied to different forms of network, for example the use of Simulated Evolution in network design, and examples of usage in different types of network will be given where possible.

Architecture

ANNs are extremely sensitive to many factors, such as the architecture, initial weights, order in which the attributes are passed through the system etc. (Bertels et al., 2001). Hornik *et al* (1989) reveal that a suitable ANN with a single hidden layer can approximate "..any measurable function to any desired degree of accuracy". This is not to say that such an ANN is the optimum. Whilst sufficient complexity is required to approximate the function (Bishop, 1993), too much complexity may result in specialisation, or the recording of each specific input, errors included (Looney, 1996). Hagiwara (1993) makes the valid point that excessive complexity also adds to the computational time and complicates possible interpretation of the structure of the network.

Bishop (1995) gives a useful analogy with fitting a polynomial to data points; with too few coefficients the polynomial will be unable to capture the underlying function, and with too many the polynomial will start to measure the noise on the data and generalisation will be reduced. The desired number of coefficients for the polynomial depends on the data to which it is being fitted, if the underlying function is for example 3rd order then a polynomial with 4 coefficients, or 3 orders will provide the best fit. This idea that the complexity of a polynomial must be optimised for each particular problem also applies to Neural Networks. If we consider the order of the network, or the number of degrees of freedom, to be controlled by the number of parameters such as nodes and weights, then selection of the most suitable architecture is equivalent to setting the desired number of degrees of freedom for the network.

Bishop further describes methods to limit excess complexity within an ANN, amongst which are mentions of constructive or deconstructive methods (growing or pruning the network structure either initially or during training). Huang and Huang (1991) describe heavily mathematical research into setting optimum architectures for ANNs with a single

hidden layer by computing the degree of complexity explicitly. Hagiwara (1993) puts forward a method of pruning which seeks to build on some of this previous work. The suggested method seeks to identify which hidden units are most active, in terms of the strength of weight between other nodes, degree of activation and how much it contributes to activation of subsequent nodes, and to remove the least active nodes. Hagiwara concludes that identification and removal of nodes with low weight connections is most effective, and reports good increases in generality with networks pruned in this manner.

An attempt to allow a network to prune itself has been proposed by Werbos (1988) where a decay factor is included in the weight change term. This decay factor gradually reduces the weights which are not contributing to the output of the network and are excluded from any major weight updates. The gradual reduction over numerous cycles will eventually reduce less-active weights to zero, thus simplifying the network.

Hastie *et al* (2001) discuss this idea of a weight term further, and suggest that the weight decay does not have to result in the removal of weights, merely to prevent them from becoming excessive. This weight term helps to prevent overfitting by driving certain weights to zero based upon an iteratively- derived weight decay factor that is simply added to the error function. However, Hastie et al concede that a similar effect (in terms of preventing overfitting) can be obtained by monitoring how accurately a separate *test* dataset (a small sample extracted from the training data) is modelled during the training process, and terminating training when the test set accuracy begins to decrease. It is assumed that it is at this point that the network is beginning to overtrain or overfit the data.

Mehrotra *et al* (1991) approach the problem of training of errors in ANNs from a different angle. Specialisation is referred to as the recording of instances together with their associated errors, analogous to trying to fit a 2nd order data dispersion with a 3rd order curve, however it entirely possible that errors may actually be incorporated by dividing the output space incorrectly.

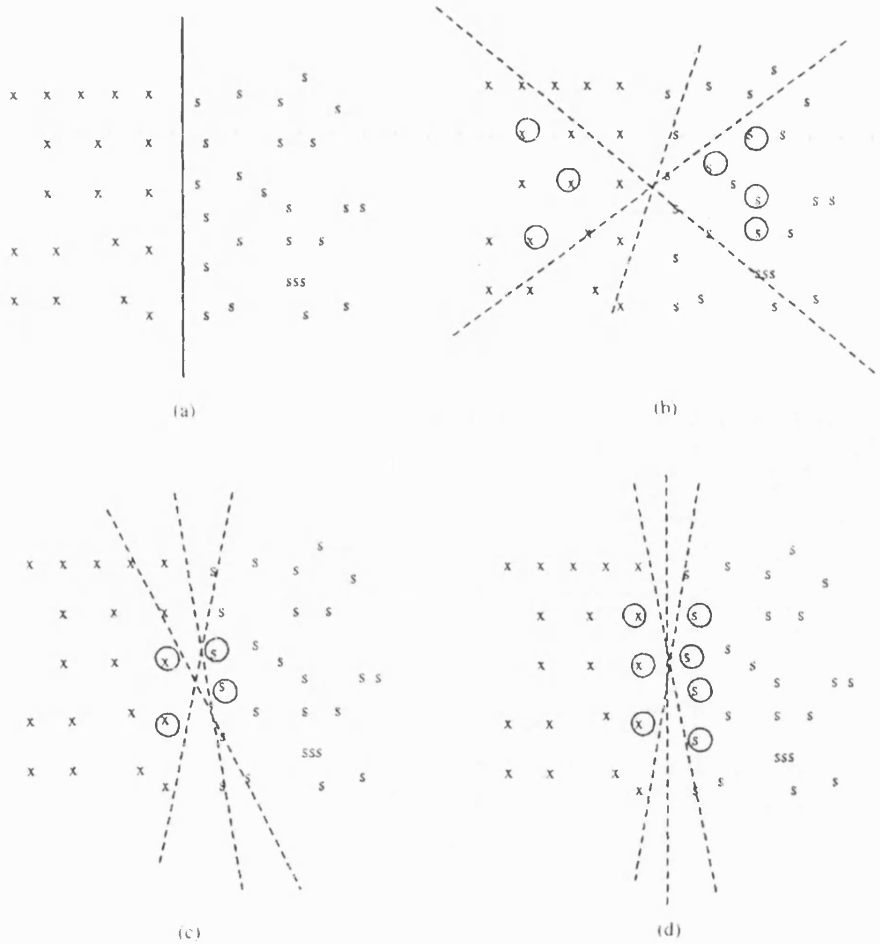


Figure 57 Illustration of Subspace Division Errors (Mehrotra et al, 1991)

Figure 57 shows how it is possible to divide clearly delineated samples incorrectly if an unrepresentative subset of the data is used in training. The illustration depicts a population of data points for two distinct classes ('x' and 's') which are to be divided by a hyperplane. Section (a) indicates the desired position of the hyperplane, where the two separate classes of instance are accurately divided. Section (b) indicates the possible positions of the hyperplane where the hyperplane is arranged only to divide the subset of circled instances, which is does accurately but with poor results for the remaining population. By increasing the number of boundary samples, as in sections (c) and (d), the hyperplane is much more confined and the hyperplane divides the population much more accurately. This is the thrust of the work by Mehrotra *et al.*, where research is focussed on quantifying the exact number of samples required for successful classification. To this end, there are numerous assumptions on the distribution of the data; in order to ensure sufficient boundary samples are included in the extracted training subset it is necessary to estimate the proportion of boundary samples to 'normal', more central-

subspace located samples. An assumption of a uniform distribution is argued to be unrealistic unless there is sufficient evidence to verify this, and it is conceded within the discussed work that the number of instances required will increase for non-uniformly distributed datasets. From this, it is useful to note that accurate training depends on incorporating sufficient boundary samples within each subset, although it is felt that the uninformed assumption of a uniform distribution cannot be guaranteed in many applications.

Transparency

A criticism levelled at Neural Networks is their inability to communicate the structure of a model, in effect they are 'black boxes' from which a decision is made without any indication of the rationale behind it or any explanation as to the structure of the data. It should be noted that such information does exist, but requires some method of interpretation of the weights, activations and connectivity of the network. There has been a vast amount of recent research into network interpretation, typically in the form of allowing logical rules to be extracted. A useful survey and critique of methods presented pre-1995 has been presented by Andrews *et al* (1995) which breaks the proposed approaches down into various genres.

Lu *et al* (1996) present what is arguably a common rationale for extracting rules, and will be discussed as an example. In this case, logical rules are extracted from trained and pruned (redundant portions removed or amalgamated) single-hidden-layer networks. This approach seeks to create rules relating hidden layer activations to output, and inputs to hidden layer activation. In order to minimise the resulting rule sets, the hidden layer activations are first clustered into groups and a single discrete value is used to represent all activations within the group. In this way only a small number of discrete activations are used within the hidden layer. The two rule sets can then be amalgamated, giving rules that relate the inputs to the outputs. Fu (1999) and more recently Fu and Shortliffe (2000) have developed a similar approach to include the idea of a certainty factor, a factor which assesses the importance and accuracy of individual rules in order to produce a reduced set consisting of only the most significant rules. Gupta et al (1999) also consider the importance of each rule, although the ranking is incorporated into the construction of rules by arranging node connection matrices in order of activation. Taha and Ghosh (1999) present several developments, such as a method of encompassing prior

knowledge into a network, a method to increase or reduce granularity (the level of detail of the extracted rules), and a method of evaluating rules to deduce salient and accurate information (as an alternative to the certainty factor approach of Fu and Shortliffe).

There is a great deal of current work into deducing suitable methods of rule extraction from ANNs. Taha and Ghosh concede that more work is required on ways to establish the suitability and adequacy of extracted rules, and how best to combine both the information from the trained network and the induced rules in order to refine the knowledge and to ensure there is 'truth maintenance', or consistency of knowledge within the domain. There has been no clear indication as to which method of rule extraction is the most suitable for use in this research, it is anticipated that more successful methods will become more prevalent over time, and may therefore be considered in further work. However, until a clear consensus is reached or a demonstrably successful method has been devised, it is argued that it would be more beneficial to focus attention within ANN information extraction elsewhere.

Sensitivity Analysis

The extraction of information may be addressed by considering the network response, in preference to attempting to make sense of the network structure. The idea of measuring the sensitivity of network output in response to changes in network input allows for the degree of influence exerted by an input parameter to be evaluated. This form of sensitivity analysis has been successfully implemented in practice, having benefits of simple application and easily understandable results.

There are two different forms of sensitivity analysis applied to ANNs, both with markedly different ambitions. The first form of sensitivity analysis seeks to ensure that a network has little sensitivity to small changes in input or weighting, in effect ensuring that noise does not influence the performance of the network (Zeng and Yeung, 2001). This analysis may investigate sensitivity to changes in both the weightings of the network and the inputs to the network (Oh and Lee, 1995).

The second form of sensitivity analysis seeks to extract information from a network rather than maintain the accuracy of prediction. Such methods are most commonly discussed in literature with a more practical focus. Shelley and Stevenson (2000) present a method of investigating ANNs that model well completion methods in geological engineering. These well completion methods vary depending upon the prevailing

geological conditions, and the most suitable method of completion is selected based upon a subjective consideration of such conditions, often in the face of conflicting information. ANN models were created to give a prediction of production output based upon these conditions and the method of well completion. Each parameter in the trained network was perturbed by +/-10% and the complete dataset was passed through the network. The effect upon network prediction was summed across all of these instances, and the information used to deduce the nature of changes that would result in the greatest production improvement. Rademan *et al* (1996) describe the modelling of operations within chemical engineering, using the example of a leaching process to illustrate their approach. The LVQ algorithm was used, a variation of the SOM algorithm described previously. The sensitivity analysis takes a different tack, comparing the response of the network over the complete dataset to the response when one input parameter at a time is omitted from the dataset. This difference in sensitivity analysis is algorithm-independent, it is equivalent to perturbing each output to zero as opposed to increasing the value. Both methods are arbitrary, and in neither case was the method of perturbation justified. The Clementine software package (SPSS, 2002) automatically performs a sensitivity analysis upon construction of an ANN model, and instead of perturbing each input by a pre-defined amount it holds each parameter in turn to the maximum value seen in the dataset whilst passing the remaining data through. Along with the method of excluding one parameter at a time as proposed by Rademan *et al*, this approach has the benefit of minimising the possibility of presenting an infeasible permutation to the network: if a parameter is artificially increased by an arbitrary amount there is a chance that such an increase would fall outside of what could practically occur. This could lead to inaccurate results as the model would be used in extrapolation instead of interpolation, which falls outside of the remit of an ANN model as it simply learns a function by examination of previous cases, and hence cannot be effectively used in situations that fall outside of the coverage of these previous examples. This form of sensitivity analysis can also be used during the training process to indicate which input parameters can be excluded or deleted, thus allowing for the architecture to be iteratively improved (Engelbrecht and Cleote, 1998), (Ng and Yeung, 2002)

Sensitivity analysis can also be used in bespoke situations to improve network performance. Engelbrecht (2001) uses the sensitivity of a network to specific training patterns to pursue a form of adaptive training in cases of classification, where a network

is assigned a step activation function for each binary output (for a definition of a step output see Figure 52). In cases where the network passes through the step threshold, it can be inferred that the pattern being presented is near a decision boundary and, as described by Mehrotra in Figure 57, it therefore contains information that will be most beneficial for network training.

Reconciliation of Missing and Erroneous Data

Robust handling of missing or distorted data is one of the strengths of the ANN approach, but when this data is used to train recurrent ANNs the effects of such errors are multiplied as the erroneous data is fed back into the network, causing more disturbance to training. Meert (1998) suggests the use of two recurrent networks in series, where the first network effectively pre-processes the data and estimates the values of the missing data. This smoothes the data for use in the second, modelling network. This approach was tested within the chemical process industry, where the inclusion of the first pre-processing network was seen to increase the performance of the network. Himmelblau and Karjala (1996)describe the use of an Elman network, a form of recurrent network, to predict step-wise input and output values for two chemical industry problems. Error was introduced into a training dataset, and from this data the network predicted the values of the data for the parameters at the next time step. The actual values, reconciled (estimated) values and the error-included values were compared, indicating that the network produced values which could accurately follow the true value of the process and was not unduly influenced by errors within the data.

The two works previously described tackle problems within on-line systems, where there is both a need to deal with data quickly and to incorporate some idea of time. Where such concerns are not applicable, it is feasible to utilise the ideas of these approaches to reconcile data for use in feed-forward networks, reducing data error and allowing the use of input instances that might be deemed too erroneous or incomplete for use. This is beneficial in this research, addressing the problem of low data volume and reducing the negative effect on generalisation of errors within the data.

12.1.8 Applications of ANNs

The almost completely exclusive use of electronic transaction in finance has resulted in the production of huge amounts of data, which can be usefully examined for many tasks, such as forecasting. There are many practical applications of neural networks in this

field, with an overview of techniques by Smith and Gupta (2000) andsurveys of literature by Wong and Selvi (1998) and Wong, Lai and Lam (2000). The descriptions of various techniques and their practical uses as described by Smith and Gupta give a good grounding in the use of neural networks for specific problems.

12.1.9 Applications in DM-Based areas

There are other examples of ANN implementation which are of more interest to the engineer. Of particular interest to this research are two applications which have a methodology closely related to CRISP-DM, although neither work is explicitly described as DM. The first of these two works, by Malinov *et al* (2001), seeks to predict the various properties of titanium alloys given their various compositions and treatments. Certain inefficiencies of the back-propagation algorithm are noted (such as the alteration of weights for each instance tending to undo the changes for previous instances), and another method, the Levenburg-Marquardt algorithm is proposed. This method is also proposed by Ray *et al* (1996) who suggest that this method requires less training time than standard back-propagation. This view is also agreed with by Hagan and Menhaj (1994) with the proviso that it is computationally more expensive and is only suitable for smaller networks. This method is similar to backpropagation, using a different method of evaluating error and computing the required weight and bias changes. Ray *et al* (1996) use an ANN with this algorithm to optimise the design of container ships and estimate added mass coefficients and, like Malinov *et al*, the results obtained from these models are good. Hagan and Menhaj compare the Levenburg-Marquardt algorithm to two different back-propagation algorithms (one with a variable learning rate) and obtain results which indicate superior performance of the Levenburg-Marquardt algorithm for smaller networks. The successful use of this algorithm within a DM-based implementation, and the suggestion that it is more efficient than back-propagation for smaller networks, suggests that this algorithm may be suitable for the problem at hand.

The second DM-oriented example of Reich and Barai (2000) includes a more illustrative example which describes the use of ANNs to optimise a marine propeller. The developed network maps a selection of 5 from 8 possible input parameters onto 3 output parameters via the use of 301 known instances. Focus is placed more on the accuracy of the models than the actual implementation, and various methods such as stacked generalisation and ANN ensembles are used to improve the generalisation of the model. The technique of

stacked generalisation consists of two algorithms, the first (designated level 0) is used to map an input x onto an output y, and has an associated error between the output y and the desired output y0. The second algorithm, designated level 1, uses the outputs of the level 0 models as inputs and seeks to combine the predictions of the level 0 models into one, final prediction. This technique is not widely used, as it is difficult to analyse and the method of implementation is problematic (Witten and Frank, 2000). The training of the level 1 model is awkward, as the model can easily be trained simply to use one particular input for all circumstances. Whilst this particular method is not ideal, it raises the important point that improvements in modelling can be accomplished by the use of multiple models, something that will be covered in later sections.

There are certain problems where the properties of neural networks are closely related to traditional methods of solution. Such an example is given by Lou and Perez (1995), where the energy-minimisation solution given by the use of a Hopfield Network is directly comparable to the total energy function in the Finite Element Stiffness Matrix method as used for structural analysis. Use of the training algorithm as applied to the Finite Element matrix allows for the parallel processing capabilities of neural networks to be exploited.

12.1.10 Applications and Issues in Engineering

Huang and Wanstedt (1998) use a standard back-propagation net to address three geological engineering problems for which the current method is not ideal. The current methods are based on approximate theoretical modelling, where physical representations are created with limited data. The ANN approach was shown to be more robust than traditional methods, mainly due to the absence of any assumption of the relationships between parameters. This approach also allows for simple updating of the models given new data, significant in areas with sparse data such as this. The identified problems have a similar nature to the problem in hand; the relationships between parameters are not well defined, the is low volume of data and the 'input' data is dispersed and uncontrolled. The reported success of the ANN approach to tackle such problems strongly suggests that this approach is suitable for investigation.

Two similar examples within the geological field are given by Goh (1995), where a back-propagation ANN is used to predict soil parameters for foundation design and the load capacity of hidden piles. Values for these parameters are commonly derived from

empirical relationships and laboratory testing, with statistical tests to infer relationships from the results. The two networks had different architectures, the first had 3 input nodes, 4 hidden nodes (optimised by continually adding nodes until there was no increase in generalisation) and 1 output node, whilst the second had 8 input, 3 hidden and 1 output node. The number of training instances used were 73 and 59 respectively and validation instances 29 and 35 respectively. This is similar to the number of instances available for this research. The reported results were good, for the second problem the correlation coefficient[43] for the ANN was 0.97 compared to 0.61, 0.76 and 0.89 for the 3 most commonly used experimental/statistical methods, thus indicating the usefulness of the ANN approach on limited datasets.

12.1.11 Applications Within Design and Manufacture

Zhang and Fuh (1998) describe research into an ANN to predict the final cost of a product given the various features designed into it. A standard back-propagation algorithm was used with an architecture decided by trial-and error. The final network had 21 input nodes (the number of identified, discernable features), 8 hidden nodes and 1 output node (the cost). A total of 66 different products were used to provide the training, validation and test data, with 32 for training 17 for both validation and testing. This is significantly less per node than for the example of Goh described previously, although erroneous validation instances (where the difference between predicted output and actual output exceeded a pres-set amount) were added to the training set under the premise that a feature contained within this instance was not present in other training instances. Despite the difficulty of direct comparison between percentage accuracy and a correlation coefficient, it can be seen that the accuracy is not as impressive as that of the ANN created by Goh, with only 88% of the validation instances being predicted to within 10% of the actual cost. The rather subjective delineation of features which serve as input suggests that the input data is subject to a degree of uncertainty and error. Further error is perhaps introduced by the increase in network complexity, the lower volume of training data and arguably the more complex domain.

[43] The measure of agreement between 2 sets of data, ranging from –1 (inverse) to 1 (identical)

An example of a network other than a back-propagation feed-forward is given by Smith *et al* (1996) who describe the use of a Hopfield Network to optimise the production in a motorcar assembly plant. The results obtained were favourable compared to traditional techniques in the field, although certain modifications were incorporated to reduce entry into local minima. This use of such networks is perhaps the most common, although Hopfield Networks can be used for tasks such as classification (Potter, 1995) they are primarily used for optimisation problems.

The comparison between the examples of Zhang and Fu and of Goh suggests that the outputs of networks with vague input, low volume of training data which characterise a domain with complex relationships between parameters (indicated by the necessity for more architecturally complex networks) will be less accurate. The problems of complex domain cannot be simply remedied, and it is difficult to address the problems of error in input data, save for removing obvious errors which has a detrimental knock-on effect for data quantity. The minimum quantity of training data is suggested by Hammerstrom (1993) to be between 5 and 10 instances for each weight of the network, however this estimate is useful only if the architecture of the network is already known and optimised. It is for this reason that the success of ANNs in modelling manufacturing data cannot be predicted, as there is scarce understanding of any of these issues. Such methods have been successfully implemented and therefore a potentially useful method of modelling, however it is suggested that the success of such an approach depends upon the necessary complexity of the network, which will be influenced to a large extent by the complexity of the domain, and by the quantity of data available in such a domain.

12.2 Simulated Evolution

Inspiration from biological phenomena in Machine Learning is not limited to Neural Networks. The field of Simulated Evolution, as the name implies, emulates the principles of optimisation in Darwinian evolution. This seeks to gradually improve the performance of a population of possible solutions to a given problem over numerous iterations (or generations). This is performed in a similar way to natural evolution, by mating and/or mutating individual models to produce 'offspring' with different properties, and then relying on 'survival of the fittest' to ensure the survival of and reproduction from the most suitable models. In this way the overall accuracy of the population of models increases with each generation.

12.2.1 Introduction and Brief History

Holland (1975) is widely accepted to be the instigator of research into Simulated Evolution, although useful applications of this research were not apparent until almost twenty years later (Wasserman, 1993). The term Genetic Algorithm is typically used to describe the field of Simulated Evolution, although this term in fact describes one facet of three in this field (Pohlheim, 1994), the other two being Evolutionary Algorithms and Evolutionary Programming. These methods operate on different levels of biological metaphor; Genetic Algorithms simulate changes on a chromosomal level, Evolutionary Algorithms operate on an individual level and Evolutionary Programming on a species level (Fogel, 1994). The interchangability of terms mirrors the interchangability of the approaches, there is no necessity that the implementation of these techniques follow the behaviour of the evolutionary process, researchers are free to incorporate aspects of each approach in their application (Tsoukalas and Uhrig, 1997).

The close relationship between Simulated Evolution and Neural Networks has led to research into methods to incorporate the benefits of both methods into one approach (Wasserman, 1993), for instance MacLeod and Maxwell (2000) expand the architecture of a simple Neural Network using a Simulated Evolution algorithm until the network is sufficiently complex to fulfil its required purpose. This avenue will be explored in later sections.

12.2.2 Motivating Biological Principles

The vehicles for biological evolution are chromosomes, the 'blueprints' of living organisms. These chromosomes dictate the features of living creatures, and are determined by the previous generation. Modification of these chromosomes between generations allows for the adaptation of the specific descriptive features of an organism. These changes may act to improve or impair the fundamental 'fitness' of a creature, however natural selection will act to cause those creatures within a species with superior features (and therefore superior chromosomes) to reproduce more often than inferior creature, thus ensuring the survival of superior chromosomes over inferior ones. This 'survival of the fittest' process is defined as Darwinian natural selection. The chromosomes adapt during reproduction in various ways, either by mixing the chromosomes of parents, or by mutation.

12.2.3 Operation of Simulated Evolution

A simple example of an optimisation problem within the engineering domain is sufficient to elucidate the basic underlying principle of Simulated Evolution. The lift and drag of a wing are dependent upon a similar set of parameters such as chord length, wingspan, camber etc. An ideal wing has maximum lift and minimum drag, both to maximise aerodynamic efficiency and to reduce weight, and there is a degree of trade-off between them. An initial estimate of parameters, perhaps using historical or empirical values, is sufficient to provide a starting point, and typically a number of different designs are formulated. The lift and drag can then be evaluated for each configuration to indicate its 'fitness'. From these configurations, a new generation can be created by various operators, such as cross-over, reproduction and mutation. These operations are used to produce a new population of a specified size, with a range of fitnesses determined by reconstructing and evaluating the new configurations. By selecting the fittest offspring and removing the poorest, the new generation will represent an improvement on the parent generation. This process can be repeated until a certain level of fitness is reached. A suitable measure of fitness must be decided beforehand, typically constructed from a measure of accuracy and a penalty term for non-viable or impractical parameters.

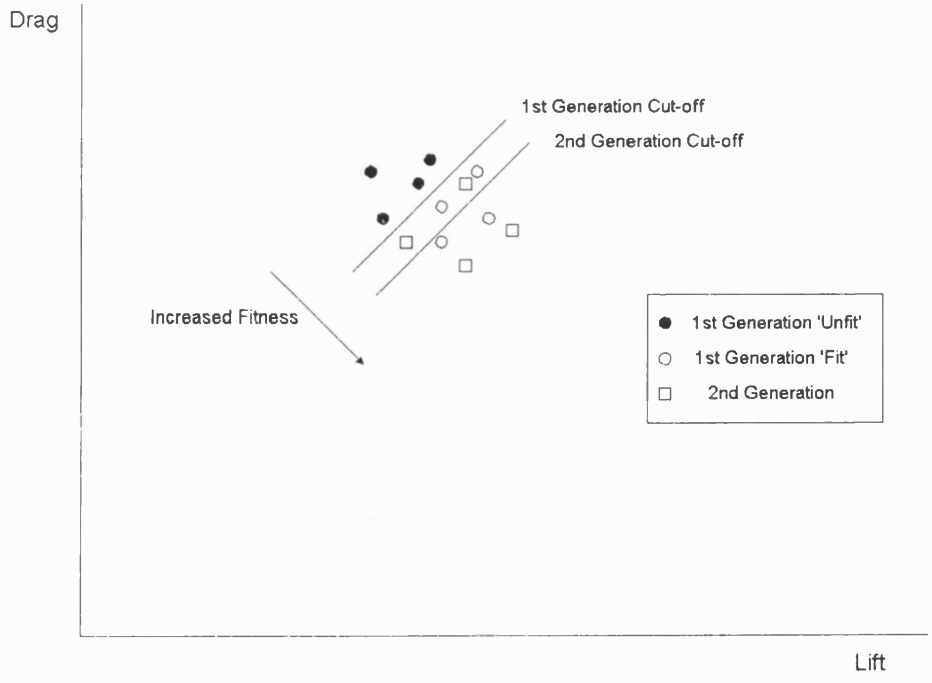


Figure 58 Illustration of Simulated Evolution Optimisation

Figure 58 shows a plot of 2 successive generations of 8 wing configurations. The optimum solution is located in the bottom right, with maximum lift and minimum drag. the 1st generation consists of 8 configurations with slightly different parameters, although there is no need to group these and more random configurations could have been used. The fittest 4 of the 1st generation act as parents for the 2nd generation, with the least fit 4 (indicated by a black fill) being killed off. In this way there is a progression of the population towards the optimum area.

Now the principle of Simulated Evolution has been covered, it is useful to understand the various operations that are available to create subsequent generations from parents. These operations, cross-over, selection and mutation, follow the paradigms of chromosomal change, individual change and species change respectively.

Cross-over

The parameters from each 'parent' are encoded into strings, analogous to genes, by concatenating each parameter in turn into either a 'real-coded' string, using the actual value of the parameter, or a series of bits (which is much more biologically plausible). By dividing each string at a preset point along its length and crossing over the strings at that point, joining the first part of the gene of one parent with the second part of the other parent, offspring are produced with the intention of finding one with the combined strengths of both parents, at the expense of the fitness of the second child. This process separates the evaluation from the optimisation, as the actual genetic operation and optimisation takes place on the concatenated string representation of the parameters of the wing, whilst the fitness is evaluated by arranging the wing according to the recombined string. This entails the use of an interpretation function, which maps the string to a set of corresponding parameters. Typically only the fittest configurations are selected for use as parents.

Selection

This operator seeks to obtain a certain portion of the population to carry on to the next generation, ensuring that the fittest configurations are not killed off. There are numerous methods achieving this, ranging from the 'roulette wheel' approach, where fitter configurations are biased to be more likely to be selected for use in the next generation, to 'elite' selection where a specified number of the fittest specimens are passed on. Whilst this operator does not explicitly optimise the configurations as it cannot change

any of the parameters, it merely ensures that the fittest are retained in subsequent populations for other future operations.

Mutation

The various constituent parameters are randomly varied by a predetermined amount, and the resultant configurations are evaluated for fitness, with a certain proportion of the fittest being retained to form the subsequent population. In this way the entire population can move towards an optimum solution, as only those that represent an improvement upon the earlier generation are used to populate the following generation.

The different operators are used most appropriately during different stages of the process (Pham and Karaboga, 1999b). Cross-over tends to be most effective in the early stages, where it is possible to incorporate the most attractive properties of 2 relatively sub-optimal parents. Mutation tends to be of most benefit in the latter stages of optimisation, where cross–over merely regenerate the current parents (Fogel, 1994), where it 'fine-tunes' the population and prevents stagnation in local minima. Whilst it is possible to use incorporate the binary or real-coded string approach for all 3 operators, it is not essential to do this for selection and mutation. For selection, the highest ranked individuals may be selected, whilst for mutation the parameters may be randomly modified in a defined manner with specified magnitudes and rate (number of individual parameter modifications).

12.2.4 Applications of Simulated Evolution

Whilst there are numerous examples of Simulated Evolution within engineering, especially manufacture (Ball et al., 1993), (Lazzerini and Marcelloni, 2000) and expanding take-up within Data Mining (Kamrani et al., 2001), this approach is not immediately applicable to the problem at hand. As in the case of the measure of quality in the Taguchi method, it is necessary to be able to evaluate the performance of each member of the population and, more importantly, control the parameters of each member. This research investigates the variance of parameters within manufacturing tolerance, and how these variances influence later performance of the artefact. As variance within tolerance is essentially uncontrollable, it is not possible to utilise Simulated Evolution directly in this research, although it may be used to supplement other approaches.

The performance of an ANN is dependent largely on the selected architecture, and constructive or deconstructive methods which have been introduced to deduce a suitable arrangement tend to be restrictive, either in terms of the final solution network they will allow (Fang and Xi, 1997) or on what type of network these methods can be implemented on (Angeline et al., 1994).

Computation of the required complexity (such as Huang and Huang, 1991, described previously) has been limited by the inability to transfer the conclusions of such work between different types of network, and more seriously by the lack of a guarantee that such a network will be able to select appropriate weights or to converge (Hagiwara, 1993). It is possible to iteratively add and remove nodes to obtain the most efficient architecture, however manual iteration of a network is impractical in terms of time, accuracy and interpreting which changes are most beneficial to the system – the operator will be iterating relatively blindly, with little objectivity or idea of direction. Simulated Evolution is in effect a form of directed random search (Nearchou, 1999), and perform optimisation by massively parallel iteration, properties making it well suited to automating the optimisation of ANNs.

There are various strands of research into the use of Simulated Evolution in ANN development, where the architecture and the weights of a network may be optimised either individually or concurrently. Perhaps the most useful aspect of this approach is the architecture optimisation, as other methods have certain prohibitive limitations. The concurrent approach is eminently sensible, as by combining the optimisation operations into one process there is information regarding a common error minimisation. Numerous authors concede the suitability of the back-propagation algorithm for determining suitable weights for a feed-forward network (Blanco et al., 2001), (Maniezzo, 1994), in most situations it tends to be faster and more stable than evolved optimisation (Kitano, 1990). Simulated Evolution has been used for weight optimisation in networks where this algorithm cannot be effectively used, for example in training the feedback loops of recurrent networks (Pham and Karaboga, 1999a). The back-propagation algorithm can be used for setting the weights of the feed-forward elements of such networks, but parameters for the recurrent (feedback loop) links are usually adjusted by a trial-and–error process. A further use of Simulated Evolution is in selection of a training algorithm (Pham and Karaboga, 1999b), typically in situations where the back-propagation algorithm or one of its derivatives is not viable.

12.2.5 Applications of Simulated Evolution in ANN Design

The method of architecture optimisation is simple and relatively uniform across most studies. A population of ANNs are created with various topologies and are trained by whichever method is preferred. Each network is evaluated and ranked according to a fitness score (which may be different from a simple test of accuracy). A new generation may then be constructed from the fittest using a selected operator or group of operators, and if none of the new population meet the desired stopping criteria the process is repeated. Angeline *et al* (1994) consider which operators may best be utilised for this approach, also approached by Fang and Xi (1997), and conclude cross-over is ineffective and mutation is most suitable for the task (Angeline *et al.* use the term Genetic Algorithm for a cross-over approach and Evolutionary Algorithm for mutation). The rationale behind this conclusion lies in the separation between the actual genetic operation and the construction and evaluation of the resultant network. The genetic operation of cross-over between genes occurs in the recombination space, and this is where the actual search takes place. The assembly of a new network on the strength of this recombination occurs in the evaluation space, requiring the additional complication of selection of a suitable interpretation function between the two spaces. On top of the complication of devising a suitable method for encoding the parameters of the network, there is also no guarantee that the devised network topology will be any more efficient, accurate or even if it will be viable (Angeline et al., 1994). Maniezzo (1994) has approached this by modifying the method of encoding parameter information into binary strings, but this method has seen little take-up as yet.

A comparison between various classifiers is given by Jagielska *et al* (1999), where, along with C4.5 and rough sets, fuzzy ANNs are compared against a GA-based fuzzy logic rule induction package for three different classification problems. The fuzzy approach has not been covered thus far, as Jagielska et al freely admit there has been little take-up of this approach in areas outside of fuzzy rule based controllers, and that few examples exist where fuzzy methods have been compared to other methods on real-life datasets. As this project is more concerned with the novel application of well-established techniques it is not considered viable to incorporate methods without prior indication of their suitability. The comparison between these methods differs from other GA investigations in that the GA is not used to optimise the ANN, rather it is used to optimise aspects of the fuzzy rule induction - details of which the reader is directed to the work by Carse *et al* (1996). The

ANN, a single hidden layer back-propagation network, was tested with various architectures and using two separate rule extraction algorithms. No pruning of the ANN was undertaken, rather the rule induction algorithms were used to simplify the rule base deduced from the full network. It was concluded that the various extracted rules and related accuracies were consistent between models, a result which, although not stated in the published work, suggests a strong underlying pattern within the data. The conclusions indicate that the GA-based fuzzy rule approach performed the most consistently over the 3 tests. It is argued that the successful implementation of ANNs is highly dependent on the architecture, and the manual modification of network architecture without consideration or iterative investigation brings us back to the problem of blind, non-subjective search for a suitable network. Confidence in the conclusions must therefore be tempered.

12.2.6 Examples of Simulated Evolution within ANNs

The use of Simulated Evolution in the design of ANNs is not limited to feed-forward Neural Networks, there are also examples of use in Recurrent Networks which are by their nature more complex than feed-forward. The training of the recurrent loops within such networks has been shown to be problematic when using back-propagation, and there are numerous examples of the use of Genetic Algorithms in place of this algorithm. Pham and Karaboga (1999b) introduce a method of training a modified Elman Net using Genetic Algorithms, and apply this method to the development of a fuzzy logic controller. Blanco *et al* (2001) develop an ANN using Genetic Algorithms for use in predicting fuzzy membership of classes (giving a measure of how much an instance belongs to a certain class or how strong a relationship between parameters is, in other words using classes with fuzzy range boundaries as opposed to crisp). Both Pham and Karaboga and Blanco *et al* use Genetic Algorithms to set the weights of all connections within the network, as the difficulties posed in architecture evolution in terms of encoding a variable architecture are not present,. It is possible to simply use the real values of the individual weights as the encoded parameters, thus avoiding the difficulties in encoding such information as a binary string and the resultant problems associated with evaluating these strings.

The previous examples of training using Simulated Evolution all incorporate networks with fixed architecture. The use of Simulated Evolution to optimise the architecture of

ANNs is arguably a more valuable tool, and whilst there are numerous examples of this there are certain differences of opinion as to the most suitable method of implementation. This conflict is to do with the specific operator used to evolve the network, as shown in the previous section Maniezzo (1994) proposes the use of cross-over and mutation (by using Genetic Algorithms) whilst both Angeline *et al* (1994) and Fang and Xi (1997) suggest that the requirement of encoding parameters in order to perform cross-over, and the difficulties in evaluating the results of this, mean that mutation should be the sole operator (in effect, recommending Evolutionary Programming).

All three parties use different methods of evaluation; Maniezzo and Angeline *et al* both opt for the use of specific problems, in the case of Maniezzo Rumelhart's test suite (Rumelhart et al., 1986) is selected, which incorporates the exclusive-or problem and other similar logical problems. Angeline *et al* use a similar indicator involving the development of a net to trigger a small system between a variety of states depending upon the value of certain inputs. Fang and Xi evaluate their approach by mapping two non-linear functions, with a simple function for the feed-forward network and a more complex example for a recurrent network. Maniezzo takes the evaluation one stage further by training an animat (an autonomous computational agent) to play a simple computer game, consisting of a two-dimensional 'playing field' with a goal at either end and a ball which must be manipulated into one of these goals (in effect a simple game of football). Seven inputs and five outputs were identified, defining the number of required nodes in the input and output layers respectively, and a Genetic Algorithm was used to optimise the number of hidden nodes. The behaviour (a function of topology and weight adjustments) of the network was seen to depend on the parameters selected for use, such as the speed of movement and the specified type of manipulation of the ball, however good results were noted.

All three methods of evaluation share the problem of control; all of the 'test suite' data is obtained from well-defined data and there is no indication of the possible performance of the approaches on 'real' data, which incorporates noise and may be incomplete or lacking quantity or density. A further point of interest is the size and complexity of the created networks; both Fang and Xi and Angeline *et al* created recurrent networks with between 5 and 10 nodes, depending on parameters used, whereas Maniezzo's animat network has greater than 12 nodes (the exact numbers are not quoted). Whilst of greater size, this network is feed-forward and therefore has fewer connection weights – it is unclear which

validation problem is more 'difficult' and would indicate the benefits of one individual methodology over another. The use of encoding has been shown to be possible on larger networks by Maniezzo although it may be argued that the expected network architecture could be estimated from previous works, such a priori knowledge serving as a good starting point. The difficulties of deducing a system to suitably encode a network in order to allow cross-over suggests that mutation is a more suitable operator, although cross-over may be of use in situations where a reasonable idea of a suitable topology is known beforehand and the encoding can be tailored around this structure.

12.3 Decision Tree Induction (DTI)

The problems of lack of clarity and coherency of explanation within Artificial Neural Networks can be eased by the use of Decision Tree Induction (DTI), where the structure of the data and the intermediate 'ground-level' decisions can readily be observed. An example of a decision tree created by DTI is shown in Figure 59, which classifies outcomes into 3 categories, A, B and C. In order to predict which class a new instance will fall into, by starting at the root node (the node at the very top of the tree) and following branches according the result of logical statements at the nodes a leaf will be reached. These leaves are labelled according to which class they represent.

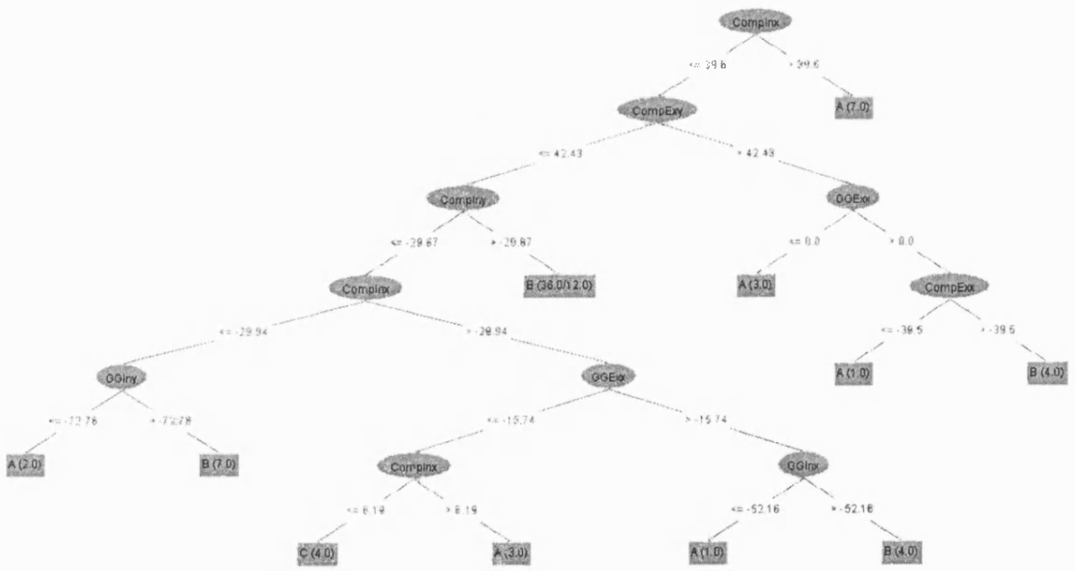


Figure 59 Sample Decision Tree

12.3.1 Operation of DTI

There are various algorithms for the induction of decision trees, the seminal algorithm among these is C4.5 which has been developed over the past 20 years by Quinlan (1986). A further development of this algorithm, C5.0, has recently been published but is released as proprietary software and hence the integration of other techniques or external development is limited, as is some of the detail of implementation. In the Clementine DM software package the C5.0 algorithm has been included and contains Boosting, a method of combining predictions from multiple models which is discussed in later sections. This method of model combination was not previously intrinsically supported by the C4.5 algorithm[44], although the C4.5 algorithm could be adapted to allow Boosting to be used (Quinlan, 1996a).

The approach of the algorithm is termed 'divide-and-conquer', where the choice of which attribute to select as the root node and for each subsequent node is decided by how effectively that attribute will split the data. Westphal and Blaxton (1998) provide a simple example describing classification of various car types according to customer age and sex. It may be argued that whether a car is 2-door or 4-door will have a large effect on the type of customer, whereas whether the car has air-conditioning or not will be less divisive. In this case, it will be more useful to select the number of doors as the root node in order to effectively split the data.

The underlying mechanism of the DTI algorithm is simple. A parameter is selected to take the role of the root node, and if this value is continuous a value is also selected to define the splitting criteria for that parameter. In the case of discrete parameters, the splitting criterion is already defined, as each discrete value is intrinsically different from another. The training dataset is then segregated by testing each parameter against the splitting criterion for the parameter in question, and will be passed to separate branches depending upon the result of this test, forming new subsets of the data. This process is repeated, with new branches and further subsets of data, until each instance within the

[44] The vagaries of the differences between C4.5 and C5.0 will not entered into, although the key differences are given on the commercial website for C5.0, http://www.rulequest.com

subset is of the same class, in which case development of that part of the tree is terminated.

The most complicated aspect of the DTI algorithm is the method used to select both the parameter and criterion to use for the split. In order to produce comprehensible trees with as small a structure as possible, it is beneficial to split the data such that instances of the same class reach the same branch in as few stages as possible.

Although knowledge obtained from a DTI approach gives more insights into the structure of the data, there are certain limitations. Fu and Shortliffe (2000) state that such an approach '..searches incompletely through a complete hypothesis space..', as the algorithm works downwards through the tree there is a significant amount of knowledge lost, which cannot be encapsulated within that specific model. DTI models are also intrinsically unstable, where small deviations within a training data set may lead to different split criteria and hence considerable change within the created model (Hastie et al., 2001) as any changes in the higher tiers of the tree will have effects on all preceding tiers.

12.3.2 Underlying Function of DTI

As mentioned earlier, the C5.0 algorithm is a development of the C4.5 algorithm and as such will be used as the basis for DTI modelling in this research. The mathematical background of the ANN approach was discussed in order to indicate how certain improvements such as momentum and learning rate can be used, and whilst there are no methods of DTI improvement that require an understanding of the functioning of the DTI algorithm, a brief mathematical description will be given for the sake of completeness. This will be described based upon the earlier C4.5 algorithm.

There are three principal decisions that a DTI algorithm must make, the first is to decide what a parameter will form the root node (and what value for the parameter will decide the split), when to stop growing the tree, and how to retrospectively reduce the complexity of the tree to reduce the possibility of overfitting (as seen in ANNs). Perhaps the most significant aspect is the decision regarding the root node, as this decides what information the tree will contain whereas the other decisions act more to assist in improving generality whilst maintaining accuracy.

Information Theory

The underlying principle of DTI is to partition data instances based upon class, or which range they fall into. There is also a requirement to partition the data in a concise manner, and hence there is a need to decide how much benefit is obtained during any particular partition. Information Theory suggests that there is an optimum way of describing information based upon the number of bits that are necessary, where this optimum number of bits is termed the entropy[45]. In telecommunications terms (where Information Theory was initially developed by Shannon, 1948) Information Theory regards information as being only those symbols that are unknown or uncertain to the receiver, and those messages that contain the greatest information, or greatest uncertainty to the receiver, are those of greatest entropy. In this manner, entropy can be used as a measure of the information obtained by a receiver, and of the conciseness of the method of communication.

Application of Information Theory to C4.5 Algorithm

The ideas of Information Theory can be successfully transferred into DTI. There is a need to consider how much information is obtained by a particular logical split, something termed the gain ratio.

$$Info(D) = -\sum_{j=1}^{C} p(D,j) \times \log_2(p(D,j))$$

The above expression indicates the residual uncertainty about which class an instance in a set D belongs, where C is the number of classes or ranges and $p(D,j)$ is the proportion of cases in set D that belong to the jth class or range. It is then necessary to divide the data using a test T, which produces mutually exclusive outcomes T1, T2, T3,...,Tn which partition set D into subsets D1, D2, D3,...,Dn, where Dn contains those cases that have the outcome Tn. The information gained by each test T, with k outcomes is given below.

$$Gain(D,T) = Info(D) - \sum_{n=1}^{k} \frac{|D_n|}{|D|} Info(D_n)$$

[45] Although not interchangeable with the thermodynamic definition of entropy, the use of the phrase highlights the similarities that both measures share in terms of randomness or uncertainty.

This idea of gain tends to favour those parameters that have large number of values, and where Gain(D,T) is maximal if there is one case in each of the subsets Dn (in cases where the parameter has a distinct value for each instance). To compensate for this it is possible to define split information, which tends to increase with the number of outcomes for a test, as follows.

$$Split(D,T) = -\sum_{n=1}^{k} \frac{|D_n|}{|D|} \times \log_2\left(\frac{|D_i|}{|D|}\right)$$

The gain ratio then becomes Gain(D,T)/Split(D,T). This gain ratio is then computed for each possible test within the dataset, and among those with at least average gain the split or test with maximum gain ratio is selected for use as the rot node.

A description of the C5.0 algorithm is given by Romanowski and Nagi (2001), and the only difference of note is the use of the term entropy when describing the residual uncertainty. It is unclear whether their description refers to C4.5 or to C5.0 as examination of their references indicate that they refer to a paper named 'Improved use of continuous attributes in C5.' whereas the correct title for the journal in question replaces C5 with C4.5. The differences in terminology may also be explained by a more faithful use of the original descriptions of the phenomena as proposed by Shannon (1948). In light of this error, and when considering the near identical descriptions, it is not clear which algorithm is under discussion. A separate argument might suggest that Romanowski and Nagi give a correct summary of the C5.0 algorithm, which would suggest that the differences in algorithm are minimal and C5.0 simply represents a incremental development of C4.5. Regardless of this situation, it is important to note that C5.0 is a development of C4.5 and hence it is argued that there will be many similarities between these algorithms, with essential variations attributable to improvements within the newer algorithm. It is also important to note that the precise functioning of each algorithm is not important to this research, where all that is required is a sound comprehension of the underlying processes that these algorithms use. In this sense, it is suggested that the processes will be identical for C4.5 and C5.0 for the purposes of this research.

12.3.3 Development of DTI

Many authors have highlighted the difficulties C4.5 has in handling continuous variables (such as Wang, 1999) although this has been addressed in further research (Quinlan,

1996c). In order to maintain one of the strengths of DTI, that of clarity of explanation of model structure, there have been developments in methods to reduce complexity in trees produced by a range of algorithms, C4.5 included (Quinlan, 1999). It is suggested that these developments have been incorporated into the C5.0 algorithm.

Emphasis within Machine Learning research has focused upon improving the prediction accuracy of a model with little attention paid to other forms of cost (Ling et al., 2004). Turney (2000) considers two other forms of cost, the first being due to misclassification and the second to test costs. Misclassification has different costs associated with different errors. For example, failing to predict an illness in a patient given a range of symptoms has significantly higher cost than incorrectly diagnosing illness in a healthy patient. In the second example, the cost of obtaining data for test can be factored into analysis, where cost is expended to obtain suitable data and complete any missing parameters for creation of the model. This consideration of test cost has not been introduced into C5.0, and is not intrinsically related to improvement in model performance, and will not be discussed further.

The C5.0 algorithm incorporates a function allowing for misclassification costs to be adjusted, thus weighting the creation of model to ensure that certain forms of error are avoided at the expense of others. The use of misclassification costs is not straightforward, where the specific costings to use are subjective and there is also a risk that they can be 'fudge factors' that are simply used to enforce a desired response from the model without addressing the key requirements (Weiss and Kulikowski, 1991). There is also the problem that the specific cost of each error is not always clear, and that different participants in the DM analysis might have different interpretations of these costs (Ting and Zheng, 1998).

In this research, emphasis will be placed upon an interpretation of the model, and less upon the predictions given by the model. Misclassification costs consider the effect a given form of error has, and are thus related to the prediction as opposed to structure of a model. When considered alongside the difficulty in correctly assigning these costs, it is argued that such costs are not suitable for use in this research.

12.3.4 Applications of Decision Tree Induction

The C4.5 algorithm has been widely accepted as the benchmark for classification schema and many new approaches use Decision Trees created using C4.5 as a control for

comparison. Of the literature previously discussed, Table 38 shows the schemas which have been compared directly to C4.5. There is an abundance of ANN/Rule extraction methods in this list, primarily as decision explanation in the form of rules is becoming more and more necessary, and it is this area that ANNs are weak. As C4.5 is considered ideal in this situation, it is perfectly sensible to compare and contrast the approaches to fully validate the new approach.

Name	Date	Compared Method
Lu *et al*	1996	ANN & Rule Extraction Engine
Fu	1999	ANN & Rule Extraction Engine
Taha & Ghosh	1999	ANN & Rule Extraction Engine
Gupta *et al*	1999	ANN & Rule Extraction Engine
Jagielska *et al*	1999	GA-based Fuzzy logic/Fuzzy ANN
Fu & Shortliffe	2000	ANN & Rule Extraction Engine

Table 38 Examples of Approaches where C4.5 used as Control

The prevalence of C4.5 as a control algorithm in various Machine Learning literature suggests that it is arguably the benchmark for DTI algorithms, and validates its selection for use in this research.

The range of application of DTI is perhaps not as clear as for Simulated Evolution and ANNs, as many applications will be referred to directly by algorithm name, for which there are numerous notable examples within DTI such as CART (Breiman et al., 1984), and CN2 (Clark and Niblett, 1989).

Chapter 13 Appendix C - Combination of Multiple Machine Learning Models

Of the numerous Machine Learning algorithms that are potentially of use, each one has associated benefits and drawbacks. Thus it is not always clear which will provide the most accurate predictions on new data until such models have been created, optimised and validated. Even at this point it is suggested to be inefficient to select one technique over another and lose all the benefits of the discarded method. It is more sensible to incorporate all of the available techniques in some form, and allow predictions from each to be considered. A further benefit of incorporating multiple models stems from the sensitivity of the algorithms to certain parameters; by using models assembled in different ways it is possible to avoid certain restrictions in the individual models.

A range of methods to combine the outputs of different models have been proposed. The technique of stacked generalisation has previously been described when applied to the combination of multiple ANN models (see section 12.1.9), and has been shown to be difficult to analyse and perhaps even more difficult to implement. The methods of Bagging and Boosting have been suggested as useful methods to combine decisions obtained from different decision tree induction models (Zheng and Webb, 1998). These two methods have restrictions, classical implementations of each method require the use of the same learning algorithm for each model (preventing different algorithms from being combined), and combining multiple models may act to prevent the explanation of results possible from an individual model (Hong and Weiss, 2001).

13.1 Bagging

The term 'Bagging' was initially coined by Breiman (Breiman, 1996) as an acronym for '**B**ootstrap **Agg**regat**ing**', a technique of producing multiple models from data sampled using a bootstrap technique (for details of bootstrap sampling see Efron and Tibshirani, 1993). The underlying idea of Bagging is to create multiple models (each constructed from individual, essentially random subsets of the training data) and then to combine the various predictions into one unifying prediction. This technique takes advantage of the

instabilities of various machine learning algorithms, in particular the high dependency upon training instances used for ANN modelling. The variation of training data will typically result in some differences between models with varying degrees of accuracy (Witten and Frank, 2000), and by selecting the final prediction that occurs most often (allowing each model to 'vote') the final prediction will be more representative of the data and less subject to instabilities within the modelling technique (Breiman, 1996), (Zheng and Webb, 1998).

The bootstrapping method used in Bagging is a statistically valid method of extracting representative subsets of training data. A specified number of subsets are defined each of the same size as the training set, and instances are selected to fill these subsets using random selection. An important point to note is the replacement of instances, selection for use in a subset does not entail deletion from the training set, in this way an instance may be selected more than once to appear in a given subset (Zheng and Webb, 1998). On average, 63% of the original training set will appear in any given subset (Domingos, 1997).

The success of Bagging depends upon the degree of instability within the model, 'if perturbing the learning set can cause significant changes in the predictor constructed, then Bagging can improve accuracy' (Breiman, 1996). Where the reverse is true, there are also cases where accuracy may be reduced (Witten and Frank, 2000). The differences between models created by different datasets is termed variance, whereas a consistent error amongst all models is termed bias (the 'persistent' error among a dataset). Bagging is only successful with problems of variance, although work is underway to identify ways to improve the handling of problems of bias (Domingos, 1997), (Breiman, 2001).

An important aspect of Bagging is the use of a baseline algorithm, a common algorithm that is used to create each individual model from the respective data subset. The principle of Bagging is that an aggregate prediction, taken by a vote of individual models, will *probably* be more accurate than any given individual vote – by using an individual model there is a possibility of selecting the 'wrong' data to use and the model will not be as accurate as perhaps possible. Whilst this is useful, it is suggested that the use of different algorithms may be applicable, as this would have a similar effect as using different subsets of data. There is no published work proving or disproving the validity of this argument, and so it is suggested that by using Bagging together with varying

algorithms, and using the idea of voting to select a final prediction, the overall accuracy can be improved. This idea will be further developed in Chapter 6.

13.2 Boosting

The creation of an effective Boosting algorithm has generally been credited to Freund and Schapire (1997), and their *AdaBoost* algorithm often serves as a benchmark for other such schema, as can be seen in work by Quinlan (1996a), (1996b) and Zheng and Webb (1998).

Boosting, much like Bagging, is based on the premise that a 'weak' learning algorithm, one with an accuracy not much greater than random, can be 'boosted' into a more accurate 'strong' algorithm (Freund and Schapire, 1999b). The basic principle of Boosting seeks to force the training algorithm to focus on more 'erroneous' instances, that is the instances it fails to handle accurately. This is accomplished by assigning a weight value to each input instance which indicates how accurately it has been trained. This weight can then be used to indicate to the algorithm how much effort needs to be expended in correctly training this instance, with more erroneous instances receiving more attention. In effect, the weighting for each instance influences the information gain equation described in section 12.3.2. A new model is created, where the tests to ascertain which attribute and which value to split upon are biased by the weightings for each instance, causing the new model to select tests that favour the correct classification of instances with high weightings (that were previously misclassified). It is important to note that there is no guarantee that this new model will be any more accurate overall, or that the erroneous instances will be dealt with any better. The weights are then updated in light of the relevant accuracies, and the process repeated for a specified number of cycles. The idea of weighting is extended into the voting stage; a model with a higher overall accuracy will have a higher voting weight and will therefore have more influence on the final prediction.

Various issues have been raised regarding the benefit of using Boosting, in some cases it may act to reduce overall accuracy, the precise reasons for which are the source of ongoing research (Quinlan, 1996b). Schapire and Singer (1999) suggest that this may be due to overfitting when used on smaller datasets, and argue that this may be addressed by controlling the complexity of subsequent models created by Boosting. The only

parameter that requires tuning is the number of iteration steps to use (Freund and Schapire, 1999b) and it has been found that it is possible to over train if too many steps are specified (as suggested theoretically by Jiang, 2004) although in some cases massive numbers of steps continue to improve accuracy (Freund and Schapire, 1999a). There appears to be no prescribed method of deducing the optimum number of steps, or if overfitting is likely, and it is suggested that this is something which is best found experimentally.

13.3 Applicability of Stacked Generalisation, Bagging and Boosting

The combining of models is an attractive proposition, as the effects of outliers can be mitigated against and a more accurate result can be achieved. The use of Stacked Generalisation is considered to be difficult to implement and analyse, but this is the only commonly recognised method which allows for the use of different learning algorithms. It is argued that the principle of Bagging can be applied to models created by different learning algorithms, therefore it is suggested that investigations into Bagging and stacked generalisation to combine different learning algorithms should be undertaken. The method of Boosting has been used in many applications, with agreeable results, and is therefore considered as suitable for use as a tool for obtaining aggregate predictions from numerous models. It is unclear which method will prove most accurate or useful. It is difficult to predict whether Boosting will improve accuracy (Quinlan, 1996a) and Bagging will only offer improvements if the model is sensitive to input data patterns (Breiman, 1996). Further to this, and in common with all machine learning methods, these methods can only create models from sample data and therefore training accuracy is subjective and cannot be considered as entirely representative of accuracy using fresh data. These methods are mostly useful for avoiding 'false minima' in terms of model parameter and training data selection, and despite Boosting being more theoretically rigorous it is suggested that the various methods should be tested on actual data before selecting one as the most suitable.

Applicability of Boosting to C4.5/C5.0

The method of Boosting is considered to be particularly suited to application upon DTI models. The calculation of the gain ration for deciding which set to use to split the data

relies upon an evaluation of the content of the individual subsets once the entire dataset has been subjected to this test. The Boosting algorithm requires that certain instances are weighted in order to ensure that the modelling algorithm focuses classification efforts upon those previously misclassified instances. These weights can be directly incorporated into the evaluation of the composition of the subsets, and therefore allow for Boosting to be simply incorporated into the DTI algorithm.